aws

Benutzerhandbuch

Amazon Relational Database Service



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Relational Database Service: Benutzerhandbuch

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon RDS? ... 1
Vorteile von Amazon RDS ... 1
Vergleich der Zuständigkeiten ... 2
Amazon-RDS-Modell der geteilten Verantwortung ... 3
DB-Instances ... 4
Amazon RDS-Anwendungsarchitektur: Beispiel ... 5
DB-Engine ... 6
DB-Instance-Klassen ... 7
DB-Instance-Speicher ... 8
DB-Instances in einer Amazon Virtual Private Cloud (Amazon VPC) ... 9
AWS-Regionen und Verfügbarkeitszonen ... 9
Availability Zones ... 10
Multi-AZ-Bereitstellungen ... 10
Zugriffskontrolle mit Sicherheitsgruppen ... 12
Amazon-RDS-Überwachung ... 13
Benutzeroberflächen zu Amazon RDS ... 15
AWS Management Console ... 15
Befehlszeilenschnittstelle ... 16
Amazon RDS APIs ... 16
Zahlungsmodell für Amazon RDS ... 16
Kostenloses AWS-Kontingent auf Amazon RDS ... 16
Als nächstes ... 17
Erste Schritte ... 17
Themen für Datenbank-Engines ... 18
DB-Instances ... 19
DB-Instance-Klassen ... 22
DB-Instance-Klassenarten ... 22
Unterstützte DB-Engines ... 31
Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen ... 98
Ändern Ihrer DB-Instance-Klasse ... 104
Konfigurieren des Prozessors für RDS für Oracle ... 104
Hardwarespezifikationen ... 127
DB-Instance-Speicher ... 168
Speichertypen ... 168

Bereitgestellter IOPS-Speicher ... 169
Speicher für allgemeine Zwecke ... 173
SSD-Leistungsmerkmale ... 177
Magnetischer Speicher (veraltet) ... 182
Dediziertes Protokollvolumen (DLV) ... 182
Überwachung der Datenbankleistung ... 184
Faktoren, die die Datenbankleistung beeinflussen ... 185
Regionen, Availability Zones und Local Zones ... 189
AWS Regionen ... 190
Availability Zones ... 196
Local Zones ... 196
Unterstützte RDS Amazon-Funktionen nach Region und Engine ... 199
Tabellenkonventionen ... 200
Kurzübersicht über Funktionen ... 200
Blau/Grün-Bereitstellungen ... 202
Regionsübergreifende automatisierte Backups ... 204
Regionsübergreifende Lesereplikate ... 205
Datenbankaktivitätsstreams ... 208
Dual-Stack-Modus ... 216
Exportieren von Snapshots nach S3 ... 236
IAM-Datenbankauthentifizierung ... 248
Kerberos-Authentifizierung ... 252
Multi-AZ-DB-Cluster ... 266
Performance Insights ... 276
RDS Custom ... 277
Amazon-RDS-Proxy ... 291
Integration von Secrets Manager ... 307
Null-ETL-Integrationen ... 307
Engine-native Funktionen ... 312
Abrechnung der DB-Instance für Amazon RDS ... 314
On-Demand-DB-Instances ... 316
Reservierte DB-Instances ... 318
Einrichtung ... 333
Melden Sie sich an für ein AWS-Konto ... 333
Erstellen eines Benutzers mit Administratorzugriff ... 334
Erteilen programmgesteuerten Zugriffs ... 335

Ermitteln der Anforderungen ... 337
Ermöglichen Sie Zugriff auf Ihre DB-Instance ... 340
Erste Schritte ... 344
Erstellen einer MariaDB-DB-Instance und Herstellen einer Verbindung dazu ... 345
Voraussetzungen ... 347
Erstellen Sie eine EC2 Instanz ... 347
Erstellen Sie eine MariaDB-DB-Instance ... 353
(Optional) Erstellen Sie VPC, EC2 Instanz und MariaDB-Instanz mit AWS CloudFormation . 359
Verbindung zu einer MariaDB-DB-Instance ... 361
Löschen Sie die EC2 Instance und die DB-Instance ... 365
(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation ... 365
(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion. ... 366
Erstellen einer DB-Instance von Microsoft SQL Server und Herstellen einer Verbindung ... 367
Voraussetzungen ... 369
Erstellen einer EC2-Instance ... 369
Erstellen Sie eine SQL Server-DB-Instance ... 374
(Optional) Erstellen Sie VPC, EC2 Instanz und SQL Server-Instanz mit AWS CloudFormation ... 380
Stellen Sie eine Verbindung zu Ihrer SQL Server-DB-Instance her ... 382
Arbeiten mit der Beispiel-DB-Instance ... 385
Löschen Sie die EC2 Instanz und die DB-Instance ... 386
(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation ... 387
(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion. ... 388
Erstellen einer MySQL-DB-Instance und Herstellen einer Verbindung dazu ... 389
Voraussetzungen ... 391
Erstellen einer EC2-Instance ... 391
Erstellen Sie eine MySQL-DB-Instance ... 397
(Optional) Erstellen Sie VPC, EC2 Instanz und MySQL-Instanz mit AWS CloudFormation ... 403
Verbindung zu einer MySQL-DB-Instance ... 405
Löschen Sie die EC2 Instance und die DB-Instance ... 409
(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation ... 410
(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion. ... 411
Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung ... 412

Voraussetzungen 414
Schritt 1: Erstellen Sie eine EC2 Instanz 414
Schritt 2: Erstellen einer Oracle-DB-Instance 420
(Optional) Erstellen Sie VPC, EC2 Instance und Oracle DB-Instance mit AWS CloudFormation 426
Schritt 3: Verbinden Ihres SQL-Clients mit einer Oracle-DB-Instance 428
Schritt 4: Löschen Sie die Instance und die DB-Instance EC2 432
(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation 433
(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion. 433
Erstellen einer PostgreSQL-DB-Instance und Herstellen einer Verbindung 434
Voraussetzungen 436
Erstellen Sie eine EC2 Instanz 436
Erstellen Sie eine PostgreSQL-DB-Instance 442
(Optional) Erstellen Sie VPC, EC2 Instanz und PostgreSQL-Instanz mit AWS CloudFormation 448
Herstellen einer Verbindung zu einer PostgreSQL-DB-Instance 450
Löschen Sie die EC2 Instance und die DB-Instance 454
(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation 454
(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion. 455
Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance 456
Starten Sie eine EC2 Instance, um eine Verbindung mit Ihrer herzustellen 458
Erstellen einer DB-Instance. 464
Erstellen eines Webservers 485
Tutorial: Erstellen einer Lambda-Funktion für den Zugriff auf Ihre DB-Instance von Amazon RDS 497
Voraussetzungen 498
Erstellen einer DB-Instance von Amazon RDS 498
Erstellen einer Lambda-Funktion und eines Proxys 500
Erstellen einer Funktionsausführungsrolle 501
Erstellen eines Lambda-Bereitstellungspakets 502
Aktualisieren der Lambda-Funktion 505
Testen Ihrer Lambda-Funktion in der Konsole. 507
Erstellen einer Amazon-SQS-Warteschlange 508
Erstellen einer Zuordnung von Ereignisquellen, um Ihre Lambda-Funktion aufzurufen 509

Testen und Überwachen Ihrer Einrichtung 510
Bereinigen Ihrer Ressourcen 511
Tutorials und Beispiel-Code 513
Tutorials in diesem Handbuch 513
Tutorials in anderen AWS Leitfäden 514
AWS Portal für Workshop- und Laborinhalte für Amazon RDS PostgreSQL 515
AWS Portal für Workshop- und Laborinhalte für Amazon RDS MySQL 516
Tutorials und Beispielcode in GitHub 516
Arbeitet mit AWS SDKs 516
Bewährte Methoden für Amazon RDS 518
Grundlegende Anleitungen für den Amazon RDS-Betrieb 519
RAM-Empfehlungen für DB-Instances 520
Halten Sie die Versionen der Datenbank-Engine auf dem neuesten Stand 520
AWS Datenbanktreiber 521
Verwenden von Enhanced Monitoring 521
Verwenden von -Metriken 522
Anzeigen von Leistungsmetriken 522
Auswerten von Leistungsmetriken 525
Optimieren von Abfragen 527
Best Practices für die Arbeit mit MySQL 528
Tabellengröße 528
Anzahl der Tabellen 529
Speicher-Engine 530
Best Practices für die Arbeit mit MariaDB 531
Tabellengröße 531
Anzahl der Tabellen 532
Speicher-Engine 532
Bewährte Methoden für die Arbeit mit Oracle 533
Bewährte Methoden für die Arbeit mit PostgreSQL 533
Laden von Daten in eine PostgreSQL-DB-Instance 533
Arbeiten mit der PostgreSQL-Selbstbereinigungsfunktion 534
Amazon RDS for PostgreSQL Video zu bewährten Praktiken 536
Bewährte Methoden für die Arbeit mit SQL Server 536
Video zu bewährten Methoden für Amazon RDS for SQL Server 537
Arbeiten mit DB-Parametergruppen 537
Automatisieren der Erstellung von DB-Instances 537

Video zu den neuen Funktionen 538
Programmatischer Zugriff auf Amazon RDS Amazon 539
Console-to-Code 540
Konfigurieren einer DB-Instance 541
Erstellen einer DB-Instance 542
Voraussetzungen 542
Erstellen einer DB-Instance 549
Verfügbare Einstellungen 556
Ressourcen erstellen mit AWS CloudFormation 594
RDS und AWS CloudFormation Vorlagen 594
Erfahren Sie mehr über AWS CloudFormation 594
Herstellen einer Verbindung mit einer DB-Instance 595
Suchen der Verbindungsinformationen 595
Szenarien für den Zugriff auf eine DB-Instance 599
Mit den AWS Treibern eine Verbindung zu DB-Instances herstellen 600
Herstellen einer Verbindung mit einer DB-Instance, auf der eine bestimmte DB-Engine ausführt wird 601
Verbindungen mit RDS Proxy verwalten 602
Datenbankauthentifizierungsoptionen 602
Verschlüsselte Verbindungen 603
Arbeiten mit Optionsgruppen 604
Übersicht über die Optionsgruppen 604
Erstellen einer Optionsgruppe 607
Kopieren einer Optionsgruppe 609
Hinzufügen einer Option zu einer Optionsgruppe 610
Auflisten der Optionen und Optionseinstellungen für eine Optionsgruppe 617
Ändern einer Optionseinstellung 618
Entfernen einer Option aus einer Optionsgruppe 622
Löschen einer Optionsgruppe 624
Parametergruppen 628
Übersicht über Parametergruppen 628
DB-Parametergruppen 633
DB-Cluster-Parametergruppen 649
Vergleichen von DB-Parametergruppen 664
Festlegen von DB-Parametern 664
Einen ElastiCache Cache von Amazon erstellen RDS 672

Überblick über die ElastiCache Cache-Erstellung mit den ... 672
Einen ElastiCache Cache mit Einstellungen aus einer erstellen ... 673
Automatische Migration von Datenbanken EC2 ... 677
Übersicht ... 677
Voraussetzungen ... 678
Einschränkungen ... 679
Erstellen von IAM-Ressourcen ... 680
Datenmigration einrichten ... 688
Verwaltung von Migrationen ... 690
Überwachen ... 693
Tutorial: Erstellen eines mit einem benutzerdefinierten Parameter und einer neuen Optionsgruppe ... 695
Voraussetzungen ... 696
Erstellen Sie eine benutzerdefinierte Parametergruppe ... 696
Ändern Sie den Wert der ... 697
Optionsgruppe erstellen ... 697
Option hinzufügen ... 698
Erstellen Sie Ihren ... 699
Verwalten einer DB-Instance ... 701
Anhalten einer DB-Instance ... 702
Anwendungsfälle ... 703
Unterstützte DB-Engines, -Klassen und Regionen ... 703
Support für Multi-AZ ... 704
Funktionsweise ... 704
Einschränkungen ... 706
Options- und Parametergruppen ... 706
Öffentliche IP-Adresse ... 707
Anhalten einer DB-Instance ... 707
Starten einer DB-Instance ... 709
Neustarten einer DB-Instance ... 711
Anwendungsfälle für den Neustart einer DB-Instance einem DB-Cluster ... 711
Wie funktioniert ein Neustart ... 712
Neustart in Multi-AZ ... 712
Überlegungen ... 713
Voraussetzungen ... 714
Neustarten einer DB-Instance : grundlegende Schritte ... 714

Eine EC2 Instance verbinden ... 716
Übersicht ... 717
Eine EC2 Instance verbinden ... 722
Anzeigen verbundener Rechenressourcen ... 725
Herstellen einer Verbindung mit einer DB-Instance, auf der eine bestimmte DB-Engine ausführt wird ... 726
Verbinden einer Lambda-Funktion ... 727
Übersicht ... 729
Verbinden einer Lambda-Funktion ... 740
Anzeigen verbundener Rechenressourcen ... 742
Ändern einer DB-Instance ... 744
Änderungen planen ... 746
Verfügbare Einstellungen ... 747
Warten einer DB-Instance ... 791
Überblick über die Wartungsupdates für DB-Instances ... 791
Anzeigen ausstehender Wartung ... 793
Warten der Multi-AZ-Bereitstellungen ... 795
Wartungsfenster ... 796
Anwenden von Updates ... 802
Betriebssystemupdates ... 804
Upgrade der Engine-Version ... 808
Manuelles Upgraden der Engine-Version ... 809
Automatisches Upgraden der Engine-Unterversion ... 811
Umbenennen einer DB-Instance ... 816
Umbenennen einer bestehenden DB-Instance, um diese zu ersetzen ... 817
Arbeiten mit DB-Instance-Lesereplikaten ... 820
Übersicht ... 822
Unterschiede zwischen DB-Engines ... 826
Erstellen eines Lesereplikats ... 830
Hochstufen eines Lesereplikats ... 834
Überwachen der Lesereplikation ... 840
Regionsübergreifende Lesereplikate ... 843
RDS Ressourcen taggen ... 858
Warum RDS Tags verwenden? ... 859
Wie funktionieren RDS Tags ... 860
Bewährte Methoden ... 863

Tags in DB-Snapshots kopieren ... 864
Hinzufügen und Löschen von Tags in Amazon RDS ... 865
Tutorial: Geben Sie mithilfe von Tags an, welche DB-Instances gestoppt werden sollen ... 870
ARNsbei Amazon RDS ... 874
Erstellen eines ARN ... 874
Ein bestehendes bekommen ARN ... 882
Arbeiten mit Speicher ... 885
Steigern der DB-Instance-Speicherkapazität ... 885
Automatische Kapazitätsverwaltung mit automatischer -Speicherskalierung ... 888
Upgrade des Speicherdateisystems ... 897
Ändern der bereitgestellten Einstellungen IOPS ... 899
E/A-intensive Speichermodifikationen ... 901
Ändern von Einstellungen für Allzweckspeicher (gp3) ... 902
Verwendung eines dedizierten Protokoll-Volumes (DLV) ... 905
Löschen einer DB-Instance ... 912
Voraussetzungen für das Löschen einer DB-Instance ... 912
Überlegungen beim Löschen einer DB-Instance ... 912
Löschen einer DB-Instance ... 914
Tutorial: Verwaltung einer My SQL DB-Instance ... 917
Einführung ... 917
Voraussetzungen ... 918
Fügen Sie Ihrer DB-Instance Tags hinzu ... 918
Erhöhen Sie den DB-Speicher ... 919
Erstellen von Read Replicas ... 919
Aktualisieren der Tags ... 920
Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung ... 924
Multi-AZ-DB-Instance-Bereitstellungen ... 926
Konvertierung einer DB-Instance in eine Multi-AZ-Bereitstellung ... 928
Failover einer Multi-AZ-DB-Instance ... 930
Multi-AZ-DB-Cluster-Bereitstellungen ... 936
Verfügbarkeit der Instance-Klassen für Multi-AZ-DB-Cluster ... 937
Multi-AZ-DB-Cluster-Architektur ... 937
Parametergruppen für Multi-AZ-DB-Cluster ... 939
RDSProxy mit Multi-AZ-DB-Clustern ... 939
Replikatverzögerung und Multi-AZ-DB-Cluster ... 940
Snapshots von Multi-AZ-DB-Clustern ... 943

Erstellen eines Multi-AZ-DB-Clusters ... 944
Herstellen einer Verbindung zu einem Multi-AZ-DB-Cluster ... 974
Eine AWS Rechenressource und einen Multi-AZ-DB-Cluster verbinden ... 981
Ändern eines Multi-AZ-DB-Clusters ... 1010
Aktualisierung eines Multi-AZ-DB-Clusters ... 1032
Umbenennen eines Multi-AZ-DB-Clusters ... 1035
Neustarten von Multi-AZ-DB-Clustern ... 1038
Failover eines Multi-AZ-DB-Clusters ... 1040
SQLLogische Postgre-Replikation mit Multi-AZ-DB-Clustern ... 1044
Arbeiten mit Multi-AZ-DB-Cluster-Lesereplikaten ... 1049
Einrichtung der externen Replikation von Multi-AZ-DB-Clustern ... 1062
Löschen eines Multi-AZ-DB-Clusters ... 1064
Einschränkungen von Multi-AZ-DB-Clustern ... 1067
RDS Extended Support ... 1068
Überblick über den RDS Extended Support ... 1069
RDSGebühren für erweiterten Support ... 1070
Vermeidung von Gebühren für RDS erweiterten Support ... 1071
Versionen mit erweitertem RDS-Support ... 1071
Versionsbenennung für RDS Extended Support ... 1071
Aufgaben mit RDS erweitertem Support ... 1072
Verantwortlichkeiten RDS ... 1073
Ihre Aufgaben ... 1073
Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters, ... 1074
RDSErweitertes Support-Verhalten ... 1074
Überlegungen zum RDS erweiterten Support ... 1075
Erstellen Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, mit RDS erweitertem Support. ... 1076
Registrierung für RDS Extended Support anzeigen ... 1077
Support-Daten anzeigen ... 1080
Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, ... 1083
RDSErweitertes Support-Verhalten ... 1084
Überlegungen zum RDS erweiterten Support ... 1084
Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit RDS erweitertem Support ... 1085
Verwendung von Blau/Grün-Bereitstellungen für Datenbankaktualisierungen ... 1088
Überblick über Blue/Green-Bereitstellungen ... 1089

Verfügbarkeit von Regionen und Versionen ... 1090
Vorteile ... 1090
Workflow ... 1091
Autorisieren des Zugriffs ... 1095
Einschränkungen und Überlegungen ... 1096
Bewährte Methoden ... 1105
Postgre-Replikationsmethoden SQL ... 1110
Erstellen einer Blau/Grün-Bereitstellung ... 1111
Vorbereiten einer Blau-Grün-Bereitstellung ... 1112
Angeben von Änderungen ... 1114
Verzögertes Laden und Speicherinitialisierung ... 1117
Erstellen einer Blau/Grün-Bereitstellung ... 1119
Verfügbare Einstellungen ... 1121
Anzeigen einer Blau/Grün-Bereitstellung ... 1126
Umstellen einer Blau/Grün-Bereitstellung ... 1131
Umstellungs-Timeout ... 1132
Integrationsschutz der Umstellung ... 1132
Umstellungsaktionen ... 1133
Bewährte Methoden für die Umstellung ... 1134
Überprüfung der CloudWatch Metriken vor dem Switchover ... 1136
Überwachung der Replikatverzögerung vor dem Switchover ... 1136
Umstellen einer Blau/Grün-Bereitstellung ... 1137
Nach der Umstellung ... 1140
Löschen einer Blau/Grün-Bereitstellung ... 1142
Daten sichern, wiederherstellen und exportieren ... 1146
Einführung in Backups ... 1147
Backup-Speicher ... 1147
Verwaltung automatisierter Backups ... 1149
Backup-Fenster ... 1149
Aufbewahrungszeitraum für Backups ... 1152
Aktivieren von automatisierten Backups ... 1153
Aufbewahren automatisierter Backups ... 1155
Löschen aufbewahrter automatisierter Backups ... 1158
Meine SQL Speicher-Engines werden nicht unterstützt ... 1161
Nicht unterstützte MariaDB-Speicher-Engines ... 1163
Regionsübergreifende automatisierte Backups ... 1164

Verwaltung manueller Backups ... 1183
Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance ... 1184
Erstellen eines Multi-AZ-DB-Cluster-Snapshots ... 1187
Löschen eines DB-Snapshots ... 1189
Wiederherstellung auf einer DB-Instance ... 1192
Wiederherstellung aus einem Snapshot ... 1193
Überlegungen ... 1195
Point-in-time Wiederherstellung ... 1200
Wiederherstellen eines Multi-AZ-DB-Clusters zu einer bestimmten Zeit ... 1206
Wiederherstellen von einem Snapshot in einem Multi-AZ-DB-Cluster ... 1210
Wiederherstellen über einen Snapshot eines Multi-AZ-DB-Clusters in einer DB-Instance ... 1213
Tutorial: Wiederherstellen einer DB-Instance aus einem DB-Snapshot ... 1216
Kopieren eines DB-Snapshots ... 1221
Kopieren eines DB-Snapshots ... 1221
Einschränkungen ... 1232
Überlegungen ... 1233
Freigeben eines DB Snapshots ... 1244
Freigeben eines Snapshots ... 1246
Freigeben öffentlicher Snapshots ... 1249
Freigeben verschlüsselter Snapshots ... 1252
Das Teilen von Snapshots wird beendet ... 1256
Exportieren von DB-Snapshot-Daten nach Amazon S3 ... 1258
Übersicht über das Exportieren von Snapshot-Daten ... 1259
Einrichten des Zugriffs auf einen S3-Bucket ... 1260
Exportieren eines DB-Snapshots ... 1266
Verfügbarkeit von Regionen und Versionen ... 1270
Einschränkungen ... 1270
Überwachung von Snapshot-Exporten ... 1272
Abbrechen eines Snapshot-Exports ... 1274
Fehlernachrichten ... 1275
Behebung von Postgre-Berechtigungsfehlern SQL ... 1277
Konventionen zur Benennung von Dateien ... 1278
Datenkonvertierung ... 1279
Verwenden AWS Backup ... 1290
Überwachen von Metriken in einer DB-Instance ... 1291
Überwachungsplan ... 1291

Leistungsbasislinie 1292
Richtlinien zur Leistung 1292
Überwachungstools 1294
Automatisierte Überwachungstools 1294
Manuelle Überwachungstools 1296
Status der anzeigen 1298
Anzeigen von Amazon RDS DB-Instance-Status 1299
Empfehlungen von Amazon RDS 1306
Betrachtungsempfeh 1308
Empfehlungen anwenden 1316
Empfehlungen werden abgelehnt 1322
Abgelehnte Empfehlungen in aktive ändern 1324
Referenz zu Empfehlungen 1326
Metriken in der RDS Amazon-Konsole anzeigen 1352
Das Performance Insights Insights-Dashboard anzeigen 1353
Auswahl der neuen Überwachungsansicht auf der Registerkarte Überwachung 1354
Auswahl der neuen Überwachungsansicht auf der Seite Performance Insights 1356
Ein benutzerdefiniertes Dashboard erstellen 1358
Auswahl des vorkonfigurierten Dashboards 1361
Überwachung von RDS mit CloudWatch 1363
Überblick über Amazon RDS, und Amazon CloudWatch 1364
CloudWatch Metriken anzeigen 1366
Performance Insights Insights-Metriken exportieren nach CloudWatch 1371
CloudWatch Alarme erstellen 1377
Tutorial: Einen CloudWatch Alarm für eine Verzögerung bei der DB-Cluster-Replikation erstellen 1377
Überwachung mit Database Insights 1385
Preisgestaltung 1385
Unterstützung für Engine, Region und Instance-Klasse 1385
Den erweiterten Modus einschalten 1390
Den Standardmodus einschalten 1395
Überwachen Sie langsame Abfragen 1400
Überlegungen 1403
Überwachung von DB-Last mit Performance Insights 1404
Überblick über Performance Insights 1405
Aktivieren und Deaktivieren von Performance Insights 1419

Leistungsschema für MariaDB oder MySQL 1424
Performance Insights-Richtlinien 1430
Analyse der Metriken mit dem Performance Insights-Dashboard 1443
Proaktive Empfehlungen von Performance Insights anzeigen 1495
Metriken mit Performance Insights abrufen API 1498
Protokollieren von Performance Insights Insights-Aufrufen mit AWS CloudTrail 1524
VPCEndpunkte ()AWS PrivateLink 1526
Analyse der Leistung mit Guru DevOps für RDS 1531
Vorteile von DevOps Guru für RDS 1531
Wie RDS funktioniert DevOps Guru for 1533
Guru einrichten für DevOps RDS 1534
Überwachen vom Betriebssystem mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung) 1543
Überblick über „Enhanced Monitoring“ (Erweiterte Überwachung) 1543
Einrichten und Aktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung) 1545
Betriebssystemmetriken in der RDS Konsole anzeigen 1551
Betriebssystemmetriken mithilfe von CloudWatch Protokollen anzeigen 1555
RDSReferenz zu 1557
CloudWatch Metriken für RDS 1557
CloudWatch Abmessungen für RDS 1580
CloudWatch Metriken für Performance Insights 1581
Zählermetriken für Performance Insights 1584
SQL-Statistiken für Performance Insights 1622
Betriebssystemmetriken in „Erweiterte Überwachung“ 1640
Überwachen von Ereignissen, Protokollen und Datenbankaktivitäts-Streams 1656
Protokolle, Ereignisse und Streams in der RDS Amazon-Konsole anzeigen 1657
Überwachung von 1661
Übersicht der Ereignisse für Amazon RDS 1661
RDSAmazon-Ereignisse anzeigen 1663
Mit RDS Amazon-Event-Benachrichtigungen arbeiten 1667
Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird 1693
Amazon RDS-Ereigniskategorien und Ereignisnachrichten 1699
Überwachung von RDS 1751
Anzeigen und Auflisten von Datenbank-Protokolldateien 1751
Herunterladen einer Datenbank-Protokolldatei 1752
Überwachen einer Datenbank-Protokolldatei 1754

In CloudWatch Logs veröffentlichen ... 1755
Lesen von Protokolldateiinhalten mit REST ... 1758
Db2-Datenbank-Protokolldateien ... 1760
MariaDB-Datenbank-Protokolldateien ... 1765
Protokolldateien der Microsoft SQL Server-Datenbank ... 1781
MySQL-Datenbank-Protokolldateien ... 1787
Oracle-Datenbank-Protokolldateien ... 1805
PostgreSQL-Datenbankprotokolldateien ... 1816
Überwachung von CloudTrail ... 1831
CloudTrail Integration mit Amazon RDS ... 1831
Amazon-Protokolldateieinträge ... 1832
Überwachung von RDS mithilfe von Datenbankaktivitätsstreams ... 1837
Übersicht ... 1837
Konfigurieren der einheitlichen Prüfung für Oracle ... 1844
Konfiguration der SQL Serverüberwachung ... 1845
Starten eines Datenbankaktivitäts-Streams ... 1847
Ändern eines Datenbankaktivitäts-Streams ... 1850
Abrufen des Status eines Aktivitätsstreams ... 1853
Stoppen eines Datenbankaktivitäts-Streams ... 1854
Überwachen von Aktivitäts-Streams ... 1855
Beispiele für IAM-Richtlinien für Aktivitätsstreams ... 1898
Überwachung von Bedrohungen mit GuardDuty RDS Protection ... 1902
Amazon RDS Custom ... 1904
Herausforderung für Datenbankanpassung ... 1904
RDS Benutzerdefiniertes Managementmodell und Vorteile ... 1906
Modell der geteilten Verantwortung in RDS Custom ... 1906
Support-Perimeter und nicht unterstützte Konfigurationen in RDS Custom ... 1909
Hauptvorteile von RDS Custom ... 1909
Benutzerdefinierte RDS Architektur ... 1910
VPC ... 1911
RDS Kundenspezifische Automatisierung und Überwachung ... 1912
Amazon S3 ... 1917
AWS CloudTrail ... 1918
Sicherheit in RDS Custom ... 1919
So verwaltet RDS Custom Aufgaben sicher in Ihrem Namen ... 1919
SSL-Zertifikate ... 1920

Schützen Sie Ihren Amazon S3 S3-Bucket vor dem Problem des verwirrten Stellvertreters 1920
Rotierende Anmeldeinformationen für RDS Custom for Oracle ... 1922
Arbeiten mit RDS Custom for Oracle ... 1927
RDS Benutzerdefiniert für Oracle-Workflow ... 1927
Datenbankarchitektur für Amazon RDS Custom für Oracle ... 1933
Verfügbarkeit und Unterstützung von Funktionen für RDS Custom for Oracle ... 1935
Anforderungen und Einschränkungen für RDS Custom for Oracle ... 1938
Einrichten Ihrer RDS Custom for Oracle-Umgebung ... 1942
Arbeiten mit CEVs for RDS Custom for Oracle ... 1964
Konfigurieren einer DB-Instance von RDS Custom für Oracle ... 1997
Verwaltung einer RDS benutzerdefinierten für Oracle DB-Instance ... 2018
Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Oracle ... 2037
Mit RDS Custom for Oracle-Replikaten arbeiten ... 2057
Sicherung und Wiederherstellung einer RDS Custom for Oracle DB-Instance ... 2068
Mit Optionsgruppen in RDS Custom for Oracle arbeiten ... 2080
Migration zu Custom for RDS Oracle ... 2089
Upgrade einer RDS benutzerdefinierten DB-Instance für Oracle ... 2090
Fehlerbehebung für RDS Custom für Oracle ... 2104
Bekannte Probleme für RDS Custom for Oracle ... 2129
Mit RDS Custom for SQL Server arbeiten ... 2133
RDSBenutzerdefinierter Arbeitsablauf für SQL Server ... 2133
Anforderungen und Einschränkungen für RDS Custom for SQL Server ... 2136
Einrichten Ihrer RDS Custom for SQL Server-Umgebung ... 2215
Bringen Sie Ihre eigenen Medien mit RDS Custom for SQL Server ... 2242
Arbeiten mit CEVs für RDS Custom für SQL Server ... 2244
Eine RDS Custom for SQL Server-DB-Instance erstellen und eine Verbindung zu ihr herstellen ... 2268
Verwaltung einer RDS Custom for SQL Server-DB-Instance ... 2281
Arbeiten mit Microsoft Active Directory mit RDS Custom for SQL Server ... 2296
Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Server SQL ... 2325
Sicherung und Wiederherstellung einer RDS Custom for SQL Server-DB-Instance ... 2342
Kopieren eines DB-Snapshots vom Typ RDS Custom for SQL Server ... 2360
Migrieren einer lokalen Datenbank zu RDS Custom for SQL Server ... 2372
Upgrade einer RDS-Instance Custom für SQL Server ... 2376
Problembehebung für Amazon RDS Custom for SQL Server ... 2380
Amazon RDS auf AWS Outposts ... 2423

Voraussetzungen ... 2424
Support für RDS Amazon-Funktionen ... 2425
Unterstützte DB-Instance-Klassen ... 2432
IP-Adressen im Besitz des Kunden ... 2435
Verwendung von CoIPs ... 2435
Einschränkungen ... 2437
Multi-AZ-Bereitstellungen ... 2439
Arbeiten mit dem Modell der geteilten Verantwortung ... 2439
Verbessern der Verfügbarkeit ... 2439
Voraussetzungen ... 2440
Arbeiten mit API-Operationen für Amazon-EC2-Berechtigungen ... 2442
Erstellen von DB-Instances für RDS on Outposts ... 2443
Erstellen von Lesereplikaten für RDS on Outposts ... 2454
Überlegungen zum Wiederherstellen von DB-Instances ... 2457
Amazon-RDS-Proxy ... 2459
Verfügbarkeit von Regionen und Versionen ... 2460
Kontingente und Einschränkungen ... 2460
Einschränkungen bei RDS for MariaDB ... 2462
Einschränkungen für RDS für SQL Server ... 2463
Einschränkungen für MySQL ... 2464
PostgreSQL-Beschränkungen ... 2465
Planen des Verwendungsortes von RDS-Proxy ... 2466
RDSKonzepte und Terminologie von Proxys ... 2468
Überblick über die RDS Proxy-Konzepte ... 2468
Verbindungspooling ... 2470
Sicherheit ... 2470
Failover ... 2472
Transaktionen ... 2474
Erste Schritte mit RDS Proxy ... 2474
Richten Sie ein Proxy-Netzwerk ein ... 2475
Datenbankanmeldedaten einrichten ... 2478
Konfigurieren der IAM-Authentifizierung ... 2482
Einen Proxy erstellen ... 2486
Einen Proxy anzeigen ... 2495
Verbinden über RDS Proxy ... 2497
Einen RDS Proxy verwalten ... 2501

Ändern eines RDS Proxy ... 2502
Hinzufügen eines Datenbankbenutzers ... 2509
Überlegungen zur RDS-Proxyverbindung ... 2510
Vermeiden Sie das Fixieren eines RDS-Proxys ... 2514
Einen RDS Proxy löschen ... 2521
Arbeiten mit RDS Proxy-Endpunkten ... 2522
Überblick über Proxy-Endpunkte ... 2523
Limits für Proxy-Endpunkte ... 2524
Proxy-Endpunkte für Multi-AZ-DB-Cluster ... 2524
Zugriff auf RDS-Datenbanken über VPCs ... 2526
Erstellen eines Proxy-Endpunktes ... 2528
Anzeigen von Proxy-Endpunkten ... 2530
Ändern eines Proxy-Endpunkts ... 2532
Löschen eines Proxy-Endpunkts ... 2533
Überwachung des RDS-Proxys mit CloudWatch ... 2535
Mit RDS Proxy-Ereignissen arbeiten ... 2543
RDSProxy-Ereignisse ... 2544
Fehlerbehebung für RDS Proxy ... 2547
Überprüfen der Konnektivität für einen Proxy ... 2547
Häufige Probleme und Lösungen ... 2549
Problembehandlung für RDS for MySQL ... 2550
Problembehandlung für RDS for PostgreSQL ... 2552
Verwenden von RDS Proxy mit AWS CloudFormation ... 2558
Null-ETL-Integrationen ... 2560
Vorteile ... 2562
Die wichtigsten Konzepte ... 2562
Einschränkungen ... 2563
Allgemeine Einschränkungen ... 2563
Einschränkungen von RDS für MySQL ... 2564
Einschränkungen von RDS für PostgreSQL ... 2565
Beschränkungen von RDS for Oracle ... 2565
Einschränkungen für Amazon Redshift ... 2566
Amazon SageMaker AI Einschränkungen bei Lakehouse ... 2566
Kontingente ... 2566
Unterstützte Regionen ... 2567
Erste Schritte mit Null-ETL-Integrationen ... 2567

Schritt 1: Erstellen einer benutzerdefinierten DB--Parametergruppe ... 2568
Schritt 2: Wählen oder erstellen Sie einen ... 2569
Schritt 3a: Erstellen Sie ein Ziel-Data Warehouse ... 2570
Richten Sie eine Integration mit dem ein AWS SDKs ... 2571
Schritt 3b: Erstellen Sie einen AWS Glue Katalog für die Amazon SageMaker AI Zero-ETL-Integration ... 2576
Nächste Schritte ... 2580
Erstellen von Zero-ETL-Integrationen mit Amazon Redshift ... 2580
Voraussetzungen ... 2580
Erforderliche Berechtigungen ... 2580
Erstellen von Null-ETL-Integrationen ... 2583
Verschlüsselung von Integrationen ... 2588
Nächste Schritte ... 2590
Erstellung von Zero-ETL-Integrationen mit einem Lakehouse Amazon SageMaker ... 2590
Voraussetzungen ... 2591
Erforderliche Berechtigungen ... 2591
Erstellung von Zero-ETL-Integrationen mit einem Lakehouse Amazon SageMaker ... 2594
Verschlüsselung von Integrationen ... 2599
Nächste Schritte ... 2601
Datenfilterung für Zero-ETL-Integrationen ... 2601
Format eines Datenfilters ... 2602
Filterlogik ... 2605
Priorität filtern ... 2606
RDS für MySQL-Beispiele ... 2606
— Beispiele ... 2607
Beispiele für RDS für Oracle ... 2608
Hinzufügen von Datenfiltern ... 2609
Datenfilter werden entfernt ... 2611
Daten hinzufügen und abfragen ... 2611
Eine Zieldatenbank erstellen ... 2612
Daten zum der Quelldatenbank hinzufügen ... 2612
Abfragen Ihrer Amazon RDS in Amazon Redshift ... 2614
Datentypunterschiede ... 2615
... 2634
Anzeigen und Überwachen von Null-ETL-Integrationen ... 2637
Anzeigen von Integrationen ... 2638

Überwachen mithilfe von Systemtabellen ... 2639
Überwachung mit EventBridge ... 2640
Änderung von Zero-ETL-Integrationen ... 2640
Löschen von Null-ETL-Integrationen ... 2642
Fehlerbehebung bei Null-ETL-Integrationen ... 2643
Ich kann keine Null-ETL-Integration erstellen ... 2644
Meine Integration steckt in einem Zustand von `Syncing` ... 2644
Meine Tabellen werden nicht auf Amazon Redshift repliziert ... 2644
Eine oder mehrere meiner Amazon-Redshift-Tabellen erfordern eine erneute Synchronisation ... 2645
Fehler bei der Integration, Probleme bei Lakehouse Zero-ETL-Integrationen Amazon SageMaker AI ... 2648
Db2 auf Amazon RDS ... 2650
Überblick über Db2 ... 2651
Db2-Funktionen ... 2652
Db2-Versionen ... 2656
Db2-Lizenzierung ... 2660
Db2-Instance-Klassen ... 2679
Db2-Standardrollen ... 2681
Db2-Parameter ... 2682
EBCDICSortierung ... 2688
Lokale Zeitzone von Db2 ... 2689
Voraussetzungen für DB-Instance ... 2692
Administratorkonto ... 2692
Weitere Überlegungen ... 2693
Mehrere Db2-Datenbanken ... 2694
Verbindung zu Ihrer Db2-DB-Instance herstellen ... 2696
Ermitteln des Endpunkts ... 2696
IBM Db2 CLP ... 2698
IBM CLPPlus ... 2703
DBeaver ... 2706
IBM Db2 Data Management Console ... 2710
Überlegungen zu Sicherheitsgruppen ... 2720
Sicherung von Db2-Verbindungen ... 2721
Verschlüsseln mit SSL/TLS ... 2721
Authentifizierung verwenden Kerberos ... 2729

Verwaltung Ihrer RDS für Db2-DB-Instance ... 2747
Systemaufgaben ... 2749
Datenbankaufgaben ... 2762
Integration in S3 ... 2787
Erstellen einer IAM-Richtlinie ... 2787
Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu ... 2790
Fügen Sie Ihre IAM Rolle zu Ihrer DB-Instance hinzu ... 2793
Migrieren Sie Daten zu for Db2 RDS ... 2796
Daten mit Services migrieren AWS ... 2796
Daten mit nativen Db2-Tools migrieren ... 2808
Verbund ... 2823
Homogener Verbund ... 2823
Heterogener Verband ... 2828
Arbeiten mit Db2-Replikaten ... 2833
Standby-Replikate ... 2833
Datenbank-Aktivierungen ... 2833
HARD-Konfigurationen ... 2834
Aufbewahrung von Archivprotokollen ... 2834
Ausfälle während der Db2-Replikation ... 2834
Anforderungen und Überlegungen für Db2-Replikate ... 2835
Vorbereitung der Erstellung eines Db2-Replikats ... 2837
Ein Standby-Db2-Replikat erstellen ... 2838
Arbeiten mit Db2-Replikat-Backups ... 2840
Behebung von Problemen mit der Db2-Replikation ... 2843
Optionen RDS für Db2-DB-Instances ... 2844
Db2-Auditprotokollierung ... 2845
Externe gespeicherte Prozeduren ... 2861
Java-basierte externe gespeicherte Prozeduren ... 2861
Bekannte Probleme und Einschränkungen ... 2870
Beschränkung der Authentifizierung ... 2870
Routinen ohne Umzäunung ... 2870
Nichtautomatische Speicher-Tablespaces während der Migration ... 2870
Einstellung des Parameters db2_compatibility_vector ... 2871
Migrieren von Datenbanken, die UNGÜLTIGE Pakete enthalten ... 2871
Gespeicherte RDS-Prozeduren für Db2 ... 2873
Überlegungen zu gespeicherten Prozeduren ... 2883

Erteilen und Widerrufen von Rechten ... 2884
Überwachungsrichtlinien ... 2903
Pufferpools ... 2909
Datenbanken ... 2915
Zugriff auf den Speicher ... 2945
Tablespaces ... 2948
RDSfür benutzerdefinierte Db2-Funktionen ... 2960
rdsadmin.get_task_status ... 2960
rdsadmin.list_databases ... 2965
Fehlerbehebung ... 2967
Fehler bei der Datenbankverbindung ... 2967
Ein Fehler in der Datei I/O ... 2967
Fehler bei gespeicherten Prozeduren ... 2971
MariaDB auf Amazon RDS ... 2983
MariaDB-Funktionsunterstützung ... 2985
MariaDB-Hauptversionen ... 2985
Unterstützte Speicher-Engines ... 2994
Cache-Warming ... 2996
Nicht unterstützte Features ... 2997
MariaDB-Versionen ... 2999
Unterstützte MariaDB-Nebenversionen ... 2999
Unterstützte MariaDB-Hauptversionen ... 3007
Die Datenbank-Vorschauumgebung ... 3008
MariaDB Version 11.8 in der Database Preview-Umgebung ... 3012
MariaDB Version 11.7 in der Database Preview-Umgebung ... 3012
MariaDB Version 11.4 in der Database Preview-Umgebung ... 3012
Veraltete MariaDB-Versionen ... 3013
Verbinden mit einer DB-Instance, auf der MariaDB ausgeführt wird ... 3014
Suchen der Verbindungsinformationen ... 3015
Vom Befehlszeilenclient aus wird eine Verbindung hergestellt ... 3019
Verbindung zu den Treibern herstellen AWS ... 3020
Fehlerbehebung ... 3021
Sichern von MariaDB-Verbindungen ... 3022
MariaDB-Sicherheit ... 3022
Plugins zur Passwortvalidierung ... 3024
Verschlüsseln mit/SSLTLS ... 3025

Neue /-Zertifikate SSL verwenden TLS 3030
Verbesserung der Abfrageleistung mit RDS Optimized Reads 3035
Übersicht 3035
Anwendungsfälle 3036
Bewährte Methoden 3037
Die Verwendung von 3037
Überwachen 3038
Einschränkungen 3039
Verbesserung der Schreibleistung mit RDS-optimierten Schreibvorgängen für MariaDB 3040
Übersicht 3040
Verwendung mit einer neuen Datenbank 3042
Aktivieren in einer vorhandenen Datenbank 3046
Einschränkungen 3047
Upgrades der MariaDB-DB-Engine 3048
Überlegungen 3049
Suche nach gültigen Upgrade-Zielen 3050
MariaDB-Versionsnummern 3051
RDSVersionsnummern 3053
Hauptversions-Upgrades 3054
Upgraden einer MariaDB-DB-Instance 3055
Automatische Unterversion-Upgrades 3055
Upgrade mit reduzierten Ausfallzeiten 3058
Aktualisierung einer MariaDB DB-Snapshot-Engine-Version 3063
Upgrade-Optionen für nicht unterstützte Engine-Versionen 3065
Daten in eine RDS for MariaDB-DB-Instance importieren 3067
Überlegungen zum Importieren von Daten 3072
Importieren von Daten aus einer externen Datenbank 3078
Importieren von Daten mit reduzierter Ausfallzeit 3082
Importieren von Daten aus jeder Quelle 3104
MariaDB-Replikation 3111
MariaDB liest Repliken 3112
Konfiguration der GTID basierten Replikation mit einer externen Quellinstanz 3128
Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance 3132
Optionen für MariaDB 3138
MariaDB Audit-Plugin-Support 3138

Parameter für MariaDB 3146
MariaDB-Parameter anzeigen 3146
MySQL-Parameter, die nicht verfügbar sind 3148
Daten von einem My SQL DB-Snapshot zu einer MariaDB-DB-Instance migrieren 3151
Durchführen der Migration 3151
Inkompatibilitäten zwischen MariaDB und My SQL 3153
MariaDB auf Amazon RDS – SQL-Referenz 3155
mysql.rds_replica_status 3155
mysql.rds_set_external_master_gtid 3157
mysql.rds_kill_query_id 3160
Lokale Zeitzone 3162
Bekannte Probleme und Einschränkungen für MariaDB 3165
Beschränkungen der Dateigröße 3165
Reserviertes Wort InnoDB 3167
Benutzerdefinierte Ports 3167
Performance Insights 3167
Microsoft SQL Server in Amazon RDS 3168
Häufige Verwaltungsaufgaben 3170
Einschränkungen 3173
Unterstützung für DB-Instance-Klassen 3176
Sicherheit 3181
SSLUnterstützung 3182
Verwendung SSL mit einer SQL Server-DB-Instance 3183
Konfiguration von SQL Serversicherheitsprotokollen und Chiffren 3189
Aktualisierung von Anwendungen für neue SSL TLS /-Zertifikate 3196
Compliance-Programme 3201
HIPAA 3201
Versionsunterstützung 3202
Versionsverwaltung. 3207
Feature-Unterstützung 3209
SQLFunktionen von Server 2022 3210
SQLFunktionen von Server 2019 3210
SQLFunktionen von Server 2017 3211
SQLFunktionen von Server 2016 3212
SQLFunktionen von Server 2014 3212
SQLEnde des Supports für Server 2012 bei Amazon RDS 3212

SQLEnde des Supports für Server 2008 R2 bei Amazon RDS ... 3212
CDCUnterstützung ... 3213
Nicht unterstützte und eingeschränkte Funktionsunterstützung ... 3213
Multi-AZ-Bereitstellungen ... 3215
Verwenden von TDE ... 3216
Funktionen und gespeicherte Prozeduren ... 3216
Lokale Zeitzone ... 3223
Unterstützte Zeitzonen ... 3224
Lizenzierung von SQL Server auf Amazon RDS ... 3236
Wiederherstellen von DB-Instances, die aus Lizenzgründen beendet wurden ... 3236
SQL Server Developer-Edition ... 3237
Herstellen einer Verbindung zu einer DB-Instance, die SQL Server ausführt ... 3238
Bevor Sie sich verbinden ... 3238
Finden des Endpunkts und der Portnummer der DB-Instance ... 3239
Verbindung zu Ihrer DB-Instance herstellen mit SSMS ... 3241
Herstellen einer Verbindung zu Ihrer DB-Instance mit SQL Workbench/J ... 3244
Überlegungen zu Sicherheitsgruppen ... 3246
Fehlerbehebung ... 3247
Arbeiten mit Active Directory mit RDS für SQL Server ... 3249
Arbeiten mit Self Managed Active Directory mit einer SQL Server-DB-Instance ... 3250
Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server ... 3272
Upgrades der SQL Server-DB-Engine ... 3289
Hauptversions-Upgrades ... 3290
Überlegungen zum Upgrade ... 3292
Testen eines Upgrades ... 3295
Aktualisieren einer SQL Server-DB-Instance ... 3297
Aktualisieren veralteter DB-Instances vor dem Ende des Supports ... 3297
Importieren und Exportieren von SQL-Server-Datenbanken ... 3298
Einschränkungen und Empfehlungen ... 3301
Einrichten ... 3303
Verwenden von nativer Sicherung und Wiederherstellung ... 3308
Komprimieren von Sicherungsdateien ... 3327
Fehlerbehebung ... 3328
Importieren und Exportieren von SQL Server-Daten mithilfe anderer Methoden ... 3332
Verwenden von BCP unter Linux ... 3347
SQL Server-Lesereplikate ... 3353

Konfigurieren von Read Replicas für SQL Server ... 3353
Read-Replica-Einschränkungen mit SQL Server ... 3354
Überlegungen zu Optionen ... 3355
Synchronisieren von Datenbankbenutzern und Objekten ... 3357
Fehlerbehebung ... 3359
Multi-AZ für Server RDS SQL ... 3360
Hinzufügen von Multi-AZ zu einer SQL Server-DB-Instance ... 3361
Multi-AZ aus einer SQL Server-DB-Instance entfernen ... 3362
Einschränkungen, Hinweise und Empfehlungen ... 3362
Festlegen des Standorts der sekundären Instance ... 3366
Migration zu Always On AGs ... 3367
Zusätzliche Funktionen für SQL Server ... 3369
Verwenden der Kennwortrichtlinie mit einer SQL Server-DB-Instance ... 3370
Amazon S3-Integration ... 3378
Verwenden von Database Mail ... 3401
Instance-Speicher-Support für tempdb ... 3417
Verwenden erweiterter Ereignisse ... 3420
Zugriff auf Transaktionsprotokoll-Backups ... 3424
Optionen für SQL Server ... 3471
Auflisten der verfügbaren Optionen für Versionen und Editionen von SQL Server ... 3473
Mit Oracle OLEDB verknüpfte Server ... 3475
Verknüpfte Server mit Teradata ODBC ... 3488
Native Sicherung und Backup ... 3498
Transparente Datenverschlüsselung in ... 3503
SQLServerprüfung ... 3517
SQLServeranalyse-Services ... 3528
SQL Server Integration Services ... 3559
SQLServer Reporting Services ... 3583
Microsoft Distributed Transaction Coordinator ... 3604
Häufige DBA-Aufgaben ... 3623
Zugriff auf die Datenbank tempdb ... 3625
Analysieren der Datenbank-Workload mit Database Engine Tuning Advisor ... 3629
Ändern des `db_owner`- in das `rdsa`-Konto für Ihre Datenbank ... 3634
Verwaltung von Kollationen und Zeichensätzen ... 3635
Erstellen eines Datenbankbenutzers ... 3644
Bestimmen eines Wiederherstellungsmodells ... 3645

Ermitteln der letzten Failover-Zeit ... 3646
Behebung von PITR-Fehlern aufgrund von LSN-Lücken ... 3647
Verweigern oder erlauben Sie das Anzeigen von Datenbanknamen ... 3648
Deaktivieren von schnellen Einfügungen ... 3649
Verwerfen einer SQL Server-Datenbank ... 3650
Umbenennen einer Multi-AZ-Datenbank ... 3651
Die Rollenmitgliedschaft für den Masterbenutzer wird zurückgesetzt db_owner ... 3651
Wiederherstellen von DB-Instances, die aus Lizenzgründen beendet wurden ... 3652
Übergang einer Datenbank von zu OFFLINE ONLINE ... 3653
Verwenden von CDC ... 3653
Verwenden von SQL Server Agent ... 3657
Arbeiten mit SQL Server-Protokollen ... 3664
Arbeiten mit Trace- und Dump-Dateien ... 3666
Mein SQL bei Amazon RDS ... 3668
Unterstützung von MySQL-Funktionen ... 3671
MySQL-Hauptversionen ... 3671
Unterstützte Speicher-Engines ... 3674
Verwenden von memcached und anderen Optionen ... 3674
InnoDB-Cache-Initialisierung ... 3675
Inklusive Sprachänderungen für MySQL 8.4 ... 3676
Nicht unterstützte Features ... 3679
MySQL-Versionen ... 3681
Nebenversionen ... 3681
Hauptversionen ... 3688
Versionen mit erweitertem RDS-Support ... 3690
Umgebung mit Datenbankvorschau ... 3694
9.3 (Vorschau) ... 3698
9.2 (Vorschau) ... 3698
9.1 (Vorschau) ... 3698
Veraltete Versionen ... 3698
Verbinden mit einer DB-Instance, auf der MySQL ausgeführt wird ... 3700
Suchen der Verbindungsinformationen ... 3701
Installation des Befehlszeilenclients ... 3705
Vom Befehlszeilenclient aus wird eine Verbindung hergestellt ... 3705
Von My SQL Workbench aus eine Verbindung herstellen ... 3706
Verbindung zu den AWS Treibern herstellen ... 3709

Fehlerbehebung 3710
Sichern von MySQL-Verbindungen 3712
Überprüfung des Passworts 3715
Verschlüsseln mit/SSLTLS 3716
Neue SSL TLS /-Zertifikate verwenden 3720
KerberosAuthentifizierung wird verwendet für MySQL 3726
Verbesserung der Abfrageleistung mit RDS optimierten Lesevorgängen 3742
Übersicht 3742
Anwendungsfälle 3743
Bewährte Methoden 3744
Die Verwendung von 3745
Überwachen 3745
Einschränkungen 3746
Verbesserung der Schreibleistung mit RDS Optimized Writes for My SQL 3747
Übersicht 3040
Verwendung mit einer neuen Datenbank 3748
Aktivieren in einer vorhandenen Datenbank 3753
Einschränkungen 3754
Upgrades der MySQL-DB-Engine 3755
Überlegungen 3756
Suche nach gültigen Upgrade-Zielen 3757
MySQL-Versionsnummern 3758
RDS-Versionsnummern 3760
Hauptversions-Upgrades 3761
Testen eines Upgrades 3767
Upgraden einer MySQL-DB-Instance 3769
Automatische Unterversion-Upgrades 3769
Upgrade mit reduzierten Ausfallzeiten 3772
Aktualisierung einer MySQL-DB-Snapshot-Engine-Version 3777
Upgrade-Optionen für nicht unterstützte Engine-Versionen 3779
Daten in eine RDS for MySQL-DB-Instance importieren 3782
Überlegungen zum Importieren von Daten 3787
Wiederherstellen eines Backups in einer MySQL-DB-Instance 3793
Importieren von Daten aus einer externen Datenbank 3809
Importieren von Daten mit reduzierter Ausfallzeit 3814
Importieren von Daten aus jeder Quelle 3836

MySQL-Replikation 3843
Meine SQL Read Replicas 3844
GTIDbasierte Replikation 3862
Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance 3871
Konfiguration der Replikation mit mehreren Quellen 3876
Konfiguration von aktiv-aktiven Clustern 3885
Anwendungsfälle 3885
Einschränkungen und Überlegungen für Active-Active-Cluster 3886
Vorbereitung für einen Cross-Active-Active-Cluster VPC 3890
Erforderliche Parametereinstellungen für Active-Active-Cluster 3892
Konvertierung einer DB-Instance in einen Active-Active-Cluster 3895
Einrichtung eines neuen Active-Active-Clusters 3901
Hinzufügen einer DB-Instance zu einem Active-Active-Cluster 3908
Überwachung von aktiv-aktiven Clustern 3911
Beenden der Gruppenreplikation in einem aktiv-aktiven Cluster 3912
Umbenennen einer DB-Instance in einem Active-Active-Cluster 3913
Entfernen einer DB-Instance aus einem Active-Active-Cluster 3914
Daten aus einer My SQL DB-Instance exportieren 3915
Bereiten Sie eine externe Datenbank vom Typ Meine SQL Datenbank vor 3915
Bereiten Sie die My SQL DB-Quell-Instance vor 3917
Kopieren der Datenbank 3918
Abschließen des Exportvorgangs 3920
Optionen für Meine SQL 3922
MariaDB-Audit-Plug-In 3923
memcached 3932
Parameter für My SQL 3938
Allgemeine DBA Aufgaben für My SQL 3941
Grundlegendes zu vordefinierten Benutzern 3941
Rollenbasiertes Berechtigungsmodell 3942
Dynamische Privilegien 3945
Beenden einer Sitzung oder Abfrage 3949
Überspringen von Fehlern für die aktuelle Replikation 3950
Verbessern Sie die Wiederherstellungszeiten nach einem Absturz 3952
Verwalten des globalen Statusverlaufs 3955
Konfiguration der Größe des Pufferpools und der Redo-Log-Kapazität 3957

Lokale Zeitzone ... 3959
Bekannte Probleme und Einschränkungen ... 3962
Reserviertes Wort InnoDB ... 3962
Vollständiges Storage-Verhalten ... 3962
Inkonsistente Größe des InnoDB-Buffer-Pools ... 3963
Index-Merge-Optimierung zeigt falsche Ergebnisse an ... 3964
MySQL-Parameterausnahmen für Amazon RDS-DB-Instances ... 3965
MySQL-Dateigrößenlimits in Amazon RDS ... 3966
MySQL Keyring-Plugin wird nicht unterstützt ... 3968
Benutzerdefinierte Ports ... 3969
Einschränkungen bei gespeicherten MySQL-Prozeduren ... 3969
GTID-basierte Replikation mit einer externen Quell-Instance ... 3969
MySQL-Standardauthentifizierungs-Plugin ... 3969
Überschreiben von innodb_buffer_pool_size ... 3970
Upgrade von MySQL 5.7 auf MySQL 8.4 ... 3970
InnoDB-Seitenkomprimierung ... 3970
RDSfür meine SQL gespeicherten Prozeduren ... 3971
Erfassung und Pflege der globalen Statushistorie ... 3972
Konfiguration, Starten und Stoppen der Binärprotokollreplikation (Binlog) ... 3975
Beenden einer Sitzung oder Abfrage ... 4019
Verwaltung von aktiv-aktiven Clustern ... 4021
Verwaltung der Replikation mit mehreren Quellen ... 4026
Transaktionen replizieren mit GTIDs ... 4049
Rotieren der Abfrage-Logs ... 4052
Konfiguration des Binärprotokolls einrichten und anzeigen ... 4054
Wärmen des InnoDB-Caches ... 4059
Oracle in Amazon RDS ... 4061
Oracle-Übersicht ... 4062
Oracle-Funktionen ... 4063
Oracle-Versionen ... 4067
Oracle-Lizenzen ... 4074
Oracle-Benutzer und -Berechtigungen ... 4078
Oracle Instance-Klassen ... 4079
Oracle-Datenbankarchitektur ... 4089
Oracle-Parameter ... 4090
Oracle-Zeichensätze ... 4093

Oracle-Beschränkungen ... 4097
Herstellen der Verbindung zu Ihrer Oracle-DB-Instance ... 4100
Ermitteln des Endpunkts ... 4100
SQL Developer ... 4102
SQL*Plus ... 4105
Überlegungen zu Sicherheitsgruppen ... 4106
Dedizierte und gemeinsam genutzte Serverprozesse ... 4107
Fehlersuche ... 4107
Ändern von sqlnet.ora-Parametern für Oracle ... 4109
Sichern von Oracle-Verbindungen ... 4115
Verschlüsseln mit SSL ... 4115
Verwendung neuer SSL/TLS-Zertifikate ... 4116
Verschlüsseln mit NNE ... 4120
Konfigurieren der Kerberos-Authentifizierung ... 4121
Konfigurieren des UTL_HTTP-Zugriffs ... 4142
Arbeiten mit CDBs ... 4155
Überblick über CDBs ... 4155
Konfiguration einer CDB ... 4161
Sichern und Wiederherstellen einer CDB ... 4169
Konvertieren einer Nicht-CDB in eine CDB ... 4170
Umwandlung von Single-Tenant in Multi-Tenant ... 4173
Hinzufügen einer RDS-für-Oracle-Tenant-Datenbank zu Ihrer CDB-Instance ... 4176
Ändern einer Tenant-Datenbank von RDS für Oracle ... 4179
Löschen einer Tenant-Datenbank von RDS für Oracle aus Ihrer CDB ... 4181
Anzeigen von Details zu einer Tenant-Datenbank ... 4184
Aktualisieren Ihrer CDB ... 4189
Administration Ihrer Oracle-DB-Instance ... 4190
Systemaufgaben ... 4205
Datenbankaufgaben ... 4233
Protokollaufgaben ... 4264
RMANAufgaben ... 4278
Oracle-Scheduler-Aufgaben ... 4317
Probleme diagnostizieren ... 4327
Weitere Aufgaben ... 4337
Konfiguration erweiterter Funktionen von RDS für Oracle ... 4353
Konfigurieren des Instance-Speichers ... 4353

Aktivieren von HugePages ... 4365
Aktivieren erweiterter Datentypen ... 4368
Importieren von Daten zu Oracle ... 4372
Importieren mit Oracle SQL Developer ... 4373
Migrieren mithilfe von Oracle Transportable Tablespaces ... 4373
Importieren mit Oracle Data Pump ... 4391
Import unter Verwendung von Oracle-Export/-Import ... 4409
Importieren mit Oracle SQL*Loader ... 4410
Migrieren mit materialisierten Oracle-Ansichten ... 4412
Arbeiten mit Oracle-Replikaten ... 4415
Übersicht über Oracle-Replikate ... 4415
Anforderungen und Überlegungen zu Oracle-Replikaten ... 4418
Vorbereiten der Erstellung eines Oracle-Replikats ... 4422
Erstellen eines bereitgestellten Oracle-Replikats ... 4424
Ändern des Replikatmodus ... 4426
Arbeiten mit Oracle-Replikat-Backups ... 4427
So führen Sie eine Oracle Data Guard-Umschaltung aus ... 4430
Fehlerbehebung bei Oracle-Replikaten ... 4438
Optionen für Oracle ... 4440
Übersicht über Oracle-DB-Optionen ... 4440
Amazon S3-Integration ... 4443
Oracle Application Express (APEX) ... 4471
EFSAmazon-Integration ... 4496
Java Virtual Machine (JVM) ... 4514
Enterprise Manager ... 4519
Label Security ... 4545
Ortung ... 4549
Native Netzwerkverschlüsselung (NNE) ... 4554
OLAP ... 4571
Secure Sockets Layer (SSL) ... 4575
Dreidimensional ... 4588
SQLT ... 4593
Statspack ... 4603
Zeitzone ... 4607
Automatische Aktualisierung der Zeitzonendatei ... 4613
Transparente Datenverschlüsselung (TDE) ... 4624

UTL_MAIL ... 4629
XML DB ... 4633
Aktualisieren der Oracle-DB-Engine ... 4634
Übersicht über Oracle-Upgrades ... 4634
Hauptversions-Upgrades ... 4642
Unterversion-Upgrades ... 4643
Überlegungen zum Upgrade ... 4650
Testen eines Upgrades ... 4654
Aktualisierung einer RDS für Oracle-DB-Instance ... 4655
Aktualisieren eines Oracle-DB-Snapshots ... 4657
Tools und Software von Drittanbietern für Oracle ... 4660
Oracle verwenden GoldenGate ... 4661
Verwenden des Oracle Repository Creation Utility ... 4681
Konfigurieren von CMAN ... 4689
Installieren einer Siebel-Datenbank auf Oracle auf Amazon RDS ... 4692
Versionshinweise zu Oracle-Database-Engine-Releases ... 4697
PostgreSQL in Amazon RDS ... 4698
Häufige Verwaltungsaufgaben ... 4700
Arbeiten mit der Datenbank-Vorschauumgebung ... 4704
Nicht in der Datenbank-Vorschauumgebung unterstützte Features ... 4705
SQLPostgre-Version 17 in der Database Preview-Umgebung ... 4705
Erstellen einer neuen DB-Instance in der Datenbank-Vorschauumgebung ... 4705
PostgreSQL-Versionen ... 4708
Veraltete PostgreSQL-Versionen ... 4708
Release-Prozess für RDS für PostgreSQL ... 4710
Vorteile des inkrementellen Release-Prozesses von RDS für PostgreSQL ... 4710
Verwaltung von Release-Updates ... 4710
Versionen der Postgre-Erweiterung SQL ... 4712
Einschränkung der Installation von Postgre-Erweiterungen SQL ... 4712
Vertrauenswürdige SQL Postgre-Erweiterungen ... 4714
Postgre-Funktionen SQL ... 4716
Benutzerdefinierte Datentypen und Aufzählungen ... 4717
Ereignisauslöser ... 4717
Riesige Seiten ... 4718
Logische Replikation durchführen ... 4719
RAMDiskette für das stats_temp_directory ... 4727

Tablespaces 4728
Kollationen für und andere Mainframe-Migrationen EBCDIC 4728
Verwaltung der logischen Steckplatzsynchronisierung 4734
Verbinden mit einer PostgreSQL-Instance 4736
Den PSQL-Client installieren 4737
Suchen der Verbindungsinformationen 4737
Wird verwendet pgAdmin , um eine Verbindung zu einer RDS for SQL Postgre-DB-Instance herzustellen 4740
Verwenden Sie psql, um eine Verbindung zu Ihrer RDS for Postgre-DB-Instance herzustellen SQL 4742
Mit dem Treiber wird eine Verbindung zu RDS for Postgre SQL hergestellt AWS JDBC 4744
SQLMit dem AWS Python-Treiber eine Verbindung zu RDS for Postgre herstellen 4744
Fehlerbehebung bei Verbindungen mit Ihrer RDS für PostgreSQL-Instance 4744
Verbindungen sichern mit/SSLTLS 4747
Verwenden von SSL mit einer PostgreSQL-DB-Instance 4747
Aktualisierung von Anwendungen zur Verwendung neuer SSL /-Zertifikate TLS 4755
Verwenden der Kerberos-Authentifizierung 4760
Verfügbarkeit von Regionen und Versionen 4761
Übersicht über die Kerberos-Authentifizierung 4761
Einrichtung 4762
Verwaltung eines RDS für PostgreSQL-DB-Instance) in einer Active Directory-Domäne 4777
Herstellen einer Verbindung mithilfe der Kerberos-Authentifizierung 4778
Verwendung eines benutzerdefinierten DNS Servers für ausgehenden Netzwerkzugriff 4782
Aktivieren der benutzerdefinierten DNS Auflösung 4782
Deaktivieren der benutzerdefinierten DNS Auflösung 4782
Einen benutzerdefinierten DNS Server einrichten 4782
Upgrades der PostgreSQL-DB-Engine 4785
Überlegungen 4787
Suche nach gültigen Upgrade-Zielen 4788
Postgre-Versionsnummern SQL 4789
RDSVersionsnummern 4790
Auswählen eines Hauptversions-Upgrades 4790
Durchführen eines Hauptversions-Upgrades 4800
Automatische Unterversion-Upgrades 4809
Aktualisierung von Postgre-Erweiterungen SQL 4812
Aktualisierung einer SQL Postgre-DB-Snapshot-Engine-Version 4814

Arbeiten mit Read Replicas in RDS for PostgreSQL 4817
Logische Dekodierung auf einer Read Replica 4817
Read Replica-Beschränkungen unter PostgreSQL 4821
Konfiguration von Read Replicas mit PostgreSQL 4822
Verwenden von kaskadierenden Lesereplikaten 4825
Erstellen von regionsübergreifenden kaskadierenden Read Replicas 4826
Funktionsweise der Replikation für verschiedene RDS-for-PostgreSQL-Versionen 4828
Überwachen und Optimieren des Replikationsprozesses 4833
Problembehandlung für RDS for PostgreSQL Read Replica 4836
Verbesserung der Abfrageleistung mit RDS Optimized Reads 4837
Übersicht über RDS-optimierte Lesevorgänge in PostgreSQL 4837
Anwendungsfälle 4838
Bewährte Methoden 4839
Die Verwendung von 4839
Überwachen 4840
Einschränkungen 4841
Daten in Postgre importieren SQL 4842
Import einer SQL Postgre-Datenbank aus einer Amazon-Instance EC2 4845
Verwenden des Befehls\ copy zum Importieren von Daten in eine Tabelle auf einer SQL Postgre-DB-Instance 4847
Importieren von Daten aus Amazon S3 in RDS für PostgreSQL 4849
Transport von SQL Postgre-Datenbanken zwischen 4869
Exportieren von PostgreSQL-Daten nach Amazon S3 4879
Installieren der Erweiterung 4880
Übersicht über das Exportieren zu S3 4881
Angeben des Amazon S3-Dateipfads für den Export 4882
Einrichten des Zugriffs auf einen Amazon S3-Bucket 4883
Exportieren von Abfragedaten mithilfe der Funktion aws_s3.query_export_to_s3 4888
Funktionsreferenz 4891
Fehlerbehebung beim Zugriff auf Amazon S3 4896
Aufrufen einer Lambda-Funktion von RDS for PostgreSQL 4897
Schritt 1: Konfigurieren ausgehender Verbindungen 4898
Schritt 2: Konfigurieren von IAM für Ihre Instance und Lambda 4899
Schritt 3: Installieren der Erweiterung 4901
Schritt 4: Verwenden von Lambda-Hilfsfunktionen 4901
Schritt 5: Aufrufen einer Lambda-Funktion 4902

Schritt 6: Erteilen von Berechtigungen für Benutzer ... 4904
Beispiele: Aufrufen von Lambda-Funktionen ... 4904
Fehlermeldungen von Lambda-Funktionen ... 4907
Lambda-Funktion und Parameterreferenz ... 4909
Häufige DBA-Aufgaben für RDS for PostgreSQL ... 4914
Kollationen werden in Postgre unterstützt RDS SQL ... 4915
Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen ... 4916
Behandlung toter Verbindungen in PostgreSQL ... 4932
Arbeiten mit der PostgreSQL-Selbstbereinigung ... 4935
Mechanismen der Protokollierung ... 4985
Verwalten temporärer Dateien mit PostgreSQL ... 4986
Verwenden von pgBadger für die Protokollanalyse mit PostgreSQL ... 4993
PGSnapper Für die Überwachung von PostgreSQL verwenden ... 4993
Arbeiten mit Parametern ... 4993
Optimierung mit Warteereignissen für RDS für PostgreSQL ... 5015
Grundlegende Konzepte für die Optimierung von RDS für PostgreSQL ... 5016
Warteereignisse von RDS für PostgreSQL ... 5021
Kunde: ClientRead ... 5023
Kunde: ClientWrite ... 5027
CPU ... 5029
io:BuffileRead und io:BuffileWrite ... 5036
E/A:DataFileRead ... 5044
IO: WALWrite ... 5053
IPC:Parallele Warteereignisse ... 5056
Lock:advisory ... 5063
Lock:extend ... 5066
Lock:Relation ... 5069
Lock:transactionid ... 5072
Lock:tuple ... 5076
LWLock:BufferMapping (LWLock:buffer_mapping) ... 5080
LWLock:BufferIO (IPC:BufferIO) ... 5083
LWLock:buffer_content (BufferContent) ... 5085
LWLock:lock_manager (:lockmanager) LWLock ... 5087
LWLock:pg_stat_statements ... 5093
Timeout:PgSleep ... 5097
Timeout:VacuumDelay ... 5098

Optimiert RDSfür Postgre SQL mit proaktiven Erkenntnissen von Amazon DevOps Guru 5101
Die Datenbank läuft seit langem inaktiv in Transaktionsverbindung 5101
Verwenden von PostgreSQL-Erweiterungen 5105
Verwenden von Funktionen von orafce 5106
Verwenden der Amazon RDS Delegated Extension Support für Postgre SQL 5108
Verwalten von Partitionen mit der Erweiterung pg_partman 5124
Wird pgAudit zum Protokollieren von Datenbankaktivitäten verwendet 5131
Planen der Wartung mit der Erweiterung pg_cron 5145
Verwenden von pglogical, um Daten zu synchronisieren 5156
Verwenden von „pgactive“ zur Erstellung der Aktiv-Aktiv-Replikation 5171
Reduzieren von überflüssigen Daten mit der Erweiterung pg_repack 5248
Aktualisierung und Verwendung PLV8 5254
Verwendung von PL/Rust zum Schreiben von Funktionen in der Sprache Rust 5257
Verwaltung von Geodaten mit Post GIS 5262
Unterstützte ausländische Daten-Wrapper in RDS SQL 5272
Verwenden der Erweiterung log_fdw 5272
Verwenden von postgres_fdw für den Zugriff auf externe Daten 5275
Arbeiten mit einer Datenbank von My SQL 5276
Arbeiten mit einer Oracle-Datenbank 5280
Arbeiten mit einer Serverdatenbank SQL 5284
Arbeiten mit Trusted Language Extensions für PostgreSQL 5288
Terminologie 5289
Anforderungen für die Verwendung von Trusted Language Extensions 5290
Einrichten von Trusted Language Extensions 5293
Übersicht über Trusted Language Extensions 5297
Erweiterungen erstellen TLE 5299
Löschen Ihrer TLE Erweiterungen aus einer Datenbank 5304
Deinstallieren von Trusted Language Extensions 5306
Verwenden von PostgreSQL-Haken mit Ihren TLE-Erweiterungen 5307
Verwendung benutzerdefinierter Datentypen in Trusted Language Extensions 5313
Funktionsreferenz für Trusted Language Extensions 5314
Hakenreferenz für Trusted Language Extensions 5328
Codebeispiele 5331
Grundlagen 5342
Hello Amazon RDS 5343
Erlernen der Grundlagen 5353

Aktionen 5473
Szenarien 5608
Erstellen Sie einen Tracker für Aurora-Serverless-Arbeitsaufgaben 5609
Serverless-Beispiele 5613
Herstellen einer Verbindung mit einer Amazon-RDS-Datenbank in einer Lambda-Funktion 5613
Sicherheit 5634
Datenbankauthentifizierung 5636
Passwortauthentifizierung 5637
IAM-Datenbankauthentifizierung 5637
Kerberos-Authentifizierung 5638
Passwortverwaltung mit RDS und Secrets Manager 5640
Einschränkungen 5640
Übersicht 5641
Vorteile 5642
Berechtigungen 5642
Durchsetzung des RDS 5643
Das Passwort für die DB-Instance wird verwaltet 5645
Das Passwort für die Tenant-Datenbank verwalten 5649
Verwaltungskennwort für Multi-AZ-DB-Cluster 5653
Rotierende Geheimnisse für die DB-Instance 5657
Rotierende Geheimnisse für Multi-AZ-DB-Cluster 5659
Geheime Details anzeigen 5661
Details zu Geheimnissen für Multi-AZ-DB-Cluster anzeigen 5665
Details zu Geheimnissen für Tenant-Datenbanken anzeigen 5669
Verfügbarkeit von Regionen und Versionen 5672
Datenschutz 5672
Datenverschlüsselung 5673
Richtlinie für den Datenverkehr zwischen Netzwerken 5706
Identity and Access Management 5708
Zielgruppe 5708
Authentifizierung mit Identitäten 5709
Verwalten des Zugriffs mit Richtlinien 5713
So RDS arbeitet Amazon mit IAM 5715
Beispiele für identitätsbasierte Richtlinien 5724
AWS verwaltete Richtlinien 5748
Richtlinienaktualisierungen 5756

Serviceübergreifende Confused-Deputy-Prävention ... 5781
IAM-Datenbankauthentifizierung ... 5784
Fehlersuche ... 5838
Protokollierung und Überwachung ... 5840
Compliance-Validierung ... 5843
Ausfallsicherheit ... 5844
Backup und Backup ... 5844
Replikation ... 5844
Failover ... 5845
Sicherheit der Infrastruktur ... 5846
Sicherheitsgruppen ... 5846
Öffentliche Zugänglichkeit ... 5846
VPC-Endpunkte (AWS PrivateLink) ... 5848
Überlegungen ... 1527
Verfügbarkeit ... 1527
Erstellen eines Schnittstellen-VPC-Endpunkts ... 1527
Erstellen einer VPC-Endpunktrichtlinie ... 1528
Bewährte Methoden für die Gewährleistung der Sicherheit ... 5852
Zugriffskontrolle mit Sicherheitsgruppen ... 5853
Überblick über VPC-Sicherheitsgruppen ... 5853
Sicherheitsgruppenszenario ... 5854
Erstellen einer VPC-Sicherheitsgruppe ... 5855
Verknüpfen mit einer DB-Instance ... 5856
Berechtigungen von Hauptbenutzerkonten ... 5856
Service-verknüpfte Rollen ... 5861
Berechtigungen von serviceverknüpften Rollen für Amazon RDS ... 5861
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom ... 5865
Servicebezogene Rollenberechtigungen für Amazon RDS Beta ... 5866
Servicebezogene Rolle für Amazon RDS Preview ... 5867
Amazon RDS mit Amazon verwenden VPC ... 5869
Arbeiten mit einer DB-Instance in einer VPC ... 5869
Aktualisierung der VPC für eine DB-Instance ... 5889
Szenarien für den Zugriff auf einen in einem VPC ... 5890
Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur) ... 5897
Tutorial: Erstellen einer VPC zur Verwendung mit einer DB-Instance (Dual-Stack-Modus) . 5905
Verschieben einer DB-Instance in eine VPC ... 5917

Kontingente und Beschränkungen 5920
Kontingente in Amazon RDS 5920
Benennungseinschränkungen in Amazon RDS 5927
Maximale Anzahl von Datenbankverbindungen 5928
Limits für Dateigrößen in Amazon RDS 5931
Fehlerbehebung 5932
Verbindung zur -DB-Instance kann nicht hergestellt werden 5932
Testen der DB-Instanceverbindung 5935
Fehlerbehebung bei der Verbindungsauthentifizierung 5936
Sicherheitsprobleme 5936
Fehlermeldung „Failed to retrieve account attributes, certain console functions may be impaired (Fehler beim Abrufen von Kontoattributen, bestimmte Konsolenfunktionen können beeinträchtigt sein).“ 5936
Problembehandlung bei inkompatiblem Netzwerkstatus 5937
Ursachen 5937
Auflösung 5937
Zurücksetzen des Besitzerpassworts der DB-Instance 5939
Ausfall oder Neustart einer DB-Instance 5939
Parameteränderungen wirken sich nicht aus 5941
Zu wenig Speicher für DB-Instance 5941
Es sind nicht genügend DB-Instances verfügbar 5943
Probleme mit freisetzbarem Speicher in RDS 5944
MySQL- und MariaDB-Probleme 5944
Maximale Anzahl von MySQL- und MariaDB-Verbindungen 5945
Diagnostizieren und Auflösen des Status "incompatible-parameters" für ein Speicherlimit .. 5945
Diagnose und Lösung bei Verzögerungen zwischen Read Replicas (Lesereplikaten) 5948
Diagnose und Lösung eines Fehlers bei einer MySQL oder MariaDB Read Replica 5950
Erstellen von Auslösern mit aktivierter Binärprotokollierung erfordert SUPER-Berechtigung 5952
Diagnose und Behebung von Wiederherstellungsfehlern point-in-time 5954
Fehler „Replication stopped (Replikation gestoppt)“ 5954
Erstellung von Read Replica fehlgeschlagen oder Replikationsunterbrechungen mit schwerwiegendem Fehler 1236 5955
Die Read Replica-Replikation kann die Metadatenstruktur nicht initialisieren 5956
Der Aufbewahrungszeitraum für Backups kann nicht auf 0 gesetzt werden 5956
Amazon RDS-API-Referenz 5957

Verwenden der Abfrage-API 5957
Abfrageparameter 5957
Authentifizierung von Abfrageanforderungen 5958
Fehlerbehebung bei Anwendungen 5958
Fehler bei Abrufen 5958
Tipps zur Problembehebung 5959
Dokumentverlauf 5960
Frühere Aktualisierungen 6154
AWS Glossar 6187
..... 6188

Was ist Amazon Relational Database Service (Amazon RDS)?

Amazon Relational Database Service (Amazon RDS) ist ein Webservice, der das Einrichten, Betreiben und Skalieren einer relationalen Datenbank in der AWS Cloud vereinfacht. Dieser Service bietet kostengünstige und anpassbare Kapazität für eine Branchenstandards entsprechende relationale Datenbank sowie die Verwaltung gängiger Datenbankaufgaben.

> **Note**
>
> Diese Anleitung behandelt Amazon RDS-Datenbank-Engines abgesehen von Amazon Aurora. Weitere Informationen zum Verwenden von Amazon Aurora finden Sie im Amazon Aurora-Benutzerhandbuch.

Wenn Sie mit AWS Produkten und Dienstleistungen noch nicht vertraut sind, können Sie sich anhand der folgenden Ressourcen weiterbilden:

- Einen Überblick über alle AWS Produkte finden Sie unter Was ist Cloud Computing?
- Amazon Web Services bietet eine Reihe von Datenbankdiensten an. Weitere Informationen zu den verschiedenen verfügbaren Datenbankoptionen in AWS finden Sie unter Auswählen eines AWS - Datenbankservice und Ausführen von Datenbanken in AWS.

Vorteile von Amazon RDS

Amazon RDS ist ein verwalteter Datenbankdienst. Es ist für die meisten Verwaltungsaufgaben verantwortlich. Durch den Wegfall langwieriger manueller Prozesse können Sie sich mit Amazon RDS auf Ihre Anwendung und Ihre Benutzer konzentrieren.

Amazon RDS bietet die folgenden Hauptvorteile gegenüber Datenbankbereitstellungen, die nicht vollständig verwaltet werden:

- Sie können Datenbank-Engines verwenden, mit denen Sie bereits vertraut sind: IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle Database und PostgreSQL.
- Amazon RDS verwaltet Backups, Software-Patching, automatische Fehlererkennung und Backup.

- Sie können automatisierte Sicherungen aktivieren oder manuell eigene Sicherungs-Snapshots erstellen. Mit diesen Sicherungen können Sie eine Datenbank wiederherstellen. Der Amazon RDS-Wiederherstellungsprozess arbeitet zuverlässig und effizient.
- Sie können eine hohe Verfügbarkeit mit einer primären DB-Instance und einer synchronen sekundären DB-Instance erreichen, auf die Sie bei Problemen ein Failover durchführen können. Sie können auch Lesereplikate verwenden, um die Leseskalierung zu erhöhen.
- Zusätzlich zur Sicherheit in Ihrem Datenbankpaket können Sie den Zugriff kontrollieren, indem Sie AWS Identity and Access Management (IAM) verwenden, um Benutzer und Berechtigungen zu definieren. Sie können Ihre Datenbanken auch schützen, indem Sie sie in einer virtuellen privaten Cloud speichern.

Vergleich der Zuständigkeiten mit Bereitstellungen bei Amazon EC2 und lokalen Bereitstellungen

Wir empfehlen Amazon RDS als Standardoption für die meisten relationalen Datenbankbereitstellungen. Die folgenden Alternativen haben den Nachteil, dass Sie mehr Zeit mit der Verwaltung von Software und Hardware verbringen müssen:

Bereitstellung vor Ort

Wenn Sie einen Server kaufen, erhalten Sie CPU, Arbeitsspeicher, Speicher und IOPS, alles zusammen gebündelt. Sie übernehmen die volle Verantwortung für den Server, das Betriebssystem und die Datenbanksoftware.

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) bietet skalierbare Rechenkapazität in der AWS Cloud. Anders als bei einem lokalen Server sind CPU, Arbeitsspeicher, Speicher und IOPS getrennt, sodass Sie sie unabhängig voneinander skalieren können. AWS verwaltet die Hardwareschichten, wodurch ein Teil des Verwaltungsaufwands für einen lokalen Datenbankserver entfällt.

Der Nachteil beim Betrieb einer Datenbank bei Amazon EC2 besteht darin, dass Sie anfälliger für Benutzerfehler sind. Wenn Sie beispielsweise das Betriebssystem oder die Datenbanksoftware manuell aktualisieren, können Sie versehentlich Ausfallzeiten der Anwendung verursachen. Möglicherweise verbringen Sie Stunden damit, jede Änderung zu überprüfen, um ein Problem zu identifizieren und zu beheben.

In der folgenden Tabelle werden die Verwaltungsmodelle für lokale Datenbanken EC2, Amazon und Amazon RDS verglichen.

Funktion	On-Premises-Verwaltung	EC2 Verwaltung von Amazon	Amazon RDS-Verwaltung
Anwendungsoptimierung	Customer	Customer	Customer
Skalierung	Customer	Customer	AWS
Hohe Verfügbarkeit	Customer	Customer	AWS
Datenbank-Backups	Customer	Customer	AWS
Patchen von Datenbanksoftware	Customer	Customer	AWS
Installieren der Datenbanksoftware	Customer	Customer	AWS
Patchen des Betriebssystems (OS)	Customer	Customer	AWS
Betriebssysteminstallation	Customer	Customer	AWS
Serverwartung	Customer	AWS	AWS
Hardware-Lebenszyklus	Customer	AWS	AWS
Strom, Netzwerk und Kühlung	Customer	AWS	AWS

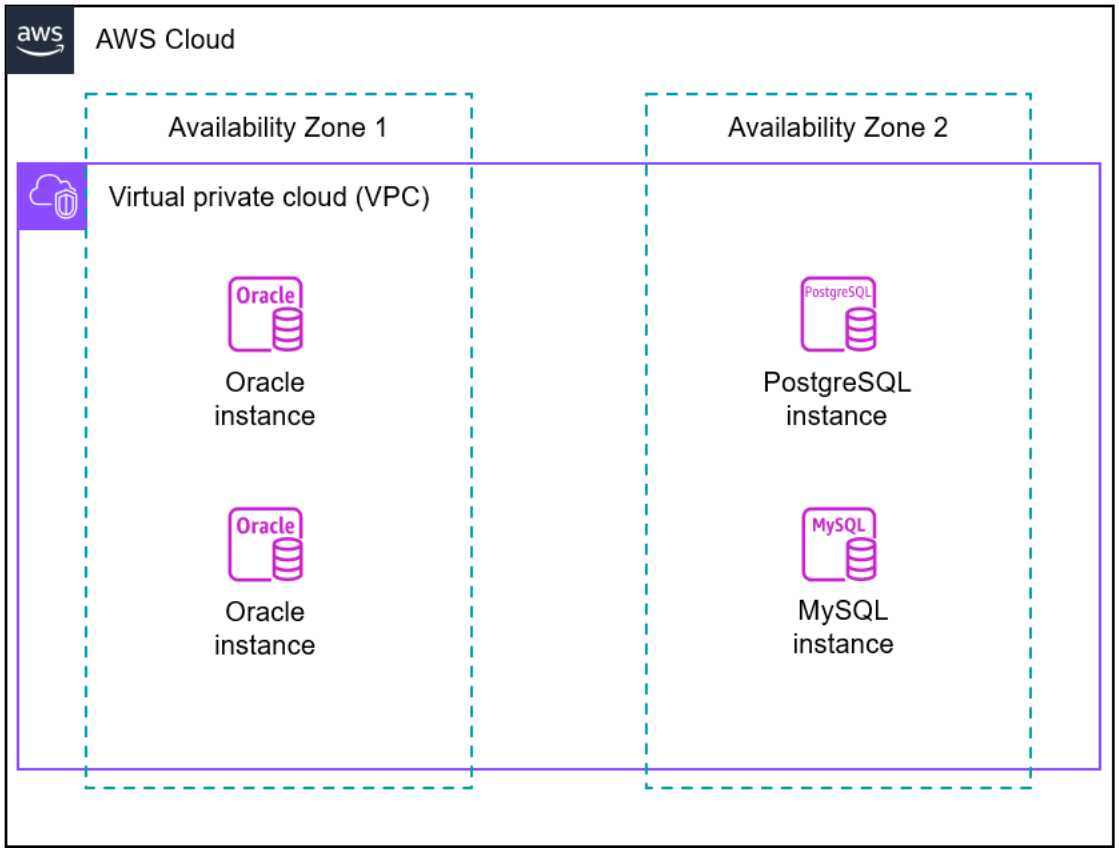
Amazon-RDS-Modell der geteilten Verantwortung

Amazon RDS ist für das Hosting der Softwarekomponenten und der Infrastruktur von DB-Instances und DB-Clustern verantwortlich. Sie sind für die Abfrageoptimierung verantwortlich, d. h. für den Prozess der Anpassung von SQL-Abfragen, um die Leistung zu verbessern. Die Abfrageleistung hängt erheblich vom Datenbankdesign, der Datengröße, der Datenverteilung, der Anwendungs-Workload und den Abfragemustern ab, die stark variieren können. Überwachung und Optimierung

sind hochgradig individualisierte Prozesse, für die Sie für Ihre RDS-Datenbanken verantwortlich sind. Sie können Erkenntnisse zur Amazon-RDS-Leistung und andere Tools verwenden, um problematische Abfragen zu identifizieren.

Amazon RDS DB-Instances

Eine DB-Instance ist eine isolierte Datenbankumgebung in der AWS Cloud. Die Grundbausteine für Amazon RDS sind Datenbank-Instances. Ihre DB-Instance kann mehrere von Benutzern erstellte Datenbanken enthalten. Das folgende Diagramm zeigt eine Virtual Private Cloud (VPC), die zwei Availability Zones enthält, wobei jede AZ zwei DB-Instances enthält.



Sie können auf Ihre DB-Instances zugreifen, indem Sie dieselben Tools und Anwendungen verwenden, die Sie mit einer eigenständigen Datenbank-Instance verwenden. Sie können eine DB-Instance erstellen und ändern, indem Sie die AWS Command Line Interface (AWS CLI), die Amazon RDS API oder die AWS Management Console verwenden.

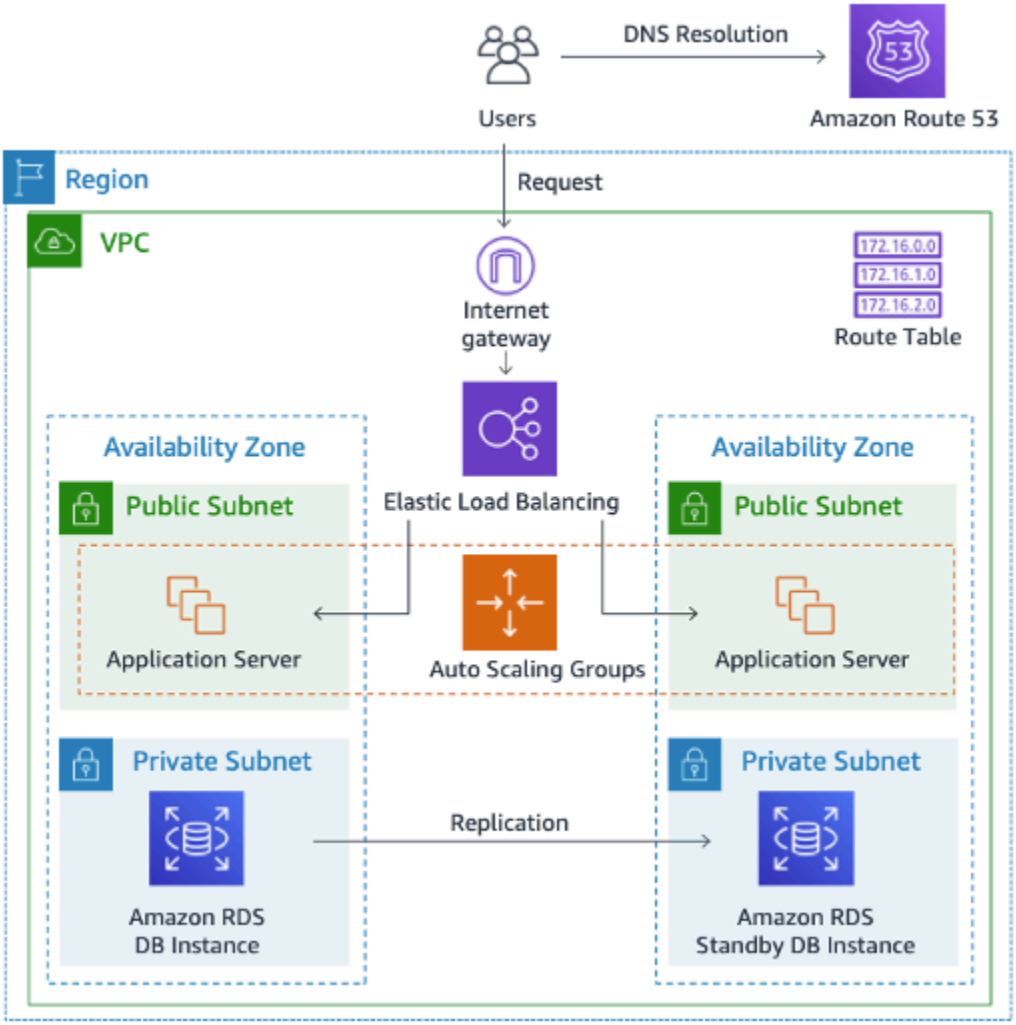
Themen

- Amazon RDS-Anwendungsarchitektur: Beispiel
- DB-Engine

- DB-Instance-Klassen
- DB-Instance-Speicher
- DB-Instances in einer Amazon Virtual Private Cloud (Amazon VPC)

Amazon RDS-Anwendungsarchitektur: Beispiel

Die folgende Abbildung zeigt einen typischen Anwendungsfall einer dynamischen Website, die Amazon RDS-DB-Instances als Datenbankspeicher verwendet:



Die Hauptkomponenten der vorherigen Architektur sind wie folgt:

Elastic Load Balancing

AWS leitet Benutzerdatenverkehr über Elastic Load Balancing weiter. Ein Load Balancer verteilt Workloads über mehrere Datenverarbeitungs-Ressourcen, wie z. B. virtuelle Server. In diesem

Beispielanwendungsfall leitet der Elastic Load Balancer Client-Anfragen an Anwendungsserver weiter.

Anwendungsserver

Anwendungsserver interagieren mit RDS-DB-Instances. Ein Anwendungsserver AWS wird in der Regel auf EC2 Instances gehostet, die skalierbare Rechenkapazität bieten. Die Anwendungsserver befinden sich in öffentlichen Subnetzen mit unterschiedlichen Availability Zones (AZs) innerhalb derselben Virtual Private Cloud (VPC).

RDS-DB-Instances

Die EC2 Anwendungsserver interagieren mit RDS-DB-Instances. Die DB-Instances befinden sich in privaten Subnetzen innerhalb verschiedener Availability Zones (AZs) innerhalb derselben Virtual Private Cloud (VPC). Da die Subnetze privat sind, sind keine Anfragen aus dem Internet zulässig.

Die primäre DB-Instance repliziert auf eine andere DB-Instance, die als Read Replica bezeichnet wird. Beide DB-Instances befinden sich in privaten Subnetzen innerhalb der VPC, was bedeutet, dass Internetnutzer nicht direkt auf sie zugreifen können.

DB-Engine

Eine DB-Engine ist die spezifische relationale Datenbanksoftware, die auf der DB-Instance ausgeführt wird. Amazon RDS unterstützt die folgenden Datenbank-Engines:

- IBM Db2

  Weitere Informationen finden Sie unter Amazon RDS für Db2.
- MariaDB

  Weitere Informationen finden Sie unter Amazon RDS für MariaDB.
- Microsoft SQL Server

  Weitere Informationen finden Sie unter Amazon RDS for Microsoft SQL Server.
- MySQL

  Weitere Informationen finden Sie unter Amazon RDS für mich SQL.
- Oracle Database

Weitere Informationen finden Sie unter Amazon RDS für Oracle.

- PostgreSQL

  Weitere Informationen finden Sie unter Amazon RDS für PostgreSQL.

Jede DB-Engine verfügt über eigene unterstützte Funktionen und jede Version einer DB-Engine kann bestimmte Funktionen enthalten. Die Support für Amazon RDS-Funktionen variiert AWS-Regionen je nach Version der einzelnen DB-Engine. Informationen zur Überprüfung der Funktionen in verschiedenen Engine-Versionen und Regionen finden Sie unter Unterstützte Funktionen in Amazon RDS by AWS-Region und DB Engine.

Darüber hinaus verfügt jede DB-Engine über eine Reihe von Parametern in einer DB-Parametergruppe, die das Verhalten der von ihr verwalteten Datenbanken steuert. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

DB-Instance-Klassen

Die DB-Instance-Klasse bestimmt die Berechnungs- und Speicherkapazität einer DB-Instance. Eine DB-Instance-Klasse besteht sowohl aus dem DB-Instance-Klassentyp als auch aus der -größe. Amazon RDS unterstützt die folgenden Instance-Klassentypen, wobei das Sternchen (*) für die Generation, das optionale Attribut und die Größe steht:

- Allgemeiner Zweck — db.m*
- Speicheroptimiert — db.z*, db.x*, db.r*
- Für Rechenleistung optimiert — db.c*
- Extrem hohe Leistung — db.t*

Jede Instance-Klasse bietet unterschiedliche Rechen-, Arbeitsspeicher- und Speicherfunktionen. Beispielsweise ist db.m7g ein universeller DB-Instance-Klassentyp der 7. Generation, der von Graviton3-Prozessoren angetrieben wird. AWS Wenn Sie eine DB-Instance erstellen, geben Sie eine DB-Instance-Klasse wie db.m7g.2xlarge an, wobei 2xlarge die Größe ist. Weitere Informationen zu den Hardwarespezifikationen für die verschiedenen Instance-Klassen finden Sie unter. Hardware-Spezifikationen für DB-Instance-Klassen

Sie können die DB-Instance-Klasse auswählen, die Ihren Anforderungen am besten entspricht. Wenn sich Ihre Anforderungen im Laufe der Zeit ändern, können Sie Ihre DB-Instance-Klasse ändern. Sie

könnten beispielsweise Ihre db.m7g.2xlarge-Instance auf db.m7g.4xlarge hochskalieren. Weitere Informationen finden Sie unter .

> **Note**
>
> Informationen zu den Preisen für DB-Instance-Klassen finden Sie im Abschnitt "Preise" auf der Amazon RDS-Produktseite.

DB-Instance-Speicher

Amazon EBS bietet Volumes für eine dauerhafte Speicherung auf Blockebene, die Sie einer ausgeführten Instance zuordnen können. DB-Instance-Speicher gibt es in den folgenden Typen:

- Allzweck (SSD)

  Dieser kostengünstige Speichertyp ist ideal für eine Vielzahl von Workloads, die auf mittelgroßen DB-Instances ausgeführt werden. Allzweck-Speicher eignet sich am besten für Entwicklungs- und Testumgebungen.
- Bereitgestellte IOPS (PIOPS)

  Dieser Speichertyp wurde entwickelt, um den I/O-intensive workloads, particularly database workloads, that require low I/O latency and consistent I/O Durchsatzanforderungen gerecht zu werden. Bereitgestellter IOPS-Speicher eignet sich am besten für Produktionsumgebungen.
- Magnetic

  Amazon RDS unterstützt Magnetspeicher aus Gründen der Abwärtskompatibilität. Wir empfehlen generell die Nutzung von Allzweck-SSD-Speicher oder SSD-Speicher mit bereitgestellten IOPS.

Die Speicherarten unterscheiden sich in Leistungsmerkmalen und Preis. Sie können Ihre Speicherleistung und Ihre Kosten an die Anforderungen Ihrer Datenbank anpassen.

Jede DB-Instance hat abhängig vom Speichertyp und der von ihr unterstützten Datenbank-Engine minimale und maximale Speicheranforderungen. Es ist wichtig, über genügend Speicher zu verfügen, damit Ihre Datenbanken Platz haben, um zu wachsen. Ausreichender Speicher stellt außerdem sicher, dass Funktionen für die DB-Engine Platz haben, um Inhalte und Protokolleinträge zu schreiben. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.

DB-Instances in einer Amazon Virtual Private Cloud (Amazon VPC)

Sie können eine DB-Instance in einer virtuellen privaten Cloud (VPC) mit dem Amazon Virtual Private Cloud (Amazon VPC)-Service laufen lassen. Wenn Sie eine VPC verwenden, haben Sie die Kontrolle über Ihre virtuelle Netzwerkumgebung. Sie können Ihren eigenen IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen und Zugriffskontrolllisten konfigurieren.

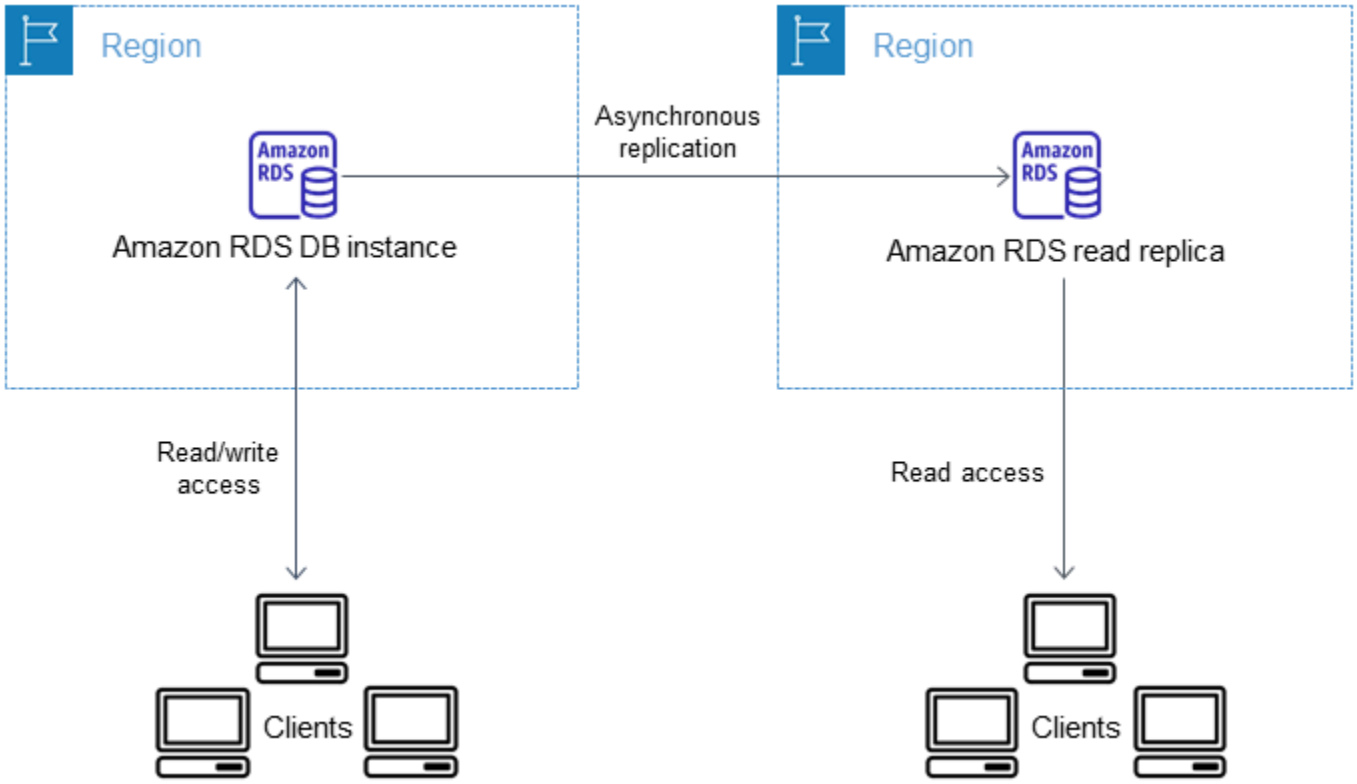
Die Grundfunktionalität von Amazon RDS bleibt dieselbe, egal ob sie in einer VPC läuft oder nicht. Amazon RDS verwaltet Backups, Software-Patching, automatische Fehlererkennung und Backup. Es fallen keine zusätzlichen Kosten für das Ausführen Ihrer DB-Instance in einer VPC an. Weitere Informationen zur Verwendung der Amazon VPC mit RDS finden Sie unter Amazon VPC und Amazon RDS.

Amazon RDS verwendet Network Time Protocol (NTP), um die Uhrzeit auf DB-Instances zu synchronisieren.

AWS-Regionen und Verfügbarkeitszonen

Amazon Cloud Computing-Ressourcen sind in hochverfügbaren Rechenzentren in verschiedenen Regionen der Welt untergebracht (zum Beispiel in Nordamerika, Europa oder Asien). Jeder Rechenzentrumsstandort wird als bezeichnet AWS-Region. Mit Amazon RDS können Sie Ihre DB-Instances in mehreren Regionen erstellen.

Das folgende Szenario zeigt eine RDS-DB-Instance in einer Region, die asynchron auf eine Standby-DB-Instance in einer anderen Region repliziert wird. Wenn eine Region nicht mehr verfügbar ist, ist die Instance in der anderen Region weiterhin verfügbar.



Availability Zones

Jede AWS Region enthält mehrere unterschiedliche Standorte, die als Availability Zones oder bezeichnet AZs werden. Jede Availability Zone ist so aufgebaut, dass sie von Fehlern in anderen Availability Zones nicht betroffen ist. Jede ist so konzipiert, dass sie kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben AWS Region bietet. Indem Sie DB-Instances in separaten Availability Zones starten, können Sie Ihre Anwendungen vor dem Ausfall eines einzelnen Standorts schützen. Weitere Informationen finden Sie unter Regionen, Availability Zones und Local Zones.

Multi-AZ-Bereitstellungen

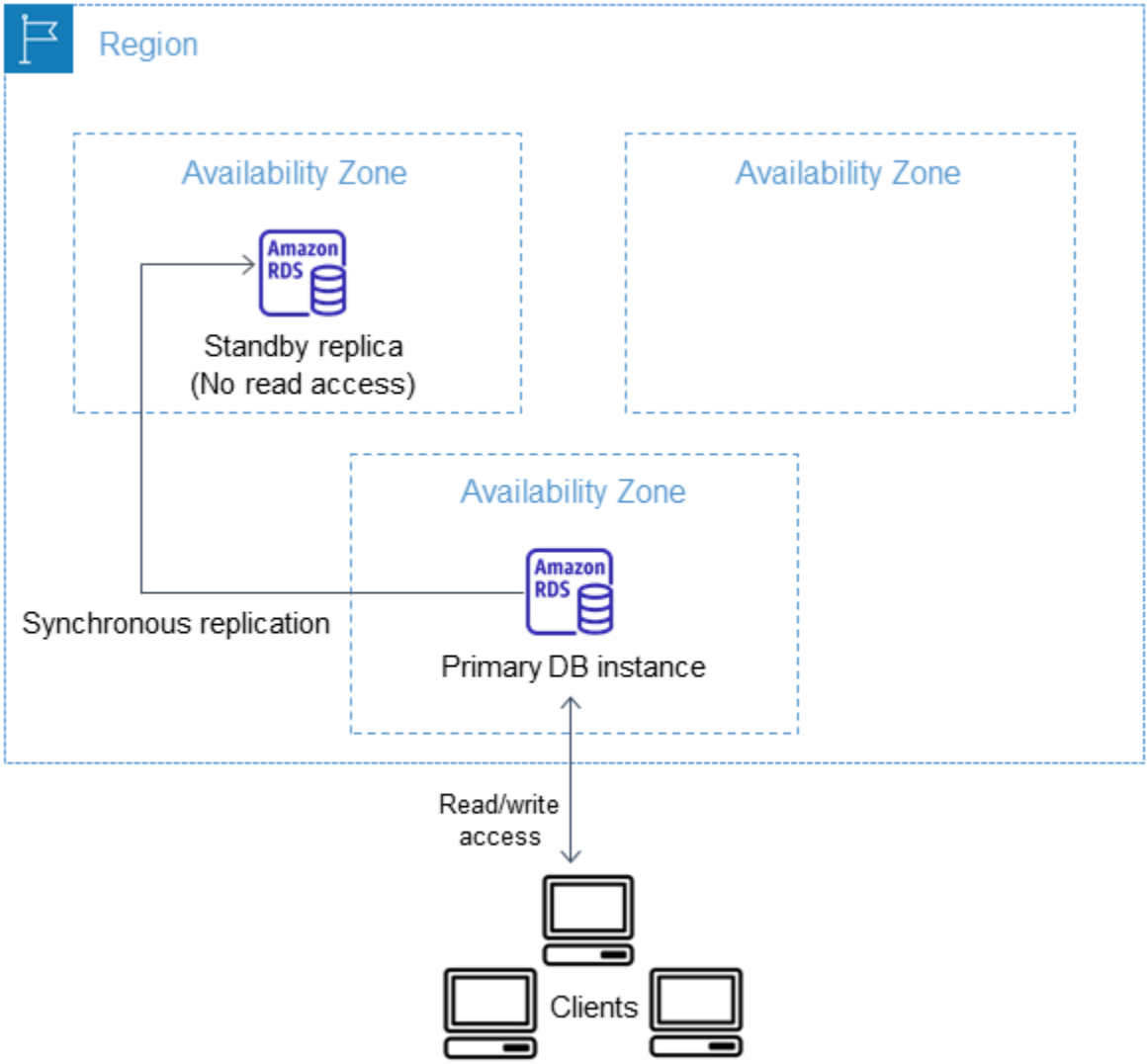
Sie können Ihre DB-Instance in mehreren Verfügbarkeitsbereichen ausführen, eine Option, die als Multi-AZ-Bereitstellung bezeichnet wird. Wenn Sie diese Option wählen, stellt Amazon automatisch eine oder mehrere sekundäre Standby-DB-Instances in einer anderen AZ bereit und verwaltet sie. Ihre primäre DB-Instance wird über die Availability Zones auf jede sekundäre DB-Instance repliziert.

Eine Multi-AZ-Bereitstellung bietet die folgenden Vorteile:

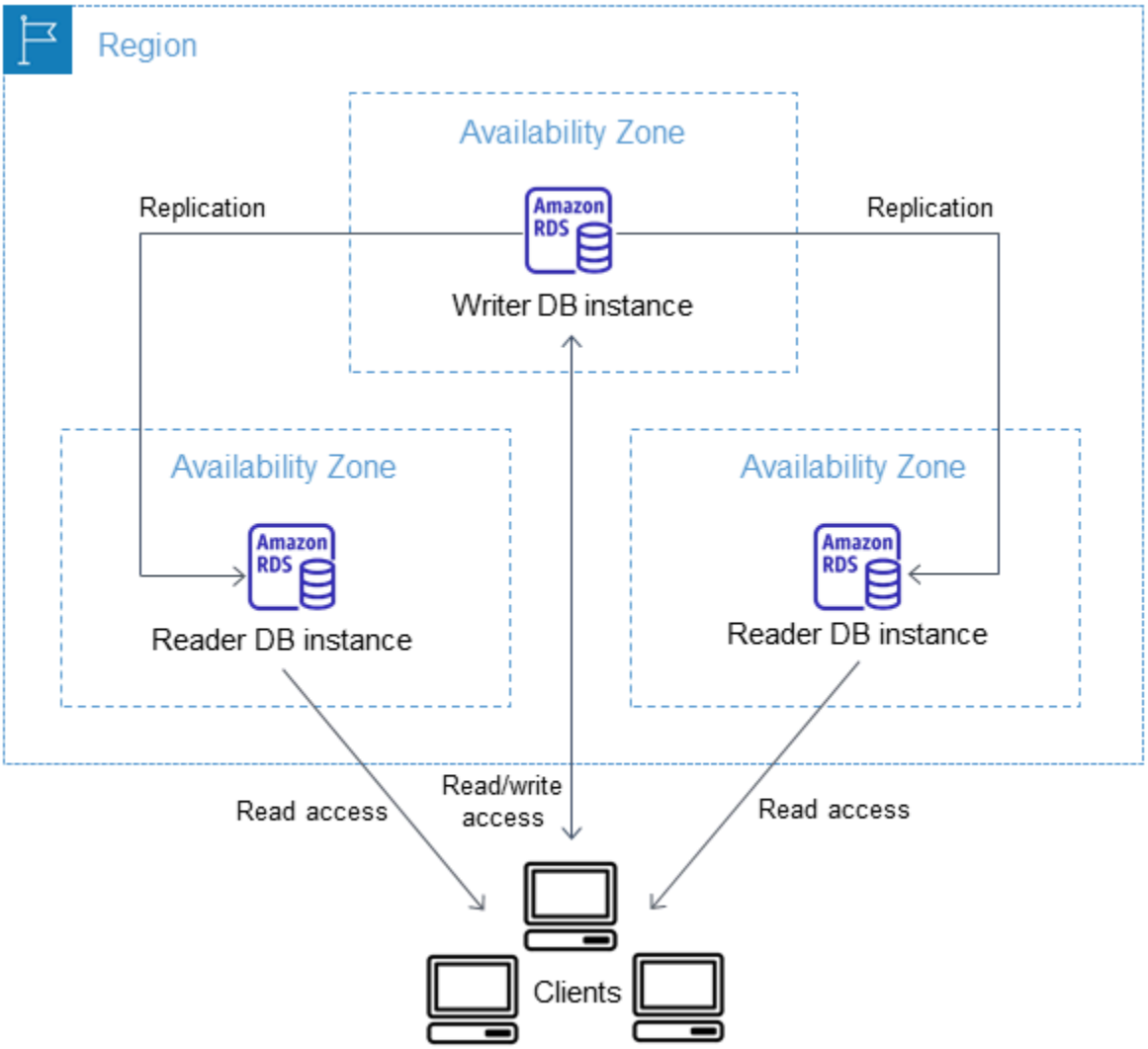
- Bereitstellung von Datenredundanz und Failover-Unterstützung

- Beseitigung von Einfrierungen I/O
- Minimierung von Latenzspitzen bei System-Backups
- Bereitstellung von Lesetraffic auf sekundären DB-Instances (nur Bereitstellung von Multi-AZ-DB-Clustern)

Das folgende Diagramm zeigt eine Multi-AZ-DB-Instance-Bereitstellung, bei der Amazon RDS automatisch ein synchrones Standby-Replikat in einer anderen Availability Zone bereitstellt und verwaltet. Die Replikatdatenbank verarbeitet keinen Leseverkehr.



Das folgende Diagramm zeigt eine Multi-AZ-DB-Cluster-Bereitstellung, die über eine Writer-DB-Instance und zwei Reader-DB-Instances in drei separaten Availability Zones derselben verfügt. AWS-Region Alle drei DB-Instances können Leseverkehr verarbeiten.



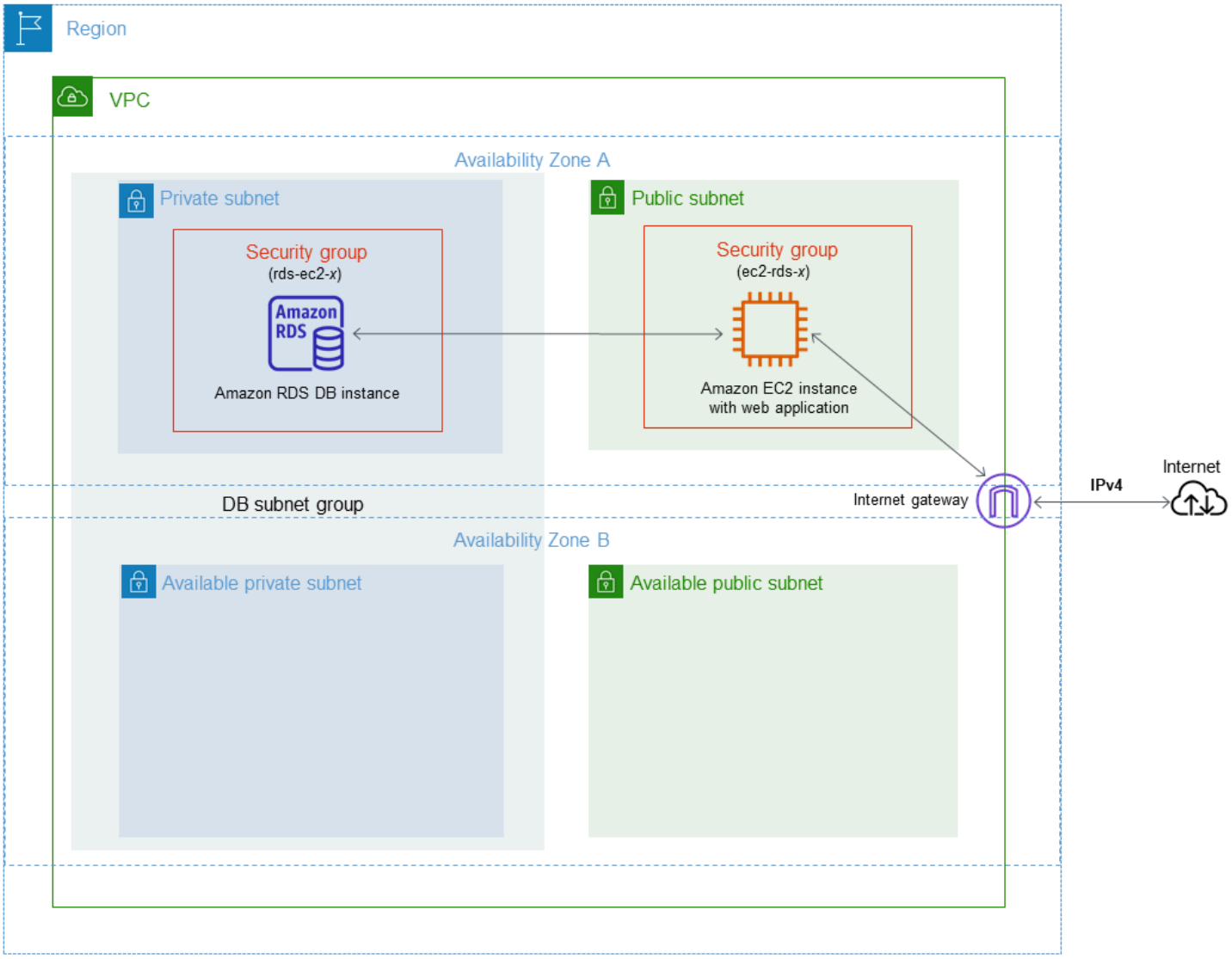
Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Zugriffskontrolle mit Sicherheitsgruppen

Eine Sicherheitsgruppe steuert den Zugriff auf eine DB-Instance, indem sie den Zugriff auf von Ihnen angegebene IP-Adressbereiche oder EC2 Amazon-Instances ermöglicht. Sie können eine Sicherheitsgruppe auf eine oder mehrere DB-Instances anwenden.

Eine DB-Instance in einer VPC wird häufig verwendet, um Daten mit einem Anwendungsserver in derselben VPC gemeinsam zu nutzen. Im folgenden Beispiel wird die VPC-Sicherheitsgruppe verwendet`ec2-rds-x`, um Regeln für eingehenden Datenverkehr zu definieren, die die IP-Adressen der Client-Anwendung als Quelle verwenden. Der Anwendungsserver gehört zu dieser Sicherheitsgruppe. Eine zweite Sicherheitsgruppe mit dem Namen `rds-ec2-x` gibt `ec2-rds-x`

als Quelle an und ist an eine RDS-DB-Instance angehängt. Gemäß den Sicherheitsgruppenregeln können Client-Anwendungen nicht direkt auf die DB-Instance zugreifen, aber die EC2 Instance kann auf die DB-Instance zugreifen.



Weitere Informationen zu Sicherheitsgruppen finden Sie unter Sicherheit in Amazon RDS.

Amazon-RDS-Überwachung

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon RDS, und Ihren anderen AWS Lösungen. AWS bietet verschiedene Überwachungstools, um Amazon RDS zu beobachten, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen.

Sie können die Leistung und den Zustand Ihrer DB-Instances mithilfe verschiedener automatisierter und manueller Tools verfolgen:

Status und Empfehlungen der Amazon RDS-DB-Instance

Mithilfe der Amazon RDS-Konsole oder der RDS-API können Sie sich Details zum aktuellen Status Ihrer Instance anzeigen lassen. AWS CLI Sie können auch auf automatisierte Empfehlungen für Datenbankressourcen wie DB-Instances, Read Replicas und DB-Parametergruppen reagieren. Weitere Informationen finden Sie unter Empfehlungen von Amazon RDS.

CloudWatch Amazon-Metriken für Amazon RDS

Sie können den CloudWatch Amazon-Service verwenden, um die Leistung und den Zustand einer DB-Instance zu überwachen. CloudWatch Leistungsdiagramme werden in der Amazon RDS-Konsole angezeigt. Amazon RDS sendet automatisch Metriken an CloudWatch jede Minute für jede aktive Datenbank. Es fallen keine zusätzlichen Gebühren für Amazon RDS-Metriken an CloudWatch.

Mithilfe von CloudWatch Amazon-Alarmen können Sie sich eine einzelne Amazon RDS-Metrik über einen bestimmten Zeitraum ansehen. Anschließend können Sie eine oder mehrere Aktionen basierend auf dem Wert der Metrik relativ zu einem von Ihnen festgelegten Schwellenwert ausführen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mit Amazon CloudWatch.

Amazon RDS Performance Insights und Betriebssystemüberwachung

Performance Insights bewertet die Auslastung Ihrer Datenbank und bestimmt, wann und wo Maßnahmen ergriffen werden müssen. Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS. Amazon RDS Enhanced Monitoring betrachtet Metriken für das Betriebssystem in Echtzeit. Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Integriert AWS-Services

Amazon RDS ist in Amazon EventBridge, Amazon CloudWatch Logs und Amazon DevOps Guru integriert. Weitere Informationen finden Sie unter Metriken in einer RDS Amazon-Instance überwachen.

Benutzeroberflächen zu Amazon RDS

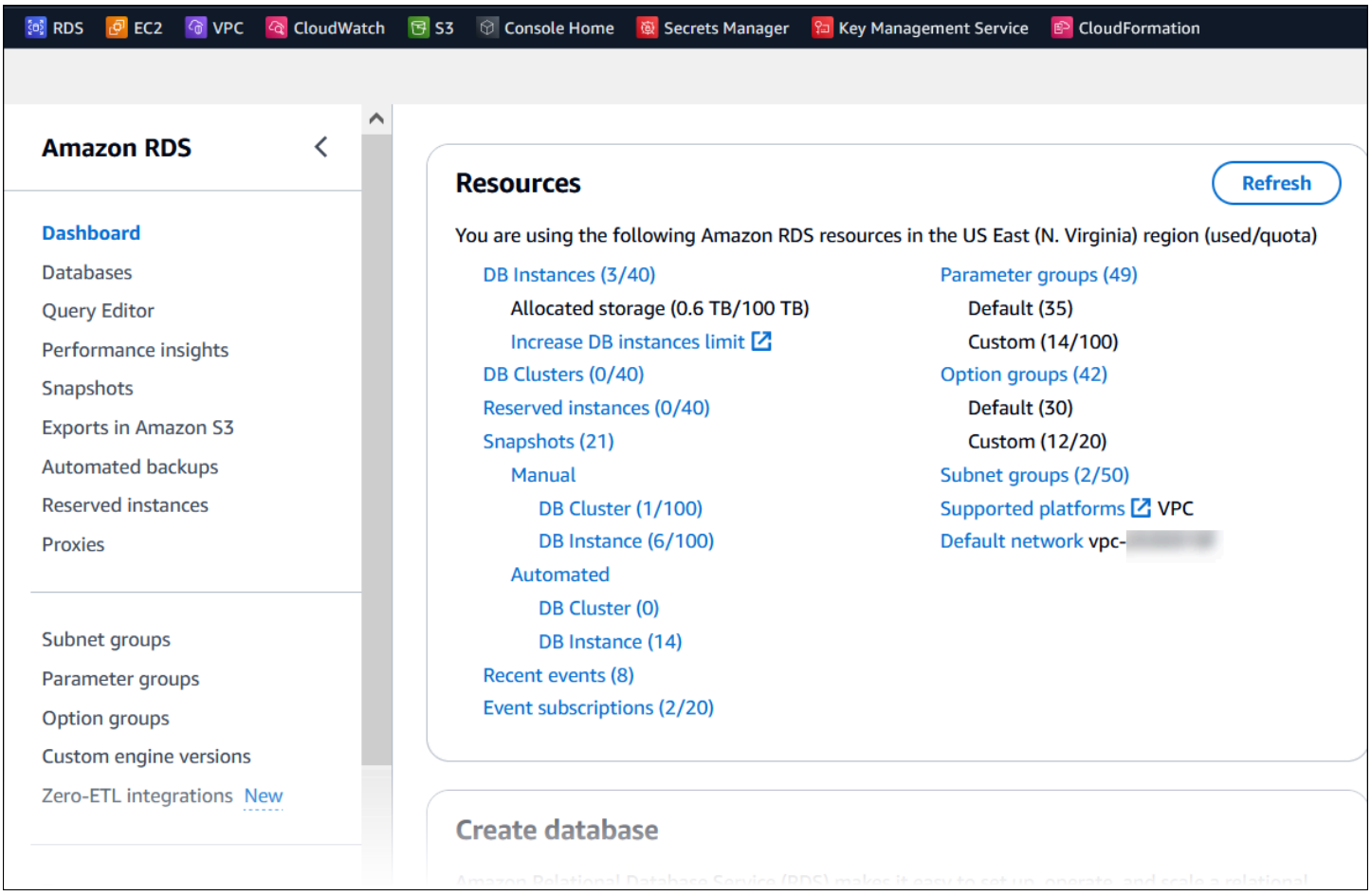
Sie können auf verschiedene Weise mit Amazon RDS interagieren.

Themen

- AWS Management Console
- Befehlszeilenschnittstelle
- Amazon RDS APIs

AWS Management Console

Das AWS Management Console ist eine einfache webbasierte Benutzeroberfläche. Sie können Ihre DB-Instances von der Konsole aus verwalten, ohne dass eine Programmierung erforderlich ist. Um auf die Amazon RDS-Konsole zuzugreifen, melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.



Befehlszeilenschnittstelle

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um interaktiv auf die Amazon RDS-API zuzugreifen. Informationen zur AWS CLI Installation von finden Sie unter Installation der AWS Befehlszeilenschnittstelle. Informationen zur Verwendung von AWS CLI für RDS finden Sie in der AWS Command Line Interface Referenz für Amazon RDS.

Amazon RDS APIs

Wenn Sie ein Entwickler sind, können Sie mithilfe von programmgesteuert auf Amazon RDS zugreifen. APIs Weitere Informationen finden Sie unter Amazon RDS-API-Referenz.

Für die Anwendungsentwicklung empfehlen wir, eines der AWS Software Development Kits () SDKs zu verwenden. Sie AWS SDKs verarbeiten grundlegende Details wie Authentifizierung, Wiederholungslogik und Fehlerbehandlung, sodass Sie sich auf Ihre Anwendungslogik konzentrieren können. AWS SDKs sind für eine Vielzahl von Sprachen verfügbar. Weitere Informationen finden Sie unter Tools für Amazon Web Services.

AWS bietet außerdem Bibliotheken, Beispielcode, Tutorials und andere Ressourcen, die Ihnen den Einstieg erleichtern. Weitere Informationen finden Sie unter Beispiel-Code und Bibliotheken.

Zahlungsmodell für Amazon RDS

Wenn Sie Amazon RDS verwenden, können Sie wählen, ob Sie On-Demand-DB-Instances oder reservierte DB-Instances verwenden möchten. Weitere Informationen finden Sie unter DB-Instance-Abrechnung für Amazon RDS.

Informationen zu Amazon RDS-Preisen finden Sie auf der Amazon RDS-Produktseite.

Kostenloses AWS-Kontingent auf Amazon RDS

Sie können es Kostenloses AWS-Kontingent auf Amazon RDS mit den folgenden Engines und DB-Instance-Klassen verwenden:

- Engine-Typen — MariaDB, MySQL, PostgreSQL oder SQL Server Express Edition
- DB-Instance-Klassen
  - t3.micro — Für alle Motortypen
  - t4g.micro — Für alle Engine-Typen außer SQL Server Express Edition

Sie können die folgenden Funktionen, Ressourcen und Aktionen in Kostenloses AWS-Kontingent Amazon RDS nicht verwenden:

- Optionen für benutzerdefinierte Amazon RDS-Engines
- Reservierte DB-Instances
- Andere Bereitstellungsoptionen als Single-AZ
- Snapshots migrieren
- Abfrage-Editor
- Aurora-Lesereplikate erstellen

Diese Einschränkungen können sich im Zuge der Weiterentwicklung des Kostenloses AWS-Kontingent Programms ändern. Weitere Informationen zu finden Sie Kostenloses AWS-Kontingent unter Erkunden Sie AWS Dienste mit Kostenloses AWS-Kontingent. Weitere Informationen zum kostenlosen Kontingent für Amazon RDS finden Sie unter Kostenloses Kontingent für Amazon RDS.

Wenn Sie vor dem 17. Juli 2025 Kostenloses AWS-Kontingent Kunde waren, können Sie das kostenlose Kontingent für Amazon RDS weiter nutzen, bis die 12 Monate Ihrer kostenlosen Nutzung abgelaufen sind. Während dieses Zeitraums haben Sie weiterhin Anspruch auf die folgende Nutzung:

- 750 Stunden pro Monat mit den Single-AZ-Instance-Klassen db.t3.micro und db.t4g.micro, auf denen MySQL, MariaDB oder PostgreSQL auf Amazon RDS ausgeführt werden
- 750 Stunden pro Monat einer db.t3.micro-Instance-Klasse, in der SQL Server Express Edition auf Amazon RDS ausgeführt wird

Nach dem 17. Juli 2025 nutzen alle neuen Kostenloses AWS-Kontingent Ressourcen, auf denen Sie neue Ressourcen erstellen, das neue AWS Angebot. Kostenloses AWS-Kontingent

Als nächstes

Der vorhergehende Abschnitt hat Ihnen die grundlegenden Infrastruktur-Komponenten von RDS vorgestellt. Was sollten Sie als nächstes tun?

Erste Schritte

Erstellen Sie eine DB-Instance mithilfe der Anleitungen in Erste Schritte mit Amazon RDS.

Themen für Datenbank-Engines

Sie können die für einen bestimmten Datenbank-Engine spezifischen Informationen in den folgenden Abschnitten erhalten:

- Amazon RDS für Db2
- Amazon RDS für MariaDB
- Amazon RDS for Microsoft SQL Server
- Amazon RDS für mich SQL
- Amazon RDS für Oracle
- Amazon RDS für PostgreSQL

Amazon RDS DB-Instances

Eine DB-Instance ist eine isolierte Datenbankumgebung, die in der Cloud ausgeführt wird. Sie ist der Grundbaustein für Amazon RDS. Eine DB-Instance kann mehrere benutzerseitig erstellte Datenbanken enthalten. Auf DB-Instances kann mit denselben Client-Tools und Anwendungen zugegriffen werden wie bei einer eigenständigen Datenbank-Instance. DB-Instances lassen sich einfach mit den AWS Befehlszeilentools, Amazon RDS-API-Operationen oder dem erstellen und ändern AWS Management Console.

> **Note**
>
> Amazon RDS unterstützt den Zugriff auf Datenbanken mit jeder beliebigen Standard-SQL-Client-Anwendung. Amazon RDS erlaubt keinen direkten Host-Zugriff außer mit RDS Custom.

Sie können bis zu 40 Amazon RDS-DB-Instances haben, mit folgenden Einschränkungen:

- 10 für jede SQL Server-Edition (Enterprise, Standard, Web und Express) im Rahmen des Modells "License-included".
- 10 für Oracle im Rahmen des Modells "license-included"
- 40 für Db2 im Rahmen des Lizenzmodells bring-your-own-license "" (BYOL)
- 40 für MySQL, MariaDB oder PostgreSQL.
- 40 für Oracle im Rahmen des Lizenzmodells "bring-your-own-license" (BYOL)

> **Note**
>
> Wenn Sie für Ihre Anwendung mehr DB-Instances benötigen, können Sie mithilfe dieses Formulars zusätzliche DB-Instances anfordern.

Jede DB-Instance hat eine DB-Instance-Kennung. Dieser vom Kunden bereitgestellte Name identifiziert die DB-Instance eindeutig, wenn sie mit der Amazon RDS-API und den AWS CLI Befehlen interagiert. Die DB-Instance-ID muss für diesen Kunden in einer AWS Region eindeutig sein.

Die DB-Instance-Kennung ist Teil des DNS-Hostnamens, der Ihrer Instance von RDS zugewiesen wurde. Wenn Sie beispielsweise db1 als DB-Instance-Kennung angeben, weist RDS automatisch einen DNS-Endpunkt für Ihre Instance zu. Ein Beispiel für einen Endpunkt ist *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, wobei *db1* Ihre Instance-ID ist.

Im Beispielendpunkt *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com *abcdefghijkl* ist die Zeichenfolge ein eindeutiger Bezeichner für eine bestimmte Kombination aus AWS-Region und AWS-Konto. Die Kennung *abcdefghijkl* im Beispiel wird intern von RDS generiert und ändert sich nicht für die angegebene Kombination aus Region und Konto. Somit teilen alle Ihre DB-Instances in dieser Region dieselbe feste Kennung. Beachten Sie die folgenden Merkmale der festen Kennung:

- Wenn Sie Ihre DB-Instance umbenennen, ist der Endpunkt anders, die feste Kennung ist jedoch identisch. Wenn Sie beispielsweise *db1* in *renamed-db1* umbenennen, ist der neue Instance-Endpunkt *renamed-db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com.
- Wenn Sie eine DB-Instance mit derselben DB-Instance-Kennung löschen und neu erstellen, ist der Endpunkt identisch.
- Wenn Sie dasselbe Konto verwenden, um eine DB-Instance in einer anderen Region zu erstellen, unterscheidet sich die intern generierte Kennung, da die Region unterschiedlich ist, wie in *db2*.*mnopqrstuvwx*.*us-west-1*.rds.amazonaws.com.

Jede DB Instance unterstützt eine Datenbank-Engine. Amazon RDS unterstützt derzeit Db2-, MySQL-, MariaDB-, PostgreSQL-, Oracle-, Microsoft SQL Server- und Amazon Aurora Aurora-Datenbank-Engines.

Wenn Sie eine DB-Instance erstellen, muss bei einigen Datenbank-Engines ein Datenbankname angegeben werden. Eine DB-Instance kann mehrere Datenbanken oder eine einzelne Oracle-Datenbank mit mehreren Schemas hosten. Die Angabe für den Datenbanknamen richtet sich nach der Datenbank-Engine:

- Für die Db2-Datenbank-Engine ist der Datenbankname der Name der Datenbank, die in Ihrer DB-Instance gehostet wird. Dies ist ein optionales Feld. Sie können später eine Datenbank erstellen, indem Sie die `rdsadmin.create_database` gespeicherte Prozedur aufrufen. Weitere Informationen finden Sie unter the section called “Erstellen einer Datenbank”.
- Bei MySQL- und MariaDB-Datenbank-Engines ist der Datenbankname der Name der Datenbank, die auf Ihrer DB-Instance gehostet wird. Wenn in einer DB-Instance mehrere Datenbanken gehostet werden, müssen sie in dieser Instance einen eindeutigen Namen haben.

- Bei der Oracle-Datenbank-Engine wird über den Datenbanknamen der Wert von ORACLE_SID festgelegt. Dieser Wert muss angegeben werden, wenn eine Verbindung zu der Oracle RDS-Instance hergestellt wird.
- Bei der Microsoft SQL Server-Datenbank-Engine wird der Parameter für den Datenbanknamen nicht unterstützt.
- Bei der PostgreSQL-Datenbank-Engine ist der Datenbankname der Name der Datenbank, die auf Ihrer DB-Instance gehostet wird. Die Angabe eines Datenbanknamens bei der Erstellung einer DB-Instance ist nicht obligatorisch. Wenn in einer DB-Instance mehrere Datenbanken gehostet werden, müssen sie in dieser Instance einen eindeutigen Namen haben.

Amazon RDS erstellt im Rahmen des Erstellungsprozesses ein Masterbenutzerkonto für die DB-Instance. Dieser Master-Benutzer verfügt über die Berechtigungen, um Datenbanken zu erstellen und über diesen Datenbanken CREATE-, DELETE-, SELECT-, UPDATE- und INSERT-Operationen auszuführen. Sie müssen das Masterbenutzerpasswort festlegen, wenn Sie eine DB-Instance erstellen. Sie können es jedoch jederzeit ändern AWS CLI, indem Sie die Amazon RDS-API-Operationen oder die verwenden AWS Management Console. Sie können Aufgaben wie das Ändern des Passworts für den Masterbenutzer oder die Verwaltung der Benutzer auch unter Verwendung von Standard-SQL-Befehlen durchführen.

> **Note**
>
> Diese Anleitung behandelt Nicht-Aurora-Amazon RDS-Datenbank-Engines. Weitere Informationen zum Verwenden von Amazon Aurora finden Sie im Amazon Aurora-Benutzerhandbuch.

Die DB-Instance-Klasse bestimmt die Berechnungs- und Speicherkapazität einer Amazon-RDS -DB-Instance. Die benötigte DB-Instance-Klasse richtet sich nach Ihren Rechen- und Speicheranforderungen.

Eine DB-Instance-Klasse besteht sowohl aus dem DB-Instance-Klassentyp als auch aus der -größe. Beispielsweise ist db.r6g ein speicheroptimierter DB-Instance-Klassentyp, der von Graviton2-Prozessoren angetrieben wird. AWS Innerhalb des db.r6g-Instance-Klassentyps ist db.r6g.2xlarge eine DB-Instance-Klasse. Die Größe dieser Klasse ist 2xlarge.

Weitere Informationen zu Preisen der Instance-Klassen erhalten Sie unter Amazon RDS-Preise.

Weitere Informationen zu DB-Instance-Klassentypen, unterstützten DB-Engines, unterstützten DB-Engines, Änderung Ihrer DB-Instance-Klasse AWS-Regionen, Konfiguration des Prozessors für RDS for Oracle oder Hardwarespezifikationen für DB-Instance-Klassen finden Sie in den folgenden Abschnitten.

Themen

- DB-Instance-Klassenarten
- Unterstützte DB-Engines für DB-Instance-Klassen
- Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen
- Ändern Ihrer DB-Instance-Klasse
- Konfigurieren des Prozessors für eine DB-Instance-Klasse in RDS für Oracle
- Hardware-Spezifikationen für DB-Instance-Klassen

DB-Instance-Klassenarten

Amazon RDS unterstützt DB-Instance-Klassen für die folgenden Anwendungsfälle:

- Allgemeine Zwecke
- RAM-optimiert
- Für Datenverarbeitung optimiert
- Spitzenlastleistung
- Optimierte Lesevorgänge

Weitere Informationen zu EC2 Amazon-Instance-Typen finden Sie unter Instance-Typen in der EC2 Amazon-Dokumentation.

Allzweck-Instance-Klassen

Die folgenden Allzweck-DB-Instance-Klassen sind verfügbar:

- db.m8g — Universell einsetzbare DB-Instance-Klassen, die auf Graviton4-Prozessoren basieren. AWS Diese Instance-Klassen bieten ausgewogene Rechen-, Arbeitsspeicher- und Netzwerkleistung für eine breite Palette universeller Workloads. Im Vergleich zu AWS Graviton3-basierten M7G-Instances der siebten Generation bieten diese neuen Klassen größere Instance-Größen mit bis zu dreimal mehr V und Arbeitsspeicher. CPUs

  Sie können eine DB-Instance so modifizieren, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton4-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.
- db.m7i — Universell einsetzbare DB-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der 4. Generation basieren. Die db.m7i-Instances sind SAP-zertifiziert und eignen sich ideal für die Unterstützung von Unternehmensanwendungen, die größere Instance-Größen oder eine hohe kontinuierliche CPU-Auslastung benötigen. Diese Instance-Klassen bieten ausgewogene Rechen-, Arbeitsspeicher- und Netzwerkleistung für eine breite Palette universeller Workloads. Dieser Instance-Klassentyp bietet bis zu 40.000 Mbit/s EBS-Bandbreite und bis zu 50 Gbit/s Netzwerkbandbreite. Die db.m7i-Instances bieten im Vergleich zu db.m6i-Instances ein um bis zu 15% besseres Preis-Leistungs-Verhältnis.
- db.m7g — Universell einsetzbare DB-Instance-Klassen, die auf Graviton3-Prozessoren basieren. AWS Diese Instance-Klassen bieten ausgewogene Rechen-, Arbeitsspeicher- und Netzwerkleistung für eine breite Palette universeller Workloads.

  Sie können eine DB-Instance so ändern, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton3-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.
- db.m6g — Universell einsetzbare DB-Instance-Klassen, die auf Graviton2-Prozessoren basieren. AWS Diese Instances bieten ausgewogene Rechen-, Arbeitsspeicher- und Netzwerkleistung für eine breite Palette universeller Workloads. Die db.m6gd-Instance-Klassen verfügen über lokalen SSD-Speicher NVMe auf Blockebene für Anwendungen, die lokalen Speicher mit hoher Geschwindigkeit und niedriger Latenz benötigen.

Sie können eine DB-Instance so ändern, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton2-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.

- db.m6i – Allzweck-Instance-Klassen, die mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation betrieben werden Diese Instances sind SAP-zertifiziert und eignen sich ideal für Workloads wie Backend-Server, die Unternehmensanwendungen unterstützen, Gaming-Server, Caching-Flotten und Anwendungsentwicklungsumgebungen. Die Instance-Klassen db.m6id und db.m6idn bieten bis zu 7,6 TB lokalen SSD-Speicher, während db.m6in reinen EBS-Speicher bietet. NVMe Die Klassen „db.m6in“ und „db.m6idn“ bieten bis zu 200 Gbit/s Netzwerkbandbreite.
- db.m5 – Allzweck-DB-Instance-Klassen der neuesten Generation, die ein ausgewogenes Verhältnis von Rechen-, Speicher- und Netzwerkressourcen bieten und für viele Anwendungen eine gute Wahl sind. Die Instance-Klasse db.m5d bietet basierten SSD-Speicher, der physisch mit dem Hostserver verbunden ist. NVMe Die Instance-Klassen db.m5 bieten mehr Rechenkapazität als die vorherigen Instance-Klassen db.m4. Sie werden vom AWS -Nitro System angetrieben, einer Kombination aus dedizierter Hardware und leichtem Hypervisor.
- db.m4 – Allzweck-DB-Instance-Klassen, die mehr Rechenkapazität bieten als die vorherigen Instance-Klassen db.m3.

  Für die DB-Engines von RDS für Oracle unterstützt Amazon RDS keine DB-Instance-Klassen vom Typ „db.m4" mehr. Wenn Sie zuvor DB-Instances von RDS für Oracle vom Typ „db.m4" erstellt haben, aktualisiert Amazon RDS diese DB-Instances automatisch auf gleichwertige DB-Instance-Klassen vom Typ „db.m5".

  Für die DB-Engines RDS for MariaDB, RDS for MySQL, RDS for SQL Server und RDS for PostgreSQL hat Amazon RDS den end-of-support Prozess für diese DB-Instance-Klasse nach dem folgenden Zeitplan gestartet. Für alle RDS-DB-Instances, die diese Instance-Klasse verwenden, empfehlen wir, so bald wie möglich ein Upgrade auf eine DB-Instance-Klasse der neueren Generation durchzuführen.

Aktion oder Empfehlung	Datum
Ab diesem Datum begann Amazon RDS, Instances , die db.m4 verwenden, automatisch auf die db.m5-Instance-Klasse der neueren Generation zu aktualisieren. Das Erstellen von DB-Instances	1. Juni 2024

Aktion oder Empfehlung	Datum
mithilfe der db.m4-Instance-Klasse wird nicht mehr unterstützt.	
Amazon RDS beendet die Unterstützung für db.m4.	31. Dezember 2024

- db.m3 – Allzweck-DB-Instance-Klassen, die mehr Rechenkapazität bieten als die vorherigen Instance-Klassen db.m1.

  Für die DB-Engines RDS for MariaDB, RDS for MySQL und RDS for PostgreSQL hat Amazon RDS den end-of-life Prozess für db.m3-DB-Instance-Klassen nach dem folgenden Zeitplan gestartet, der Upgrade-Empfehlungen enthält. Für alle RDS-DB-Instances, die db.m3-DB-Instance-Klassen verwenden, empfehlen wir, so bald wie möglich ein Upgrade auf eine DB-Instance-Klasse der höheren Generation durchzuführen.

Aktion oder Empfehlung	Datumsangaben
Sie können keine RDS-DB-Instances mehr erstellen , die die DB-Instance-Klassen db.m3 verwenden.	Jetzt
Amazon RDS startete automatische Upgrades von RDS-DB-Instances, die DB-Instance-Klassen vom Typ „db.m3" verwenden, auf DB-Instance-Klassen vom Typ „db.m5“.	1. Februar 2023

Typen arbeitsspeicheroptimierter Instance-Klassen

Die speicheroptimierte Z-Familie unterstützt die folgenden Instance-Klassen:

- db.z1d – Optimierte Instance-Klassen für speicherintensive Anwendungen. Diese Instance-Klassen bieten eine hohe Rechenkapazität und einen großen Arbeitsspeicher. Hochfrequente Z1D-Instances bieten eine konstante All-Core-Frequenz von bis zu 4,0. GHz

Die speicheroptimierte X-Familie unterstützt die folgenden Instance-Klassen:

- db.x2g — Instance-Klassen, die für speicherintensive Anwendungen optimiert sind und auf Graviton2-Prozessoren basieren. AWS Diese Instance-Klassen bieten niedrige Kosten pro GiB Speicher.

  Sie können eine DB-Instance so ändern, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton2-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.
- db.x2i – Optimierte Instance-Klassen für speicherintensive Anwendungen. Die Instance-Klassentypen db.x2iedn und db.x2idn werden von skalierbaren Intel-Xeon-Prozessoren der dritten Generation (Ice Lake) betrieben. Sie umfassen bis zu 3,8 TB lokalen NVMe SSD-Speicher, bis zu 100 Gbit/s Netzwerkbandbreite und bis zu 4 TiB (db.x2iden) oder 2 TiB (db.x2idn) Arbeitsspeicher. Der Typ db.x2iezn wird von skalierbaren Intel Xeon Prozessoren der zweiten Generation (Cascade Lake) mit einer All-Core-Turbofrequenz von bis zu 4,5 GHz und bis zu 1,5 TiB Speicher angetrieben.
- db.x1 – Optimierte Instance-Klassen für speicherintensive Anwendungen. Diese Instance-Klassen bieten einen der niedrigsten Preise pro GiB RAM unter den DB-Instance-Klassen und bis zu 1.952 GiB DRAM-basierten Instance-Speicher. Der Istance-Klassentyp db.x1e bietet bis zu 3.904 GiB DRAM-basierten Instance-Speicher.

Die speicheroptimierte R-Familie unterstützt die folgenden Typen von Instance-Klassen:

- db.r8g — Instanzklassen, die von Graviton4-Prozessoren unterstützt werden. AWS Diese Instance-Klassen eignen sich ideal für die Ausführung speicherintensiver Workloads in Open-Source-Datenbanken wie MySQL und PostgreSQL. Diese Instances bieten größere Instance-Größen mit bis zu dreimal mehr V CPUs und Arbeitsspeicher als die Graviton3-basierten db.r7g-Instances der siebten Generation. AWS

  Sie können eine DB-Instance so modifizieren, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton4-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.
- db.r7g — Instance-Klassen, die auf Graviton3-Prozessoren basieren. AWS Diese Instance-Klassen eignen sich ideal für die Ausführung speicherintensiver Workloads in Open-Source-Datenbanken wie MySQL und PostgreSQL.

  Sie können eine DB-Instance so ändern, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton3-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.

- db.r7i — Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der 4. Generation basieren. Diese Instanzklassen sind SAP-zertifiziert und eignen sich ideal für die Ausführung speicherintensiver Workloads in Open-Source-Datenbanken wie MySQL und PostgreSQL. Sie können eine DB-Instance so modifizieren, dass sie eine der DB-Instance-Klassen verwendet, die auf skalierbaren Intel Xeon Prozessoren der 4. Generation basieren. Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.
- db.r6g — Instance-Klassen, die auf Graviton2-Prozessoren basieren. AWS Diese Instance-Klassen eignen sich ideal für die Ausführung speicherintensiver Workloads in Open-Source-Datenbanken wie MySQL und PostgreSQL. Der Typ db.r6gd bietet lokalen SSD-Speicher auf Blockebene für Anwendungen, die lokalen NVMe Speicher mit hoher Geschwindigkeit und niedriger Latenz benötigen.

  Sie können eine DB-Instance so ändern, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton2-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.
- db.r6i – Instance-Klassen mit Unterstützung der skalierbaren Intel-Xeon-Prozessoren der 3. Generation. Diese Instanzklassen sind SAP-zertifiziert und eignen sich ideal für speicherintensive Workloads in Open-Source-Datenbanken wie MySQL und PostgreSQL. Die Instanzklassen db.r6id, db.r6in und db.r6idn haben ein Verhältnis von 8:1 und einen maximalen Arbeitsspeicher von 1 TiB. memory-to-vCPU Die Klassen db.r6id und db.r6idn bieten bis zu 7,6 TB direkt angeschlossenen SSD-Speicher, wohingegen db.r6in reinen EBS-Speicher bietet. NVMe Die Klassen „db.r6idn“ und „db.r6in“ bieten bis zu 200 Gbit/s Netzwerkbandbreite.
- db.r5b – Instance-Klassen, die für durchsatzintensive Anwendungen speicheroptimiert sind. db.r5b-Instances werden vom AWS Nitro-System unterstützt und bieten eine Bandbreite von bis zu 60 Gbit/s und 260.000 IOPS EBS-Leistung. Dies ist die schnellste Blockspeicherleistung aller Zeiten. EC2
- db.r5d — Instanzklassen, die für niedrige Latenz, sehr hohe I/O Zufallsleistung und hohen sequentiellen Lesedurchsatz optimiert sind.
- db.r5 – Optimierte Instance-Klassen für speicherintensive Anwendungen. Diese Instance-Klassen bieten eine verbesserte Netzwerk- -Leistung. Sie werden vom AWS Nitro System betrieben, einer Kombination aus dedizierter Hardware und leichtem Hypervisor.
- db.r4 – Instance-Klassen, die im Vergleich zu früheren db.r3-Instance-Klassen ein verbessertes Netzwerk bieten.

  Für die DB-Engines von RDS for Oracle hat Amazon RDS den end-of-life Prozess für db.r4-DB-Instance-Klassen nach dem folgenden Zeitplan gestartet, der Upgrade-Empfehlungen enthält. Für

RDS for Oracle DB-Instances, die db.r4-Instance-Klassen verwenden, empfehlen wir, so bald wie möglich ein Upgrade auf eine Instance-Klasse der höheren Generation durchzuführen.

Aktion oder Empfehlung	Datumsangaben
Sie können keine DB-Instances von RDS für Oracle mehr erstellen, die die DB-Instance-Klassen db.r4 verwenden.	Jetzt
Amazon RDS startete automatische Upgrades von DB-Instances von RDS für Oracle, die DB-Instance-Klassen vom Typ „db.r4" verwenden, auf DB-Instance-Klassen vom Typ „db.r5“.	17. April 2023

Für die DB-Engines RDS for MariaDB, RDS for MySQL, RDS for SQL Server und RDS for PostgreSQL hat Amazon RDS den end-of-support Prozess für diese DB-Instance-Klasse nach dem folgenden Zeitplan gestartet. Für alle RDS-DB-Instances, die diese Instance-Klasse verwenden, empfehlen wir, so bald wie möglich ein Upgrade auf eine DB-Instance-Klasse der neueren Generation durchzuführen.

Aktion oder Empfehlung	Datumsangaben
Ab diesem Datum begann Amazon RDS, Instances, die db.r4 verwenden, automatisch auf die db.r5-Instance-Klasse der neueren Generation zu aktualisieren. Das Erstellen von DB-Instances mithilfe der db.m4-Instance-Klasse wird nicht mehr unterstützt.	1. Juni 2024
Amazon RDS beendet die Unterstützung für db.r4.	31. Dezember 2024

- db.r3 – Instance-Klassen, die Speicheroptimierung bieten.

  Für die DB-Engines RDS for MariaDB, RDS for MySQL und RDS for PostgreSQL hat Amazon RDS den end-of-life Prozess für db.r3-DB-Instance-Klassen nach dem folgenden Zeitplan gestartet, der Upgrade-Empfehlungen enthält. Für alle RDS-DB-Instances, die db.r3-DB-Instance-Klassen

verwenden, empfehlen wir, so bald wie möglich ein Upgrade auf eine DB-Instance-Klasse der höheren Generation durchzuführen.

Aktion oder Empfehlung	Datumsangaben
Sie können keine RDS-DB-Instances mehr erstellen , die die DB-Instance-Klassen db.r3 verwenden.	Jetzt
Amazon RDS startete automatische Upgrades von RDS-DB-Instances, die DB-Instance-Klassen vom Typ „db.r3" verwenden, auf DB-Instance-Klassen vom Typ „db.r5“.	1. Februar 2023

Für die Datenverarbeitung optimierter Instance-Klassentyp

Die folgenden rechenoptimierten Instanzklassentypen sind verfügbar:

- db.c6gd — Instanzklassen, die sich ideal für die Ausführung fortgeschrittener rechenintensiver Workloads eignen. Diese Instance-Klassen werden von AWS Graviton2-Prozessoren unterstützt und bieten lokalen SSD-Speicher auf Blockebene für Anwendungen, NVMe die lokalen Speicher mit hoher Geschwindigkeit und niedriger Latenz benötigen.

  > **Note**
  >
  > Die c6gd-Instance-Klassen werden nur für Multi-AZ-DB-Cluster-Bereitstellungen unterstützt. Sie sind die einzige Instance-Klasse, die für Multi-AZ-DB-Cluster unterstützt wird, die diese Instance-Größe bieten. `medium` Weitere Informationen finden Sie unter the section called “Multi-AZ-DB-Cluster-Bereitstellungen”.

Instance-Klassen mit Spitzenlastleistung

Die folgenden DB-Instance-Klassentypen mit Spitzenlastleistung sind verfügbar:

- db.t4g — Allzweck-Instance-Klassen, die auf ARM-basierten Graviton2-Prozessoren basieren. AWS Diese Instance-Klassen bieten ein besseres Preis-Leistungs-Verhältnis als die DB-Instance-Klassen mit Spitzenlastleistung der vorherigen Generation für eine breite Palette von Allzweck-Workloads mit Spitzenleistung. Amazon RDS db.t4g-Instances sind für den unbegrenzten Modus

konfiguriert. Das bedeutet, dass sie gegen eine zusätzliche Gebühr während eines 24-Stunden-Zeitfensters Burst-Leistung über die Baseline hinaus bieten können.

Sie können eine DB-Instance so ändern, dass sie eine der DB-Instance-Klassen verwendet, die von Graviton2-Prozessoren unterstützt werden. AWS Führen Sie dazu die gleichen Schritte wie bei jeder anderen Änderung der DB-Instance aus.

- db.t3: Instance-Klassen, die ein Basisleistungsniveau bieten, mit der Möglichkeit, die volle CPU-Auslastung zu erreichen. Die db.t3-T3-Instances sind für den unbegrenzten Modus konfiguriert. Diese Instance-Klassen bieten mehr Rechenkapazität als die vorherigen db.t2-Instance-Klassen. Sie werden vom AWS -Nitro System angetrieben, einer Kombination aus dedizierter Hardware und leichtem Hypervisor.
- db.t2: Instance-Klassen, die ein Basisleistungsniveau bieten, mit der Möglichkeit, die volle CPU-Auslastung zu erreichen. Die db.t2-Instances sind für den Unlimited-Modus konfiguriert. Wir empfehlen, diese Instance-Klassen ausschließlich für das Entwickeln und Testen von Servern oder Nicht-Produktionsservern zu verwenden.

  Für die DB-Engines RDS for MariaDB, RDS for MySQL, RDS for SQL Server und RDS for PostgreSQL hat Amazon RDS den end-of-support Prozess für diese DB-Instance-Klasse nach dem folgenden Zeitplan gestartet. Für alle RDS-DB-Instances, die diese Instance-Klasse verwenden, empfehlen wir, so bald wie möglich ein Upgrade auf eine DB-Instance-Klasse der neueren Generation durchzuführen.

Aktion oder Empfehlung	Datumsangaben
Ab diesem Datum begann Amazon RDS, Instances, die db.t2 verwenden, automatisch auf die db.t3-Instance-Klasse der neueren Generation zu aktualisieren. Das Erstellen von DB-Instances mithilfe der db.t2-Instance-Klasse wird nicht mehr unterstützt.	1. Juni 2024
Amazon RDS beendet die Unterstützung für db.t2.	31. Dezember 2024

> **Note**
>
> Die DB-Instance-Klassen, die das AWS Nitro-System verwenden (db.m5, db.r5, db.t3), werden bei kombinierter Lese- und Schreiblast gedrosselt.

Informationen zu den Hardware-Spezifikationen für DB-Instance-Klassen finden Sie unter Hardware-Spezifikationen für DB-Instance-Klassen .

Optimiert: Liest Instance-Klassentypen

Folgende Instance-Klassentypen für optimierte Lesevorgänge sind verfügbar:

- db.r6gd — Instanzklassen, die auf Graviton2-Prozessoren basieren. AWS Diese Instance-Klassen eignen sich ideal für die Ausführung speicherintensiver Workloads und bieten lokalen SSD-Speicher auf Blockebene für Anwendungen, die lokalen NVMe Speicher mit hoher Geschwindigkeit und niedriger Latenz benötigen.
- db.r6id – Instance-Klassen mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation. Diese Instance-Klassen sind SAP-zertifiziert und eignen sich ideal für speicherintensive Workloads. Sie bieten einen maximalen Speicher von 1 TiB und bis zu 7,6 TB direkt angeschlossenen SSD-Speicher NVMe.

Unterstützte DB-Engines für DB-Instance-Klassen

Im Folgenden finden Sie DB-Engine-spezifische Überlegungen zu DB-Instance-Klassen:

Db2

Die Unterstützung von DB-Instance-Klassen variiert je nach Version und Edition von Db2. Weitere Informationen zur Unterstützung der Instance-Klasse nach Versionen und Editionen, siehe Amazon RDS für Db2-Instance-Klassen.

Microsoft SQL Server

Die Unterstützung von DB-Instance-Klassen variiert je nach Version und Edition von SQL Server. Weitere Informationen zur Unterstützung der Instance-Klasse nach Versionen und Editionen, siehe Unterstützung für DB-Instance-Klassen für Microsoft SQL Server.

Oracle

Die Unterstützung von DB-Instance-Klassen variiert je nach Version und Edition der Oracle-Datenbank. RDS for Oracle unterstützt zusätzliche speicheroptimierte Instance-Klassen. Diese Klassen haben Namen in der Form db.r5. *instance_size*.tpc .mem. *threads_per_core ratio* Informationen zur vCPU-Anzahl und Speicherzuweisung für jede optimierte Klasse finden Sie unter Unterstützte RDS für Oracle-DB-Instance-Klassen.

RDS Custom

Informationen zu den unterstützten DB-Instance-Klassen in RDS Custom finden Sie unter Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle und Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server.

In der folgenden Tabelle finden Sie Detailinformationen zu den unterstützten Amazon RDS-DB-Instance-Klassen für jede Amazon RDS-DB-Engine. Die Zelle für jede Engine enthält einen der folgenden Werte:

Ja

Die Instance-Klasse wird für alle Versionen der DB-Engine unterstützt.

Nein

Die Instance-Klasse wird für die DB-Engine nicht unterstützt.

specific-versions

Die Instance-Klasse wird nur für die angegebenen Datenbankversionen der DB-Engine unterstützt.

Amazon RDS verbietet regelmäßig Haupt- und Nebenversionen der DB-Engine. AWS-Regionen Möglicherweise bieten nicht alle Unterstützung für frühere Engine-Versionen. Informationen zu den aktuell unterstützten Versionen finden Sie in den Themen zu den einzelnen DB-Engines: Db2-Versionen, MariaDB-Versionen, Microsoft SQL Server-Versionen, MySQL-Versionen, Oracle-Versionenund PostgreSQL-Versionen.

Themen

- Unterstützte DB-Engines für allgemeine Instance-Klassen
- Unterstützte DB-Engines für speicheroptimierte Instance-Klassen
- Unterstützte DB-Engines für rechenoptimierte Instanzklassen
- Unterstützte DB-Engines für Instance-Klassen mit hoher Leistung
- Unterstützte DB-Engines für Instanzklassen mit optimierten Lesevorgängen

Unterstützte DB-Engines für allgemeine Instance-Klassen

Die folgenden Tabellen zeigen die unterstützten Datenbanken und Datenbankversionen für die Allzweck-Instance-Klassen.

db.m8g — Allzweck-Instanzklassen, die auf Graviton4-Prozessoren basieren AWS

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m8g.4 xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.2 xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.1 xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.1 xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
		höher und 10.4.29 und höher				höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.8x large	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.4x large	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.2x large	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m8g.xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m8g.large	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

db.m7i — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der 4. Generation basieren

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m7i.48xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, nur EE	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.24xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL,	PostgreSQL Version 17.1 und

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
					nur EE	höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.16xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, nur EE	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.12xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, nur EE	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.8xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, nur EE	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m7i.4xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.2xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.xlarge	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.m7i.large	Db2 11,5	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

db.m7g – Allzweck-Instance-Klassen mit AWS -Graviton3-Prozessoren

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.16 xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.m7g.12 xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.m7g.8x large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.m7g.4x large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.2x large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.m7g.xl arge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.m7g.la rge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen

db.m6g – Allzweck-Instance-Klassen mit AWS -Graviton2-Prozessoren

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6g.1(xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						12.7 und höhere 12-Versionen
db.m6g.1 xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.m6g.8 large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.m6g.4 large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.m6g.2 large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.m6g.xl arge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6g.large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen

db.m6gd — Allzweck-Instanzklassen, die auf Graviton2-Prozessoren und SSD-Speicher basieren AWS

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6gd.16xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL-Versionen 17, 16, 15 und 14; 13.7 und höhere 13-Versionen; und 13.4
db.m6gd.12xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL-Versionen 17, 16, 15 und 14; 13.7 und höhere 13-Versionen; und 13.4
db.m6gd.8xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen; 13.7 und höhere 13-Versionen; und 13.4

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6gd.4xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen; 13.7 und höhere 13-Versionen; und 13.4
db.m6gd.2xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen; 13.7 und höhere 13-Versionen; und 13.4
db.m6gd.xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen; 13.7 und höhere 13-Versionen; und 13.4
db.m6gd.large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen; 13.7 und höhere 13-Versionen; und 13.4

db.m6id — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren und SSD-Speicher der dritten Generation basieren

Instance-Klasse	Db	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
db.m6id.32xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.24xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.16xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.12xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.8xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7

Instance-Klasse	Db	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
						und höhere 13-Versionen
db.m6id.4xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.2xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6id.large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen

db.m6idn – Allzweck-Instance-Klassen mit skalierbaren Intel-Xeon-Prozessoren der dritten Generation, SSD-Speicher und Netzwerkoptimierung

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6idn.32xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.24xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.16xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.12xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.8xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.4xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
						Versionen und 13.7 und höhere 13-Versionen
db.m6idn.2xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.m6idn.large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Neir	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen

db.m6in — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der dritten Generation und Netzwerkoptimierung basieren

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6in.32xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und

Instance-Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
						höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.m6in.24xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.m6in.16xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.m6in.12xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen

Instance-Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.m6in.8xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.m6in.4xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.m6in.2xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen

Instance-Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.m6in.xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.m6in.large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen

db.m6i — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der dritten Generation basieren

Instance-Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Oracle	PostgreSQL
db.m6i.32xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.24xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.16 xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.12 xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.8x large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.4x large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.2x large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.xl arge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen
db.m6i.la rge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Oracle Databa 19c	Alle verfügbaren Versionen

db.m5d — Allzweck-Instance-Klassen, die auf Intel Xeon Platinum-Prozessoren und SSD-Speicher basieren

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.24xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.m5d.16xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.m5d.12xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.m5d.8xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.m5d.4xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.m5d.2xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
						und höhere 13-Versionen und 13.4
db.m5d.xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.m5d.large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

db.m5 — Allzweck-Instanzklassen 2,5 Intel Xeon Platinum-Prozessoren GHz

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MyS	Oracl	PostgreSQL
db.m5.24xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.m5.16xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.m5.12xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MyS	Oracl	PostgreSQL
db.m5.8xlarge	Nei	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.m5.4xlarge	Nei	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.m5.2xlarge	Nei	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.m5.xlarge	Nei	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.m5.large	Nei	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen

db.m4 — Allzweck-Instanzklassen mit Intel Xeon-Prozessoren

Instance-Klasse	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.16large	Ne	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veralte gekennzeichnet	Als veraltet gekennzeichnet

Instance-Klasse	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.10 large	Ne	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veralte gekennzeichnet	Als veraltet gekennzeichnet
db.m4.4x arge	Ne	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veralte gekennzeichnet	Als veraltet gekennzeichnet
db.m4.2x arge	Ne	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veralte gekennzeichnet	Als veraltet gekennzeichnet
db.m4.xla rge	Ne	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veralte gekennzeichnet	Als veraltet gekennzeichnet
db.m4.lar ge	Ne	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veralte gekennzeichnet	Als veraltet gekennzeichnet

db.m3 – Allzweck-Instance-Klassen

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.2xl arge	Nein	Nein	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.xlarge	Nein	Nein	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet
db.m3.large	Nein	Nein	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet
db.m3.medium	Nein	Nein	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet

Unterstützte DB-Engines für speicheroptimierte Instance-Klassen

Die folgenden Tabellen zeigen die unterstützten Datenbanken und Datenbankversionen für die speicheroptimierten Instance-Klassen.

db.z1d – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.z1d.1xlarge	Nein	Nein	Ja	Nein	Ja	Nein
db.z1d.6large	Nein	Nein	Ja	Nein	Ja	Nein
db.z1d.3large	Nein	Nein	Ja	Nein	Ja	Nein
db.z1d.2large	Nein	Nein	Ja	Nein	Ja	Nein

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.z1d.xlarge	Nein	Nein	Ja	Nein	Ja	Nein
db.z1d.large	Nein	Nein	Ja	Nein	Ja	Nein

db.x2g — speicheroptimierte Instanzklassen, die auf Graviton2-Prozessoren basieren AWS

Instance Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.x2g.1 xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen
db.x2g.1 xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen
db.x2g.8 large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen
db.x2g.4 large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen
db.x2g.2 large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und

Instance Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
						12.7 und höhere 12-Versio nen
db.x2g.x arge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen
db.x2g.l arge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen

db.x2idn – arbeitsspeicheroptimierte Instance-Klassen, die mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation betrieben werden

Instance-Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Oracle	PostgreSQL
db.x2idn. 32xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	nur Enterp e Editior	PostgreSQL 15-Versionen, 14.6 und 13.9
db.x2idn. 24xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	nur Enterp e Editior	PostgreSQL 15-Versionen, 14.6 und 13.9
db.x2idn. 16xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4	nur Enterp	PostgreSQL 15-Versionen, 14.6 und 13.9

Instance-Klasse	DI	MariaDB	Micros SQL Serve	MySQL	Oracle	PostgreSQL
				und 8.0	e Editio	

db.x2iedn — speicheroptimierte Instanzklassen mit lokaler NVMe Basis, unterstützt durch skalierbare Intel Xeon Prozessoren der dritten Generation SSDs

Instance-Klasse	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .32xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterpris e- und Standard-Edition, SQL Server 2014 12.00 und höher.	MySQL 8.4 und 8.0	nur Enterp e Editio	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.x2iedn .24xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterpris e- und Standard-Edition, SQL Server 2014 12.00 und höher.	MySQL 8.4 und 8.0	nur Enterp e Editio	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.x2iedn .16xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterpris e- und Standard-Edition, SQL Server	MySQL 8.4 und 8.0	nur Enterp e Editio	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

Instance-Klasse	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
			2014 12.00 und höher.			
db.x2iedn .8xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterprise- und Standard-Edition, SQL Server 2014 12.00 und höher.	MySQL 8.4 und 8.0	nur Enterprise Edition	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.x2iedn .4xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterprise- und Standard-Edition, SQL Server 2014 12.00 und höher.	MySQL 8.4 und 8.0	Enterprise Edition und Standa Edition (2SE2	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.x2iedn .2xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterprise- und Standard-Edition, SQL Server 2014 12.00 und höher.	MySQL 8.4 und 8.0	Enterprise Edition und Standa Edition (2SE2	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

Instance-Klasse	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nur Enterprise- und Standard-Edition, SQL Server 2014 12.00 und höher.	MySQL 8.4 und 8.0	Enterp e Editior und Standa Editior (2SE2	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

db.x2iezn — arbeitsspeicheroptimierte Instance-Klassen, die von skalierbaren Intel-Xeon-Prozessoren der 2. Generation betrieben werden

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iezn .8xlarge	Nein	Nein	Nein	Nein	nur Enterprise Edition	Nein
db.x2iezn .6xlarge	Nein	Nein	Nein	Nein	nur Enterprise Edition	Nein
db.x2iezn .4xlarge	Nein	Nein	Nein	Nein	Enterprise Edition und Standard Edition 2 () SE2	Nein
db.x2iezn .2xlarge	Nein	Nein	Nein	Nein	Enterprise Edition und Standard Edition (2SE2)	Nein

db.x1e – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1e.32xlarge	Nein	Nein	Ja	Nein	Veraltet (1)	Nein
db.x1e.16xlarge	Nein	Nein	Ja	Nein	1 ist veraltet	Nein
db.x1e.8xlarge	Nein	Nein	Ja	Nein	1 ist veraltet	Nein
db.x1e.4xlarge	Nein	Nein	Ja	Nein	1 ist veraltet	Nein
db.x1e.2xlarge	Nein	Nein	Ja	Nein	Veraltet 1	Nein
db.x1e.xlarge	Nein	Nein	Ja	Nein	1 ist veraltet	Nein

[1] Sie können mit der X1-Instance-Klassenfamilie kein RDS mehr für Oracle-DB-Instances erstellen. Wenn Sie derzeit X1-Klassen verwenden, wechseln Sie so bald wie möglich zu einer Instance-Klasse der neuen Generation. Ab dem 22. Januar 2025 beginnt RDS mit automatisierten Upgrades in Ihrem definierten Wartungsfenster. Während des Upgrades wählt RDS den entsprechenden X2IEDN-Instanztyp und aktualisiert ihn. Weitere Informationen finden Sie im re:Post-Artikel Amazon RDS for Oracle beendet den Support für X1-Datenbankinstanzen am 22. Januar 2025.

db.x1 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1.32xlarge	Nein	Nein	Ja	Nein	Veraltet (1)	Nein
db.x1.16xlarge	Nein	Nein	Ja	Nein	1 ist veraltet	Nein

[1] Sie können mit der X1-Instance-Klassenfamilie kein RDS mehr für Oracle-DB-Instances erstellen. Wenn Sie derzeit X1-Klassen verwenden, wechseln Sie so bald wie möglich zu einer Instance-Klasse der neuen Generation. Ab dem 22. Januar 2025 beginnt RDS mit automatisierten Upgrades in Ihrem definierten Wartungsfenster. Während des Upgrades wählt RDS den entsprechenden X2IEDN-Instanztyp und aktualisiert ihn. Weitere Informationen finden Sie im re:Post-Artikel Amazon RDS for Oracle beendet den Support für X1-Datenbankinstanzen am 22. Januar 2025.

db.r8g — speicheroptimierte Instance-Klassen, die auf Graviton4-Prozessoren basieren AWS

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r8g.48xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r8g.24xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r8g.16xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r8g.12xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r8g.8xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r8g.4xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r8g.2xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r8g.xlarge	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r8g.large	Nein	MariaDB 11.4.3 und höher, 10.11.7 und höher, 10.6.13 und höher, 10.5.20 und höher und 10.4.29 und höher	Nein	MySQL 8.0.32 und höher	Nein	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

db.r7i — speicheroptimierte Instanzklassen, die auf skalierbaren Intel Xeon Prozessoren der 4. Generation basieren

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.r7i.48xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.24xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL	PostgreSQL Version 17.1 und

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
						höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.16xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.12xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.8xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.r7i.4xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.2xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.xlarge	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher
db.r7i.large	Db 11	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.0.32 und höher	Nur BYOL, alle Edition	PostgreSQL Version 17.1 und höher, 16.1 und höher, 15.4 und höher, 14.9 und höher und 13.11 und höher

db.r7g — speicheroptimierte Instanzklassen, die auf Graviton3-Prozessoren basieren AWS

Instance Klasse	DB	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r7g.1 xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.r7g.1 xlarge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.r7g.8 large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.r7g.4 large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.r7g.2 large	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen
db.r7g.xl arge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-

Instance Klasse	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						Versionen und 13.4 und höhere 13-Versionen
db.r7g.la rge	Ne	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.4 und höhere 13-Versionen

db.r6g — speicheroptimierte Instanzklassen, die auf Graviton2-Prozessoren basieren AWS

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6g.16 xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r6g.12 xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r6g.8x large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r6g.4x large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13;

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						und 12.7 und höhere 12-Versionen
db.r6g.2x large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r6g.xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r6g.large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen

db.r6gd — speicheroptimierte Instanzklassen, die auf Graviton2-Prozessoren basieren AWS

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.16xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.12xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versio

Instance-Klasse	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						nen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.8 xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.4 xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.2 xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.x large	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.l arge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

db.r6id – arbeitsspeicheroptimierte Instance-Klassen, die mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation betrieben werden

Instance-Klasse	Db2	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.32xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.24xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.16xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.12xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.8xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-

Instance-Klasse	Db2	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
					und BYOL	Versionen und 13.7 und höhere 13-Versio nen
db.r6id.4 xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen
db.r6id.2 xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen
db.r6id.x large	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen
db.r6id.l arge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen

db.r6idn – arbeitsspeicheroptimierte Instance-Klassen, die mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation betrieben werden

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6idn.32xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6idn.24xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6idn.16xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6idn.12xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6idn.8xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6idn.4xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6idn.2xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6idn.xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen

db.r6in – arbeitsspeicheroptimierte Instance-Klassen, die mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation betrieben werden

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.32xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.24xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.r6in.16xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.r6in.12xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.r6in.8xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYO	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.4xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.r6in.2xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.r6in.xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen
db.r6in.large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYC	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.3 und höhere 14-Versionen, 13.7 und höhere 13-Versionen, 12.11 und höhere 12-Versionen und 11.16 und höhere 11-Versionen

db.r6i — speicheroptimierte Instanzklassen, die für viel Arbeitsspeicher, Speicherplatz und I/O vorkonfiguriert sind

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.8xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Nur EE	Nein
db.r6i.8xlarge.tpc2.mem3x	Nein	Nein	Nein	Nein	Nur EE	Nein
db.r6i.6xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Nur EE	Nein
db.r6i.4xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.4xlarge.tpc2.mem3x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.4xlarge.tpc2.mem2x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.2xlarge.tpc2.mem8x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.2xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r6i.2xlarge.tpc1.mem2x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.xlarge.tpc2.mem2x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein
db.r6i.large.tpc1.mem2x	Nein	Nein	Nein	Nein	EE und BYOL SE2	Nein

db.r6i – arbeitsspeicheroptimierte Instance-Klassen

Instance Klasse	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.3 xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.2 xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen,

Instance Klasse	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
						12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.1(xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.1: xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.8: large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen

Instance Klasse	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.4x large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.2x large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.xlarge	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen
db.r6i.large	Ja	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16-, 15- und 14-Versionen, 13.4 und höhere 13-Versionen, 12.8 und höhere 12-Versionen, 11.13 und höhere 11-Versionen und 10.21 und höhere 10-Versionen

db.r5d – arbeitsspeicheroptimierte Instance-Klassen

Instance Klasse	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r5d.2 xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r5d.1 xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r5d.1 xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r5d.8 large	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r5d.4 large	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r5d.2 large	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7

Instance Klasse	D	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
						und höhere 13-Versionen und 13.4
db.r5d.x arge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r5d.la rge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

db.r5b – arbeitsspeicheroptimierte Instance-Klassen, die für hohen Arbeitsspeicher, Speicher und I/O vorkonfiguriert sind

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.8xlarge.tpc2.mem3x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.6xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.4xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Ja	Nein

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.4xlarge.tpc2.mem3x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.4xlarge.tpc2.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.2xlarge.tpc2.mem8x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.2xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.2xlarge.tpc1.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.xlarge.tpc2.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5b.large.tpc1.mem2x	Nein	Nein	Nein	Nein	Ja	Nein

db.r5b – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r5b.24xlarge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r5b.16xlarge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r5b.12xlarge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r5b.8xlarge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	>Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r5b.4xlarge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.r5b.2xlarge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7

Instance-Klasse	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
						und höhere 12-Versio nen
db.r5b.xl arge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen
db.r5b.la rge	Nei	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Ja	MySQL 8.4 und 8.0	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versio nen

db.r5 – arbeitsspeicheroptimierte Instance-Klassen, die für hohen Arbeitsspeicher, Speicher und I/O vorkonfiguriert sind

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.12xlarge.tpc 2.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.8xlarge.tpc2 .mem3x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.6xlarge.tpc2 .mem4x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.4xlarge.tpc2 .mem4x	Nein	Nein	Nein	Nein	Ja	Nein

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.4xlarge.tpc2.mem3x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.4xlarge.tpc2.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.2xlarge.tpc2.mem8x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.2xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.2xlarge.tpc1.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.xlarge.tpc2.mem4x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.xlarge.tpc2.mem2x	Nein	Nein	Nein	Nein	Ja	Nein
db.r5.large.tpc1.mem2x	Nein	Nein	Nein	Nein	Ja	Nein

db.r5 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaD	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.r5.24xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen

Instance-Klasse	Db2	MariaD	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.r5.16xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.r5.12xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.r5.8xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.r5.4xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.r5.2xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.r5.xlarge	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen
db.r5.large	Nein	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12 und 11; 10.17 und höhere 10-Versionen; und 9.6.22 und höhere 9-Versionen

db.r4 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r4.16xlarge	Nein	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet
db.r4.8xlarge	Nein	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet
db.r4.4xlarge	Nein	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet
db.r4.2xlarge	Nein	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet
db.r4.xlarge	Nein	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet
db.r4.large	Nein	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet	Als veraltet gekennzeichnet

db.r3 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQ	Oracle	PostgreSQL
db.r3.8xlarge**	Nei	Alle MariaDB-Versionen 10.6, 10.5, 10.4 und 10.3	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet
db.r3.4xlarge	Nei	Alle MariaDB-Versionen 10.6, 10.5, 10.4 und 10.3	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet
db.r3.2xlarge	Nei	Alle MariaDB-Versionen 10.6, 10.5, 10.4 und 10.3	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet
db.r3.xlarge	Nei	Alle MariaDB-Versionen 10.6, 10.5, 10.4 und 10.3	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet
db.r3.large	Nei	Alle MariaDB-Versionen 10.6, 10.5, 10.4 und 10.3	Als veraltet gekennzeichnet	Ja	Veraltet	Als veraltet gekennzeichnet

Unterstützte DB-Engines für rechenoptimierte Instanzklassen

Die folgenden Tabellen zeigen die unterstützten Datenbanken und Datenbankversionen für die rechenoptimierten Instance-Klassen.

db.c6gd — für die Datenverarbeitung optimierte Instance-Klassen (nur für Multi-AZ-DB-Cluster-Bereitstellungen)

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.c6gd.16xlarge	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
						Versionen; 13.4 und 13.7 und höhere 13-Versionen
db.c6gd.12xlarge	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen
db.c6gd.8xlarge	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen
db.c6gd.4xlarge	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen
db.c6gd.2xlarge	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen
db.c6gd.xlarge	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen
db.c6gd.large	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.c6gd.medium	Nein	Nein	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL 17-, 16- und 15-Versionen; 14.5 und höhere 14-Versionen; 13.4 und 13.7 und höhere 13-Versionen

Unterstützte DB-Engines für Instance-Klassen mit hoher Leistung

Die folgenden Tabellen zeigen die unterstützten Datenbanken und Datenbankversionen für die Instance-Klassen mit Burstable-Performance.

db.t4g — Instance-Klassen mit hoher Leistung, die auf Graviton2-Prozessoren basieren AWS

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.t4g.2xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.t4g.xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.t4g.large	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.t4g.medium	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13;

Instance-Klasse	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
				und 8.0		und 12.7 und höhere 12-Versionen
db.t4g.klein	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen
db.t4g.micro	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nein	Alle PostgreSQL-Versionen 17, 16, 15, 14 und 13; und 12.7 und höhere 12-Versionen

db.t3 – Instance-Klassen mit Spitzenlastleistung

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.t3.2xlarge	Ja	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12, 11 und 10; und 9.6.22 und höhere 9-Versionen
db.t3.xlarge	Ja	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12, 11 und 10; und 9.6.22 und höhere 9-Versionen
db.t3.large	Ja	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12, 11 und 10; und 9.6.22 und höhere 9-Versionen

Instance-Klasse	Db2	Maria	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.t3.medium	Ja	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12, 11 und 10; und 9.6.22 und höhere 9-Versionen
db.t3.small	Ja	Ja	Ja	Ja	Ja	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12, 11 und 10; und 9.6.22 und höhere 9-Versionen
db.t3.micro	Nein	Ja	Ja	Ja	Nur auf Oracle Database 12c Release 1 (12.1.0.2), das veraltet ist	Alle PostgreSQL-Versionen 17, 16, 15, 14, 13, 12, 11 und 10; und 9.6.22 und höhere 9-Versionen

db.t2 – Instance-Klassen mit Spitzenlastleistung

Instance-Klasse	Db2	MariaDB	Microsof SQL Server	MySQL	Oracle	PostgreSQL
db.t2.2xlarge	Nein	Als veraltet gekennzeichnet	Nein	Als veraltet gekennzeichnet	Als veraltet gekennz chnet	Als veraltet gekennzeichnet
db.t2.xlarge	Nein	Als veraltet gekennzeichnet	Nein	Als veraltet gekennzeichnet	Als veraltet gekennz chnet	Als veraltet gekennzeichnet

Instance-Klasse	Db2	MariaDB	Microsof SQL Server	MySQL	Oracle	PostgreSQL
db.t2.lar ge	Nein	Als veraltet gekennzeichnet	Als veraltet gekennz chnet	Als veraltet gekennze chnet	Als veraltet gekenn chnet	Als veraltet gekennzeichnet
db.t2.med ium	Nein	Als veraltet gekennzeichnet	Als veraltet gekennz chnet	Als veraltet gekennze chnet	Als veraltet gekenn chnet	Als veraltet gekennzeichnet
db.t2.sma ll	Nein	Als veraltet gekennzeichnet	Als veraltet gekennz chnet	Als veraltet gekennze chnet	Als veraltet gekenn chnet	Als veraltet gekennzeichnet
db.t2.mic ro	Nein	Als veraltet gekennzeichnet	Als veraltet gekennz chnet	Als veraltet gekennze chnet	Als veraltet gekenn chnet	Als veraltet gekennzeichnet

Unterstützte DB-Engines für Instanzklassen mit optimierten Lesevorgängen

Die folgenden Tabellen zeigen die unterstützten Datenbanken und Datenbankversionen für die Instance-Klassen von Optimized Reads.

db.r6gd — speicheroptimierte Instanzklassen, die Optimized Reads unterstützen und auf Graviton2-Prozessoren basieren AWS

Instance-Klasse	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.6xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.2xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.xlarge	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.large	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versio

Instance-Klasse	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
						nen, 13.7 und höhere 13-Versionen und 13.4
db.r6gd.large	N	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nei	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen, 13.7 und höhere 13-Versionen und 13.4

db.r6id — speicheroptimierte Instanzklassen, die optimierte Lesevorgänge unterstützen und auf skalierbaren Intel Xeon Prozessoren der dritten Generation basieren

Instance-Klasse	Db2	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.32xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.24xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.16xlarge	Nein	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-

Instance-Klasse	Db2	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
					und BYOL	Versionen und 13.7 und höhere 13-Versio nen
db.r6id.12xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen
db.r6id.8xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur EE und BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen
db.r6id.4xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen
db.r6id.2xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versio nen

Instance-Klasse	Db2	MariaDB	Microso SQL Server	MySQL	Oracle	PostgreSQL
db.r6id.xlarge	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen
db.r6id.large	Neir	MariaDB 11.4, 10.11, 10.6, 10.5 und 10.4	Nein	MySQL 8.4 und 8.0	Nur BYOL	Alle PostgreSQL 17-, 16- und 15-Versionen, 14.5 und höhere 14-Versionen und 13.7 und höhere 13-Versionen

Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen

Zur Bestimmung der DB-Instance-Klassen, die von jeder DB-Engine in einer bestimmten AWS-Region unterstützt werden, stehen mehrere Ansätze zur Verfügung. Sie können die AWS Management ConsoleAmazon RDS-Preisseite oder den Befehl describe-orderable-db-instance-options für AWS Command Line Interface (AWS CLI) verwenden.

> **Note**
>
> Wenn Sie Operationen mit dem ausführen AWS Management Console, werden automatisch die unterstützten DB-Instance-Klassen für eine bestimmte DB-Engine, DB-Engine-Version und AWS-Region angezeigt. Beispiele für Vorgänge, die Sie ausführen können, sind das Erstellen und Ändern einer DB-Instance.

Inhalt

- Verwenden Sie die Amazon RDS-Preisseite, um die Unterstützung der DB-Instance-Klasse zu ermitteln in AWS-Regionen

- Verwenden Sie die AWS CLI , um die Unterstützung der DB-Instance-Klasse zu ermitteln in AWS-Regionen
  - Auflistung der DB-Instance-Klassen, die von einer bestimmten DB-Engine-Version in einer AWS-Region unterstützt werden
  - Auflisten der DB-Engine-Versionen, die eine bestimmte DB-Instance-Klasse in einer AWS-Region unterstützen
  - Liste AWS-Regionen , die eine bestimmte DB-Engine und Instance-Klasse unterstützen

Verwenden Sie die Amazon RDS-Preisseite, um die Unterstützung der DB-Instance-Klasse zu ermitteln in AWS-Regionen

Sie können die Seite Amazon RDS Pricing verwenden, um die DB-Instance-Klassen zu bestimmen, die von jeder DB-Engine in einer bestimmten AWS-Region unterstützt werden.

So verwenden Sie die Preisseite, um die DB-Instance-Klassen zu bestimmen, die von jeder Engine in einer Region unterstützt werden

1. Gehen Sie zu Amazon RDS Pricing.
2. Wählen Sie im Bereich AWS -Preisrechner für Amazon RDS die Option Jetzt Ihre maßgeschneiderte Kostenschätzung erstellen aus.
3. Wählen Sie unter Region auswählen eine AWS-Region aus.
4. Geben Sie im Feld Service suchen **`Amazon RDS`** ein.
5. Wählen Sie für Ihre Konfigurationsoption und DB-Engine Konfigurieren aus.
6. Verwenden Sie den Abschnitt für kompatible Instances, um sich die unterstützten DB-Instance-Klassen anzusehen.
7. (Optional) Wählen Sie andere Optionen im Rechner und dann Zusammenfassung speichern und anzeigen oder Service speichern und hinzufügen aus.

Verwenden Sie die AWS CLI , um die Unterstützung der DB-Instance-Klasse zu ermitteln in AWS-Regionen

Sie können den verwenden AWS CLI , um zu ermitteln, welche DB-Instance-Klassen für bestimmte DB-Engines und DB-Engine-Versionen in einem unterstützt AWS-Region werden. Die folgende Tabelle zeigt die gültigen DB-Engine-Werte.

Engine-Namen	Engine-Werte in CLI-Befehlen	Weitere Informationen zu den Versionen
Db2	`db2-ae` `db2-se`	Versionen von Db2 auf Amazon RDS
MariaDB	`mariadb`	MariaDB auf Amazon-RDS-Versionen

Microsoft SQL Server	`sqlserver-ee`
	`sqlserver-se`
	`sqlserver-ex`
	`sqlserver-web`
		Microsoft SQL Server-Versionen auf Amazon RDS

MySQL	`mysql`	MySQL in Amazon RDS-Versionen
Oracle	`oracle-ee` `oracle-se2`	Versionshinweise für Amazon RDS for Oracle
PostgreSQL	`postgres`	Verfügbare PostgreSQL-Datenbankversionen

Hinweise zu AWS-Region Namen finden Sie unterAWS Regionen.

Die folgenden Beispiele zeigen, wie Sie AWS-Region mithilfe des AWS CLI Befehls describe-orderable-db-instance-options die Unterstützung für DB-Instance-Klassen ermitteln können.

> **Note**
>
> Um die Ausgabe einzuschränken, zeigen die folgenden Beispiele nur Ergebnisse für den Speichertyp General Purpose SSD (`gp2`). Bei Bedarf können Sie den Speichertyp in den Befehlen in General Purpose SSD (`gp3`), Provisioned IOPS (`io1`), Provisioned IOPS Block Express (`io2`) oder magnetic (`standard`) ändern.

Themen

- Auflistung der DB-Instance-Klassen, die von einer bestimmten DB-Engine-Version in einer AWS-Region unterstützt werden
- Auflisten der DB-Engine-Versionen, die eine bestimmte DB-Instance-Klasse in einer AWS-Region unterstützen
- Liste AWS-Regionen , die eine bestimmte DB-Engine und Instance-Klasse unterstützen

Auflistung der DB-Instance-Klassen, die von einer bestimmten DB-Engine-Version in einer AWS-Region unterstützt werden

Führen Sie den folgenden Befehl aus, um die DB-Instance-Klassen aufzulisten, die von einer bestimmten DB-Engine-Version unterstützt werden. AWS-Region

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version
 \
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" \
    --output text \
    --region region
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version
 ^
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" ^
    --output text ^
    --region region
```

Der folgende Befehl listet beispielsweise die unterstützten DB-Instance-Klassen für Version 13.6 der RDS-for-PostgreSQL-DB-Engine in USA Ost (Nord-Virginia) auf.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4
 \
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" \
```

```
    --output text \
    --region us-east-1
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4
 ^
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" ^
    --output text ^
    --region us-east-1
```

Auflisten der DB-Engine-Versionen, die eine bestimmte DB-Instance-Klasse in einer AWS-Region unterstützen

Um die DB-Engine-Versionen aufzulisten, die eine bestimmte DB-Instance-Klasse in einer AWS-Region unterstützen, führen Sie den folgenden Befehl aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class \
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
    --output text \
    --region region
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class ^
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
    --output text ^
    --region region
```

Der folgende Befehl listet beispielsweise die DB-Engine-Versionen der RDS for PostgreSQL-DB-Engine auf, welche die db.r5.large DB-Instance-Klasse in US East (N. Virginia) unterstützen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large \
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
    --output text \
    --region us-east-1
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large ^
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
    --output text ^
    --region us-east-1
```

Liste AWS-Regionen , die eine bestimmte DB-Engine und Instance-Klasse unterstützen

Das folgende Bash-Skript listet alle auf AWS-Regionen , die die angegebene Kombination aus DB-Engine und Instance-Klasse unterstützen.

```
#!/usr/bin/env bash
# Usage: check_region_support.sh <db-engine> <db-instance-class>

if [ $# -ne 2 ]; then
  echo "Usage: $0 <db-engine> <db-instance-class>"
  exit 1
fi
ENGINE="$1"
INSTANCE_CLASS="$2"
REGIONS=$(aws ec2 describe-regions --query "Regions[].RegionName" --output text)
for region in $REGIONS; do
  supported_count=$(aws rds describe-orderable-db-instance-options \
    --region "$region" \
    --engine "$ENGINE" \
    --db-instance-class "$INSTANCE_CLASS" \
    --query 'length(OrderableDBInstanceOptions)' \
    --output text 2>/dev/null || echo "0")
  if [ "$supported_count" -gt 0 ]; then
    echo "$region supports $INSTANCE_CLASS for $ENGINE."
  else
    echo "$region doesn't support $INSTANCE_CLASS for $ENGINE."
```

```
  fi
done
```

Die folgende Beispielausgabe überprüft die Regionsunterstützung für mithilfe der Instance-Klasse db.r8g.large.

```
./check_region_support.sh mysql db.r8g.large
ap-south-1 doesn't support db.r8g.large for mysql.
eu-north-1 doesn't support db.r8g.large for mysql.
eu-west-3 doesn't support db.r8g.large for mysql.
eu-west-2 doesn't support db.r8g.large for mysql.
eu-west-1 doesn't support db.r8g.large for mysql.
ap-northeast-3 doesn't support db.r8g.large for mysql.
ap-northeast-2 doesn't support db.r8g.large for mysql.
ap-northeast-1 doesn't support db.r8g.large for mysql.
ca-central-1 doesn't support db.r8g.large for mysql.
sa-east-1 doesn't support db.r8g.large for mysql.
ap-southeast-1 doesn't support db.r8g.large for mysql.
ap-southeast-2 doesn't support db.r8g.large for mysql.
eu-central-1 supports db.r8g.large for mysql.
us-east-1 supports db.r8g.large for mysql.
us-east-2 supports db.r8g.large for mysql.
us-west-1 doesn't support db.r8g.large for mysql.
us-west-2 supports db.r8g.large for mysql.
```

Ändern Ihrer DB-Instance-Klasse

Sie können die CPU und den verfügbaren Speicher einer DB-Instance ändern, indem Sie ihre DB-Instance-Klasse ändern. Um die DB-Instance-Klasse zu ändern, modifizieren Sie Ihre DB-Instance, indem Sie die Anweisungen in befolge Ändern einer Amazon RDS DB-Instance.

Konfigurieren des Prozessors für eine DB-Instance-Klasse in RDS für Oracle

Amazon RDS-DB-Instance-Klassen unterstützen die Intel Hyperthreading-Technologie, die es ermöglicht, mehrere Threads gleichzeitig auf einem einzigen Intel Xeon CPU-Kern auszuführen. Jeder Thread wird als virtuelle CPU (vCPU) auf der DB-Instance dargestellt. Eine DB-Instance hat eine Standardanzahl von CPU-Kernen, die je nach DB-Instance-Klassevariiert. Beispielsweise hat eine DB-Instance-Klasse db.m4.xlarge standardmäßig zwei CPU-Kerne und zwei Threads pro Kern — insgesamt vier V. CPUs

> **Note**
>
> Jede vCPU ist ein Hyperthread eines Intel Xeon CPU-Kerns.

Themen

- Überblick über die Prozessorkonfiguration für RDS für Oracle
- DB-Instance-Klassen, welche die Prozessorkonfiguration unterstützen
- Festlegen der CPU-Kerne und -Threads pro CPU-Kern für eine DB-Instance-Klasse

Überblick über die Prozessorkonfiguration für RDS für Oracle

Wenn Sie RDS für Oracle verwenden, finden Sie in der Regel eine DB-Instance-Klasse mit einer Kombination aus Arbeitsspeicher und Anzahl von v, die Ihren Workloads entspricht. CPUs Sie können jedoch auch die folgenden Prozessorfunktionen angeben, um Ihre RDS for Oracle-DB-Instance für bestimmte Workloads oder Geschäftsanforderungen zu optimieren:

- Anzahl der CPU-Kerne – Sie können die Anzahl der CPU-Kerne für die DB-Instance anpassen. Sie könnten dies tun, um die Lizenzkosten Ihrer Software mit einer DB-Instance zu optimieren, die genügend RAM für speicherintensive Workloads, aber weniger CPU-Kerne hat.
- Threads pro Kern – Sie können die Intel Hyperthreading-Technologie deaktivieren, indem Sie einen einzelnen Thread pro CPU-Kern angeben. Sie können dies für bestimmte Workloads tun, z. B. für High Performance Computing (HPC)-Workloads.

Sie können die Anzahl der CPU-Kerne und Threads für jeden Kern separat steuern. Sie können eines oder beides in einer Anfrage festlegen. Nachdem eine Einstellung mit einer DB-Instance verknüpft wurde, bleibt die Einstellung so lange bestehen, bis Sie diese ändern.

Die Prozessoreinstellungen für eine DB-Instance sind mit Snapshots der DB-Instance verknüpft. Wenn ein Snapshot wiederhergestellt wird, verwendet seine wiederhergestellte DB-Instance die bei der Erstellung des Snapshots verwendeten Prozessorfunktion-Einstellungen.

Wenn Sie die DB-Instance-Klasse für eine DB-Instance mit nicht standardmäßigen Prozessoreinstellungen ändern, müssen Sie bei der Änderung entweder Standardprozessoreinstellungen oder explizit Prozessoreinstellungen angeben. Diese Anforderung stellt sicher, dass Sie sich über die Lizenzkosten von Drittanbietern im Klaren sind, die bei der Modifikation der DB-Instance entstehen können.

Es entstehen keine zusätzlichen Kosten oder reduzierte Gebühren für die Angabe von Prozessorfunktionen auf einer RDS-für-Oracle-DB-Instance. Sie werden genauso berechnet wie DB-Instances, die mit Standard-CPU-Konfigurationen gestartet werden.

DB-Instance-Klassen, welche die Prozessorkonfiguration unterstützen

Sie können die Anzahl der CPU-Kerne und Threads pro Kern nur konfigurieren, wenn die folgenden Bedingungen erfüllt sind:

- Sie konfigurieren eine RDS-für-Oracle-DB-Instance. Informationen über die von verschiedenen Oracle-Datenbank-Editionen unterstützten DB-Instance-Klassen finden Sie unter RDS für Oracle-DB-Instance-Klassen.
- Ihre DB-Instance verwendet die Bring Your Own License (BYOL)-Lizenzierungsoption von RDS für Oracle. Weitere Informationen über Oracle-Lizenzoptionen finden Sie unter RDSfür Oracle-Lizenzoptionen.
- Ihre DB-Instance gehört nicht zu den Instance-Klassen db.r5 oder db.r5b mit vordefinierten Prozessorkonfigurationen. Diese Instance-Klassen haben Namen in der Form db.r5. *instance_size*.tpc .mem oder db.r5b*threads_per_core*. *ratio* *instance_size*.tpc *threads_per_core* .mem*ratio*. Beispielsweise ist db.r5b.xlarge.tpc2.mem4x mit 2 Threads pro Kern (tpc2) und 4x so viel Speicher wie die Standard-Instance-Klasse db.r5b.xlarge vorkonfiguriert. Sie können die Prozessorfunktionen dieser optimierten Instance-Klassen nicht konfigurieren. Weitere Informationen finden Sie unter Unterstützte RDS für Oracle-DB-Instance-Klassen.

Sie können den folgenden AWS CLI Befehl verwenden, um die Standardwerte für VCPUs, Kerne, Threads pro Kern und die gültige Anzahl von Kernen für eine Instanzklasse anzuzeigen. Ersetzen Sie *r7i.48xlarge* den Beispielbefehl durch den Namen Ihrer Instanzklasse.

```
aws ec2 describe-instance-types \
    --instance-types r7i.48xlarge \
    --query '{
        DefaultVCPUs: InstanceTypes[0].VCpuInfo.DefaultVCpus,
        DefaultCores: InstanceTypes[0].VCpuInfo.DefaultCores,
        DefaultThreadsPerCore: InstanceTypes[0].VCpuInfo.DefaultThreadsPerCore,
        ValidCores: InstanceTypes[0].VCpuInfo.ValidCores
    }' \
    --output json
```

In der folgenden Tabelle finden Sie die DB-Instance-Klassen, welche die Festlegung einer bestimmten Anzahl von CPU-Kernen und CPU-Threads pro Kern unterstützen. Sie können auch den Standardwert und die gültigen Werte für die Anzahl der CPU-Kerne und CPU-Threads pro Kern für jede DB-Instance-Klasse finden.

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.m6i – arbeitsspeicheroptimierte Instance-Klassen					
db.m6i.large	2	1	2	1	1, 2
db.m6i.xlarge	4	2	2	2	1, 2
db.m6i.2xlarge	8	4	2	2, 4	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m6i.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m6i.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m6i.24 xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m6i.32 xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40,	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
				42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	
db.m5 – Allzweck-Instance-Klassen					
db.m5.large	2	1	2	1	1, 2
db.m5.xlarge	4	2	2	2	1, 2
db.m5.2xlarge	8	4	2	2, 4	1, 2
db.m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5.12x large	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5.16x large	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5.24x large	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m5d – Allzweck-Instance-Klassen					
db.m5d.large	2	1	2	1	1, 2
db.m5d.xlarge	4	2	2	2	1, 2
db.m5d.2x large	8	4	2	2, 4	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.m5d.4x large	16	8	2	2, 4, 6, 8	1, 2
db.m5d.8x large	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5d.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m5d.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5d.24 xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m4 – Allzweck-Instance-Klassen					
db.m4.10x large	40	20	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	1, 2
db.m4.16x large	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r7i — speicheroptimierte Instanzklassen					
db.r7i.large	2	1	2	1	1, 2
db.r7i.xlarge	4	2	2	1, 2	1, 2
db.r7i.2 x groß	8	4	2	1, 2, 3, 4	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.r7i.4x groß	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r7i.8xgroß	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.r7i.12xlarge	48	24	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	1, 2
db.r7i.16xlarge	64	32	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32	1, 2
db.r7i.24x groß	96	48	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48	1, 2
db.r7i.48x groß	192	96	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.r6i – arbeitsspeicheroptimierte Instance-Klassen					
db.r6i.large	2	1	2	1	1, 2
db.r6i.xlarge	4	2	2	1, 2	1, 2
db.r6i.2xlarge	8	4	2	2, 4	1, 2
db.r6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.r5 – arbeitsspeicheroptimierte Instance-Klassen					
db.r5.large	2	1	2	1	1, 2
db.r5.xlarge	4	2	2	2	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.r5.2xlarge	8	4	2	2, 4	1, 2
db.r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5 – arbeitsspeicheroptimierte Instance-Klassen					
db.r5b.large	2	1	2	1	1, 2
db.r5b.xlarge	4	2	2	2	1, 2
db.r5b.2xlarge	8	4	2	2, 4	1, 2
db.r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5b.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.r5b.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5b.24 xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5d – arbeitsspeicheroptimierte Instance-Klassen					
db.r5d.large	2	1	2	1	1, 2
db.r5d.xlarge	4	2	2	2	1, 2
db.r5d.2xlarge	8	4	2	2, 4	1, 2
db.r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5d.12 xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r5d.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5d.24 xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r4 – arbeitsspeicheroptimierte Instance-Klassen					

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.r4.large	2	1	2	1	1, 2
db.r4.xlarge	4	2	2	1, 2	1, 2
db.r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r3 – arbeitsspeicheroptimierte Instance-Klassen					
db.r3.large	2	1	2	1	1, 2
db.r3.xlarge	4	2	2	1, 2	1, 2
db.r3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2idn – arbeitsspeicheroptimierte Instance-Klassen					
db.x2idn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.x2idn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.x2idn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iedn – arbeitsspeicheroptimierte Instance-Klassen					
db.x2iedn.xlarge	4	2	2	1, 2	1, 2
db.x2iedn.2xlarge	8	4	2	2, 4	1, 2
db.x2iedn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.x2iedn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2iedn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x2iedn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.x2iedn .32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iezn – arbeitsspeicheroptimierte Instance-Klassen					
db.x2iezn .2xlarge	8	4	2	2, 4	1, 2
db.x2iezn .4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.x2iezn .6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.x2iezn .8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2iezn .12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.x1 – arbeitsspeicheroptimierte Instance-Klassen					
db.x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.x1e – arbeitsspeicheroptimierte Instance-Klassen					

DB-Instance-Klasse	Standard v CPUs	Standard-CPU-Kerne	Standard-Threads pro Kern	Gültige Anzahl der CPU-Kerne	Gültige Anzahl der Threads pro Kern
db.x1e.xlarge	4	2	2	1, 2	1, 2
db.x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.x1e.16 xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1e.32 xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.z1d – arbeitsspeicheroptimierte Instance-Klassen					
db.z1d.large	2	1	2	1	1, 2
db.z1d.xlarge	4	2	2	2	1, 2
db.z1d.2xlarge	8	4	2	2, 4	1, 2
db.z1d.3xlarge	12	6	2	2, 4, 6	1, 2
db.z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.z1d.12 xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

> **Note**
>
> Sie können AWS CloudTrail damit Änderungen an der Prozesskonfiguration von Amazon RDS for Oracle Oracle-DB-Instances überwachen und prüfen. Weitere Informationen zur Verwendung von CloudTrail finden Sie unter überwachen Amazon RDS API ruft an AWS CloudTrail.

Festlegen der CPU-Kerne und -Threads pro CPU-Kern für eine DB-Instance-Klasse

Sie können die Anzahl der CPU-Kerne und Threads pro Kern für die DB-Instance-Klasse konfigurieren, wenn Sie die folgenden Operationen durchführen:

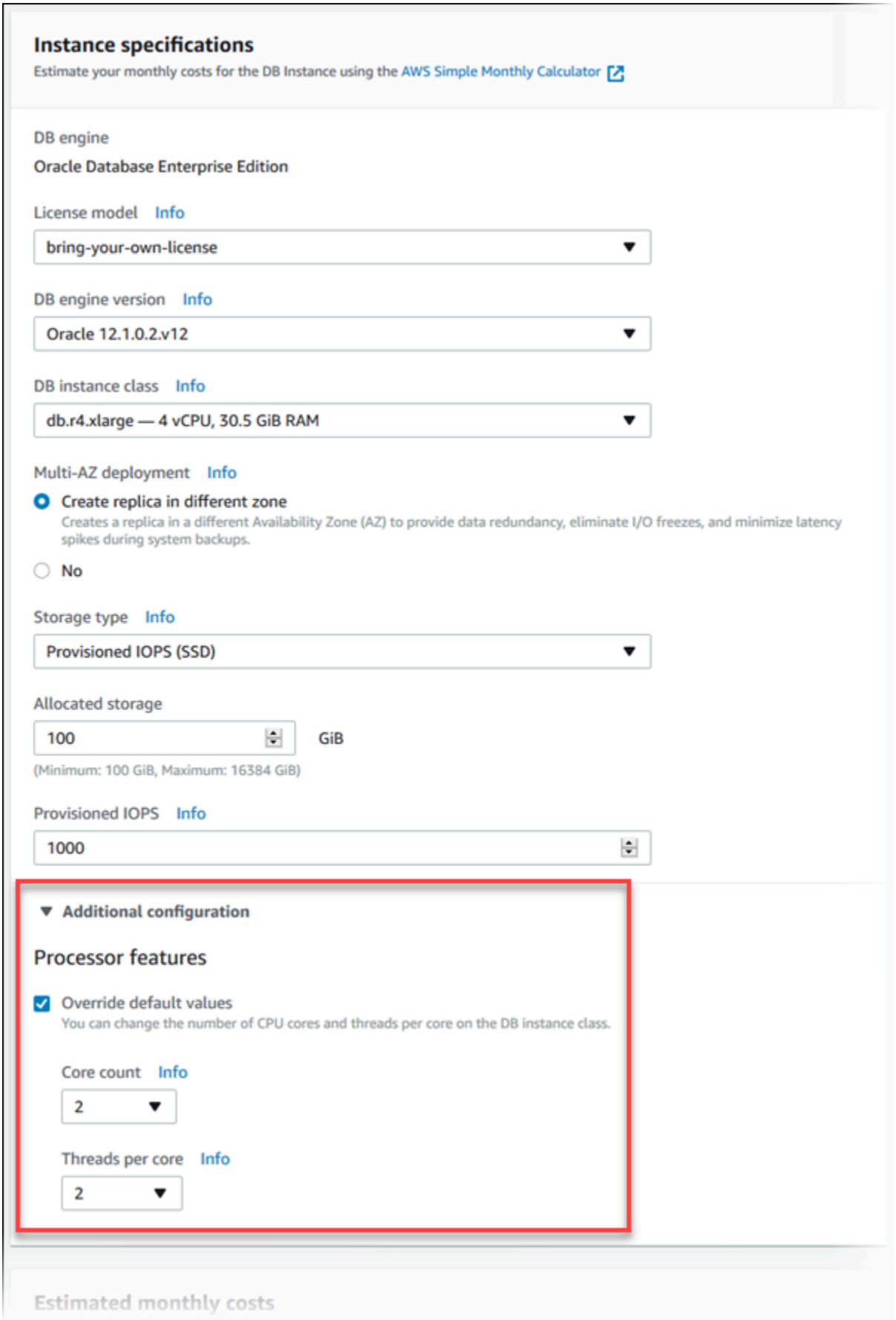
- Erstellen einer Amazon RDS-DB-Instance
- Ändern einer Amazon RDS DB-Instance
- Wiederherstellung auf einer DB-Instance
- Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS

> **Note**
>
> Wenn Sie eine DB-Instance ändern, um die Anzahl der CPU-Kerne oder Threads pro Kern zu konfigurieren, kommt es zu einem kurzen Ausfall der DB-Instance.

Sie können die CPU-Kerne und die Threads pro CPU-Kern für eine DB-Instance-Klasse mithilfe der AWS Management Console AWS CLI, der oder der RDS-API festlegen.

Konsole

Wenn Sie eine DB-Instance erstellen, ändern oder wiederherstellen, legen Sie die DB-Instance-Klasse in der fes AWS Management Console. Der Abschnitt Instance-Spezifikationen zeigt Optionen für den Prozessor. Das folgende Abbild zeigt Die Prozessorfunktionen-Optionen.

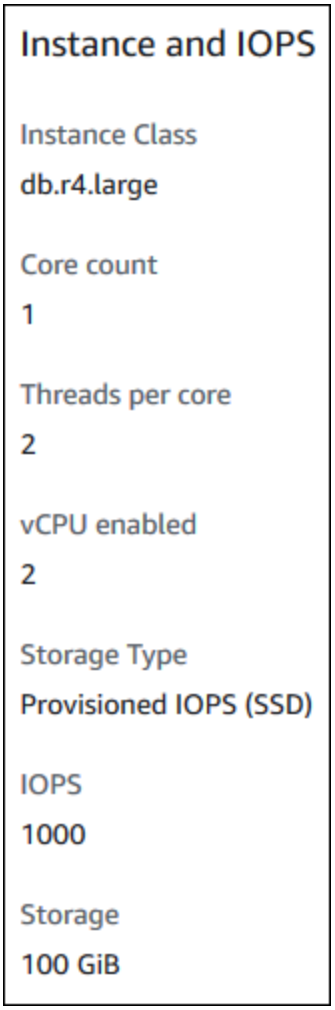

Instance specifications
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator
DB engine
Oracle Database Enterprise Edition
License model Info
bring-your-own-license
DB engine version Info
Oracle 12.1.0.2.v12
DB instance class Info
db.r4.xlarge — 4 vCPU, 30.5 GiB RAM
Multi-AZ deployment Info
Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
No
Storage type Info
Provisioned IOPS (SSD)
Allocated storage
100
GiB
(Minimum: 100 GiB, Maximum: 16384 GiB)
Provisioned IOPS Info
1000
Additional configuration
Processor features
Override default values
You can change the number of CPU cores and threads per core on the DB instance class.
Core count Info
2
Threads per core Info
2
Estimated monthly costs

Setzen Sie die folgenden Optionen auf die entsprechenden Werte für Ihre DB-Instance-Klasse unter Prozessorfunktionen:

- Core-Anzahl – Legen Sie mit dieser Option die Anzahl der CPU-Kerne fest. Der Wert muss gleich oder kleiner als die maximale Anzahl von CPU-Kernen für die DB-Instance-Klasse sein.
- Threads pro Kern – Geben Sie 2 an, um mehrere Threads pro Kern zuzulassen, oder 1, um mehrere Threads pro Kern zu verbieten.

Wenn Sie eine DB-Instance ändern oder wiederherstellen, können Sie auch die CPU-Kerne und die Threads pro CPU-Kern auf die Standardeinstellungen für die Instance-Klasse setzen.

Wenn Sie die Details zu einer DB-Instance in der Konsole anzeigen, können Sie die Prozessorinformationen für ihre DB-Instance-Klasse auf dem Tab Configuration (Konfiguration) anzeigen. Das folgende Bild zeigt eine DB-Instance-Klasse mit einem CPU-Kern und mehreren Threads pro Kern.



Bei Oracle DB-Instances werden die Prozessorinformationen nur für Bring Your Own License (BYOL) DB-Instances angezeigt.

AWS CLI

Sie können die Prozessorfunktionen für eine DB-Instance festlegen, wenn Sie einen der folgenden AWS CLI -Befehle ausführen:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-db-snapshot
- restore-db-instance-from-3
- restore-db-instance-to-point-in-time

Um den Prozessor einer DB-Instance-Klasse für eine DB-Instance mithilfe von zu konfigurieren AWS CLI, fügen Sie die `--processor-features` Option in den Befehl ein. Geben Sie die Anzahl der CPU-Kerne mit dem Funktionsnamen `coreCount` an, und geben Sie mit dem Funktionsnamen `threadsPerCore` an, ob mehrere Threads pro Kern aktiviert sind.

Die Option weist die folgende Syntax auf.

```
--processor-features "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

Es folgen Beispiele für die Konfiguration des Prozessors:

Beispiele

- Einstellen der Anzahl der CPU-Kerne für eine DB-Instance
- Festlegen der Anzahl der CPU-Kerne und Deaktivieren mehrerer Threads für eine DB-Instance
- Anzeigen der gültigen Prozessorwerte für eine DB-Instance-Klasse
- Wiederherstellen der Standard-Prozessoreinstellungen für eine DB-Instance
- Wiederherstellen der Standardanzahl der CPU-Kerne für eine DB-Instance
- Wiederherstellen der Standardanzahl der Threads pro Kern für eine DB-Instance

Einstellen der Anzahl der CPU-Kerne für eine DB-Instance

Example

Das folgende Beispiel ändert `mydbinstance`, indem die Anzahl der CPU-Kerne auf 4 gesetzt wird. Die Änderungen werden mit sofort übernomme `--apply-immediately`. Wenn Sie die Änderungen beim nächsten geplanten Wartungsfenster übernehmen wollen, lassen Sie die Option `--apply-immediately` weg.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=coreCount,Value=4" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=coreCount,Value=4" ^
    --apply-immediately
```

Festlegen der Anzahl der CPU-Kerne und Deaktivieren mehrerer Threads für eine DB-Instance

Example

Das folgende Beispiel ändert `mydbinstance`, indem die Anzahl der CPU-Kerne auf 4 gesetzt wird und mehrere Threads pro Kern deaktiviert werden. Die Änderungen werden mit sofort übernomme `--apply-immediately`. Wenn Sie die Änderungen beim nächsten geplanten Wartungsfenster übernehmen wollen, lassen Sie die Option `--apply-immediately` weg.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
```

```
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" ^
    --apply-immediately
```

Anzeigen der gültigen Prozessorwerte für eine DB-Instance-Klasse

Example

Sie können die gültigen Prozessorwerte für eine bestimmte DB-Instance-Klasse anzeigen, indem Sie den Befehl describe-orderable-db-instance-options ausführen und die Instance-Klasse für die `--db-instance-class` Option angeben. Die Ausgabe für den folgenden Befehl zeigt beispielsweise die Prozessoroptionen für die Instance-Klasse db.r3.large.

```
aws rds describe-orderable-db-instance-options --engine oracle-ee --db-instance-class
 db.r3.large
```

Es folgt eine Beispielausgabe für den Befehl im JSON-Format.

```
    {
                "SupportsIops": true,
                "MaxIopsPerGib": 50.0,
                "LicenseModel": "bring-your-own-license",
                "DBInstanceClass": "db.r3.large",
                "SupportsIAMDatabaseAuthentication": false,
                "MinStorageSize": 100,
                "AvailabilityZones": [
                    {
                        "Name": "us-west-2a"
                    },
                    {
                        "Name": "us-west-2b"
                    },
                    {
                        "Name": "us-west-2c"
                    }
                ],
                "EngineVersion": "12.1.0.2.v2",
                "MaxStorageSize": 32768,
                "MinIopsPerGib": 1.0,
                "MaxIopsPerDbInstance": 40000,
                "ReadReplicaCapable": false,
                "AvailableProcessorFeatures": [
```

```
                {
                    "Name": "coreCount",
                    "DefaultValue": "1",
                    "AllowedValues": "1"
                },
                {
                    "Name": "threadsPerCore",
                    "DefaultValue": "2",
                    "AllowedValues": "1,2"
                }
            ],
            "SupportsEnhancedMonitoring": true,
            "SupportsPerformanceInsights": false,
            "MinIopsPerDbInstance": 1000,
            "StorageType": "io1",
            "Vpc": false,
            "SupportsStorageEncryption": true,
            "Engine": "oracle-ee",
            "MultiAZCapable": true
    }
```

Darüber hinaus können Sie die folgenden Befehle ausführen, um Informationen zum Prozessor der DB-Instance-Klasse zu erhalten:

- describe-db-instances— Zeigt die Prozessorinformationen für die angegebene DB-Instance an.
- describe-db-snapshots— Zeigt die Prozessorinformationen für den angegebenen DB-Snapshot an.
- describe-valid-db-instance-modifications — Zeigt die gültigen Änderungen am Prozessor für die angegebene DB-Instance an.

In der Ausgabe der vorhergehenden Befehle sind die Werte für die Prozessorfunktionen nur dann nicht null, wenn die folgenden Bedingungen erfüllt sind:

- Sie verwenden eine DB-Instance von RDS für Oracle.
- Ihre DB-Instance von RDS für Oracle unterstützt das Ändern von Prozessorwerten.
- Die aktuellen CPU-Kern- und Thread-Einstellungen sind auf Nicht-Standardwerte festgelegt.

Wenn die oben genannten Bedingungen nicht erfüllt sind, können Sie den Instance-Typ mithilfe von describe-db-instancesermitteln. Sie können die Prozessorinformationen für diesen Instance-Typ abrufen, indem Sie den EC2 Vorgang ausführen describe-instance-types.

Wiederherstellen der Standard-Prozessoreinstellungen für eine DB-Instance

Example

Das folgende Beispiel ändert `mydbinstance`, indem es ihre DB-Instance-Klasse auf die zugehörigen Standard-Prozessorwerte zurücksetzt. Die Änderungen werden mit sofort übernomme `--apply-immediately`. Wenn Sie die Änderungen beim nächsten geplanten Wartungsfenster übernehmen wollen, lassen Sie die Option `--apply-immediately` weg.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --use-default-processor-features \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --use-default-processor-features ^
    --apply-immediately
```

Wiederherstellen der Standardanzahl der CPU-Kerne für eine DB-Instance

Example

Das folgende Beispiel ändert `mydbinstance`, indem es ihre DB-Instance-Klasse auf die zugehörige Standardanzahl der CPU-Kerne zurücksetzt. Die Einstellung für Threads pro Kern wird nicht geändert. Die Änderungen werden mit sofort übernomme `--apply-immediately`. Wenn Sie die Änderungen beim nächsten geplanten Wartungsfenster übernehmen wollen, lassen Sie die Option `--apply-immediately` weg.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=coreCount,Value=DEFAULT" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=coreCount,Value=DEFAULT" ^
    --apply-immediately
```

Wiederherstellen der Standardanzahl der Threads pro Kern für eine DB-Instance

Example

Das folgende Beispiel ändert `mydbinstance`, indem es ihre DB-Instance-Klasse auf die zugehörige Standardanzahl der Threads pro Kern zurücksetzt. Die Einstellung für die Anzahl der CPU-Kerne wird nicht geändert. Die Änderungen werden mit sofort übernomme `--apply-immediately`. Wenn Sie die Änderungen beim nächsten geplanten Wartungsfenster übernehmen wollen, lassen Sie die Option `--apply-immediately` weg.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=threadsPerCore,Value=DEFAULT" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=threadsPerCore,Value=DEFAULT" ^
    --apply-immediately
```

RDS-API

Sie können die Prozessorfunktionen für eine DB-Instance festlegen, indem Sie eine der folgenden Amazon RDS API-Operationen ausführen:

- CreateDBInstance
- Modifizieren SieDBInstance
- Wiederherstellen DBInstance von DBSnapshot
- DBInstanceAus S3 wiederherstellen
- RestoreDBInstanceToPointInTime

Um die Prozessorfunktionen einer DB-Instance-Klasse für eine DB-Instance unter Verwendung der Amazon RDS-API zu konfigurieren, nehmen Sie den Parameter `ProcessFeatures` in den Aufruf auf.

Der Parameter hat die folgende Syntax.

```
ProcessFeatures "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

Geben Sie die Anzahl der CPU-Kerne mit dem Funktionsnamen `coreCount` an, und geben Sie mit dem Funktionsnamen `threadsPerCore` an, ob mehrere Threads pro Kern aktiviert sind.

Sie können die gültigen Prozessorwerte für eine bestimmte DB-Instance-Klasse anzeigen, indem Sie den Vorgang DescribeOrderableDBInstanceOptions ausführen und die Instance-Klasse für den `DBInstanceClass` Parameter angeben. Sie können auch die folgenden Operationen verwenden:

- Beschreiben DBInstances — Zeigt die Prozessorinformationen für die angegebene DB-Instance an.
- Beschreiben DBSnapshots — Zeigt die Prozessorinformationen für den angegebenen DB-Snapshot an.
- DescribeValidDBInstanceÄnderungen — Zeigt die gültigen Änderungen am Prozessor für die angegebene DB-Instance an.

In der Ausgabe der vorhergehenden Operationen sind die Werte für die Prozessorfunktionen nur dann nicht null, wenn die folgenden Bedingungen erfüllt sind:

- Sie verwenden eine DB-Instance von RDS für Oracle.
- Ihre DB-Instance von RDS für Oracle unterstützt das Ändern von Prozessorwerten.
- Die aktuellen CPU-Kern- und Thread-Einstellungen sind auf Nicht-Standardwerte festgelegt.

Wenn die oben genannten Bedingungen nicht erfüllt sind, können Sie den Instance-Typ mithilfe von Describe ermitteln DBInstances. Sie können die Prozessorinformationen für diesen Instance-Typ abrufen, indem Sie den EC2 Vorgang ausführen DescribeInstanceTypes.

Hardware-Spezifikationen für DB-Instance-Klassen

In den in diesem Abschnitt finden Sie Hardwaredetails zu den Amazon RDS-DB-Instance-Klassen .

Informationen zur Amazon RDS-DB-Engine-Unterstützung für die einzelnen DB-Instance-Klassen finden Sie unter Unterstützte DB-Engines für DB-Instance-Klassen.

Themen

- Hardwareterminologie für DB-Instance-Klassen
- Hardwarespezifikationen für die Allzweck-Instance-Klassen
- Hardwarespezifikationen für die speicheroptimierten Instance-Klassen
- Hardwarespezifikationen für die rechenoptimierten Instance-Klassen
- Hardwarespezifikationen für Instance-Klassen mit hoher Leistung

Hardwareterminologie für DB-Instance-Klassen

Die folgende Terminologie wird zum Beschreiben der Hardwarespezifikationen für DB-Instance-Klassen verwendet:

vCPU

Die Anzahl der virtuellen Zentraleinheiten (CPUs). Eine virtuelle CPU ist eine Kapazitätseinheit, mit der Sie DB-Instance-Klassen vergleichen können. Anstatt einen bestimmten Prozessor für mehrere Monate oder Jahre zu erwerben oder zu leasen, wird jetzt Kapazität stundenweise gemietet. Unser Ziel ist es, eine konsistente und spezifische Menge an CPU-Kapazität innerhalb der Grenzen der zugrunde liegenden Hardware zur Verfügung zu stellen.

EC2-Recheneinheiten

Das relative Maß für die ganzzahlige Verarbeitungsleistung einer EC2 Amazon-Instance. Um Entwicklern den Vergleich der CPU-Kapazität zwischen verschiedenen Instance-Klassen zu erleichtern, haben wir eine Amazon EC2 Compute Unit definiert. Die Menge an CPU, die einer bestimmten Instance zugewiesen ist, wird in diesen EC2 Recheneinheiten ausgedrückt. Ein Steuergerät bietet derzeit eine CPU-Kapazität, die einem Opteron- oder GHz 2007 Xeon-Prozessor von 1,0 bis 1,2 entspricht.

Arbeitsspeicher (GiB)

Der Arbeitsspeicher (RAM) in Gibibytes, der der DB-Instance zugeteilt ist. Häufig ist das Verhältnis zwischen Arbeitsspeicher- und vCPU konsistent. Beispielsweise hat die Instance-Klasse db.r4 das gleiche Verhältnis von Speicher zu vCPU wie die Instance-Klasse db.r5. Für die meisten Anwendungsfälle bietet die Instance-Klasse db.r5 jedoch eine bessere und konsistentere Performance als die Instance-Klasse db.r4.

EBS-optimiert

Eine DB-Instance nutzt einen optimierten Konfigurations-Stack und bietet zusätzliche dedizierte Kapazität für I/O-Vorgänge. Diese Optimierung bietet die beste Leistung, indem Konflikte zwischen I/O-Vorgängen und anderem Datenverkehr von Ihrer Instance minimiert werden. Weitere Informationen zu Amazon EBS-optimierten Instances finden Sie unter Amazon EBS-optimierte Instances im Amazon-Benutzerhandbuch. EC2

Für EBS-optimierte Instances gibt es eine Baseline und eine maximale IOPS-Rate. Die maximale IOPS-Rate wird auf DB-Instance-Ebene erzwungen. Eine Reihe von EBS-Volumes, die zusammen eine IOPS-Rate haben, die über dem Maximum liegt, darf den Schwellenwert auf Instance-Ebene nicht überschreiten. Wenn die maximale IOPS-Rate für eine bestimmte DB-Instance-Klasse beispielsweise 40 000 beträgt und Sie vier 64 000 IOPS-EBS-Volumes anhängen, beträgt die maximale IOPS-Rate 40 000 statt 256 000. Informationen zum IOPS-Maximum, das für jeden EC2 Instance-Typ spezifisch ist, finden Sie unter Unterstützte Instance-Typen im EC2 Amazon-Benutzerhandbuch für Linux-Instances.

Max. EBS-Bandbreite (Mbit/s)

Die maximale EBS-Bandbreite in Megabit pro Sekunde. Dividieren Sie durch 8, um den erwarteten Durchsatz in Megabyte pro Sekunde zu erhalten.

> **Important**
>
> Allzweck-SSD (gp2)-Volumes für Amazon RDS-DB-Instances haben in den meisten Fällen eine Durchsatzgrenze von 250 MiB/s. Die Durchsatzgrenze kann jedoch je nach Volume-Größe variieren. Weitere Informationen finden Sie unter Amazon EBS-Volumetypen im EC2 Amazon-Benutzerhandbuch.

Netzwerkbandbreite

Die Netzwerkgeschwindigkeit relativ zu anderen DB-Instance-Klassen.

Hardwarespezifikationen für die Allzweck-Instance-Klassen

Die folgenden Tabellen zeigen die Rechen-, Arbeitsspeicher-, Speicher- und Bandbreitenspezifikationen für die Allzweck-Instanzklassen.

db.m8g — Allzweck-Instanzklassen, die auf Graviton4-Prozessoren basieren AWS

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m8g.48xlarge	192	—	768	Nur EBS-optimiert	40 000	50
db.m8g.24xlarge	96	—	512	Nur EBS-optimiert	30 000	40
db.m8g.16xlarge	64	—	384	Nur EBS-optimiert	20 000	30
db.m8g.12xlarge	48	—	256	Nur EBS-optimiert	15 000	22,5
db.m8g.8xgroß	32	—	128	Nur EBS-optimiert	10.000	15
db.m8g.4xgroß*	16	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.m8g.2xgroß*	8	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.m8g.xgroß*	4	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m8g.groß*	2	—	8	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.m7i — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der vierten Generation basieren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m7i.48xlarge	192	—	768	Nur EBS-optimiert	40 000	50
db.m7i.24x groß	96	—	384	Nur EBS-optimiert	30 000	37,5
db.m7i.16xlarge	64	—	256	Nur EBS-optimiert	20 000	25
db.m7i.12xlarge	48	—	192	Nur EBS-optimiert	15 000	18,75
db.m7i.8 x groß	32	—	128	Nur EBS-optimiert	10.000	12,5
db.m7i.4x groß	16	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m7i.2xgroß	8	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m7i.xlarge	4	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m7i.groß	2	—	8	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.m7g – Allzweck-Instance-Klassen mit AWS -Graviton3-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m7g.16xlarge	64	—	256	Nur EBS-optimiert	20 000	30
db.m7g.12xlarge	48	—	192	Nur EBS-optimiert	15 000	22,5
db.m7g.8xlarge	32	—	128	Nur EBS-optimiert	10.000	15
db.m7g.4xlarge	16	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.m7g.2xlarge*	8	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.m7g.xlarge*	4	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m7g.large*	2	—	8	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.m6g – Allzweck-Instance-Klassen mit AWS -Graviton2-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6g.16xlarge	64	—	256	Nur EBS-optimiert	19.000	25

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6g.12xlarge	48	—	192	Nur EBS-optimiert	13.500	20
db.m6g.8xlarge	32	—	128	Nur EBS-optimiert	9 000	12
db.m6g.4xlarge	16	—	64	Nur EBS-optimiert	4.750	Bis zu 10
db.m6g.2xlarge*	8	—	32	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.m6g.xlarge*	4	—	16	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.m6g.large*	2	—	8	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

db.m6gd — Allzweck-Instanzklassen, die auf Graviton2-Prozessoren und SSD-Speicher basieren AWS

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6gd.16xlarge	64	—	256	2 NVMe x 1900 SSD	19.000	25
db.m6gd.12xlarge	48	—	192	2 x 1425 SSD NVMe	13.500	20

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6gd.8xlarge	32	—	128	1 x 1900 SSD NVMe	9 000	12
db.m6gd.4x groß*	16	—	64	1 x 950 SSD NVMe	4.750	Bis zu 10
db.m6gd.2 x groß*	8	—	32	1 x 474 SSD NVMe	Bis zu 4750.	Bis zu 10
db.m6gd.xgroß*	4	—	16	1 x 237 SSD NVMe	Bis zu 4750.	Bis zu 10
db.m6gd.groß*	2	—	8	1 x 118 SSD NVMe	Bis zu 4750.	Bis zu 10

db.m6id — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der dritten Generation und SSD-Speicher basieren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6id.32xlarge	128	—	512	4 NVMe x 1900 SSD	40 000	50
db.m6id.24xlarge	96	—	384	4 x 1425 SSD NVMe	30 000	37,5
db.m6id.16xlarge	64	—	256	2 x 1900 SSD NVMe	20 000	25

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6id.12xlarge	48	—	192	2 x 1425 SSD NVMe	15 000	18,75
db.m6id.8xlarge	32	—	128	1 x 1900 SSD NVMe	10.000	12,5
db.m6id.4xlarge*	16	—	64	1 x 950 NVMe SSD	Bis zu 10 000*	Bis zu 12,5
db.m6id.2xlarge*	8	—	32	1 x 474 NVMe SSD	Bis zu 10 000*	Bis zu 12,5
db.m6id.xlarge*	4	—	16	1 x 237 NVMe SSD	Bis zu 10 000*	Bis zu 12,5
db.m6id.large*	2	—	8	1 x 118 NVMe SSD	Bis zu 10 000*	Bis zu 12,5

db.m6idn – Allzweck-Instance-Klassen mit skalierbaren Intel-Xeon-Prozessoren der dritten Generation, SSD-Speicher und Netzwerkoptimierung

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6idn.32xlarge	128	—	512	4 x 1900 NVMe SSD	80 000	200
db.m6idn.24xlarge	96	—	384	4 x 1425 SSD NVMe	60 000	150

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6idn.16xlarge	64	—	256	2 x 1900 SSD NVMe	40 000	100
db.m6idn.12xlarge	48	—	192	2 x 1425 SSD NVMe	30 000	75
db.m6idn.8xlarge	32	—	128	1 x 1900 SSD NVMe	20 000	50
db.m6idn.4xlarge*	16	—	64	1 x 950 NVMe SSD	Bis zu 20 000*	Bis zu 50
db.m6idn.2xlarge*	8	—	32	1 x 474 NVMe SSD	Bis zu 20 000*	Bis zu 40
db.m6idn.xlarge*	4	—	16	1 x 237 NVMe SSD	Bis zu 20 000*	Bis zu 30
db.m6idn.large*	2	—	8	1 x 118 NVMe SSD	Bis zu 20 000*	Bis zu 25

db.m6in — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der dritten Generation und Netzwerkoptimierung basieren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6in.32xlarge	128	—	512	Nur EBS-optimiert	80 000	200

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6in.24xlarge	96	—	384	Nur EBS-optimiert	60 000	150
db.m6in.16xlarge	64	—	256	Nur EBS-optimiert	40 000	100
db.m6in.12xlarge	48	—	192	Nur EBS-optimiert	30 000	75
db.m6in.8xlarge	32	—	128	Nur EBS-optimiert	20 000	50
db.m6in.4xlarge*	16	—	64	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 50
db.m6in.2xlarge*	8	—	32	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 40
db.m6in.xlarge*	4	—	16	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 30
db.m6in.large*	2	—	8	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 25

db.m6i — Allzweck-Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der dritten Generation basieren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m6i.32xlarge	128	—	512	Nur EBS-optimiert	40 000	50
db.m6i.24xlarge	96	—	384	Nur EBS-optimiert	30 000	37,5
db.m6i.16xlarge	64	—	256	Nur EBS-optimiert	20 000	25
db.m6i.12xlarge	48	—	192	Nur EBS-optimiert	15 000	18,75
db.m6i.8xlarge	32	—	128	Nur EBS-optimiert	10.000	12,5
db.m6i.4xlarge*	16	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m6i.2xlarge*	8	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m6i.xlarge*	4	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.m6i.large*	2	—	8	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.m5d — Allzweck-Instance-Klassen, die auf Intel Xeon Platinum-Prozessoren und SSD-Speicher basieren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m5d.24xlarge	96	345	384	4 x 900 SSD NVMe	19.000	25
db.m5d.16xlarge	64	262	256	4 x 600 NVMe SSD	13.600	20
db.m5d.12xlarge	48	173	192	2 x 900 NVMe SSD	9.500	10
db.m5d.8xlarge	32	131	128	2 x 600 NVMe SSD	6.800	10
db.m5d.4xlarge	16	61	64	2 x 300 NVMe SSD	4.750	Bis zu 10
db.m5d.2xlarge*	8	31	32	1 x 300 NVMe SSD	Bis zu 4750.	Bis zu 10
db.m5d.xlarge*	4	15	16	1 x 150 NVMe SSD	Bis zu 4750.	Bis zu 10
db.m5d.large*	2	10	8	1 x 75 NVMe SSD	Bis zu 4750.	Bis zu 10

db.m5 – Allzweck-Instance-Klassen mit Intel-Xeon-Platinum-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m5.24xlarge	96	345	384	Nur EBS-optimiert	19.000	25
db.m5.16xlarge	64	262	256	Nur EBS-optimiert	13.600	20
db.m5.12xlarge	48	173	192	Nur EBS-optimiert	9.500	10
db.m5.8xlarge	32	131	128	Nur EBS-optimiert	6.800	10
db.m5.4xlarge	16	61	64	Nur EBS-optimiert	4.750	Bis zu 10
db.m5.2xlarge*	8	31	32	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.m5.xlarge*	4	15	16	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.m5.large*	2	10	8	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

db.m4 — Allzweck-Instance-Klassen mit skalierbaren Intel Xeon Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m4.16xlarge	64	188	256	Nur EBS-optimiert	10.000	25
db.m4.10xlarge	40	124.5	160	Nur EBS-optimiert	4.000	10
db.m4.4xlarge	16	53.5	64	Nur EBS-optimiert	2.000	Hoch
db.m4.2xlarge	8	25.5	32	Nur EBS-optimiert	1.000	Hoch
db.m4.xlarge	4	13	16	Nur EBS-optimiert	750	Hoch
db.m4.large	2	6,5	8	Nur EBS-optimiert	450	Mittel

db.m3 – Allzweck-Instance-Klassen

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m3.2xlarge	8	26	30	Nur EBS-optimiert	1.000	Hoch
db.m3.xlarge	4	13	15	Nur EBS-optimiert	500	Hoch
db.m3.large	2	6,5	7,5	Nur EBS	—	Mittel

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.m3.medium	1	3	3,75	Nur EBS	—	Mittel

* Diese DB-Instance-Typen können die maximale Leistung über 30 Minuten mindestens einmal alle 24 Stunden unterstützen. Weitere Informationen zur Basisleistung der zugrunde liegenden EC2 Instance-Typen finden Sie unter Amazon EBS-optimierte Instances im EC2 Amazon-Benutzerhandbuch.

Hardwarespezifikationen für die speicheroptimierten Instance-Klassen

Die folgenden Tabellen zeigen die Rechen-, Arbeitsspeicher-, Speicher- und Bandbreitenspezifikationen für die speicheroptimierten Instance-Klassen.

db.z1d – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.z1d.12xlarge	48	271	384	2 x 900 SSD NVMe	14.000	25
db.z1d.6xlarge	24	134	192	1 x 900 NVMe SSD	7.000	10
db.z1d.3xlarge	12	75	96	1 x 450 NVMe SSD	3.500	Bis zu 10
db.z1d.2xlarge	8	53	64	1 x 300 NVMe SSD	2 333	Bis zu 10
db.z1d.xlarge*	4	28	32	1 x 150 NVMe SSD	Bis zu 2,333	Bis zu 10

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.z1d.large*	2	15	16	1 x 75 NVMe SSD	Bis zu 2,333	Bis zu 10

db.x2g — speicheroptimierte Instanzklassen mit Graviton2-Prozessoren AWS

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x2g.16xlarge	64	—	1024	Nur EBS-optimiert	19.000	25
db.x2g.12xlarge	48	—	768	Nur EBS-optimiert	14.250	20
db.x2g.8xlarge	32	—	512	Nur EBS-optimiert	9.500	12
db.x2g.4xlarge	16	—	256	Nur EBS-optimiert	4.750	Bis zu 10
db.x2g.2xlarge	8	—	128	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.x2g.xlarge	4	—	64	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.x2g.large	2	—	32	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

db.x2idn — speicheroptimierte Instanzklassen mit skalierbaren Intel Xeon Prozessoren der dritten Generation

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x2idn.32xlarge	128	—	2 048	2 x 1900 SSD NVMe	80 000	100
db.x2idn.24xlarge	96	—	1 536	2 x 1425 SSD NVMe	60 000	75
db.x2idn.16xlarge	64	—	1,024	1 x 1900 SSD NVMe	40 000	50

db.x2iedn — speicheroptimierte Instanzklassen mit lokalen SSDs, skalierbaren Intel Xeon NVMe Prozessoren der dritten Generation

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x2iedn.32xlarge	128	—	4.096	2 NVMe x 1900 SSD	80 000	100
db.x2iedn.24xlarge	96	—	3.072	2 x 1425 SSD NVMe	60 000	75
db.x2iedn.16xlarge	64	—	2 048	1 x 1900 SSD NVMe	40 000	50
db.x2iedn.8xlarge	32	—	1,024	1 x 950 NVMe SSD	20 000	25

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x2iedn.4xlarge	16	—	512	1 x 475 NVMe SSD	Bis zu 20 000*	Bis zu 25
db.x2iedn.2xlarge	8	—	256	1 x 237 NVMe SSD	Bis zu 20 000*	Bis zu 25
db.x2iedn.xlarge	4	—	128	1 x 118 NVMe SSD	Bis zu 20 000*	Bis zu 25

db.x2iezn — speicheroptimierte Instance-Klassen mit skalierbaren Intel Xeon Prozessoren der zweiten Generation

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x2iezn.12xlarge	>48	—	1 536	Nur EBS-optimiert	19.000	100
db.x2iezn.8xlarge	32	—	1,024	Nur EBS-optimiert	12.000	75
db.x2iezn.6xlarge	24	—	768	Nur EBS-optimiert	Bis zu 9 500	50
db.x2iezn.4xlarge	16	—	512	Nur EBS-optimiert	Bis zu 4750.	Bis zu 25
db.x2iezn.2xlarge	8	—	256	Nur EBS-optimiert	Bis zu 3 170	Bis zu 25

db.x1e – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x1e.32xlarge	128	340	3.904	Nur EBS-optimiert	14.000	25
db.x1e.16xlarge	64	179	1.952	Nur EBS-optimiert	7.000	10
db.x1e.8xlarge	32	91	976	Nur EBS-optimiert	3.500	Bis zu 10
db.x1e.4xlarge	16	47	488	Nur EBS-optimiert	1.750	Bis zu 10
db.x1e.2xlarge	8	23	244	Nur EBS-optimiert	1.000	Bis zu 10
db.x1e.xlarge	4	12	122	Nur EBS-optimiert	500	Bis zu 10

db.x1 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.x1.32xlarge	128	349	1.952	Nur EBS-optimiert	14.000	25
db.x1.16xlarge	64	174,5	976	Nur EBS-optimiert	7.000	10

db.r8g — speicheroptimierte Instanzklassen mit Graviton4-Prozessoren AWS

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r8g.48xlarge	192	—	1536	Nur EBS-optimiert	40 000	50
db.r8g.24xlarge	96	—	768	Nur EBS-optimiert	30 000	40
db.r8g.16xlarge	64	—	512	Nur EBS-optimiert	20 000	30
db.r8g.12xlarge	48	—	384	Nur EBS-optimiert	15 000	22,5
db.r 8g.8 x groß	32	—	256	Nur EBS-optimiert	10.000	15
db.r8g.4xgroß*	16	—	128	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.r8g.2xgroß*	8	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.r8g.xgroß*	4	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r8g.groß*	2	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.r7i — speicheroptimierte Instance-Klassen, die auf skalierbaren Intel Xeon Prozessoren der vierten Generation basieren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r7i.48xlarge	192	—	1536	Nur EBS-optimiert	40 000	50
db.r7i.24x groß	96	—	768	Nur EBS-optimiert	30 000	37,5
db.r7i.16xlarge	64	—	512	Nur EBS-optimiert	20 000	25
db.r7i.12xlarge	48	—	384	Nur EBS-optimiert	15 000	18,75
db.r7i.8 x groß	32	—	256	Nur EBS-optimiert	10.000	12,5
db.r7i.4x groß	16	—	128	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r7i.2xgroß	8	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r7i.xlarge	4	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r7i.groß	2	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.r7g – arbeitsspeicheroptimierte Instance-Klassen mit AWS -Graviton3-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r7g.16xlarge	64	—	512	Nur EBS-optimiert	20 000	30
db.r7g.12xlarge	48	—	384	Nur EBS-optimiert	15 000	22,5
db.r7g.8xlarge	32	—	256	Nur EBS-optimiert	10.000	15
db.r7g.4xlarge	16	—	128	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.r7g.2xlarge*	8	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 15
db.r7g.xlarge*	4	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r7g.large*	2	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.r6g – arbeitsspeicheroptimierte Instance-Klassen mit AWS -Graviton2-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6g.16xlarge	64	—	512	Nur EBS-optimiert	19.000	25

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6g.12xlarge	48	—	384	Nur EBS-optimiert	13.500	20
db.r6g.8xlarge	32	—	256	Nur EBS-optimiert	9 000	12
db.r6g.4xlarge	16	—	128	Nur EBS-optimiert	4.750	Bis zu 10
db.r6g.2xlarge*	8	—	64	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.r6g.xlarge*	4	—	32	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.r6g.large*	2	—	16	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

db.r6gd — speicheroptimierte Instanzklassen mit Graviton2-Prozessoren und SSD-Speicher AWS

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6gd.16xlarge	64	—	512	2 NVMe x 1900 SSD	19.000	25
db.r6gd.12xlarge	48	—	384	2 x 1425 SSD NVMe	13.500	20
db.r6gd.8xlarge	32	—	256	1 x 1900 SSD NVMe	9 000	12

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6gd.4xlarge	16	—	128	1 x 950 NVMe SSD	4.750	Bis zu 10
db.r6gd.2xlarge	8	—	64	1 x 474 NVMe SSD	Bis zu 4750.	Bis zu 10
db.r6gd.xlarge	4	—	32	1 x 237 NVMe SSD	Bis zu 4750.	Bis zu 10
db.r6gd.large	2	—	16	1 x 118 NVMe SSD	Bis zu 4750.	Bis zu 10

db.r6id — speicheroptimierte Instance-Klassen mit skalierbaren Intel Xeon Prozessoren der dritten Generation und SSD-Speicher

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6id.32xlarge	128	—	1,024	NVMe 4 x 1900 SSD	40 000	50
db.r6id.24xlarge	96	—	768	4 x 1425 SSD NVMe	30 000	37,5
db.r6id.16xlarge	64	—	512	2 x 1900 SSD NVMe	20 000	25
db.r6id.12xlarge	48	—	384	2 x 1425 SSD NVMe	15 000	18,75

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6id.8xlarge	32	—	256	1 x 1900 SSD NVMe	10.000	12,5
db.r6id.4xlarge*	16	—	128	1 x 950 SSD NVMe	Bis zu 10 000*	Bis zu 12,5
db.r6id.2xlarge*	8	—	64	1 x 474 SSD NVMe	Bis zu 10 000*	Bis zu 12,5
db.r6id.xlarge*	4	—	32	1 x 237 SSD NVMe	Bis zu 10 000*	Bis zu 12,5
db.r6id.large*	2	—	16	1 x 118 SSD NVMe	Bis zu 10 000*	Bis zu 12,5

db.r6idn — speicheroptimierte Instance-Klassen mit skalierbaren Intel Xeon Prozessoren der dritten Generation, SSD-Speicher und Netzwerkoptimierung

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6idn.32xlarge	128	—	1,024	NVMe 4x1900 SSD	80 000	200
db.r6idn.24xlarge	96	—	768	4 x 1425 SSD NVMe	60 000	150
db.r6idn.16xlarge	64	—	512	2 x 1900 SSD NVMe	40 000	100

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6idn.12xlarge	48	—	384	2 x 1425 SSD NVMe	30 000	75
db.r6idn.8xlarge	32	—	256	1 x 1900 SSD NVMe	20 000	50
db.r6idn.4xlarge*	16	—	128	1 x 950 SSD NVMe	Bis zu 20 000*	Bis zu 50
db.r6idn.2xlarge*	8	—	64	1 x 474 SSD NVMe	Bis zu 20 000*	Bis zu 40
db.r6idn.xlarge*	4	—	32	1 x 237 SSD NVMe	Bis zu 20 000*	Bis zu 30
db.r6idn.large*	2	—	16	1 x 118 SSD NVMe	Bis zu 20 000*	Bis zu 25

db.r6in — speicheroptimierte Instance-Klassen mit skalierbaren Intel Xeon Prozessoren der dritten Generation und Netzwerkoptimierung

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6in.32xlarge	128	—	1,024	Nur EBS-optimiert	80 000	200
db.r6in.24xlarge	96	—	768	Nur EBS-optimiert	60 000	150

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6in.16xlarge	64	—	512	Nur EBS-optimiert	40 000	100
db.r6in.12xlarge	48	—	384	Nur EBS-optimiert	30 000	75
db.r6in.8xlarge	32	—	256	Nur EBS-optimiert	20 000	50
db.r6in.4xlarge*	16	—	128	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 50
db.r6in.2xlarge*	8	—	64	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 40
db.r6in.xlarge*	4	—	32	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 30
db.r6in.large*	2	—	16	Nur EBS-optimiert	Bis zu 20 000*	Bis zu 25

db.r6i — Speicheroptimierte Oracle-Instance-Klassen, die für hohe Speicher-, Speicher- und I/O-Anforderungen vorkonfiguriert sind

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6i.8xlarge.tpc2.mem4x	32	—	1024	Nur EBS-optimiert	40 000	50

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6i.8xlarge.tpc2.mem3x	32	—	768	Nur EBS-optimiert	30 000	37,5
db.r6i.6xlarge.tpc2.mem4x	24	—	768	Nur EBS-optimiert	30 000	37,5
db.r6i.4xgroß.tpc2.mem4x	16	—	512	Nur EBS-optimiert	20 000	25
db.r6i.4xlarge.tpc2.mem3x	16	—	384	Nur EBS-optimiert	15 000	18,75
db.r6i.4xlarge.tpc2.mem2x	16	—	256	Nur EBS-optimiert	10.000	12,5
db.r6i.2xgroß.tpc2.mem8x	8	—	512	Nur EBS-optimiert	20 000	12,5
db.r6i.2xlarge.tpc2.mem4x	8	—	256	Nur EBS-optimiert	10.000	12,5
db.r6i.2xgroß.tpc1.mem2x	8	—	128	Nur EBS-optimiert	Bis zu 10 000*	12,5
db.r6i.xlarge.tpc2.mem4x	4	—	128	Nur EBS-optimiert	Bis zu 10 000*	12,5
db.r6i.xlarge.tpc2.mem2x	4	—	64	Nur EBS-optimiert	Bis zu 10 000*	12,5
db.r6i.large.tpc1.mem2x	2	—	32	Nur EBS-optimiert	Bis zu 10 000*	12,5

db.r6i – arbeitsspeicheroptimierte Instance-Klassen mit skalierbaren Intel-Xeon-Prozessoren der 3. Generation

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r6i.32xlarge	128	—	1,024	Nur EBS-optimiert	40 000	50
db.r6i.24xlarge	96	—	768	Nur EBS-optimiert	30 000	37,5
db.r6i.16xlarge	64	—	512	Nur EBS-optimiert	20 000	25
db.r6i.12xlarge	48	—	384	Nur EBS-optimiert	15 000	18,75
db.r6i.8xlarge	32	—	256	Nur EBS-optimiert	10.000	12,5
db.r6i.4xlarge*	16	—	128	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r6i.2xlarge*	8	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r6i.xlarge*	4	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5
db.r6i.large*	2	—	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 12,5

db.r5d – arbeitsspeicheroptimierte Instance-Klassen mit Intel-Xeon-Platinum-Prozessoren und SSD-Speicher

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r5d.24xlarge	96	347	768	4 x 900 SSD NVMe	19.000	25
db.r5d.16xlarge	64	264	512	4 x 600 NVMe SSDs	13.600	20
db.r5d.12xlarge	48	173	384	2 x 900 NVMe SSD	9.500	10
db.r5d.8xlarge	32	132	256	2 x 600 NVMe SSDs	6.800	10
db.r5d.4xlarge	16	71	128	2 x 300 NVMe SSD	4.750	Bis zu 10
db.r5d.2xlarge*	8	38	64	1 x 300 NVMe SSD	Bis zu 4750.	Bis zu 10
db.r5d.xlarge*	4	19	32	1 x 150 NVMe SSD	Bis zu 4750.	Bis zu 10
db.r5d.large*	2	10	16	1 x 75 NVMe SSD	Bis zu 4750.	Bis zu 10

db.r5b – von Oracle arbeitsspeicheroptimierte Instance-Klassen, die für hohen Arbeitsspeicher, Speicher und I/O vorkonfiguriert sind

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r5b.8xlarge.tpc2.mem3x	32	—	768	Nur EBS-optimiert	60 000	25
db.r5b.6xlarge.tpc2.mem4x	24	—	768	Nur EBS-optimiert	60 000	25
db.r5b.4xlarge.tpc2.mem4x	16	—	512	Nur EBS-optimiert	40 000	20
db.r5b.4xlarge.tpc2.mem3x	16	—	384	Nur EBS-optimiert	30 000	10
db.r5b.4xlarge.tpc2.mem2x	16	—	256	Nur EBS-optimiert	20 000	10
db.r5b.2xlarge.tpc2.mem8x	8	—	512	Nur EBS-optimiert	40 000	20
db.r5b.2xlarge.tpc2.mem4x	8	—	256	Nur EBS-optimiert	20 000	10
db.r5b.2xlarge.tpc1.mem2x	8	—	128	Nur EBS-optimiert	10.000	Bis zu 10
db.r5b.xlarge.tpc2.mem4x	4	—	128	Nur EBS-optimiert	10.000	Bis zu 10
db.r5b.xlarge.tpc2.mem2x	4	—	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 10
db.r5b.large.tpc1.mem2x	2	—	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 10

db.r5b – arbeitsspeicheroptimierte Instance-Klassen mit Intel-Xeon-Platinum-Prozessoren und EBS-Optimierung

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r5b.24xlarge	96	347	768	Nur EBS-optimiert	60 000	25
db.r5b.16xlarge	64	264	512	Nur EBS-optimiert	40 000	20
db.r5b.12xlarge	48	173	384	Nur EBS-optimiert	30 000	10
db.r5b.8xlarge	32	132	256	Nur EBS-optimiert	20 000	10
db.r5b.4xlarge	16	71	128	Nur EBS-optimiert	10.000	Bis zu 10
db.r5b.2xlarge	8	38	64	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 10
db.r5b.xlarge	4	19	32	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 10
db.r5b.large	2	10	16	Nur EBS-optimiert	Bis zu 10 000*	Bis zu 10

db.r5 – von Oracle arbeitsspeicheroptimierte Instance-Klassen, die für hohen Arbeitsspeicher, Speicher und I/O vorkonfiguriert sind

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r5.12xlarge.tpc2.mem2x	48	—	768	Nur EBS-optimiert	19.000	25
db.r5.8xlarge.tpc2.mem3x	32	—	768	Nur EBS-optimiert	19.000	25
db.r5.6xlarge.tpc2.mem4x	24	—	768	Nur EBS-optimiert	19.000	25
db.r5.4xlarge.tpc2.mem4x	16	—	512	Nur EBS-optimiert	13.600	20
db.r5.4xlarge.tpc2.mem3x	16	—	384	Nur EBS-optimiert	9.500	10
db.r5.4xlarge.tpc2.mem2x	16	—	256	Nur EBS-optimiert	6.800	10
db.r5.2xlarge.tpc2.mem8x	8	—	512	Nur EBS-optimiert	13.600	20
db.r5.2xlarge.tpc2.mem4x	8	—	256	Nur EBS-optimiert	6.800	10
db.r5.2xlarge.tpc1.mem2x	8	—	128	Nur EBS-optimiert	4.750	Bis zu 10
db.r5.xlarge.tpc2.mem4x	4	—	128	Nur EBS-optimiert	4.750	Bis zu 10
db.r5.xlarge.tpc2.mem2x	4	—	64	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r5.large.tpc1.mem2x	2	—	32	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

db.r5 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r5.24xlarge	96	347	768	Nur EBS-optimiert	19.000	25
db.r5.16xlarge	64	264	512	Nur EBS-optimiert	13.600	20
db.r5.12xlarge	48	173	384	Nur EBS-optimiert	9.500	12
db.r5.8xlarge	32	132	256	Nur EBS-optimiert	6.800	10
db.r5.4xlarge	16	71	128	Nur EBS-optimiert	4.750	Bis zu 10
db.r5.2xlarge*	8	38	64	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.r5.xlarge*	4	19	32	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10
db.r5.large*	2	10	16	Nur EBS-optimiert	Bis zu 4750.	Bis zu 10

db.r4 – arbeitsspeicheroptimierte Instance-Klassen mit skalierbaren Intel-Xeon-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r4.16xlarge	64	195	488	Nur EBS-optimiert	14.000	25
db.r4.8xlarge	32	99	244	Nur EBS-optimiert	7.000	10
db.r4.4xlarge	16	53	122	Nur EBS-optimiert	3.500	Bis zu 10
db.r4.2xlarge	8	27	61	Nur EBS-optimiert	1.700	Bis zu 10
db.r4.xlarge	4	13.5	30.5	Nur EBS-optimiert	850	Bis zu 10
db.r4.large	2	7	15,25	Nur EBS-optimiert	425	Bis zu 10

db.r3 – arbeitsspeicheroptimierte Instance-Klassen

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r3.8xlarge**	32	104	244	Nur EBS	—	10
db.r3.4xlarge	16	52	122	Nur EBS-optimiert	2.000	Hoch

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.r3.2xlarge	8	26	61	Nur EBS-optimiert	1.000	Hoch
db.r3.xlarge	4	13	30,5	Nur EBS-optimiert	500	Mittel
db.r3.large	2	6,5	15,25	Nur EBS-optimiert	—	Mittel

* Diese DB-Instance-Typen können die maximale Leistung über 30 Minuten mindestens einmal alle 24 Stunden unterstützen. Weitere Informationen zur Basisleistung der zugrunde liegenden EC2 Instance-Typen finden Sie unter Amazon EBS-optimierte Instances im EC2 Amazon-Benutzerhandbuch.

** Die DB-Instance-Klasse r3.8xlarge hat keine dedizierte EBS-Bandbreite und bietet daher keine EBS-Optimierung. Für diese Instance-Klasse wird der Netzwerkdatenverkehr zusammen mit dem Amazon-EBS-Datenverkehr durch dieselbe 10-Gigabit-Netzwerkschnittstelle geleitet.

Hardwarespezifikationen für die rechenoptimierten Instance-Klassen

Die folgenden Tabellen zeigen die Rechen-, Speicher-, Speicher- und Bandbreitenspezifikationen für die rechenoptimierten Instance-Klassen.

db.c6gd — rechenoptimierte Instance-Klassen (nur für Multi-AZ-DB-Cluster-Bereitstellungen)

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.c6gd.16xlarge	64	—	128	2 x 1900 SSD NVMe	19.000	25

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.c6gd.12x groß	48	—	96	2 x 1425 SSD NVMe	13.500	20
db.c6gd.8 x groß	32	—	64	1 x 1900 SSD NVMe	9 000	12
db.c6gd.4x groß	16	—	32	1 x 950 SSD NVMe	4.750	Bis zu 10
db.c6gd.2 x groß	8	—	16	1 x 474 SSD NVMe	Bis zu 4750.	Bis zu 10
db.c6gd.xlarge	4	—	8	1 x 237 SSD NVMe	Bis zu 4750.	Bis zu 10
db.c6gd.large	2	—	4	1 x 118 SSD NVMe	Bis zu 4750.	Bis zu 10
db.c6gd.medium	1	—	2	1 x 59 SSD NVMe	Bis zu 4750.	Bis zu 10

Hardwarespezifikationen für Instance-Klassen mit hoher Leistung

Die folgenden Tabellen zeigen die Rechen-, Speicher-, Speicher- und Bandbreitenspezifikationen für die Instance-Klassen mit hoher Leistung.

db.t4g — Instance-Klassen mit Spitzenlastleistung und AWS -Graviton2-Prozessoren

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.t4g.2xlarge*	8	—	32	Nur EBS-optimiert	Bis zu 2.780	Bis zu 5
db.t4g.xlarge*	4	—	16	Nur EBS-optimiert	Bis zu 2.780	Bis zu 5
db.t4g.large*	2	—	8	Nur EBS-optimiert	Bis zu 2.780	Bis zu 5
db.t4g.medium*	2	—	4	Nur EBS-optimiert	Bis zu 2 085	Bis zu 5
db.t4g.klein*	2	—	2	Nur EBS-optimiert	Bis zu 2 085	Bis zu 5
db.t4g.micro*	2	—	1	Nur EBS-optimiert	Bis zu 2 085	Bis zu 5

db.t3 – Instance-Klassen mit Spitzenlastleistung

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.t3.2xlarge*	8	Variab	32	Nur EBS-optimiert	Bis zu 2.048	Bis zu 5
db.t3.xlarge*	4	Variab	16	Nur EBS-optimiert	Bis zu 2.048	Bis zu 5

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.t3.large*	2	Variab	8	Nur EBS-optimiert	Bis zu 2.048	Bis zu 5
db.t3.medium*	2	Variab	4	Nur EBS-optimiert	Bis zu 1.536	Bis zu 5
db.t3.small*	2	Variab	2	Nur EBS-optimiert	Bis zu 1.536	Bis zu 5
db.t3.micro*	2	Variab	1	Nur EBS-optimiert	Bis zu 1.536	Bis zu 5

db.t2 – Instance-Klassen mit Spitzenlastleistung

Instance-Klasse	vCPU	EC2-Recheneinheiten	Arbeitsspeicher (GiB)	Instance-Speicher (GB)	Max. EBS-Bandbreite (Mbit/s)	Netzwerkbandbreite (Gbit/s)
db.t2.2xlarge	8	Variab	32	Nur EBS	—	Mittel
db.t2.xlarge	4	Variab	16	Nur EBS	—	Mittel
db.t2.large	2	Variab	8	Nur EBS	—	Mittel
db.t2.Medium	2	Variab	4	Nur EBS	—	Mittel
db.t2.small	1	Variab	2	Nur EBS	—	Niedrig
db.t2.micro	1	Variab	1	Nur EBS	—	Niedrig

* Diese DB-Instance-Typen können die maximale Leistung über 30 Minuten mindestens einmal alle 24 Stunden unterstützen. Weitere Informationen zur Basisleistung der zugrunde liegenden

EC2 Instance-Typen finden Sie unter Amazon EBS-optimierte Instances im EC2 Amazon-Benutzerhandbuch.

Amazon RDS-DB-Instance-Speicher

DB-Instances für Amazon RDS for Db2, MariaDB, MySQL, PostgreSQL, Oracle und Microsoft SQL Server verwenden Amazon Elastic Block Store (Amazon EBS) -Volumes für die Datenbank- und Protokollspeicherung.

Ihre Datenbank-Workload kann die bereitgestellten IOPS möglicherweise nicht zu 100 Prozent erreichen. Weitere Informationen finden Sie unter Faktoren, die die Datenbankleistung beeinflussen.

Weitere Informationen zu Preisen für Instance-Speicher erhalten Sie unter Amazon RDS-Preise.

Themen

- Amazon RDS-Speichertypen
- Bereitgestellter IOPS SSD-Speicher
- Allzweck-SSD-Speicher
- Leistungsmerkmale von Solid-State-Drive-Speichertypen (SSD)
- Magnetischer Speicher (veraltet)
- Dediziertes Protokollvolumen (DLV)
- Überwachung der Datenbankleistung
- Faktoren, die die Datenbankleistung beeinflussen

Amazon RDS-Speichertypen

Amazon RDS bietet drei Speichertypen: Bereitgestellte IOPS-SSD (auch bekannt als io1 und io2 Block Express), Allzweck-SSD (auch bekannt als gp2 und gp3) und magnetisch (auch bekannt als Standard). Diese unterscheiden sich bei den Leistungsmerkmalen und im Preis, das bedeutet, dass Sie die Speicherleistung und -kosten an die Anforderungen der Datenbank-Workload anpassen können. Sie können Db2-, MySQL-, MariaDB-, Oracle-, SQL Server- und PostgreSQL RDS-DB-Instances mit bis zu 64 Tebibyte (TiB) Speicher erstellen. RDS für Db2 unterstützt die Speichertypen GP2 und Magnetic nicht.

In der folgenden Liste werden die drei Speichertypen beschrieben:

- Bereitgestellte IOPS-SSD — Der bereitgestellte IOPS-Speicher ist darauf ausgelegt, die Durchsatzanforderungen zu erfüllen. I/O-intensive workloads, particularly database workloads, that

require low I/O latency and consistent I/O Bereitgestellter IOPS-Speicher eignet sich am besten für Produktionsumgebungen.

Weitere Informationen über bereitgestellten IOPS-Speicher einschließlich der Speichergrößenbereiche finden Sie unter Bereitgestellter IOPS SSD-Speicher.

- Allzweck-SSD – Diese Volumes bieten kosteneffizienten Speicher, der für ein breites Spektrum an Workloads, die auf mittelgroßen DB-Instances ausgeführt werden, geeignet ist. Allzweck-Speicher eignet sich am besten für Entwicklungs- und Testumgebungen.

  Weitere Informationen über Allzweck-SSD einschließlich der Speichergrößenbereiche finden Sie unter Allzweck-SSD-Speicher.
- Magnetspeicher – Amazon RDS unterstützt auch magnetische Speicherung für die Abwärtskompatibilität. Wir empfehlen generell die Nutzung von Allzweck-SSD-Speicher oder SSD-Speicher mit bereitgestellten IOPS. Die maximal zulässige Speichermenge für DB-Instances auf Magnetspeichern beträgt 3 TiB. Weitere Informationen finden Sie unter Magnetischer Speicher (veraltet).

Bereitgestellter IOPS SSD-Speicher

Für eine Produktionsanwendung, die eine schnelle und konsistente I/O Leistung erfordert, empfehlen wir bereitgestellten IOPS-Speicher. Ein Speicher mit bereitgestellten IOPS liefert voraussagbare Leistung und konsistente niedrige Latenz. Speicher mit bereitgestellten IOPS ist für Workloads bei der Online-Transaktionsverarbeitung (OLTP) optimiert, die konsistente Performance erfordern. Speicher mit bereitgestellten IOPS trägt zur Optimierung dieser Workloads bei.

Amazon RDS bietet zwei Arten von bereitgestelltem IOPS-SSD-Speicher: io2 und io1. Wenn Sie eine DB-Instance erstellen, geben Sie die IOPS-Rate und die Größe des Volumes an. Amazon RDS stellt diese IOPS-Leistung für die DB-Instance bereit, bis Sie diese ändern.

Themen

- io2 Block Express-Speicher (empfohlen)
- io1-Speicher (vorherige Generation)
- Kombinieren von bereitgestelltem IOPS-Speicher mit Multi-AZ-Bereitstellungen oder Lesereplikaten
- Kosten für bereitgestellten IOPS-Speicher
- Optimale Leistung aus dem von Amazon RDS bereitgestellten IOPS-Speicher herausholen

io2 Block Express-Speicher (empfohlen)

Für I/O-intensive and latency-sensitive workloads, we recommend that you use Provisioned IOPS SSD io2 Block Express storage to achieve up to 256,000 I/O operations per second (IOPS). The throughput of io2 Block Express volumes varies based on the amount of IOPS provisioned per volume and on the size of the I/O laufende Operationen.

Alle RDS-io2-Volumes, die auf dem AWS Nitro-System basieren, sind io2 Block Express-Volumes und bieten eine durchschnittliche Latenz von unter einer Millisekunde. DB-Instances, die nicht auf dem Nitro-System basieren, sind io2-Volumes. AWS

Die folgende Tabelle zeigt den Bereich der bereitgestellten IOPS und den maximalen Durchsatz für jede Datenbank-Engine und jeden Speichergrößenbereich.

Datenbank-Engine	Speicherp latzbereich	Bereitgestellte IOPS-Leistung	Maximaler Durchsatz
Db2, MariaDB, MySQL und PostgreSQL	100—65.536 GiB	1.000–256.000 IOPS	16.000 MiB/s
Oracle	100—199 GiB	1.000—199.000 IOPS	4 000 MiB/s
Oracle	200—65.536 GiB	1.000–256.000 IOPS	16.000 MiB/s
SQL Server	20—65.536 GiB	1.000–256.000 IOPS	4 000 MiB/s

Für die IOPS- und Speichergrößenbereiche gelten die folgenden Einschränkungen:

- Das Verhältnis von IOPS zu zugewiesenem Speicher (in GiB) muss zwischen 0,5 und 1.000 liegen. Für DB-Instances, die nicht auf dem AWS Nitro-System basieren, muss das Verhältnis zwischen 0,5 und 500 liegen.
- Maximale IOPS können mit Volumes ab einer Größe von 256 GiB (1 000 IOPS × 256 GiB = 256 000 IOPS) bereitgestellt werden. Für DB-Instances, die nicht auf dem AWS Nitro System basieren, werden maximale IOPS bei 512 GiB erreicht (500 IOPS x 512 GiB = 256.000 IOPS).
- Der Durchsatz skaliert proportional auf bis zu 0,256 pro bereitgestelltem IOPS. MiB/s Ein maximaler Durchsatz von 4.000 IOPS MiB/s kann bei 256.000 IOPS bei einer Größe von 16 KB und 16.000 IOPS oder höher bei einer I/O Größe von 256 KiB erreicht werden. I/O Für DB-

Instances, die nicht auf dem AWS Nitro-System basieren, MiB/s kann ein maximaler Durchsatz von 2.000 bei 128.000 IOPS und einer Größe von 16 KiB erreicht werden. I/O

- Wenn Sie die automatische Speicherskalierung verwenden, gelten auch die gleichen Verhältnisse zwischen IOPS und maximalem Speicherschwellenwert (in GiB). Weitere Informationen zur automatischen Speicherskalierung finden Sie unter Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung.

Amazon RDS io2 Block Express-Volumen sind in allen AWS Handelsregionen und Regionen AWS GovCloud (USA) erhältlich. Diese Bände sind in der Region China nicht verfügbar.

io1-Speicher (vorherige Generation)

Für I/O-intensive workloads, you can use Provisioned IOPS SSD io1 storage and achieve up to 256,000 I/O operations per second (IOPS). The throughput of io1 volumes varies based on the amount of IOPS provisioned per volume and on the size of the I/O Operationen, die gerade ausgeführt werden. Wir empfehlen, io2 Block Express-Speicher zu verwenden, sofern er verfügbar ist.

Die folgende Tabelle zeigt den Bereich der bereitgestellten IOPS und den maximalen Durchsatz für jede Datenbank-Engine und jeden Speichergrößenbereich.

Datenbank-Engine	Speicherplatzbereich	Bereitgestellte IOPS-Leistung	Maximaler Durchsatz
Db2, MariaDB, MySQL und PostgreSQL	100—399 GiB	1 000–19 950 IOPS	500 MiB/s
Db2, MariaDB, MySQL und PostgreSQL	400—65.536 GiB	1.000–256.000 IOPS	4 000 MiB/s
Oracle	100—199 GiB	1 000–9 950 IOPS	500 MiB/s
Oracle	200—65.536 GiB	1.000—256.000 IOPS[1]	4 000 MiB/s
SQL Server	20—16.384 GiB	1.000—64.000 IOPS[2]	1 000 MiB/s

> **Note**
>
> [1] Für Oracle können Sie die maximalen 256.000 IOPS nur für den Instance-Typ r5b bereitstellen.
> [2] Für SQL Server ist die maximale Anzahl von 64.000 IOPS nur für Nitro-basierte Instances garantiert, die sich auf den Instance-Typen m5*, m6i, r5*, r6i und z1d befinden. Andere Instance-Typen garantieren eine Leistung von bis zu 32.000 IOPS.

Für die IOPS- und Speichergrößenbereiche gelten die folgenden Einschränkungen:

- Das Verhältnis von IOPS zu zugewiesenem Speicher (in GiB) muss bei RDS für SQL Server zwischen 1-50 und bei anderen RDS-DB-Engines zwischen 0,5 und 50 liegen.
- Wenn Sie die automatische Speicherskalierung verwenden, gelten auch die gleichen Verhältnisse zwischen IOPS und maximalem Speicherschwellenwert (in GiB).

  Weitere Informationen zur automatischen Speicherskalierung finden Sie unter Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung.

Kombinieren von bereitgestelltem IOPS-Speicher mit Multi-AZ-Bereitstellungen oder Lesereplikaten

Für OLTP-Anwendungsfälle in Produktionsumgebungen empfehlen wir die Verwendung von Multi-AZ-Bereitstellungen, da diese bei Speicher mit bereitgestellten IOPS, der schnelle und voraussagbare Leistung bereitstellt, eine erweiterte Fehlertoleranz bieten.

Sie können auch bereitgestellten IOPS-Speicher mit Read Replicas für MySQL, MariaDB oder PostgreSQL verwenden. Der Speichertyp für ein Read Replica ist von der Speicherart der primärem DB-Instance unabhängig. Beispielsweise verwenden Sie ggf. eine Allzweck-SSD für Read Replicas mit einer primären DB-Instance, die SSD-Speicher mit bereitgestellten IOPS nutzt, um Kosten zu sparen. Die Leistung Ihrer Read Replica kann sich in diesem Fall jedoch von der einer Konfiguration unterscheiden, bei der sowohl die primäre DB-Instance als auch die Read Replicas bereitgestellten IOPS-Speicher verwenden.

Kosten für bereitgestellten IOPS-Speicher

Bei Speicher mit bereitgestellten IOPS werden Ihnen die bereitgestellten Ressourcen berechnet, auch wenn Sie diese während des jeweiligen Monats nicht genutzt haben.

Weitere Informationen zu Preisen finden Sie unter Amazon RDS-Preise.

Optimale Leistung aus dem von Amazon RDS bereitgestellten IOPS-Speicher herausholen

Wenn Ihre Arbeitslast I/O eingeschränkt ist, kann die Verwendung von bereitgestelltem IOPS-Speicher die Anzahl der I/O Anfragen erhöhen, die das System gleichzeitig verarbeiten kann. Eine höhere Parallelität ermöglicht eine geringere Latenz, da I/O Anfragen weniger Zeit in einer Warteschlange verbringen. Dadurch wiederum werden Datenbank-Commits beschleunigt, was die Reaktionszeit und den Datenbankdurchsatz verbessert.

Der bereitgestellte IOPS-Speicher bietet eine Möglichkeit, I/O Kapazität durch Angabe von IOPS zu reservieren. Der Maximaldurchsatz unter Last wird allerdings, wie alle anderen Systemkapazitätsattribute auch, durch diejenige Ressource begrenzt, die zuerst verbraucht wird. Dies kann entweder Netzwerkbandbreite, CPU, Arbeitsspeicher oder eine datenbankinterne Ressource sein.

Allzweck-SSD-Speicher

Allzweckspeicher bietet kostengünstigen Speicher, der für die meisten Datenbank-Workloads akzeptabel ist, die nicht latenz- oder leistungssensitiv sind.

> **Note**
>
> Bei DB-Instances, die Allzweckspeicher verwenden, kann es zu einer wesentlich längeren Latenz kommen als bei Instances, die bereitgestellten IOPS-Speicher verwenden. Wenn Sie nach diesen Vorgängen eine DB-Instance mit minimaler Latenz benötigen, empfehlen wir die Verwendung von Bereitgestellter IOPS SSD-Speicher.

Amazon RDS bietet zwei Arten von Allzweckspeicher: GP3-Speicher (empfohlen) und GP2-Speicher (vorherige Generation).

GP3-Speicher (empfohlen)

Durch die Verwendung von GP3-Allzweck-Speichervolumes können Sie die Speicherleistung unabhängig von der Speicherkapazität anpassen. Die Speicherleistung ist die Kombination aus I/O Operationen pro Sekunde (IOPS) und der Geschwindigkeit, mit der das Speichervolume Lese- und

Schreibvorgänge ausführen kann (Speicherdurchsatz). Auf gp3-Speichervolumes bietet Amazon RDS eine Basisspeicherleistung von 3 000 IOPS und 125 MiB/s.

Für jede RDS-DB-Engine außer RDS für SQL Server gilt: Wenn die Speichergröße für GP3-Volumes einen bestimmten Schwellenwert erreicht, steigt die Basisspeicherleistung. Dies ist auf das Volume-Striping zurückzuführen, bei dem der Speicher vier Volumes anstelle von einem verwendet. RDS für SQL Server unterstützt kein Volume-Striping und hat daher keinen Schwellenwert. Für Striped-Volumes bietet Amazon RDS eine Basisspeicherleistung von 12.000 IOPS und 500 MIB/s.

Die Speicherleistung für gp3-Volumes auf DB-Engines von Amazon RDS, einschließlich des Schwellenwerts, ist in der folgenden Tabelle dargestellt.

DB-Engine	Speichergröße	Basisspeicherleistung	Bereitgestellte IOPS-Leistung	Bereich des bereitgestellten Speicherdurchsatzes
Db2, MariaDB, MySQL und PostgreSQL	20—399 GiB	3 000 IOPS/125 MiB/s	N/A	N/A
Db2, MariaDB, MySQL und PostgreSQL	400—65.536 GiB	12 000 IOPS/500 MiB/s	12 000–64 000 IOPS	500-4 000 MiB/s
Oracle	20—199 GiB	3 000 IOPS/125 MiB/s	N/A	N/A
Oracle	200—65.536 GiB	12 000 IOPS/500 MiB/s	12 000–64 000 IOPS	500-4 000 MiB/s
SQL Server	20—16.384 GiB	3 000 IOPS/125 MiB/s	3 000–16 000 IOPS	125–1 000 MiB/s

Für jede DB-Engine außer RDS für SQL Server können Sie zusätzliche IOPS und zusätzlichen Speicherdurchsatz bereitstellen, wenn die Speichergröße den Schwellenwert erreicht oder überschreitet. Für RDS für SQL Server können Sie zusätzliche IOPS und zusätzlichen Speicherdurchsatz für jede verfügbare Speichergröße bereitstellen. Für alle DB-Engines zahlen

Sie nur für die zusätzliche bereitgestellte Speicherleistung. Weitere Informationen finden Sie unter Amazon RDS – Preise.

Die zusätzlichen bereitgestellten IOPS und der Speicherdurchsatz hängen zwar nicht von der Speichergröße ab, sind jedoch miteinander verbunden. Wenn Sie die IOPS für MariaDB und MySQL auf über 32.000 erhöhen, erhöht sich der Speicherdurchsatzwert automatisch von 500. MiBps Wenn Sie beispielsweise die IOPS auf RDS für MySQL auf 40.000 festlegen, muss der Speicherdurchsatz mindestens 625 MiBps betragen. Die automatische Erhöhung erfolgt nicht für Db2-, Oracle-, PostgreSQL- und SQL Server-DB-Instances.

Für Multi-AZ-DB-Cluster legt Amazon RDS den Durchsatzwert automatisch auf der Grundlage der von Ihnen bereitgestellten IOPS fest. Sie können den Durchsatzwert nicht ändern.

Für Speicherleistungswerte für gp3-Volumes in RDS gelten die folgenden Einschränkungen:

- Das maximale Verhältnis von Speicherdurchsatz zu IOPS beträgt 0,25 für alle unterstützten DB-Engines.
- Das Mindestverhältnis von IOPS zu zugewiesenem Speicher (in GiB) liegt für RDS für SQL Server bei 0,5. Für die anderen unterstützten DB-Engines gibt es kein Mindestverhältnis.
- Das maximale Verhältnis von IOPS zu Speicherdurchsatz beträgt 500 für alle unterstützten DB-Engines.
- Wenn Sie die automatische Speicherskalierung verwenden, gelten auch die gleichen Verhältnisse zwischen IOPS und maximalem Speicherschwellenwert (in GiB).

  Weitere Informationen zur automatischen Speicherskalierung finden Sie unter Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung.

GP2-Speicher (vorherige Generation)

Wenn Ihre Anwendungen keine hohe Speicherleistung benötigen, können Sie Allzweck-SSD-gp2-Speicher verwenden. Die I/O Basisleistung für GP2-Speicher beträgt 3 IOPS pro GiB, mit einem Minimum von 100 IOPS. Dieses Verhältnis bedeutet, dass die Leistung bei großen Volumes besser. Die Basisleistung für ein 100-GiB-Volume beträgt beispielsweise 300 IOPS. Ein Volume mit 1 000 GiB verfügt über eine Basisleistung von 3 000 IOPS.

Einzelne gp2-Volumes unter 1 000 GiB können über einen längeren Zeitraum zudem bis auf 3 000 IOPS steigen. Das I/O Volumenguthaben bestimmt die Burst-Performance. Eine detailliertere Beschreibung der Auswirkungen der Basisleistung und des I/O-Guthabensaldos auf die Leistung

finden Sie im Beitrag Understanding Burst vs. Baseline Performance with Amazon RDS and gp2 im AWS Datenbank-Blog.

Bei vielen Workloads wird die Burst-Balance nicht ausgeschöpft. Bei einigen Workloads kann das Burst-Speicherguthaben von 3.000 IOPS jedoch aufgebraucht sein. Planen Sie Ihre Speicherkapazität daher so, dass sie den Anforderungen Ihrer Workloads entspricht.

Bei GP2-Volumes mit mehr als 4.000 GiB ist die Basisleistung höher als die Burst-Leistung. Für solche Volumes ist die Spitzenlast irrelevant, da die Ausgangsleistung besser ist als die 3.000 IOPS-Spitzenlastleistung. Bei DB-Instances bestimmter Engines und Größen wird der Speicher durch Striping jedoch auf vier Volumes verteilt, was den vierfachen Basisdurchsatz und die vierfache IOPS-Spitzenleistung eines einzelnen Volumes bietet.

Die Speicherleistung für GP2-Volumes verschiedener Speichergrößen auf Amazon RDS-DB-Engines ist in der folgenden Tabelle dargestellt.

DB-Engine	Größe des RDS-Speichers	Bereich der Basis-IOPS	Bereich des Basisdurchsatzes	IOPS-Spitzenleistung
MariaDB, MySQL und PostgreSQL	5—399 GiB	100-1 197 IOPS	128-250 MiB/s	3,000
MariaDB, MySQL und PostgreSQL	400—1.335 GiB	1 200-4 005 IOPS	512—1.000 MiB/s	12.000
MariaDB, MySQL und PostgreSQL	1.336—3.999 GiB	4 008-11 997 IOPS	1 000 MiB/s	12.000
MariaDB, MySQL und PostgreSQL	4.000—65.536 GiB	12 000–64 000 IOPS	1 000 MiB/s	N/A[1]
Oracle	20—199 GiB	100-597 IOPS	128-250 MiB/s	3,000

DB-Engine	Größe des RDS-Speichers	Bereich der Basis-IOPS	Bereich des Basisdurchsatzes	IOPS-Spitzenleistung
Oracle	200—1.335 GiB	600-4 005 IOPS	512—1.000 MiB/s	12.000
Oracle	1.336—3.999 GiB	4 008-11 997 IOPS	1 000 MiB/s	12.000
Oracle	4.000—65.536 GiB	12 000–64 000 IOPS	1 000 MiB/s	N/A[1]
SQL Server	20—333 GiB	100-999 IOPS	128-250 MiB/s	3,000
SQL Server	334—999 GiB	1 002–2 997 IOPS	250 MiB/s	3,000
SQL Server	1.000—16.384 GiB	3 000–16 000 IOPS	250 MiB/s	N/A[1]

[1] Die Ausgangsleistung des Volumes übersteigt die maximale Burst-Leistung.

Leistungsmerkmale von Solid-State-Drive-Speichertypen (SSD)

In der folgenden Tabelle werden Anwendungsfälle und Leistungsmerkmale pro Volume für die von Amazon RDS verwendeten SSD-Speichervolumes beschrieben.

Merkmal	Bereitgestellte IOPS (io2 Block Express)	Bereitgestellte IOPS (io1)	Allzweck (gp3)	Allzweck (gp2)
Beschreibung	Höchste Leistung innerhalb des RDS-Speicherportfolios (IOPS, Durchsatz, Latenz)	Konsistente Speicherleistung (IOPS, Durchsatz, Latenz)	Flexibilität bei der unabhängigen Bereitstellung von Speicher, IOPS und Durchsatz	Bietet burstfähige IOPS Bietet ein günstiges Preis-Leistungs-Ve

Merkmal	Bereitgestellte IOPS (io2 Block Express)	Bereitgestellte IOPS (io1)	Allzweck (gp3)	Allzweck (gp2)
	Konzipiert für latenzempfindliche, transaktionale Workloads	Konzipiert für latenzempfindliche, transaktionale Workloads	Bietet ein günstiges Preis-Leistungs-Verhältnis für ein breites Spektrum an Transaktions-Workloads	rhältnis für ein breites Spektrum an Transaktions-Workloads
Anwendungsfälle	Geschäftskritische Transaktionsworkloads, die eine Latenz von unter einer Millisekunde und eine anhaltende IOPS-Leistung von bis zu 256.000 IOPS erfordern	Transaktions-Workloads, die anhaltende IOPS-Leistung von bis zu 256 000 IOPS erfordern	Breites Spektrum an Workloads, die auf mittelgroßen relationalen development/test Datenbanken in Umgebungen ausgeführt werden	Breites Spektrum an Workloads, die auf mittelgroßen relationalen Datenbanken in Umgebungen ausgeführt werden development/test
Latency	In weniger als einer Millisekunde, konsistent in 99,9% der Fälle bereitgestellt	Einstellige Millisekunde, die in 99,9 % der Fälle konstant bereitgestellt wird	Einstellige Millisekunde, die in 99 % der Fälle konstant bereitgestellt wird	Einstellige Millisekunde, die in 99 % der Fälle konstant bereitgestellt wird
Volume-Größe	100—65.536 GiB	100—65.536 GiB (20—16.384 GiB auf RDS für SQL Server)	20—65.536 GiB (16.384 GiB auf RDS für SQL Server)	20—65.536 GiB (16.384 GiB auf RDS für SQL Server)

Merkmal	Bereitgestellte IOPS (io2 Block Express)	Bereitgestellte IOPS (io1)	Allzweck (gp3)	Allzweck (gp2)
Maximale IOPS	256 000	256 000 (64 000 in RDS für SQL Server)	64 000 (16 000 in RDS für SQL Server)	64 000 (16 000 in RDS für SQL Server) **Note** Sie können IOPS nicht direkt auf dem gp2-Speicher bereitstellen. IOPS variiert je nach Größe des zugewiesenen Speichers.

Merkmal	Bereitgestellte IOPS (io2 Block Express)	Bereitgestellte IOPS (io1)	Allzweck (gp3)	Allzweck (gp2)
Maximaler Durchsatz	Skaliert auf Basis der bereitgestellten IOPS bis zu 4 000 MB/s Der Durchsatz lässt sich proportional auf bis zu 0,256 pro bereitgestelltem IOPS skalieren. MiB/s Ein maximaler Durchsatz von 4.000 IOPS MiB/s kann bei 256.000 IOPS bei einer Größe von 16 KB und 16.000 IOPS oder höher bei einer I/O Größe von 256 KiB erreicht werden. I/O Für Instanzen, die nicht auf dem AWS Nitro-System basieren, MiB/s kann ein maximaler Durchsatz von 2.000 bei 128.000	Skaliert auf Basis der bereitgestellten IOPS bis zu 4 000 MB/s	Stellen Sie zusätzlichen Durchsatz von bis zu 4.000 MB/s (1000 MB/s auf RDS für SQL Server) bereit	1000 MB/s (250 MB/s auf RDS für SQL Server)

Merkmal	Bereitgestellte IOPS (io2 Block Express)	Bereitgestellte IOPS (io1)	Allzweck (gp3)	Allzweck (gp2)
	IOPS und einer Größe von 16 KiB erreicht werden. I/O			
AWS CLI und RDS-API-Name	io2	io1	gp3	gp2

Automatisches Striping auf allen SSD-Volumes

Wenn Sie „Allzweck-SSD“ oder „Bereitgestellte IOPS-SSD“ auswählen, verteilt Amazon RDS je nach ausgewählter Engine und angeforderter Speichermenge Daten automatisch per Striping auf mehrere Volumes, um die Leistung zu verbessern, wie in der folgenden Tabelle dargestellt.

Datenbank-Engine	Amazon-RDS-Speichergröße	Anzahl der bereitgestellten Volumes
Db2	Weniger als 400 GiB	1
Db2	400—65.536 GiB	4
MariaDB, MySQL und PostgreSQL	Weniger als 400 GiB	1
MariaDB, MySQL und PostgreSQL	400—65.536 GiB	4
Oracle	Weniger als 200 GiB	1
Oracle	200—65.536 GiB	4
SQL Server	Any	1

Auswirkungen auf die Leistung, wenn Sie ein SSD-Volume ändern

Wenn Sie ein Allzweck-SSD- oder Bereitgestelltes IOPS-SSD-Volume ändern, durchläuft es eine Reihe von Zuständen. Solange sich das Volume im `optimizing` Status befindet, liegt die Leistung Ihres Volumes zwischen den Quell- und Zielkonfigurationsspezifikationen. Die Leistung des Volumes in der Übergangszeit wird nicht geringer sein als die niedrigere der beiden Spezifikationen.

Wenn Sie den Speicher einer Instance so ändern, dass er von einem Volume auf vier Volumes wechselt, oder wenn Sie eine Instance mithilfe von Magnetspeicher ändern, verwendet Amazon RDS die Elastic Volumes-Funktion nicht. Stattdessen stellt Amazon RDS neue Volumes bereit und verschiebt die Daten transparent vom alten Volume auf die neuen Volumes. Dieser Vorgang verbraucht eine erhebliche Menge an IOPS und einen erheblichen Durchsatz sowohl des alten als auch des neuen Volumes. Abhängig von der Größe des Volumes und der Menge der Datenbank-Arbeitslast, die während der Änderung anfällt, kann dieser Vorgang eine hohe Menge an IOPS verbrauchen, die I/O Latenz erheblich erhöhen und mehrere Stunden dauern, bis der Vorgang abgeschlossen ist, solange die RDS-Instance im `Modifying` Status bleibt.

Basiswerte und maximale IOPS-Raten für EBS-optimierte Instances

Für EBS-optimierte Instances gibt es eine Baseline und eine maximale IOPS-Rate. Die maximale IOPS-Rate wird auf DB-Instance-Ebene erzwungen. Eine Reihe von EBS-Volumes, die zusammen eine IOPS-Rate haben, die über dem Maximum liegt, darf den Schwellenwert auf Instance-Ebene nicht überschreiten. Wenn die maximale IOPS-Rate für eine bestimmte DB-Instance-Klasse beispielsweise 40 000 beträgt und Sie vier 64 000 IOPS-EBS-Volumes anhängen, beträgt die maximale IOPS-Rate 40 000 statt 256 000. Informationen zum IOPS-Maximum, das für jeden EC2 Instance-Typ spezifisch ist, finden Sie unter Unterstützte Instance-Typen im EC2 Amazon-Benutzerhandbuch für Linux-Instances.

Magnetischer Speicher (veraltet)

Amazon RDS unterstützt Magnetspeicher nicht mehr. Wir empfehlen, für Ihre Speicheranforderungen eine Allzweck-SSD oder eine bereitgestellte IOPS-SSD zu verwenden.

Dediziertes Protokollvolumen (DLV)

Sie können ein dediziertes Log-Volume (DLV) für eine DB-Instance verwenden, die Provisioned IOPS (PIOPS) -Speicher verwendet, indem Sie die Amazon RDS-Konsole oder die Amazon AWS CLI RDS-API verwenden. Ein DLV verschiebt Transaktionsprotokolle und MySQL/MariaDB

Redo-Logs und Binärprotokolle der PostgreSQL-Datenbank auf ein Speichervolume, das von dem Volume getrennt ist, das die Datenbanktabellen enthält. Ein DLV macht die Protokollierung von Transaktionsschreibvorgängen effizienter und konsistenter. DLVs eignen sich ideal für Datenbanken mit großem zugewiesenem Speicher, hohen I/O IOPS-Anforderungen (IOPS) oder latenzempfindlichen Workloads.

DLVs werden für PIOPS-Speicher (io1 und io2 Block Express) unterstützt und werden mit einer festen Größe von 1.024 GiB und 3.000 bereitgestellten IOPS erstellt.

> **Note**
>
> DLVs werden für Allzweckspeicher (gp2 und gp3) nicht unterstützt.

Amazon RDS unterstützt DLVs AWS-Regionen insgesamt die folgenden Versionen:

- MariaDB 10.6.7 und höhere 10-Versionen
- MySQL 8.0.28 und höhere 8.0-Versionen, MySQL 8.4.3 und höhere 8.4-Versionen
- PostgreSQL 13.10 und höher 13 Versionen, 14.7 und höher 14 Versionen, 15.2 und höher 15 Versionen und 16.1 und höher 16 Versionen

RDS unterstützt Multi-AZ-Bereitstellungen. DLVs Wenn Sie eine Multi-AZ-Instance ändern oder erstellen, wird ein DLV sowohl für die primäre als auch für die sekundäre Instance erstellt.

RDS unterstützt DLVs mit Read Replicas. Wenn für die primäre DB-Instance ein DLV aktiviert ist, verfügen alle Lesereplikate, die nach der Aktivierung des DLV erstellt wurden, auch über ein DLV. Für alle Lesereplikate, die vor dem Wechsel zu DLV erstellt wurden, wird diese Funktion nicht aktiviert, sofern sie nicht ausdrücklich entsprechend geändert wurde. Wir empfehlen, dass alle Lesereplikate, die vor der Aktivierung von DLV einer primären Instance angefügt wurden, ebenfalls manuell so geändert werden, dass sie über ein DLV verfügen.

Nachdem Sie die DLV-Einstellung für eine DB-Instance geändert haben, muss die DB-Instance neu gestartet werden.

Informationen zur Aktivierung eines DLV finden Sie unter. Verwendung eines dedizierten Protokoll-Volumes (DLV)

Überwachung der Datenbankleistung

Amazon RDS bietet mehrere Metriken, mit deren Hilfe Sie die Leistung der DB-Instance ermitteln können. Sie können die Metriken auf der Übersichtsseite Ihrer Instance in der Amazon RDS Management Console anzeigen. Sie können Amazon auch verwenden CloudWatch , um diese Metriken zu überwachen. Weitere Informationen finden Sie unter Metriken in der RDS Amazon-Konsole anzeigen. Die erweiterte Überwachung bietet detailliertere I/O Messwerte. Weitere Informationen finden Sie unterÜberwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Die folgenden Metriken sind nützlich, um die Leistung Ihrer DB-Instance zu überwachen:

- `DiskQueueDepth`— Die Anzahl der I/O Anfragen in der Warteschlange, die darauf warten, bearbeitet zu werden. Dies sind I/O Anfragen, die von der Anwendung eingereicht, aber nicht an das Gerät gesendet wurden, da das Gerät gerade andere I/O Anfragen bearbeitet. Zeit in der Warteschlange ist ein Teil der Latenz- und Verarbeitungszeit (nicht als Metrik verfügbar). Diese Metrik gibt die durchschnittliche Warteschlangentiefe für einen bestimmten Zeitraum an. Amazon RDS meldet die Warteschlangentiefe in Intervallen von 1 Minute. Übliche Werte für die Warteschlangentiefe liegen zwischen null und mehreren hundert.
- `EBSByteBalance%`— Der Prozentsatz der im Burst-Bucket Ihrer RDS-Datenbank verbleibenden Durchsatz-Credits. Diese Metrik ist nur für die grundlegende Überwachung verfügbar. Der Metrikwert basiert auf dem Durchsatz aller Volumes, einschließlich des Root-Volumes, und nicht nur auf den Volumes, die Datenbankdateien enthalten.

  Wenn sich diese Metrik Null nähert, bedeutet dies, dass Ihrer DB-Instance die Rechenkapazität ausgeht. Wenn dies regelmäßig passiert, sollten Sie ein Upgrade auf eine größere Instance-Klassengröße in Betracht ziehen, z. B. von db.r6g.large auf db.r6g.xlarge. Weitere Informationen finden Sie unter DB-Instance-Klasse.
- `ReadIOPS`und I/O — `WriteIOPS` Die Anzahl der pro Sekunde abgeschlossenen Operationen. Diese Metrik gibt die durchschnittliche IOPS für einen bestimmten Zeitraum an. Amazon RDS meldet Lese- und Schreib-IOPS getrennt in 1-Minuten-Intervallen. `TotalIOPS`ist die Summe der Lese- und Schreib-IOPS. Übliche Werte für IOPS liegen zwischen null und mehreren zehntausend pro Sekunde.

  Wenn sich Ihre `TotalIOPS` Werte regelmäßig dem Wert für bereitgestellte IOPS nähern, den Sie für Ihre DB-Instance festgelegt haben, sollten Sie eine Erhöhung der bereitgestellten IOPS (Speichertypen io1, io2 Block Express und gp3) in Betracht ziehen.

Die gemessenen IOPS-Werte sind unabhängig von der Größe der einzelnen Operationen. I/O operation. This means that when you measure I/O performance, make sure to look at the throughput of the instance, not simply the number of I/O

- `ReadLatency`und `WriteLatency` — Die verstrichene Zeit zwischen der Einreichung einer I/O Anfrage und deren Abschluss. Diese Metrik gibt die durchschnittliche Latenz für einen bestimmten Zeitraum an. Amazon RDS gibt Lese- und Schreib-Latenz separat für 1-minütige Intervalle an. Übliche Werte für die Latenz liegen im Millisekundenbereich (ms).
- `ReadThroughput`und `WriteThroughput` — Die Anzahl der Byte pro Sekunde, die auf oder von der Festplatte übertragen werden. Diese Metrik gibt den durchschnittlichen Durchsatz für einen bestimmten Zeitraum an. Amazon RDS meldet den Lese- und Schreibdurchsatz getrennt in 1-Minuten-Intervallen in Einheiten von Byte pro Sekunde (maximale Bandbreite des B/s). Typical values for throughput range from zero to the I/O Kanals).

  Wenn sich Ihre Durchsatzwerte regelmäßig dem maximalen Durchsatz für Ihre DB-Instance nähern, sollten Sie erwägen, mehr Speicherdurchsatz bereitzustellen, wenn Sie den Speichertyp gp3 verwenden.

Faktoren, die die Datenbankleistung beeinflussen

Systemaktivitäten, Datenbank-Workload und DB-Instance-Klasse können sich auf die Datenbankleistung auswirken.

Themen

- Systemaktivitäten
- Datenbank-Workload
- DB-Instance-Klasse

Systemaktivitäten

Die folgenden systembezogenen Aktivitäten verbrauchen I/O Kapazität und können die Leistung der DB-Instance verringern, während sie ausgeführt werden:

- Standby-Erstellung mit Multi-AZ
- Read Replica-Erstellung
- Ändern von Speichertypen

Datenbank-Workload

In einigen Fällen führt der Entwurf Ihrer Datenbank oder Anwendung zu Gleichzeitigkeitsproblemen, Sperren oder anderen Datenbankkonflikten. Dann können Sie die gesamte bereitgestellte Bandbreite möglicherweise nicht direkt nutzen. Außerdem kann es zu folgenden Situationen bezüglich des Workloads kommen:

- Die Durchsatzgrenze des zugrunde liegenden Instance-Typs wird erreicht.
- Die Warteschlangentiefe ist durchweg kleiner als 1, da Ihre Anwendung nicht genügend I/O Vorgänge ausführt.
- In der Datenbank treten Abfragekonflikte auf, obwohl ein Teil der I/O Kapazität ungenutzt ist.

In einigen Fällen gibt es keine Systemressource, die an oder nahe einem Limit liegt, und das Hinzufügen von Threads erhöht die Datenbanktransaktionsrate nicht. In solchen Fällen ist der Engpass höchstwahrscheinlich ein Konflikt in der Datenbank. Die verbreitetsten Formen sind Zeilensperr- und Indexseitensperrkonflikte, aber zahlreiche andere Möglichkeiten bestehen ebenso. Falls dies Ihre Situation beschreibt, sollten Sie den Rat eines Experten für die Optimierung der Datenbankleistung einholen.

DB-Instance-Klasse

Um Ihre Amazon RDS-DB-Instance optimal zu nutzen, sollten Sie einen aktuellen Instance-Typ mit ausreichend Bandbreite für Ihren Speichertyp verwenden. Sie können beispielsweise Amazon-EBS-optimierte Instances und Instances mit einer 10-Gigabit-Netzwerkanbindung nutzen.

> **Important**
>
> Je nach der von Ihnen verwendeten Instance-Klasse kann die IOPS-Leistung unter dem Maximum liegen, das Sie mit RDS bereitstellen können. Spezifische Informationen zur IOPS-Leistung für DB-Instance-Klassen finden Sie unter Amazon EBS — Optimized Instances im EC2 Amazon-Benutzerhandbuch. Wir empfehlen, dass Sie die maximale IOPS-Anzahl für die Instance-Klasse bestimmen, bevor Sie einen bereitgestellten IOPS-Wert für Ihre DB-Instance festlegen.

Wir empfehlen Ihnen, Instances der aktuellen Generation zu verwenden, um die bestmögliche Leistung zu erhalten. DB-Instances der vorherigen Generation können auch einen geringeren maximalen Speicherplatz haben.

Einige ältere 32-Bit-Dateisysteme haben möglicherweise geringere Speicherkapazitäten. Um die Speicherkapazität Ihrer DB-Instance zu ermitteln, können Sie den describe-valid-db-instance Befehl -modifications verwenden. AWS CLI

Die folgende Liste zeigt den maximalen Speicherplatz, auf den die meisten DB-Instance-Klassen für jede Datenbank-Engine skalieren können:

- Db2 — 64 TiB
- MariaDB: 64 TiB
- Microsoft SQL Server — 64 TiB
- MySQL: 64 TiB
- Oracle: 64 TiB
- PostgreSQL: 64 TiB

In der folgenden Tabelle werden einige Ausnahmen für maximalen Speicherplatz (in TiB) angezeigt. Alle RDS für Microsoft SQL Server-DB-Instances mit Ausnahme des io2 Block Express-Speichers haben einen maximalen Speicherplatz von 16 TiB, sodass es keine Einträge für SQL Server gibt.

Instance-Klasse	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.m3 – Standard-Instance-Klassen					
db.t4g – Instance-Klassen mit Spitzenlastleistung					
db.t4g.medium	N/A	16	16	N/A	32
db.t4g.klein	N/A	16	16	N/A	16
db.t4g.micro	N/A	6	6	N/A	6
db.t3 – Instance-Klassen mit Spitzenlastleistung					
db.t3.medium	32	16	16	32	32
db.t3.small	32	16	16	32	16
db.t3.micro	N/A	6	6	32	6

Instance-Klasse	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.t2 – Instance-Klassen mit Spitzenlastleistung					

Weitere Informationen zu allen unterstützten Instance-Klassen finden Sie unter DB-Instances der vorherigen Generation.

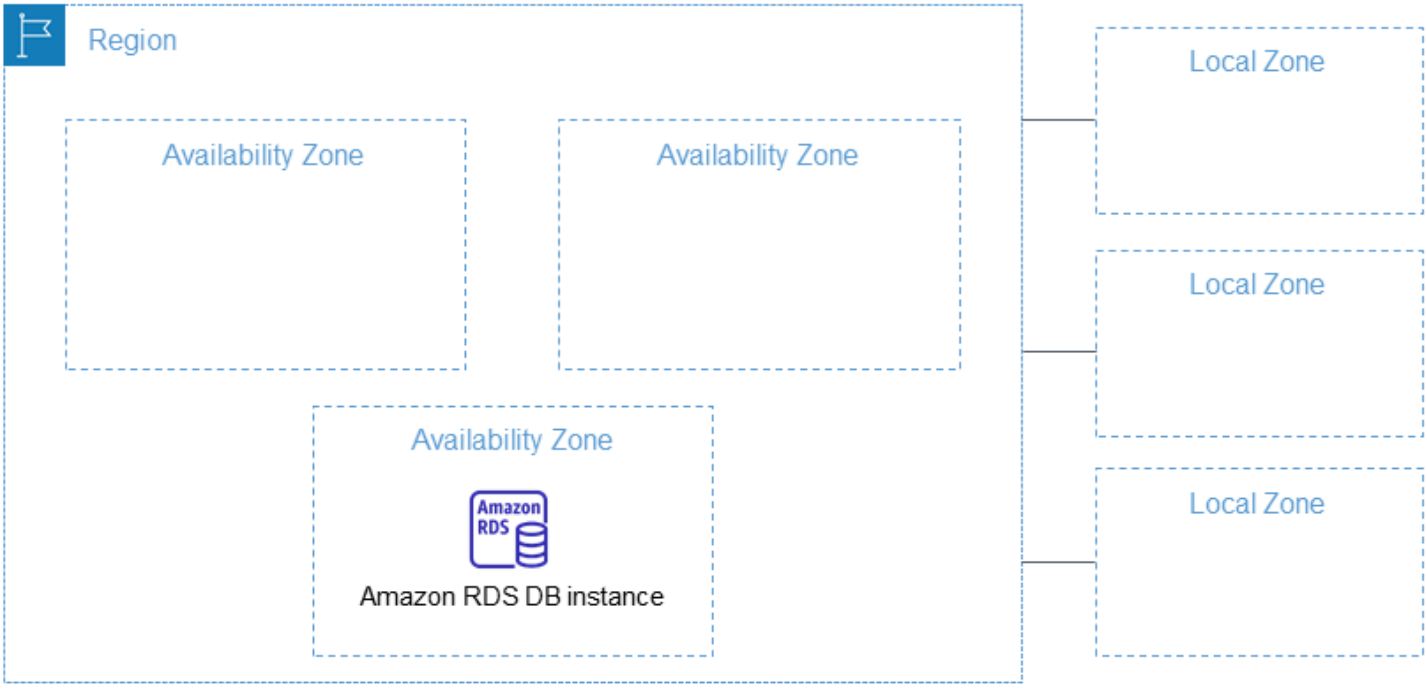
Regionen, Availability Zones und Local Zones

Amazon Cloud Computing-Ressourcen werden an mehreren Standorten weltweit gehostet. Diese Standorte bestehen aus AWS Regionen, Availability Zones und Local Zones. Jede AWS -Region ist ein separater geografischer Bereich. Jede AWS Region hat mehrere isolierte Standorte, die als Availability Zones bezeichnet werden.

> **Note**
>
> Informationen zur Suche nach den Availability Zones für eine AWS Region finden Sie unter Beschreiben Sie Ihre Availability Zones in der EC2 Amazon-Dokumentation.

Durch die Verwendung von Local Zones können Sie Ressourcen wie Computer und Speicher an mehreren Standorten näher bei Ihren Benutzern platzieren. Mithilfe von Amazon RDS können Sie Ressourcen wie DB-Instances und Daten an mehreren Standorten platzieren. Ressourcen werden nicht regionsübergreifend AWS repliziert, es sei denn, Sie tun dies ausdrücklich.

Amazon betreibt state-of-the-art hochverfügbare Rechenzentren. Obwohl selten, können Fehler auftreten, die sich auf die Verfügbarkeit von DB-Instances auswirken, die sich am selben Speicherort befinden. Wenn Sie alle Ihre DB-Instances an einem Ort hosten, der von einem solchen Ausfall betroffen ist, ist keine Ihrer DB-Instances verfügbar.



Es ist wichtig, sich daran zu erinnern, dass jede AWS Region völlig unabhängig ist. Jede von Ihnen initiierte Amazon RDS-Aktivität (z. B. das Erstellen von Datenbank-Instances oder das Auflisten verfügbarer Datenbank-Instances) wird nur in Ihrer aktuellen AWS Standardregion ausgeführt. Die AWS Standardregion kann in der Konsole oder durch Einstellen der `AWS_DEFAULT_REGION`Umgebungsvariablen geändert werden. Oder sie kann überschrieben werden, indem der `--region` Parameter mit der AWS Command Line Interface ()AWS CLI verwendet wird. Weitere Informationen finden Sie unter Konfigurieren der AWS Command Line Interface, insbesondere in den Abschnitten zu Umgebungsvariablen und zu den Befehlszeilenoptionen.

Amazon RDS unterstützt spezielle AWS Regionen namens AWS GovCloud (US). Diese wurden entwickelt, damit US-Regierungsbehörden und Kunden vertrauliche Workloads in die Cloud verschieben können. Die Regionen AWS GovCloud (US) gehen auf die spezifischen regulatorischen und Compliance-Anforderungen der US-Regierung ein. Weitere Informationen finden Sie unter Was ist AWS GovCloud (US)?

Um eine Amazon RDS-DB-Instance in einer bestimmten AWS Region zu erstellen oder mit ihr zu arbeiten, verwenden Sie den entsprechenden regionalen Service-Endpunkt.

AWS Regionen

Jede AWS Region ist so konzipiert, dass sie von den anderen AWS Regionen isoliert ist. Dieser Entwurf sorgt für die größtmögliche Fehlertoleranz und Stabilität.

Wenn Sie Ihre Ressourcen anzeigen, sehen Sie nur die Ressourcen, die mit der von Ihnen angegebenen AWS Region verknüpft sind. Das liegt daran, dass AWS Regionen voneinander isoliert sind und wir Ressourcen nicht automatisch regionsübergreifend AWS replizieren.

Verfügbarkeit in Regionen

Die folgende Tabelle zeigt die AWS Regionen, in denen Amazon RDS derzeit verfügbar ist, und den Endpunkt für jede Region.

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
USA Ost (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)
		rds-fips.us-east-2.api.aws	HTTPS
		rds.us-east-2.api.aws	HTTPS
		rds-fips.us-east-2.amazonaws.com	HTTPS
USA Ost (Nord-Virginia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
USA West (Nordkalifornien)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
USA West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
Afrika (Kapstadt)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
Asien-Pazifik (Hongkong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Asien-Pazifik (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com rds.ap-south-2.api.aws	HTTPS HTTPS
Asien-Pazifik (Jakarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com rds.ap-southeast-3.api.aws	HTTPS HTTPS
Asien-Pazifik (Malaysia)	ap-southeast-5	rds.ap-southeast-5.amazonaws.com	HTTPS
Asien-Pazifik (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS
Asien-Pazifik (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS
Asien-Pazifik (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS
Asien-Pazifik (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Asien-Pazifik (Singapur)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS
		rds.ap-southeast-1.api.aws	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS
		rds.ap-southeast-2.api.aws	HTTPS
Asien-Pazifik (Taipeh)	ap-east-2	rds.ap-east-2.amazonaws.com	HTTPS
Asien-Pazifik (Thailand)	ap-southeast-7	rds.ap-southeast-7.amazonaws.com	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
Kanada (Zentral)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
Kanada West (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Europa (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
Europa (Irland)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
Europa (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
Europa (Mailand)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS
Europa (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS
Europa (Spanien)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
Europa (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
Europa (Zürich)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
Israel (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Mexiko (Zentral)	mx-central-1	rds.mx-central-1.amazonaws.com	HTTPS
Naher Osten (Bahrain)	me-south-1	rds.me-south-1.amazonaws.com rds.me-south-1.api.aws	HTTPS HTTPS
Naher Osten (VAE)	me-central-1	rds.me-central-1.amazonaws.com rds.me-central-1.api.aws	HTTPS HTTPS
Südamerika (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com rds.sa-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (US-Ost)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (US-West)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS

Wenn Sie keinen expliziten Endpunkt festlegen, wird standardmäßig der Endpunkt USA West (Oregon) verwendet.

Wenn Sie mit einer DB-Instance arbeiten, die die API-Operationen AWS CLI oder verwendet, stellen Sie sicher, dass Sie ihren regionalen Endpunkt angeben.

Availability Zones

Wenn Sie eine DB-Instance erstellen, können Sie eine Availability Zone auswählen oder Amazon RDS eine für Sie zufällig auswählen lassen. Eine Availability Zone wird durch einen AWS Regionalcode gefolgt von einer Buchstabenkennung dargestellt (z. B.`us-east-1a`).

Verwenden Sie den describe-availability-zones EC2 Amazon-Befehl wie folgt, um die Availability Zones innerhalb der angegebenen Region zu beschreiben, die für Ihr Konto aktiviert sind.

```
aws ec2 describe-availability-zones --region region-name
```

Wenn Sie beispielsweise die Availability Zones in der Region USA Ost (Nord-Virginia) (us-east-1) beschreiben möchten, die für Ihr Konto aktiviert sind, führen Sie den folgenden Befehl aus:

```
aws ec2 describe-availability-zones --region us-east-1
```

Sie können die Availability Zones für die primären und sekundären DB-Instances in einer Multi-AZ-DB-Bereitstellung nicht auswählen. Amazon RDS wählt sie zufällig für Sie aus. Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

> **Note**
>
> Die zufällige Auswahl von Availability Zones durch RDS garantiert keine gleichmäßige Verteilung von DB-Instances zwischen Availability Zones innerhalb eines einzelnen Kontos oder einer DB-Subnetzgruppe. Sie können eine bestimmte AZ anfordern, wenn Sie eine Single-AZ-Instance erstellen oder ändern, und Sie können spezifischere DB-Subnetzgruppen für Multi-AZ-Instances verwenden. Weitere Informationen erhalten Sie unter Erstellen einer Amazon RDS-DB-Instance und Ändern einer Amazon RDS DB-Instance.

Local Zones

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede VPC aus der übergeordneten AWS -Region auf lokale Zonen erweitern. Erstellen Sie dazu ein neues Subnetz und weisen Sie es der lokalen AWS -Zone zu. Wenn Sie ein Subnetz in einer lokalen Zone erstellen, wird Ihre VPC auf diese lokale Zone erweitert. Das Subnetz in der lokalen Zone funktioniert genauso wie andere Subnetze in Ihrer VPC.

Wenn Sie eine DB-Instance erstellen, können Sie ein Subnetz in einer lokalen Zone auswählen. Local Zones haben ihre eigenen Verbindungen mit dem Internet und unterstützen AWS Direct Connect. Daher können Ressourcen, die in einer lokalen Zone erstellt wurden, von lokalen Benutzern mit Kommunikationen mit sehr geringer Latenz genutzt werden. Weitere Informationen finden Sie unter AWS -Local-Zones.

Eine lokale Zone wird beispielsweise `us-west-2-lax-1a` durch einen AWS Regionalcode dargestellt, gefolgt von einer Kennung, die den Standort angibt.

> **Note**
>
> Eine lokale Zone kann nicht in eine Multi-AZ-Bereitstellung aufgenommen werden.

So verwenden Sie eine lokale Zone

1. Aktivieren Sie die lokale Zone in der EC2 Amazon-Konsole.

   Weitere Informationen finden Sie unter Enabling Local Zones im EC2 Amazon-Benutzerhandbuch.
2. Erstellen Sie ein Subnetz in der Local Zone.

   Weitere Informationen finden Sie unter Erstellen eines Subnetzes in Ihrer VPC im Amazon-VPC-Benutzerhandbuch.
3. Erstellen Sie eine DB-Subnetzgruppe in der lokalen Zone.

   Wenn Sie eine DB-Subnetzgruppe erstellen, wählen Sie die Gruppe der Availability Zones für die lokale Zone aus.

   Weitere Informationen finden Sie unter Erstellen einer DB-Instance in einer VPC.
4. Erstellen Sie eine DB-Instance, welche die DB-Subnetzgruppe in der lokalen Zone verwendet.

   Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

> **Important**
>
> Derzeit ist Los Angeles in der Region USA West (Oregon) die einzige AWS lokale Zone, in der Amazon RDS verfügbar ist.

Unterstützte Funktionen in Amazon RDS by AWS-Region und DB Engine

Die Support für RDS Amazon-Funktionen und -Optionen variiert AWS-Regionen je nach Version der einzelnen DB-Engine. In den folgenden Abschnitten können Sie die Unterstützung und Verfügbarkeit der RDS DB-Engine-Versionen in einer bestimmten AWS-Region Version ermitteln.

Die RDS Funktionen von Amazon unterscheiden sich von den Funktionen und Optionen der Engine. Weitere Informationen zu den Engine-nativen Funktionen und Optionen finden Sie unterEngine-native Funktionen.

Unterstützte Regionen und DB-Engines

- Tabellenkonventionen
- Kurzübersicht über Funktionen
- Unterstützte Regionen und DB-Engines für Amazon RDS Blue/Green-Bereitstellungen
- Unterstützte Regionen und DB-Engines für regionsübergreifende automatisierte Backups in Amazon RDS
- Unterstützte Regionen und DB-Engines für regionsübergreifende Read Replicas in Amazon RDS
- Unterstützte Regionen und DB-Engines für Datenbankaktivitätsstreams in Amazon RDS
- Unterstützte Regionen und DB-Engines für den Dual-Stack-Modus in Amazon RDS
- Unterstützte Regionen und DB-Engines für den Export von Snapshots nach S3 in Amazon RDS
- Unterstützte Regionen und DB-Engines für die IAM-Datenbankauthentifizierung in Amazon RDS
- Unterstützte Regionen und DB-Engines für die Kerberos-Authentifizierung in Amazon RDS
- Unterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS
- Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS
- Unterstützte Regionen und DB-Engines für RDS Custom
- Unterstützte Regionen und DB-Engines für Amazon RDS Proxy
- Unterstützte Regionen und DB-Engines für die Secrets Manager Manager-Integration mit Amazon RDS
- Unterstützte Regionen und DB-Engines für Amazon RDS Zero-ETL-Integrationen
- Engine-native Funktionen in Amazon RDS

Tabellenkonventionen

In den Tabellen in den Funktionsabschnitten werden die folgenden Muster verwendet, um Versionsnummern und den Grad der Verfügbarkeit anzugeben:

- Version x.y – Nur die spezifische Version ist verfügbar.
- Version x.y und höher – Die angegebene Version und alle höheren Unterversionen der jeweiligen Hauptversion werden unterstützt. Zum Beispiel bedeutet „Version 10.11 und höher“, dass die Versionen 10.11, 10.11.1 und 10.12 verfügbar sind.

Kurzübersicht über Funktionen

In der folgenden Kurzübersichtstabelle sind alle Funktionen und verfügbaren RDS DB-Engines aufgeführt. Die Verfügbarkeit von Regionen und spezifischen Versionen wird in den späteren Funktionsabschnitten angezeigt.

Funkti	RDSfür Db2	RDS for MariaDB	RDSfür mein SQL	RDS for Oracle	RDSfür Postgre SQL	RDSfür Server SQL
Blau/ Grün - Bereits ellung	Nicht verfügbar	Verfügbar	Verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar
Regior ergreif de autom ierte Backu	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar
Regior ergreif de	Nicht verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar

Funkti	RDSfür Db2	RDS for MariaDB	RDSfür mein SQL	RDS for Oracle	RDSfür Postgre SQL	RDSfür Server SQL
Lesere kate						
Datenl aktivitä sstrea	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar	Verfügbar
Dual-Stac k-Modus	Nicht verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar
Export en eines Snaps nachc Amazo S3	Nicht verfügbar	Verfügbar	Verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar
AWS Identity and Acces Manag t (IAM) Datenl auther izierun	Nicht verfügbar	Verfügbar	Verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar
Kerber Auther izierun	Verfügbar	Nicht verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar

Funkti	RDSfür Db2	RDS for MariaDB	RDSfür mein SQL	RDS for Oracle	RDSfür Postgre SQL	RDSfür Server SQL
Multi-AZ-DB-Cluster	Nicht verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar
Perfor ce Insight	Nicht verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar
RDSB erdefir rt	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Verfügbar	Nicht verfügbar	Verfügbar
RDSP	Nicht verfügbar	Verfügbar	Verfügbar	Nicht verfügbar	Verfügbar	Verfügbar
Integra on von Secret Manag	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar	Verfügbar

Unterstützte Regionen und DB-Engines für Amazon RDS Blue/Green-Bereitstellungen

blue/green deployment copies a production database environment in a separate, synchronized staging environment. By using Amazon RDS Blue/GreenBei Bereitstellungen können Sie Änderungen an der Datenbank in der Staging-Umgebung vornehmen, ohne die Produktionsumgebung zu beeinträchtigen. Sie können beispielsweise die Haupt- oder Nebenversion der DB-Engine aktualisieren, Datenbankparameter ändern oder Schemaänderungen in der Staging-Umgebung vornehmen. Wenn Sie bereit sind, können Sie die Staging-Umgebung zur neuen

Produktionsdatenbankumgebung hochstufen. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Blaue/grüne Bereitstellungen werden in allen unterstützt. AWS-Regionen

Blaue/grüne Bereitstellungen werden von den folgenden Engines nicht unterstützt:

- RDSfür Db2
- RDSfür Server SQL
- RDS for Oracle

Themen

- Blaue/grüne Bereitstellungen mit für MariaDB RDS
- Blaue/grüne Bereitstellungen mit for My RDS SQL
- Blaue/grüne Bereitstellungen mit für Postgre RDS SQL

Blaue/grüne Bereitstellungen mit für MariaDB RDS

RDSFür MariaDB werden Blau/Grün-Bereitstellungen für die folgenden Versionen unterstützt:

- RDSfür MariaDB 11.4 (Alle verfügbaren Versionen)
- RDSfür MariaDB 10.2 und höhere 10-Versionen

Blaue/grüne Bereitstellungen mit for My RDS SQL

RDSFor MySQL: Blau/Grün-Bereitstellungen werden für die folgenden Versionen unterstützt:

- RDSfür My SQL 8.4 (Alle verfügbaren Versionen)
- RDSfür My SQL 8.0 (Alle verfügbaren Versionen)
- RDSfür My SQL 5.7 (Alle verfügbaren Versionen)

Blaue/grüne Bereitstellungen mit für Postgre RDS SQL

RDSFür Postgre SQL werden blaue/grüne Bereitstellungen für Version 11.1 und alle höheren Haupt- und Nebenversionen unterstützt.

> **Note**
>
> Unter bestimmten Bedingungen SQL verwendet Postgre RDS die logische Replikation anstelle der physischen Replikation, um die grüne Umgebung mit der blauen Umgebung synchron zu halten. Weitere Informationen finden Sie unter the section called “Postgre-Replikationsmethoden SQL”.

Unterstützte Regionen und DB-Engines für regionsübergreifende automatisierte Backups in Amazon RDS

Durch die Verwendung der Backup-Replikation in Amazon RDS können Sie Ihre RDS-DB-Instance so konfigurieren, dass Snapshots und Transaktionsprotokolle in eine Zielregion repliziert werden. Wenn die Backup-Replikation für eine DB-Instance konfiguriert ist, startet RDS eine regionsübergreifende Kopie aller Snapshots und Transaktionsprotokolle, sobald sie bereit sind. Weitere Informationen finden Sie unter Automatisierte Backups auf ein anderes replizieren AWS-Region.

Informationen darüber, AWS-Regionen wo die Backup-Replikation verfügbar ist, finden Sie unterAutomatisierte Backups auf ein anderes replizieren AWS-Region.

Themen

- Backup-Replikation mit RDS für Db2
- Backup-Replikation mit RDS für MariaDB
- Backup-Replikation mit RDS für MySQL
- Backup-Replikation mit RDS für Oracle
- Backup-Replikation mit dem RDS für PostgreSQL
- Backupreplikation mit dem RDS für SQL Server

Backup-Replikation mit RDS für Db2

Amazon RDS unterstützt die Backup-Replikation für alle derzeit verfügbaren Versionen von RDS for Db2.

Backup-Replikation mit RDS für MariaDB

Amazon RDS unterstützt die Backup-Replikation für alle derzeit verfügbaren Versionen von RDS für MariaDB.

Backup-Replikation mit RDS für MySQL

Amazon RDS unterstützt die Backup-Replikation für alle derzeit verfügbaren Versionen von RDS für MySQL.

Backup-Replikation mit RDS für Oracle

Amazon RDS unterstützt die Backup-Replikation für alle derzeit verfügbaren Versionen von RDS für Oracle.

Backup-Replikation mit dem RDS für PostgreSQL

Amazon RDS unterstützt die Backup-Replikation für alle derzeit verfügbaren Versionen von RDS für PostgreSQL.

Backupreplikation mit dem RDS für SQL Server

Amazon RDS unterstützt die Backup-Replikation für alle derzeit verfügbaren Versionen von RDS für SQL Server.

Unterstützte Regionen und DB-Engines für regionsübergreifende Read Replicas in Amazon RDS

Durch die Verwendung von regionsübergreifenden Lesereplikaten in Amazon RDS können Sie ein Lesereplikat von MariaDB, MySQL, Oracle, PostgreSQL oder SQL Server in einer anderen Region als die Quell-DB-Instance erstellen. Weitere Informationen zu regionsübergreifenden Lesereplikaten finden Sie unter Erstellen eines Lesereplikats in einer anderen AWS-Region.

Regionsübergreifende Read Replicas sind für die folgenden Engines nicht verfügbar:

- RDS für Db2

Themen

- Regionsübergreifende Read Replicas mit RDS für MariaDB
- Regionsübergreifende Read Replicas mit RDS für MySQL
- Regionsübergreifende Read Replicas mit RDS für Oracle
- Regionsübergreifende Read Replicas mit RDS für PostgreSQL
- Regionsübergreifende Lesereplikate mit RDS für SQL Server

Regionsübergreifende Read Replicas mit RDS für MariaDB

Regionsübergreifende Read Replicas mit RDS für Maria sind für alle Regionen für die folgenden Versionen verfügbar:

- RDS for MariaDB 11.4 (Alle verfügbaren Versionen)
- RDS für MariaDB 10.11 (alle verfügbaren Versionen)
- RDS für MariaDB 10.6 (alle verfügbaren Versionen)
- RDS für MariaDB 10.5 (alle verfügbaren Versionen)
- RDS für MariaDB 10.4 (alle verfügbaren Versionen)

Regionsübergreifende Read Replicas mit RDS für MySQL

Regionsübergreifende Read Replicas mit RDS für MySQL sind für alle Regionen für die folgenden Versionen verfügbar:

- RDS for MySQL 8.4 (Alle verfügbaren Versionen)
- RDS für MySQL 8.0 (alle verfügbaren Versionen)
- RDS für MySQL 5.7 (alle verfügbaren Versionen)

Regionsübergreifende Read Replicas mit RDS für Oracle

Regionsübergreifende Read Replicas für RDS for Oracle sind in allen AWS-Regionen unterstützten Datenbankversionen mit Enterprise Edition verfügbar. Replikate werden nur in der Konfiguration der CDB-Architektur ohne Mandant CDBs und in der Single-Tenant-Konfiguration unterstützt. Regionsübergreifende Read Replicas werden in der Mehrmandantenkonfiguration der CDB-Architektur nicht unterstützt.

Weitere Informationen zu zusätzlichen Anforderungen für regionsübergreifende Lesereplikate mit RDS für Oracle finden Sie unter Anforderungen und Überlegungen zu Backup und Wiederherstellung für RDS-für-Oracle-Replikate.

Regionsübergreifende Read Replicas mit RDS für PostgreSQL

Regionsübergreifende Read Replicas mit RDS für PostgreSQL sind für alle Regionen für die folgenden Versionen verfügbar:

- RDS for PostgreSQL 17 (Alle verfügbaren Versionen)

- RDS für PostgreSQL 16 (Alle verfügbaren Versionen)
- RDS für PostgreSQL 15 (alle verfügbaren Versionen)
- RDS für PostgreSQL 14 (alle verfügbaren Versionen)
- RDS für PostgreSQL 13 (alle verfügbaren Versionen)
- RDS für PostgreSQL 12 (alle verfügbaren Versionen)
- RDS für PostgreSQL 11 (alle verfügbaren Versionen)
- RDS für PostgreSQL 10 (alle verfügbaren Versionen)

Regionsübergreifende Lesereplikate mit RDS für SQL Server

Regionsübergreifende Lesereplikate mit RDS für SQL Server sind für alle Regionen mit folgenden Ausnahmen verfügbar:

- Afrika (Kapstadt)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Jakarta)
- Asien-Pazifik (Melbourne)
- Kanada West (Calgary)
- Europa (Milan)
- Europa (Spain)
- Europa (Zürich)
- Israel (Tel Aviv)
- Naher Osten (Bahrain)
- Naher Osten (VAE)

Regionsübergreifende Lesereplikate mit RDS für SQL Server sind für die folgenden Versionen verfügbar, die Microsoft SQL Server Enterprise Edition verwenden:

- RDS für SQL Server 2022
- RDS für SQL Server 2019 (Version 15.00.4073.23 und höher)
- RDS für SQL Server 2017 (Version 14.00.3281.6 und höher)
- RDS für SQL Server 2016 (Version 13.00.6300.2 und höher)

Unterstützte Regionen und DB-Engines für Datenbankaktivitätsstreams in Amazon RDS

Durch die Verwendung von Datenbankaktivitätsströmen in Amazon RDS können Sie Alarme für Auditing-Aktivitäten in Ihrer Oracle-Datenbank und SQL Serverdatenbank überwachen und einrichten. Weitere Informationen finden Sie unter Übersicht über Datenbankaktivitätsstreams.

Datenbankaktivitäts-Streams sind mit den folgenden Engines nicht verfügbar:

- RDSfür Db2
- RDS for MariaDB
- RDSfür mein SQL
- RDSfür Postgre SQL

Themen

- Datenbank-Aktivitätsstreams mit RDS für Oracle
- Datenbank-Aktivitätsstreams mit RDS für Server SQL

Datenbank-Aktivitätsstreams mit RDS für Oracle

Die folgenden Regionen und Engine-Versionen sind für Datenbank-Aktivitätsstreams mit RDS für Oracle verfügbar.

Weitere Informationen zu zusätzlichen Anforderungen für Datenbank-Aktivitätsstreams RDS für Oracle finden Sie unterÜbersicht über Datenbankaktivitätsstreams.

Region	RDSfür Oracle 21c	RDSfür Oracle 19c
USA Ost (Nord-Virginia)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
USA Ost (Ohio)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder

Region	RDSfür Oracle 21c	RDSfür Oracle 19c
		mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
USA West (Nordkali fornien)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
USA West (Oregon)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Afrika (Kapstadt)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Hongkong)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Hyderaba d)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Jakarta)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar

Region	RDSfür Oracle 21c	RDSfür Oracle 19c
Asien-Pazifik (Mumbai)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Osaka)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Seoul)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Singapur)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Sydney)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Asien-Pazifik (Tokio)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Kanada (Zentral)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2

Region	RDSfür Oracle 21c	RDSfür Oracle 19c
Kanada West (Calgary)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
China (Peking)	Nicht verfügbar	Nicht verfügbar
China (Ningxia)	Nicht verfügbar	Nicht verfügbar
Europa (Frankfurt)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Europa (Irland)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Europa (London)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Europa (Milan)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Europa (Paris)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2

Region	RDSfür Oracle 21c	RDSfür Oracle 19c
Europa (Spain)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Europa (Stockholm)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Naher Osten UAE ()	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
Südamerika (São Paulo)	Nicht verfügbar	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 und höher, entweder mit Enterprise Edition (EE) oder Standard Edition 2 () SE2
AWS GovCloud (USA Ost)	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar

Datenbank-Aktivitätsstreams mit RDS für Server SQL

Die folgenden Regionen und Engine-Versionen sind für Datenbank-Aktivitätsstreams mit RDS for SQL Server verfügbar.

Weitere Informationen zu zusätzlichen Anforderungen für Datenbank-Aktivitätsstreams RDS für SQL Server finden Sie unterÜbersicht über Datenbankaktivitätsstreams.

Region	RDSfür SQL Server 2019	RDSfür SQL Server 2017	RDSfür SQL Server 2016
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDSfür SQL Server 2019	RDSfür SQL Server 2017	RDSfür SQL Server 2016
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Ningxia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDSfür SQL Server 2019	RDSfür SQL Server 2017	RDSfür SQL Server 2016
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (UAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Unterstützte Regionen und DB-Engines für den Dual-Stack-Modus in Amazon RDS

Durch die Verwendung des Dual-Stack-Modus in RDS können Ressourcen mit einer DB-Instance über Internet Protocol Version 4 (IPv4), Internet Protocol Version 6 (IPv6) oder beide kommunizieren. Weitere Informationen finden Sie unter Dual-Stack-Modus.

Themen

- Dual-Stack-Modus mit RDS für Db2
- Dual-Stack-Modus mit RDS für MariaDB
- Dual-Stack-Modus mit RDS für MySQL
- Dual-Stack-Modus mit RDS für Oracle
- Dual-Stack-Modus mit RDS für PostgreSQL
- Dual-Stack-Modus mit RDS für SQL Server

IPv6 wird im Folgenden für keine Amazon RDS unterstützt AWS-Regionen:

- Asien-Pazifik (Hyderabad)
- Naher Osten (VAE)
- Asien-Pazifik (Malaysia)
- Asien-Pazifik (Melbourne)
- Kanada West (Calgary)
- Europa (Zürich)
- Israel (Tel Aviv)

Dual-Stack-Modus mit RDS für Db2

Die folgenden Regionen und Engine-Versionen sind für den Dual-Stack-Modus mit RDS für Db2 verfügbar.

Region	RDS für Db2 11.5				
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen				
USA Ost (Ohio)	Alle verfügbaren Versionen				
USA West (Nordkalifornien)	Alle verfügbaren Versionen				
USA West (Oregon)	Alle verfügbaren Versionen				
Afrika (Kapstadt)	Alle verfügbaren Versionen				
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen				
Asien-Pazifik (Hyderabad)	Nicht verfügbar				
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen				
Asien-Pazifik (Malaysia)	Nicht verfügbar				
Asien-Pazifik (Melbourne)	Nicht verfügbar				
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen				

Region	RDS für Db2 11.5				
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen				
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen				
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen				
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen				
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen				
Kanada (Zentral)	Alle verfügbaren Versionen				
Kanada West (Calgary)	Nicht verfügbar				
China (Peking)	Nicht verfügbar				
China (Ningxia)	Nicht verfügbar				
Europa (Frankfurt)	Alle verfügbaren Versionen				
Europa (Irland)	Alle verfügbaren Versionen				
Europa (London)	Alle verfügbaren Versionen				

Region	RDS für Db2 11.5				
Europa (Milan)	Alle verfügbaren Versionen				
Europa (Paris)	Alle verfügbaren Versionen				
Europa (Spain)	Nicht verfügbar				
Europa (Stockholm)	Alle verfügbaren Versionen				
Europa (Zürich)	Nicht verfügbar				
Israel (Tel Aviv)	Nicht verfügbar				
Naher Osten (Bahrain)	Alle verfügbaren Versionen				
Naher Osten (VAE)	Nicht verfügbar				
Südamerika (São Paulo)	Alle verfügbaren Versionen				
AWS GovCloud (US-Ost)	Nicht verfügbar				
AWS GovCloud (US-West)	Nicht verfügbar				

Dual-Stack-Modus mit RDS für MariaDB

Die folgenden Regionen und Engine-Versionen sind für den Dual-Stack-Modus mit RDS für MariaDB verfügbar.

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Dual-Stack-Modus mit RDS für MySQL

Die folgenden Regionen und Engine-Versionen sind für den Dual-Stack-Modus mit RDS für MySQL verfügbar.

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Dual-Stack-Modus mit RDS für Oracle

Die folgenden Regionen und Engine-Versionen sind für den Dual-Stack-Modus mit RDS für Oracle verfügbar.

Region	RDS für Oracle 21c	RDS für Oracle 19c
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar

Region	RDS für Oracle 21c	RDS für Oracle 19c
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Nicht verfügbar	Nicht verfügbar
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für Oracle 21c	RDS für Oracle 19c
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Dual-Stack-Modus mit RDS für PostgreSQL

Die folgenden Regionen und Engine-Versionen sind für den Dual-Stack-Modus mit RDS für PostgreSQL verfügbar.

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Dual-Stack-Modus mit RDS für SQL Server

Die folgenden Regionen und Engine-Versionen sind für den Dual-Stack-Modus mit RDS für SQL Server verfügbar.

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Unterstützte Regionen und DB-Engines für den Export von Snapshots nach S3 in Amazon RDS

Sie können RDS DB-Snapshot-Daten in einen Amazon S3-Bucket exportieren. Sie können alle Arten von DB-Snapshots exportieren – einschließlich manueller Snapshots, automatisierter System-Snapshots und vom AWS Backup erzeugte Snapshots. Nachdem die Daten exportiert wurden, können Sie die exportierten Daten direkt mit Tools wie Amazon Athena oder Amazon Redshift Spectrum analysieren. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS.

Das Exportieren von Snapshots nach S3 ist für die folgenden Engines nicht verfügbar:

- RDS für Db2
- RDS für Oracle
- RDS für SQL Server

Themen

- Exportieren von Snapshots nach S3 mit RDS für MariaDB
- Exportieren von Snapshots nach S3 mit RDS für MySQL
- Exportieren von Snapshots nach S3 mit RDS für PostgreSQL

Exportieren von Snapshots nach S3 mit RDS für MariaDB

Die folgenden Regionen und Engine-Versionen sind für die Funktion zum Exportieren von Snapshots in S3 mit RDS für MariaDB verfügbar.

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Exportieren von Snapshots nach S3 mit RDS für MySQL

Die folgenden Regionen und Engine-Versionen sind für den Export von Snapshots nach S3 mit RDS for MySQL verfügbar.

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Exportieren von Snapshots nach S3 mit RDS für PostgreSQL

Die folgenden Regionen und Engine-Versionen sind für die Funktion zum Exportieren von Snapshots in S3 mit RDS für PostgreSQL verfügbar.

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Unterstützte Regionen und DB-Engines für die IAM-Datenbankauthentifizierung in Amazon RDS

Durch die Verwendung der IAM-Datenbankauthentifizierung in Amazon RDS können Sie sich ohne Passwort authentifizieren, um eine Verbindung mit einer DB-Instance herzustellen. Stattdessen verwenden Sie ein Authentifizierungstoken. Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

Die IAM-Datenbank-Authentifizierung ist für folgende Engines und Instance-Klassen verfügbar:

- RDS für Db2
- RDS für Oracle
- RDS für SQL Server

Themen

- IAM-Datenbankauthentifizierung mit RDS für MariaDB
- IAM-Datenbankauthentifizierung mit RDS für MySQL
- IAM-Datenbankauthentifizierung mit RDS für PostgreSQL

IAM-Datenbankauthentifizierung mit RDS für MariaDB

Die folgenden Regionen und Engine-Versionen sind für die IAM-Datenbankauthentifizierung mit RDS für MariaDB verfügbar.

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Malaysia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar

IAM-Datenbankauthentifizierung mit RDS für MySQL

Die IAM-Datenbankauthentifizierung für RDS für MySQL ist in allen Regionen für die folgenden Versionen verfügbar:

- RDS for MySQL 8.4 — Alle verfügbaren Versionen

- RDS für MySQL 8.0 – alle verfügbaren Versionen
- RDS für MySQL 5.7 – alle verfügbaren Versionen

IAM-Datenbankauthentifizierung mit RDS für PostgreSQL

Die IAM-Datenbankauthentifizierung für RDS für PostgreSQL ist in allen Regionen für die folgenden Versionen verfügbar:

- RDS for PostgreSQL 17 — Alle verfügbaren Versionen
- RDS for PostgreSQL 16 — Alle verfügbaren Versionen
- RDS für PostgreSQL 15 – alle verfügbaren Versionen
- RDS für PostgreSQL 14 – alle verfügbaren Versionen
- RDS für PostgreSQL 13 – alle verfügbaren Versionen
- RDS für PostgreSQL 12 – alle verfügbaren Versionen
- RDS für PostgreSQL 11 – alle verfügbaren Versionen
- RDS für PostgreSQL 10 – alle verfügbaren Versionen

Unterstützte Regionen und DB-Engines für die Kerberos-Authentifizierung in Amazon RDS

Durch die Verwendung der Kerberos-Authentifizierung in Amazon RDS können Sie die externe Authentifizierung von Datenbankbenutzern mit Kerberos und Microsoft Active Directory unterstützen. Die Verwendung von Kerberos und Active Directory bietet die Vorteile des Single Sign-Ons und der zentralisierten Authentifizierung von Datenbankbenutzern.

Die Kerberos-Authentifizierung ist für die folgenden Engines nicht verfügbar:

- RDS for MariaDB

Obwohl die meisten AWS Regionen standardmäßig für Ihr AWS Konto aktiv sind, werden bestimmte Regionen nur aktiviert, wenn Sie sie manuell auswählen. Diese Regionen werden als Opt-in-Regionen bezeichnet. Im Gegensatz dazu werden Regionen, die standardmäßig aktiv sind, sobald Ihr AWS Konto erstellt wurde, als kommerzielle Regionen oder einfach Regionen bezeichnet. Für Opt-in-Regionen müssen Sie einen regionalisierten Service Principal des Formulars verwenden. `directoryservice.rds.`*`region_name`*`.amazonaws.com` Für Afrika (Kapstadt) müssen Sie

beispielsweise Service Principal `directoryservice.rds.af-south-1.amazonaws.com` zu Ihrer Vertrauensrichtlinie hinzufügen. Weitere Informationen finden Sie unter Kerberos-Authentifizierung.

Themen

- Kerberos-Authentifizierung mit RDS für Db2
- Kerberos-Authentifizierung mit RDS für MySQL
- Kerberos-Authentifizierung mit RDS für Oracle
- Kerberos-Authentifizierung mit RDS für PostgreSQL
- Kerberos-Authentifizierung mit dem RDS für SQL Server

Kerberos-Authentifizierung mit RDS für Db2

Die folgenden Regionen und Engine-Versionen sind für die Kerberos-Authentifizierung mit RDS für Db2 verfügbar.

Region	RDS für Db2 11.5
USA Ost (Nord-Virginia)	Alle Versionen
USA Ost (Ohio)	Alle Versionen
USA West (Nordkalifornien)	Alle Versionen
USA West (Oregon)	Alle Versionen
Afrika (Kapstadt)	Alle Versionen
Asien-Pazifik (Hongkong)	Nicht verfügbar
Asien-Pazifik (Hyderabad)	Alle Versionen
Asien-Pazifik (Jakarta)	Alle Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar
Asien-Pazifik (Melbourne)	Alle Versionen
Asien-Pazifik (Mumbai)	Alle Versionen

Region	RDS für Db2 11.5
Asien-Pazifik (Osaka)	Nicht verfügbar
Asien-Pazifik (Seoul)	Alle Versionen
Asien-Pazifik (Singapur)	Alle Versionen
Asien-Pazifik (Sydney)	Alle Versionen
Asien-Pazifik (Tokio)	Alle Versionen
Kanada (Zentral)	Alle Versionen
Kanada West (Calgary)	Nicht verfügbar
China (Peking)	Nicht verfügbar
China (Ningxia)	Nicht verfügbar
Europa (Frankfurt)	Alle Versionen
Europa (Irland)	Alle Versionen
Europa (London)	Alle Versionen
Europa (Milan)	Alle Versionen
Europa (Paris)	Nicht verfügbar
Europa (Spain)	Alle Versionen
Europa (Stockholm)	Alle Versionen
Europa (Zürich)	Alle Versionen
Israel (Tel Aviv)	Alle Versionen
Naher Osten (Bahrain)	Alle Versionen
Naher Osten (VAE)	Alle Versionen

Region	RDS für Db2 11.5
Südamerika (São Paulo)	Alle Versionen
AWS GovCloud (US-Ost)	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar

Kerberos-Authentifizierung mit RDS für MySQL

Die folgenden Regionen und Engine-Versionen sind für die Kerberos-Authentifizierung mit RDS für MySQL verfügbar.

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
USA Ost (Nord-Virginia)	Alle Versionen	Alle Versionen	Alle Versionen
USA Ost (Ohio)	Alle Versionen	Alle Versionen	Alle Versionen
USA West (Nordkalifornien)	Alle Versionen	Alle Versionen	Alle Versionen
USA West (Oregon)	Alle Versionen	Alle Versionen	Alle Versionen
Afrika (Kapstadt)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Hongkong)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Hyderabad)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Jakarta)	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Mumbai)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Osaka)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Seoul)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Singapur)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Sydney)	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Tokio)	Alle Versionen	Alle Versionen	Alle Versionen
Kanada (Zentral)	Alle Versionen	Alle Versionen	Alle Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle Versionen	Alle Versionen	Alle Versionen
China (Ningxia)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Frankfurt)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Irland)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (London)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Milan)	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7 (unter RDS Extended Support)
Europa (Paris)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Spain)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Stockholm)	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Zürich)	Alle Versionen	Alle Versionen	Alle Versionen
Israel (Tel Aviv)	Alle Versionen	Alle Versionen	Alle Versionen
Naher Osten (Bahrain)	Alle Versionen	Alle Versionen	Alle Versionen
Naher Osten (VAE)	Alle Versionen	Alle Versionen	Alle Versionen
Südamerika (São Paulo)	Alle Versionen	Alle Versionen	Alle Versionen
AWS GovCloud (US-Ost)	Alle Versionen	Alle Versionen	Alle Versionen
AWS GovCloud (US-West)	Alle Versionen	Alle Versionen	Alle Versionen

Kerberos-Authentifizierung mit RDS für Oracle

Die folgenden Regionen und Engine-Versionen sind für die Kerberos-Authentifizierung mit RDS für Oracle verfügbar.

Region	RDS für Oracle 21c	RDS für Oracle 19c
USA Ost (Nord-Virginia)	Alle Versionen	Alle Versionen
USA Ost (Ohio)	Alle Versionen	Alle Versionen

Region	RDS für Oracle 21c	RDS für Oracle 19c
USA West (Nordkalifornien)	Alle Versionen	Alle Versionen
USA West (Oregon)	Alle Versionen	Alle Versionen
Afrika (Kapstadt) (Opt-in-Region)	Alle Versionen	Alle Versionen
Asien-Pazifik (Hongkong) (Opt-in-Region)	Alle Versionen	Alle Versionen
Asien-Pazifik (Hyderabad) (Opt-in-Region)	Alle Versionen	Alle Versionen
Asien-Pazifik (Jakarta) (Opt-in-Region)	Alle Versionen	Alle Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne) (Opt-in-Region)	Alle Versionen	Alle Versionen
Asien-Pazifik (Mumbai)	Alle Versionen	Alle Versionen
Asien-Pazifik (Osaka)	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Seoul)	Alle Versionen	Alle Versionen
Asien-Pazifik (Singapur)	Alle Versionen	Alle Versionen
Asien-Pazifik (Sydney)	Alle Versionen	Alle Versionen
Asien-Pazifik (Tokio)	Alle Versionen	Alle Versionen
Kanada (Zentral)	Alle Versionen	Alle Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar
China (Peking)	Nicht verfügbar	Nicht verfügbar

Region	RDS für Oracle 21c	RDS für Oracle 19c
China (Ningxia)	Nicht verfügbar	Nicht verfügbar
Europa (Frankfurt)	Alle Versionen	Alle Versionen
Europa (Irland)	Alle Versionen	Alle Versionen
Europa (London)	Alle Versionen	Alle Versionen
Europa (Mailand) (Opt-in-Region)	Alle Versionen	Alle Versionen
Europa (Paris)	Nicht verfügbar	Nicht verfügbar
Europa (Spanien) (Opt-in-Region)	Alle Versionen	Alle Versionen
Europa (Stockholm)	Alle Versionen	Alle Versionen
Europa (Zürich) (Opt-in-Region)	Alle Versionen	Alle Versionen
Israel (Tel Aviv) (Opt-in-Region)	Alle Versionen	Alle Versionen
Naher Osten (Bahrain) (Opt-in-Region)	Alle Versionen	Alle Versionen
Naher Osten (VAE) (Opt-in-Region)	Alle Versionen	Alle Versionen
Südamerika (São Paulo)	Alle Versionen	Alle Versionen
AWS GovCloud (US-Ost)	Alle Versionen	Alle Versionen
AWS GovCloud (US-West)	Alle Versionen	Alle Versionen

Kerberos-Authentifizierung mit RDS für PostgreSQL

Die folgenden Regionen und Engine-Versionen sind für die Kerberos-Authentifizierung mit RDS für PostgreSQL verfügbar.

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
USA Ost (Nord-Virginia)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
USA Ost (Ohio)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
USA West (Nordkalifornien)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
USA West (Oregon)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Afrika (Kapstadt)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Hongkong)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
(Hyderabad)								
Asien-Pazifik (Jakarta)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Mumbai)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Osaka)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Seoul)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Singapur)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Asien-Pazifik (Sydney)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Tokio)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Kanada (Zentral)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
China (Ningxia)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Frankfurt)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Irland)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (London)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Milan)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Europa (Paris)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Spain)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Stockholm)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Zürich)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Israel (Tel Aviv)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Naher Osten (Bahrain)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Naher Osten (VAE)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Südamerika (São Paulo)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
AWS GovCloud (US-Ost)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
AWS GovCloud (US-West)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Kerberos-Authentifizierung mit dem RDS für SQL Server

Die folgenden Regionen und Engine-Versionen sind für die Kerberos-Authentifizierung mit RDS für SQL Server verfügbar.

Region	RDS für SQL Server 2022	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
USA Ost (Nord-Virginia)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
USA Ost (Ohio)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
USA West (Nordkalifornien)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
USA West (Oregon)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Afrika (Kapstadt)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Hongkong)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Hyderabad)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RDS für SQL Server 2022	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
Asien-Pazifik (Melbourne)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Mumbai)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Osaka)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Seoul)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Singapur)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Sydney)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Asien-Pazifik (Tokio)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Kanada (Zentral)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
China (Ningxia)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Frankfurt)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Irland)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (London)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Region	RDS für SQL Server 2022	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
Europa (Milan)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Paris)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Spain)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Stockholm)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Europa (Zürich)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Naher Osten (VAE)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
Südamerika (São Paulo)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
AWS GovCloud (US-Ost)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen
AWS GovCloud (US-West)	Alle Versionen	Alle Versionen	Alle Versionen	Alle Versionen

Unterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS

Eine Multi-AZ-DB-Cluster-Bereitstellung in Amazon RDS ist ein Bereitstellungsmodus für Hochverfügbarkeit von Amazon RDS mit zwei lesbaren Standby-DB-Instances. Ein Multi-AZ-DB-Cluster verfügt über eine Writer-DB-Instance und zwei Reader-DB-Instances in drei separaten Availability Zones in der selben -Region : Multi-AZ-DB-Cluster bieten hohe Verfügbarkeit, erhöhte Kapazität für Lese-Workloads und eine geringere Schreiblatenz im Vergleich zu Multi-

AZ DB-Instance-Bereitstellungen. Weitere Informationen finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS.

Multi-AZ-DB-Cluster sind mit den folgenden Engines nicht verfügbar:

- RDS für Db2
- RDS for MariaDB
- RDS für Oracle
- RDS für SQL Server

Themen

- Multi-AZ DB-Cluster mit RDS für MySQL
- Multi-AZ-DB-Cluster mit dem RDS für PostgreSQL

Multi-AZ DB-Cluster mit RDS für MySQL

Die folgenden Regionen und Engine-Versionen sind für Multi-AZ-DB-Cluster mit RDS für MySQL verfügbar.

Region	RDS für MySQL 8.4	RDS für MySQL 8.0
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Nicht verfügbar	Nicht verfügbar
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar

Sie können die verfügbaren Versionen in einer Region für eine bestimmte DB-Instance-Klasse auflisten, indem Sie AWS CLIÄndern Sie die DB-Instance-Klasse, um die verfügbaren Engine-Versionen dafür anzuzeigen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-orderable-db-instance-options \
--engine mysql \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine mysql ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Multi-AZ-DB-Cluster mit dem RDS für PostgreSQL

Die folgenden Regionen und Engine-Versionen sind für Multi-AZ-DB-Cluster mit RDS für PostgreSQL verfügbar.

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13
USA Ost (Nord-Virginia)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
USA Ost (Ohio)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
USA West (Nordkalifornien)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
USA West (Oregon)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Afrika (Kapstadt)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Hongkong)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Hyderabad)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13
Asien-Pazifik (Jakarta)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Malaysia)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Melbourne)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Mumbai)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Osaka)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Seoul)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Singapur)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13
Asien-Pazifik (Sydney)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Asien-Pazifik (Tokio)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Kanada (Zentral)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Kanada West (Calgary)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
China (Peking)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
China (Ningxia)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (Frankfurt)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13
Europa (Irland)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (London)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (Milan)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (Paris)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (Spain)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (Stockholm)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Europa (Zürich)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13
Israel (Tel Aviv)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Naher Osten (Bahrain)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Naher Osten (VAE)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
Südamerika (São Paulo)	Alle PostgreSQL 17-Versionen	Alle PostgreSQL 16-Versionen	Alle PostgreSQL 15-Versionen	Version 14.5 und höher	Version 13.4 sowie Version 13.7 und höher
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Sie können die verfügbaren Versionen in einer Region für eine bestimmte DB-Instance-Klasse auflisten, indem Sie AWS CLIÄndern Sie die DB-Instance-Klasse, um die verfügbaren Engine-Versionen dafür anzuzeigen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-orderable-db-instance-options \
--engine postgres \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine postgres ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS

Important

AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere

Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

Performance Insights in Amazon RDS erweitert die bestehenden Amazon RDS-Überwachungsfunktionen, um die Leistung Ihrer Datenbank zu veranschaulichen und zu analysieren. Mit dem Performance Insights Dashboard können Sie die Datenbankauslastung Ihrer Amazon RDS-DB-Instance visualisieren. Sie können die Auslastung auch nach Wartezeiten, SQL-Anweisungen, Hosts oder Benutzern filtern. Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.

Performance Insights ist für alle RDS-DB-Engines verfügbar, mit Ausnahme von RDS for Db2.

Für die verfügbaren DB-Engines ist Performance Insights mit allen verfügbaren Engine-Versionen und insgesamt verfügbar AWS-Regionen.

Informationen zur Unterstützung von Regionen, DB-Engine und Instance-Klassen für Performance Insights Insights-Funktionen finden Sie unterDB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen.

Unterstützte Regionen und DB-Engines für RDS Custom

Amazon RDS Custom automatisiert Aufgaben und Abläufe der Datenbankverwaltung. Durch die Verwendung von RDS Custom können Sie als Datenbankadministrator auf Ihre Datenbankumgebung und Ihr Betriebssystem zugreifen und diese anpassen. Mit RDS Custom können Sie die Anforderungen von älteren, benutzerdefinierten und verpackten Anwendungen anpassen. Weitere Informationen finden Sie unter Amazon RDS Custom.

RDS Custom wird nur für die folgenden DB-Engines unterstützt:

Themen

- Unterstützte Regionen und DB-Engines für RDS Custom for Oracle
- Unterstützte Regionen und DB-Engines für RDS Custom for SQL Server

Unterstützte Regionen und DB-Engines für RDS Custom for Oracle

Die folgenden Regionen und Engine-Versionen sind für RDS Custom für Oracle verfügbar.

Region	Oracle Database 19c	Oracle Database 18c	Oracle Datenbank 12c
USA Ost (Nord-Virginia)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
USA Ost (Ohio)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
USA West (Nordkalifornien)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
USA West (Oregon)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Afrika (Kapstadt)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Hongkong)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Asien-Pazifik (Osaka)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher

Region	Oracle Database 19c	Oracle Database 18c	Oracle Datenbank 12c
Asien-Pazifik (Seoul)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Asien-Pazifik (Singapur)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Asien-Pazifik (Sydney)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Asien-Pazifik (Tokio)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Kanada (Zentral)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Ningxia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Frankfurt)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Europa (Irland)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher

Region	Oracle Database 19c	Oracle Database 18c	Oracle Datenbank 12c
Europa (London)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Europa (Milan)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Europa (Paris)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Europa (Stockholm)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (VAE)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
Südamerika (São Paulo)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
AWS GovCloud (US-Ost)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher
AWS GovCloud (US-West)	19c mit der RU/RUR vom Januar 2021 oder höher	18c mit der RU/RUR vom Januar 2021 oder höher	12.1 und 12.2 mit der RU/RUR vom Januar 2021 oder höher

Unterstützte Regionen und DB-Engines für RDS Custom for SQL Server

Sie können RDS Custom für SQL Server bereitstellen, indem Sie entweder eine von RDS bereitgestellte Engine-Version (RPEV) oder eine benutzerdefinierte Engine-Version (CEV) verwenden:

- Wenn Sie eine RPEV verwenden, umfasst sie die Standardinstallation von Amazon Machine Image (AMI) und SQL Server. Wenn Sie das Betriebssystem anpassen oder ändern, werden Ihre Änderungen möglicherweise beim Patchen, bei der Snapshot-Wiederherstellung oder bei der automatischen Wiederherstellung nicht beibehalten.
- Wenn Sie eine CEV verwenden, wählen Sie Ihr eigenes AMI entweder mit vorinstalliertem Microsoft SQL Server oder mit SQL Server aus, den Sie mit Ihren eigenen Medien installieren. Wenn Sie ein AWS bereitgestelltes CEV verwenden, wählen Sie das neueste EC2 Amazon-Image (AMI) AWS, das über das von RDS Custom for SQL Server unterstützte kumulative Update (CU) verfügt. Bei einer CEV können Sie die Konfiguration des Betriebssystems und von SQL Server an Ihre Unternehmensanforderungen anpassen.

Die folgenden Versionen AWS-Regionen und die DB-Engine-Versionen sind für RDS Custom for SQL Server verfügbar. Die Unterstützung der Engine-Version hängt davon ab, ob Sie RDS Custom für SQL Server mit einer RPEV, einer bereitgestellten CEV oder einer von AWS oder einer vom Kunden bereitgestellten CEV verwenden.

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
USA Ost (Nord-Virginia)	SQL Server 2022 Enterprise, Standard oder Web, mit CU9, -GDR CU13, CU14 -GDR, CU15, CU16 CU17, CU18 und. CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
	-GDR, CU29 0 und CU3. CU32	-GDR, CU29 0 und CU3. CU32	-GDR, CU29 0 und CU3. CU32
USA Ost (Ohio)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
USA West (Nordkalifornien)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
USA West (Oregon)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Afrika (Kapstadt)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Asien-Pazifik (Hongkong)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Asien-Pazifik (Hyderabad)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Jakarta)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Malaysia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Melbourne)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Mumbai)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Asien-Pazifik (Osaka)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Asien-Pazifik (Seoul)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Asien-Pazifik (Singapur)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Asien-Pazifik (Sydney)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Asien-Pazifik (Tokio)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Kanada (Zentral)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Kanada West (Calgary)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
China (Ningxia)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Frankfurt)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Europa (Irland)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Europa (London)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Europa (Milan)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Paris)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Europa (Stockholm)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
Europa (Zürich)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (Bahrain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	RPEV	AWS CEV bereitgestellt	Vom Kunden bereitgestellte CEV
Südamerika (São Paulo)	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32	SQL Server 2022 Enterprise, Standard oder Web, mit, -GDR CU9 CU13, CU14 - GDR,, CU15 CU16, CU17 und. CU18 CU19 SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
AWS GovCloud (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovCloud (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Unterstützte Regionen und DB-Engines für Amazon RDS Proxy

Amazon RDS Proxy ist ein vollständig verwalteter, hochverfügbarer Datenbank-Proxy, der Anwendungen durch Bündelung und gemeinsame Nutzung etablierter Datenbankverbindungen skalierbarer macht. Weitere Informationen finden Sie unter Amazon RDS-Proxy .

RDS Proxy ist für die folgenden Engines nicht verfügbar:

- RDS für Db2
- RDS für Oracle

Themen

- RDS Proxy mit RDS für MariaDB

- RDS Proxy mit RDS für MySQL
- RDS Proxy mit RDS für PostgreSQL
- RDS-Proxy mit RDS für SQL Server

RDS Proxy mit RDS für MariaDB

Die folgenden Regionen und Engine-Versionen sind für RDS Proxy mit RDS für MariaDB verfügbar.

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Asien-Pazifik (Malaysia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Thailand)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MariaDB 11.4	RDS für MariaDB 10.11	RDS für MariaDB 10.6	RDS für MariaDB 10.5	RDS für MariaDB 10.4
Mexiko (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

RDS Proxy mit RDS für MySQL

Die folgenden Regionen und Engine-Versionen sind für RDS Proxy mit RDS für MySQL verfügbar.

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Thailand)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für MySQL 8.4	RDS für MySQL 8.0	RDS für MySQL 5.7
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Mexiko (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

RDS Proxy mit RDS für PostgreSQL

Die folgenden Regionen und Engine-Versionen sind für RDS Proxy mit RDS für PostgreSQL verfügbar.

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West	Alle verfügbar	Alle verfügbar	Alle verfügbar	Alle verfügbar	Alle verfügbar	Alle verfügbar	Alle verfügbar	Alle verfügbar

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
(Nordkalifornien)	en Versionen	en Versionen	en Versionen	en Versionen	en Versionen	en Versionen	en Versionen	en Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderaba d)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Thailand)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für PostgreSQL 17	RDS für PostgreSQL 16	RDS für PostgreSQL 15	RDS für PostgreSQL 14	RDS for PostgreSQL 13	RDS für PostgreSQL 12	RDS für PostgreSQL 11	RDS für PostgreSQL 10
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Mexiko (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

RDS-Proxy mit RDS für SQL Server

Die folgenden Regionen und Engine-Versionen sind für RDS Proxy mit RDS für SQL Server verfügbar.

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Hyderabad)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Jakarta)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Malaysia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Melbourne)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Mumbai)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Osaka)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singapur)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydney)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Thailand)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgary)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Peking)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
China (Ningxia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Frankfurt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (London)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Region	RDS für SQL Server 2019	RDS für SQL Server 2017	RDS für SQL Server 2016
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Spain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Stockholm)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Israel (Tel Aviv)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (Bahrain)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Naher Osten (VAE)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Mexiko (Zentral)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Südamerika (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-Ost)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovCloud (US-West)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen

Unterstützte Regionen und DB-Engines für die Secrets Manager Manager-Integration mit Amazon RDS

Mit AWS Secrets Manager können Sie hartcodierte Anmeldeinformationen in Ihrem Code, einschließlich Datenbankkennwörtern, durch einen API-Aufruf an Secrets Manager ersetzen, um das Geheimnis programmgesteuert abzurufen. Weitere Informationen zu Secrets Manager finden Sie im Benutzerhandbuch für AWS Secrets Manager.

Sie können angeben, dass Amazon RDS das Hauptbenutzerpasswort in Secrets Manager für eine DB-Instance von Amazon RDS oder einen Multi-AZ-DB-Cluster verwaltet. RDS generiert das Passwort, speichert es in Secrets Manager und rotiert es regelmäßig. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.

Die Secrets Manager Manager-Integration ist in allen verfügbar AWS-Regionen.

Unterstützte Regionen und DB-Engines für Amazon RDS Zero-ETL-Integrationen

RDS Zero-ETL Integrations ist eine vollständig verwaltete Lösung zur Bereitstellung von Transaktionsdaten in Amazon Redshift oder Amazon SageMaker nachdem sie in eine Amazon RDS-DB-Instance geschrieben wurden. Weitere Informationen finden Sie unter Null-ETL-Integrationen.

Themen

- Zero-ETL-Integrationen mit RDS für MySQL

Zero-ETL-Integrationen mit RDS für MySQL

Die folgenden Regionen und Engine-Versionen sind für RDS for MySQL Zero-ETL-Integrationen mit Amazon Redshift und verfügbar. Amazon SageMaker

Region	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.0	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.0
USA Ost (Nord-Virginia)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA Ost (Ohio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
USA West (Nordkalifornien)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
USA West (Oregon)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Afrika (Kapstadt)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Pazifik (Hongkong)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Paz	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.0	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.0
ifik (Hyder d)				
Asien-Paz ifik (Jakart	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Paz ifik (Malay)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Paz ifik (Melbo e)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Asien-Paz ifik (Mumb	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Asien-Paz ifik (Osaka	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar

Region	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.0	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.0
Asien-Pazifik (Seoul)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Singap)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Sydne	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Asien-Pazifik (Tokio)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada (Zentra	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Kanada West (Calgar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
China (Peking	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.0	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.0
China (Ningxi	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Frankf t)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Irland)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Londo	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Milan)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Paris)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Europa (Spain)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Europa (Stockh m)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
Europa (Zürich	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Israel (Tel Aviv)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Region	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon Redshift für RDS für MySQL 8.0	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.4	Zero-ETL-Integration mit Amazon SageMaker für RDS für MySQL 8.0
Naher Osten (Bahrai	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Nicht verfügbar	Nicht verfügbar
Naher Osten (VAE)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
Südam a (São Paulo)	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen	Alle verfügbaren Versionen
AWS GovClo (US-Ost)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar
AWS GovClo (US-West)	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar	Nicht verfügbar

Engine-native Funktionen in Amazon RDS

Amazon RDS-Datenbank-Engines unterstützen auch viele der gängigsten Engine-nativen Merkmale und Funktionen. Diese Funktionen unterscheiden sich von den auf dieser Seite aufgeführten Amazon RDS-nativen Funktionen. Einige engine-native Funktionen werden möglicherweise nur begrenzt unterstützt oder haben eingeschränkte Berechtigungen.

Weitere Informationen zu Engine-nativen Funktionen finden Sie unter:

- Funktionen von Amazon RDS für Db2
- Unterstützung MariaDB MariaDB-Funktionen bei Amazon RDS
- Unterstützung von MySQL-Funktionen in Amazon RDS
- RDS for Oracle – Funktionen
- Arbeiten mit SQL Postgre-Funktionen, die von Amazon RDS für Postgre unterstützt werden SQL
- Funktionen von Microsoft SQL Server bei Amazon RDS

DB-Instance-Abrechnung für Amazon RDS

RDSAmazon-Instances werden auf der Grundlage der folgenden Komponenten abgerechnet:

- DB-Instance-Stunden (pro Stunde): Basierend auf der DB-Instance-Klasse der DB-Instance (z. B. db.t2.small oder db.m4.large). Die Preise werden pro Stunde angegeben, aber die Rechnungen werden sekundengenau berechnet und die Zeiten werden in Dezimalform angezeigt. RDSDie Nutzung wird in Schritten von 1 Sekunde abgerechnet, mindestens jedoch 10 Minuten. Weitere Informationen finden Sie unter .
- Speicher (pro GiB pro Monat: Die Speicherkapazität, die Sie für die DB-Instance bereitstellen lassen. Bei einer Skalierung Ihrer zur Verfügung gestellten Speicherkapazität im laufenden Monat werden die Gebühren entsprechend anteilig erfasst. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.
- Input/output (I/O) requests (per 1 million requests) – Total number of storage I/OAnfragen, die Sie in einem Abrechnungszeitraum gestellt haben, für RDSAmazon-Magnetspeicher Nur DB-Cluster-Konfiguration.
- Bereitgestellt IOPS (IOPSpro Monat) — IOPS Bereitstellungsrate, unabhängig vom IOPS Verbrauch, für Amazon RDS Provisioned IOPS (SSD) und General Purpose (SSD) gp3-Speicher. Bereitgestellter Speicher für EBS Volumes wird in Schritten von 1 Sekunde abgerechnet, mindestens jedoch 10 Minuten.
- Backup-Speicher (pro GiB pro Monat) – Ein Backup-Speicher ist Speicher, der automatisierten Datenbanksicherungen und allen aktiven Datenbank-Snapshots zugeordnet ist, die Sie erstellt haben. Wenn Sie die Aufbewahrungszeit Ihrer Backups erhöhen oder zusätzliche Datenbank-Snapshots erstellen, belegt Ihre Datenbank dementsprechend mehr Backup-Speicher. Die sekundengenaue Abrechnung gilt nicht für den Backup-Speicher (gemessen in GB/Monat).

  Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.
- Datenübertragung (pro GB) — Datenübertragung in und aus Ihrer DB-Instance vom oder zum Internet und anderen Regionen. AWS Nützliche Beispiele finden Sie im AWS Blogbeitrag Exploring Data Transfer Costs for AWS Managed Databases.

Amazon RDS bietet die folgenden Kaufoptionen, damit Sie Ihre Kosten an Ihre Bedürfnisse anpassen können:

- On-Demand instances (On-Demand-Instances) – Sie bezahlen stundenweise für DB-Instance-Stunden, die Sie nutzen. Die Preise werden pro Stunde angegeben, Rechnungen werden jedoch

sekundengenau berechnet und die Zeiten werden in Dezimalform angezeigt. RDSDie Nutzung wird jetzt in Schritten von einer Sekunde abgerechnet, mindestens jedoch in 10 Minuten.

- Reserved instances (Reserved Instances) – Reservieren Sie eine DB-Instance für einen Zeitraum von einem oder drei Jahren und erhalten Sie einen im Vergleich zu den Preisen für eine On-Demand-DB-Instance erheblichen Rabatt. Durch die Nutzung von Reserved Instances können Sie mehrere Instances innerhalb einer Stunde in Betrieb nehmen, starten, löschen oder beenden und die Reserved Instance-Rabatte für alle Instances erhalten.

RDSPreisinformationen von Amazon finden Sie auf der Seite mit den RDS Amazon-Preisen.

Themen

- On-Demand-DB-Instances für Amazon RDS
- Reservierte DB-Instances für Amazon RDS

On-Demand-DB-Instances für Amazon RDS

On-Demand-DB-Instances von Amazon RDS werden basierend auf der Klasse der DB-Instance abgerechnet (z. B. db.t3.small oder db.m5.large). Informationen zu Amazon RDS-Preisen finden Sie auf der Amazon RDS-Produktseite.

Die Abrechnung für eine DB-Instance beginnt, sobald die DB-Instance verfügbar ist. Die Preise werden auf Stundenbasis aufgeführt, aber Rechnungen werden jetzt auf die Sekunde genau kalkuliert und zeigen die Zeiten im Dezimalformat an. Die Amazon-RDS-Nutzung wird pro Sekunde berechnet, wobei mindestens für 10 Minuten zu zahlen ist. Im Falle einer abrechenbaren Konfigurationsänderung, wie Skalierung der Rechen- oder Speicherkapazität, wird Ihnen eine Mindestdauer von 10 Minuten in Rechnung gestellt. Die Abrechnung wird solange fortgesetzt, bis die DB-Instance beendet wird, was beim Löschen der DB-Instance oder beim Ausfall der DB-Instance passiert.

Wenn Sie nicht mehr mit Gebühren für Ihre DB-Instance belastet werden wollen, müssen Sie diese stoppen oder löschen, um zu vermeiden, dass zusätzliche Stunden der DB-Instance in Rechnung gestellt werden. Weitere Informationen über die Zustände der DB-Instance, die Ihnen in Rechnung gestellt werden, finden Sie unter Anzeigen von Amazon RDS DB-Instance-Status.

Angehaltene DB-Instances

Während Ihre DB-Instance angehalten wird, werden nur Gebühren für bereitgestellten Speicher in Rechnung gestellt, einschließlich bereitgestellter IOPS. Es werden Ihnen auch Gebühren für den Backup-Speicher berechnet, einschließlich des Speichers für manuelle Snapshots und automatische Backups innerhalb des von Ihnen festgelegten Aufbewahrungsfensters. Für DB-Instance-Stunden werden Ihnen keine Gebühren in Rechnung gestellt.

Multi-AZ-DB-Instances

Ein Multi-AZ-Setup verbessert die Haltbarkeit und Verfügbarkeit von Daten, indem es automatisch ein synchrones Standby-Replikat in einer anderen Availability Zone bereitstellt und verwaltet. Aufgrund der zusätzlichen Ressourcen und der erhöhten Verfügbarkeit sind Multi-AZ-Bereitstellungen teurer als Single-AZ-Bereitstellungen und können aufgrund der zusätzlichen Standby-Instance und der zugehörigen Ressourcen etwa doppelt so viel kosten.

Beachten Sie die folgenden wichtigen Informationen zur Multi-AZ-Preisgestaltung:

- Rechenkosten: Wird pro DB-Instance-Stunde sowohl für die Primär- als auch für die Standby-Instance in Rechnung gestellt.

- Speicherkosten: Wird pro GB-Monat für den Speicher berechnet, der sowohl für die Primär- als auch für die Standby-Instance bereitgestellt wird.
- Datenübertragungskosten: Die Replikation zwischen der Primär- und der Standby-Instanz ist in den Kosten enthalten, es können jedoch je nach Nutzung weitere Datenübertragungsgebühren anfallen.

Um Ihre monatlichen Kosten auf der Grundlage Ihres spezifischen Anwendungsfalls genau abzuschätzen AWS-Region, können Sie den verwenden AWS -Preisrechner. Dieses Tool ermöglicht Ihnen die Eingabe Ihrer Konfigurationsdetails und bietet eine umfassende Kostenaufschlüsselung.

> **Note**
>
> Die Preise können sich ändern. Die meisten up-to-date Informationen finden Sie auf der Seite mit den Amazon RDS-Preisen.

Reservierte DB-Instances für Amazon RDS

Mit Reserved DB-Instances können Sie eine DB-Instance für eine ein- oder dreijährige Laufzeit reservieren. Reservierte DB-Instances bieten Ihnen einen deutlichen Rabatt im Vergleich zu den bedarfsorientierten Preisen für DB-Instances. Bei reservierten DB-Instances handelt es sich nicht um physische Instances, sondern um einen Fakturierungsrabatt für die Nutzung gewisser On-Demand-Instances in Ihrem Konto. Rabatte für Reserved DB-Instances sind an den Instance-Typ und die AWS-Region gebunden.

Der allgemeine Prozess für das Arbeiten mit reservierten DB-Instances ist: Zuerst Informationen über verfügbare reservierte DB-Instance-Angebote erhalten, dann ein reserviertes DB-Instance-Angebot kaufen und schließlich Informationen über Ihre vorhandenen reservierten DB-Instances erhalten.

Informationen zum Kauf reservierter DB-Instances und zur Anzeige der Abrechnung für reservierte DB-Instances finden Sie in den folgenden Abschnitten.

- Kauf reservierter DB-Instances für Amazon RDS
- Abrechnung für reservierte DB-Instances für Amazon anzeigen RDS

Übersicht über Reservierte DB-Instances

Wenn Sie eine Reserved DB-Instance in Amazon RDS kaufen, erwerben Sie eine Verpflichtung, eine reduzierte Rate für einen bestimmten DB-Instance-Typ für die Dauer der Reserved DB-Instance zu erhalten. Um eine Amazon RDS Reserved DB-Instance zu verwenden, erstellen Sie eine neue DB-Instance, genau wie bei einer On-Demand-Instance.

Die neu erstellte DB-Instance muss mit den Spezifikationen der Reserved DB-Instance in folgenden Punkten übereinstimmen:

- AWS-Region
- DB-Engine (Die Versionsnummer der DB-Engine muss nicht übereinstimmen.)
- DB-Instance-Typ
- Größe der DB-Instance (Lizenz für RDS für Microsoft SQL Server und Amazon RDS for Oracle enthalten)
- Edition (RDS für SQL Server und RDS für Oracle)
- Lizenztyp (inklusive Lizenz oder) bring-your-own-license

Wenn die Spezifikationen der neuen DB-Instance mit einer vorhandenen Reserved DB-Instance für Ihr Konto übereinstimmen, wird Ihnen der angebotene diskontierte Preis für die Reserved DB-Instance in Rechnung gestellt. Andernfalls wird der DB-Instance eine On-Demand-Rate berechnet.

Sie können eine DB-Instance, die Sie als Reserved-DB-Instance verwenden, ändern. Wenn die Änderung innerhalb der Spezifikationen der reservierten DB-Instance liegt, gilt der diskontierte Preis teilweise oder vollständig für die geänderte DB-Instance. Wenn die Änderung außerhalb der Spezifikationen liegt, z. B. die Änderung der Instance-Klasse, gilt der diskontierte Preis nicht mehr. Weitere Informationen finden Sie unter Größenflexible Reservierte DB-Instances.

Themen

- Angebotstypen
- Größenflexible Reservierte DB-Instances
- Abrechnungsbeispiel für Reserved DB-Instances
- Reserved DB-Instances für einen Multi-AZ-DB-Cluster
- Löschen einer Reserved DB-Instance

Weitere Informationen zu reservierten DB-Instances samt Preisen finden Sie unter Amazon RDS Reserved Instances.

Angebotstypen

Reservierte DB-Instances sind in drei Varianten verfügbar – Keine Vorauszahlung, Teilweise Vorauszahlung und Vollständige Vorauszahlung. Auf diese Weise können Sie die Amazon RDS-Kosten auf der Basis der erwarteten Nutzung optimieren.

> **Note**
>
> Nicht alle RDS-Instance-Klassen unterstützen alle Reserved Instance-Angebotstypen. Beispielsweise bieten einige Instance-Klassen möglicherweise nicht die Option „Keine Vorauszahlung“ an. Um die Verfügbarkeit zu überprüfen, überprüfen Sie die Reserved Instance-Angebote im AWS Management Console oder verwenden Sie den `describe-reserved-db-instances-offerings` AWS CLI Befehl.

Keine Vorabzahlung

Diese Option ermöglicht den Zugriff auf eine reservierte DB-Instance, ohne dass eine Vorauszahlung erforderlich ist. Ihre reservierte DB-Instance ohne Vorauszahlung rechnet für jede Stunde innerhalb der Laufzeit einen ermäßigten Stundensatz ab, unabhängig von der Nutzung, und es ist keine Vorauszahlung erforderlich. Diese Option ist lediglich als Reservierung für die Dauer eines Jahres verfügbar.

Teilweise Vorauszahlung

Diese Option beruht darauf, dass ein Teil der reservierten DB-Instance im Voraus bezahlt wird. Die innerhalb der Laufzeit verbleibenden Stunden werden unabhängig von der Nutzung zu einem vergünstigten Stundensatz berechnet. Diese Option ersetzt die vorherige Heavy Utilization-Option.

Komplette Vorauszahlung

Die vollständige Zahlung erfolgt zu Beginn der Laufzeit, unabhängig von der Nutzungsdauer und ohne weitere Kosten innerhalb der Restlaufzeit.

Wenn Sie die konsolidierte Fakturierung nutzen, werden alle Konten in der Organisation wie ein einziges Konto behandelt. Das bedeutet, dass alle Konten in der Organisation den stündlichen Kostenvorteil für Reserved DB-Instances erhalten können, die durch ein anderes Konto erworben wurden. Weitere Informationen zur konsolidierten Fakturierung finden Sie unter Reservierte Amazon-RDS-DB-Instances im Benutzerhandbuch AWS -Fakturierungs- und Kostenverwaltung.

Größenflexible Reservierte DB-Instances

Beim Kauf einer Reserved DB-Instance wird unter anderem die Instance-Klasse angegeben, z. B. db.r5.large. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Wenn Sie eine DB-Instance haben und diese auf größere Kapazität skalieren müssen, wird Ihre Reserved DB-Instance automatisch auf Ihre skalierte DB-Instance angewendet. Das heißt, Ihre Reserved DB-Instances werden automatisch auf alle DB-Instance-Klassengrößen angewendet. Größenflexible reservierte DB-Instances sind für DB-Instances mit derselben AWS-Region DB-Engine verfügbar. Größenflexible Reserved DB-Instances können nur innerhalb ihres Instance-Klassentyps skalieren. Eine Reserved DB-Instance für das Modell db.r5.large kann beispielsweise auf das Modell db.r5.xlarge angewendet werden, nicht aber auf db.r6g.large, da es sich bei db.r5 und db.r6g um unterschiedliche Instance-Klassentypen handelt.

Die Vorteile reservierter DB-Instances gelten sowohl für Multi-AZ- als auch für Single-AZ-Konfigurationen. Das bedeutet, dass Sie innerhalb desselben DB-Instance-Klassentyps frei zwischen

Konfigurationen wechseln können. Sie können beispielsweise von einer Single-AZ-Bereitstellung, die auf einer großen DB-Instance (vier normalisierte Einheiten pro Stunde) ausgeführt wird, zu einer Multi-AZ-Bereitstellung wechseln, die auf zwei mittleren DB-Instances ausgeführt wird (2+2 = 4 normalisierte Einheiten pro Stunde).

Größenflexible reservierte DB-Instances sind für die folgenden Amazon RDS-Datenbank-Engines verfügbar:

- RDS for MariaDB
- RDS for MySQL
- RDS für Oracle, bringen Sie Ihre eigene Lizenz mit
- RDS for PostgreSQL

Größenflexibilität gilt nicht für RDS für SQL Server und RDS für Oracle (Lizenz enthalten).

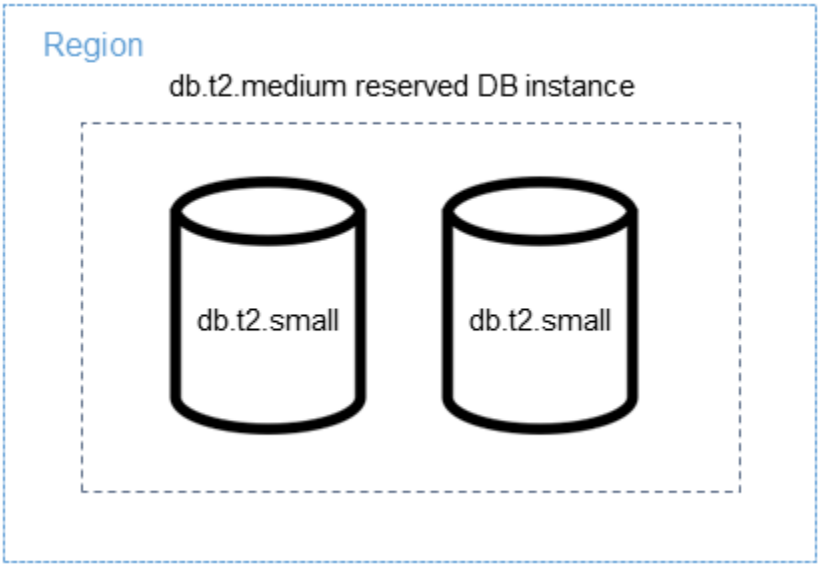
Einzelheiten zur Verwendung von größenflexiblen Reserved Instances mit Aurora finden Sie unter Reserved DB Instances für Aurora.

Sie können die Nutzung für verschiedene Reserved DB-Instance-Größen vergleichen, indem Sie normalisierte Einheiten pro Stunde verwenden. Beispielsweise entspricht eine Nutzungseinheit auf zwei db.r3.large DB-Instances acht normalisierten Nutzungseinheiten pro Stunde auf einem db.r3.small. Die folgende Tabelle zeigt die Anzahl von normalisierten Einheiten pro Stunde für jede DB-Instance-Größe.

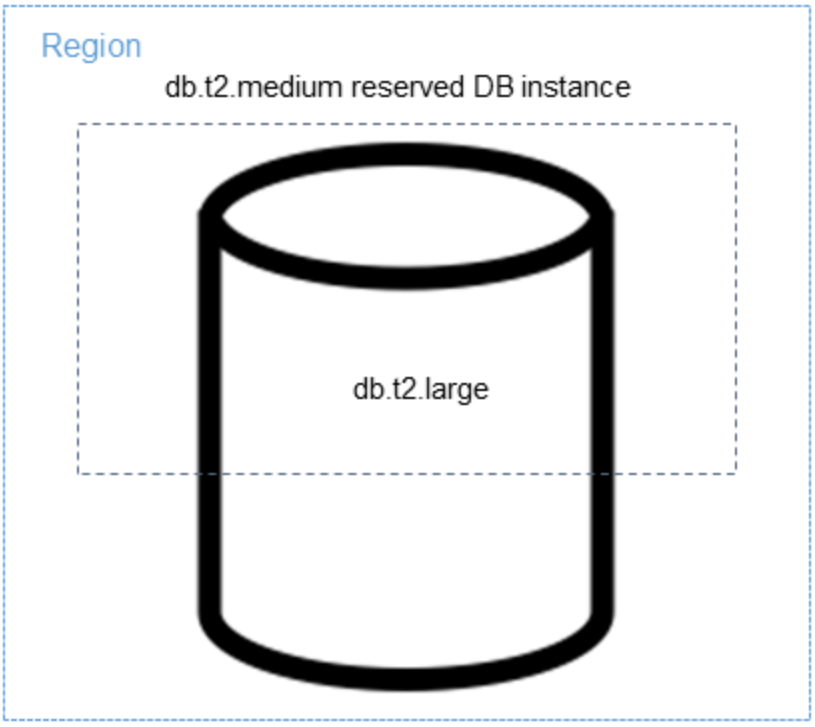
Instance-Größe	Normalisierte Single-AZ-Einheiten pro Stunde (Bereitstellung mit einer DB-Instance)	Normalisierte Einheiten der Multi-AZ-DB-Instance pro Stunde (Bereitstellung mit einer DB-Instance und einer Standby-Instance)	Normalisierte Einheiten des Multi-AZ-DB-Clusters pro Stunde (Bereitstellung mit einer DB-Instance und zwei Standby-Instances)
Micro	0.5	1	1.5
small	1	2	3
Medium	2	4	6

Instance-Größe	Normalisierte Single-AZ-Einheiten pro Stunde (Bereitstellung mit einer DB-Instance)	Normalisierte Einheiten der Multi-AZ-DB-Instance pro Stunde (Bereitstellung mit einer DB-Instance und einer Standby-Instance)	Normalisierte Einheiten des Multi-AZ-DB-Clusters pro Stunde (Bereitstellung mit einer DB-Instance und zwei Standby-Instances)
large	4	8	12
xlarge	8	16	24
2xlarge	16	32	48
4xlarge	32	64	96
6xlarge	48	96	144
8xlarge	64	128	192
10xlarge	80	160	240
12xlarge	96	192	288
16xlarge	128	256	384
24xlarge	192	384	576
32xlarge	256	512	768

Angenommen, Sie kaufen eine Reserved `db.t2.medium`-DB-Instance und haben zwei `db.t2.small`-DB-Instances in Ihrem Konto, die in der gleichen AWS-Region ausgeführt werden. In diesem Fall wird der Rabatt in vollem Umfang auf beide Instances angewendet.



Wenn in Ihrem Konto eine `db.t2.large` Instance ausgeführt wird AWS-Region, wird der Abrechnungsvorteil alternativ auf 50 Prozent der Nutzung der DB-Instance angerechnet.



Abrechnungsbeispiel für Reserved DB-Instances

Der Preis für eine Reserved DB-Instance enthält keinen Rabatt für die Kosten in Verbindung mit Speicher, Backups und I/O. Er bietet nur einen Rabatt auf die stündliche Nutzung der On-Demand-Instance. Das folgende Beispiel zeigt die monatlichen Gesamtkosten für eine Reserved DB-Instance:

- Eine Single-AZ-DB-Instance von RDS für MySQL der Klasse db.r5.large in der Region USA Ost (Nord-Virginia) mit der Option „keine Vorauszahlung“ kostet 0,12 USD pro Instance oder 90 USD pro Monat.
- 400 GiB Universal-SSD (gp2)-Speicher zu einem Preis von 0,115 USD pro GiB pro Monat oder 45,60 USD pro Monat.
- 600 GiB Sicherungsspeicher für 0,095 USD oder 19 USD pro Monat (400 GiB kostenlos).

Wenn Sie all diese Kosten (90 USD + 45,60 USD + 19 USD) mit der Reserved DB-Instance addieren, liegen die monatlichen Gesamtkosten bei 154,60 USD.

Wenn Sie sich statt für eine Reserved DB-Instance für eine On-Demand-DB-Instance entscheiden, kostet eine Single-AZ-Instance der Klasse db.r5.large von RDS für MySQL in der Region USA Ost (Nord-Virginia) 0,1386 USD pro Stunde oder 101,18 USD pro Monat. Wenn Sie die Kosten all dieser Optionen (101,18 USD + 45,60 USD + 19 USD) mit der On-Demand-DB-Instance addieren, liegen die monatlichen Gesamtkosten bei 165,78 USD. Sie sparen etwas mehr als 11 USD pro Monat, indem Sie die reservierte DB-Instance verwenden.

> **Note**
>
> Die Preise in diesem Beispiel sind Beispielpreise und entsprechen möglicherweise nicht den tatsächlichen Preisen. Informationen zu Amazon-RDS-Preisen finden Sie unter Amazon-RDS-Preise.

Reserved DB-Instances für einen Multi-AZ-DB-Cluster

Die entsprechenden Reserved DB-Instances für einen Multi-AZ-DB-Cluster können Sie wie folgt erwerben:

- Reservieren Sie drei Single-AZ-DB-Instances, die dieselbe Größe wie die Instances im Cluster haben.

- Reservieren Sie eine Multi-AZ-DB-Instance und eine Single-AZ-DB-Instance, die die gleiche Größe wie die DB-Instances im Cluster haben.

Angenommen, Sie haben einen Cluster, der aus drei DB-Instances vom Typ db.m6gd.large besteht. In diesem Fall können Sie entweder drei Single-AZ Reserved DB-Instances vom Typ db.m6gd.large oder eine Multi-AZ Reserved DB-Instance vom Typ db.m6gd.large und eine Single-AZ Reserved DB-Instance vom Typ db.m6gd.large erwerben. Bei jeder dieser Optionen wird der maximale Reserved-Instance-Rabatt für den Multi-AZ-DB-Cluster reserviert.

Alternativ können Sie größenflexible DB-Instances verwenden und eine größere DB-Instance erwerben, um kleinere DB-Instances in einem oder mehreren Clustern abzudecken. Wenn Sie beispielsweise zwei Cluster mit insgesamt sechs DB-Instances vom Typ db.m6gd.large haben, können Sie drei Single-AZ Reserved DB-Instances vom Typ db.m6gd.xl erwerben. Dadurch werden alle sechs DB-Instances in den beiden Clustern reserviert. Weitere Informationen finden Sie unter Größenflexible Reservierte DB-Instances.

Sie können DB-Instances reservieren, die dieselbe Größe wie die DB-Instances im Cluster haben, aber weniger DB-Instances als die Gesamtzahl der DB-Instances im Cluster. In diesem Fall ist der Cluster jedoch nur teilweise reserviert. Angenommen, Sie haben einen Cluster mit drei DB-Instances vom Typ db.m6gd.large und erwerben eine Multi-AZ Reserved DB-Instance vom Typ db.m6gd.large. In diesem Fall ist der Cluster nur teilweise reserviert, da nur zwei der drei Instances im Cluster durch Reserved DB-Instances abgedeckt werden. Die verbleibende DB-Instance wird zum On-Demand-Stundensatz von db.m6gd.large abgerechnet.

Weitere Informationen zu Multi-AZ-DB-Clustern finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS.

Löschen einer Reserved DB-Instance

In den Bedingungen für eine Reserved DB-Instance ist eine einjährige oder dreijährige Verpflichtung enthalten. Sie können eine Reserved DB-Instance nicht stornieren. Sie können jedoch eine DB-Instance löschen, die durch einen Rabatt für eine Reserved DB-Instance abgedeckt ist. Der Vorgang zum Löschen einer DB-Instance, für die ein Rabatt für eine Reserved DB-Instance gilt, ist der gleiche wie für jede andere DB-Instance.

Die Vorabkosten werden Ihnen in Rechnung gestellt, unabhängig davon, ob Sie die Ressourcen nutzen.

Wenn Sie eine DB-Instance löschen, die durch einen Rabatt für eine Reserved DB-Instance gedeckt ist, können Sie eine andere DB-Instance mit kompatiblen Spezifikationen starten und den ermäßigten Preis während der Reservierungslaufzeit (ein Jahr oder drei Jahre) erhalten. In diesem Fall erhalten Sie den Rabatt während des Reservierungszeitraums (ein Jahr oder drei Jahre).

Kauf reservierter DB-Instances für Amazon RDS

Sie können die AWS Management Console, und die RDS-API verwenden AWS CLI, um mit reservierten DB-Instances zu arbeiten.

Konsole

Sie können die verwenden AWS Management Console , um mit reservierten DB-Instances zu arbeiten, wie in den folgenden Verfahren gezeigt.

Preise und Informationen zu verfügbaren Angeboten für reservierte DB-Instances erhalten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Reserved instances (Reservierte Instances) aus.
3. Klicken Sie auf Reserved DB-Instance kaufen.
4. Um die Produktbeschreibung einzusehen, wählen Sie die DB-Engine und den Lizenztyp aus.
5. Wählen Sie für DB-Instance-Klasse die DB-Instance-Klasse aus.
6. Wählen Sie für Bereitstellungsoption aus, ob Sie eine Single-AZ- oder eine Multi-AZ-Bereitstellung der DB-Instance wünschen.

   > **Note**
   >
   > Wenn Sie die entsprechenden Reserved-DB-Instances für eine Multi-AZ-DB-Cluster-Bereitstellung möchten, erwerben Sie entweder drei Single-AZ Reserved DB-Instances oder eine Multi-AZ und eine Single-AZ Reserved DB-Instance. Weitere Informationen finden Sie unter Reserved DB-Instances für einen Multi-AZ-DB-Cluster.

7. Für Laufzeit wählen Sie die Zeitspanne aus, für welche die DB-Instance reserviert werden soll.
8. Wählen Sie für Angebotstyp den Angebotstyp aus.

   Nachdem Sie die Angebotsart ausgewählt haben, werden Ihnen die Preisinformationen angezeigt.

> **Important**
>
> Klicken Sie auf Abbrechen, um den Kaufvorgang für die Reserved DB-Instance abzubrechen und Kosten zu vermeiden.

Nachdem Sie Informationen über die verfügbaren reservierten DB-Instance-Angebote erhalten haben, können Sie diese Informationen verwenden, um ein Angebot zu erwerben, wie in der folgenden Vorgehensweise gezeigt.

Kauf einer reservierten DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Reserved instances (Reservierte Instances) aus.
3.
> **Important**
>
> Bevor Sie fortfahren, stellen Sie sicher, dass Sie auf der richtigen Seite sind AWS-Region. Reservierte DB-Instances sind regionsspezifisch und können nicht zwischen Regionen übertragen werden. Überprüfen Sie die Regionsauswahl in der oberen rechten Ecke der Konsole, um sicherzustellen, dass Sie die Reserved Instance in der gewünschten Region erwerben.
4. Klicken Sie auf Reserved DB-Instance kaufen.
5. Um die Produktbeschreibung einzusehen, wählen Sie die DB-Engine und den Lizenztyp aus.
6. Wählen Sie für DB-Instance-Klasse die DB-Instance-Klasse aus.
7. Wählen Sie für Multi-AZ-Bereitstellung aus, ob Sie eine Single-AZ- oder eine Multi-AZ-Bereitstellung der DB-Instance wünschen.

> **Note**
>
> Wenn Sie die entsprechenden Reserved-DB-Instances für eine Multi-AZ-DB-Cluster-Bereitstellung möchten, erwerben Sie entweder drei Single-AZ Reserved DB-Instances oder eine Multi-AZ und eine Single-AZ Reserved DB-Instance. Weitere Informationen finden Sie unter Reserved DB-Instances für einen Multi-AZ-DB-Cluster.

8. Für die Auswahl des Laufzeit wählen Sie die Zeitspanne aus, für welche die DB-Instance reserviert werden soll.
9. Wählen Sie für Angebotstyp den Angebotstyp aus.

   Nachdem Sie die Angebotsart ausgewählt haben, werden Ihnen die Preisinformationen angezeigt.
10. (Optional) Sie können den Reserved DB-Instances, die Sie kaufen, Ihre eigene Kennung zuweisen, um die Übersicht zu behalten. Geben Sie unter Reservierte ID eine Kennzeichnung für Ihre reservierte DB-Instance ein.
11. Wählen Sie Absenden aus.

    Ihre Reserved DB-Instance wird gekauft und dann in der Liste Reserved Instances angezeigt.

Nachdem Sie reservierte DB-Instances gekauft haben, erhalten Sie Informationen über Ihre reservierten DB-Instances, wie in der folgenden Vorgehensweise gezeigt.

Um Informationen zu reservierten DB-Instances für Ihr Konto zu erhalten AWS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Bereich Navigation die Option Reserved Instances (Reservierte Instances) aus.

   Es erscheinen die reservierten DB-Instances für Ihr Konto. Um detaillierte Informationen zu einer bestimmten Reserved DB-Instance anzuzeigen, wählen Sie diese in der Liste aus. Sie können dann detaillierte Informationen über diese Instance im Detailbereich am unteren Rand der Konsole anzeigen.

AWS CLI

Sie können die verwenden AWS CLI , um mit reservierten DB-Instances zu arbeiten, wie in den folgenden Beispielen gezeigt.

Example Erhalten von verfügbaren Reserved DB-Instance-Angeboten

Rufen Sie den AWS CLI Befehl auf, um Informationen über verfügbare Angebote für reservierte DB-Instances zu erhalten `describe-reserved-db-instances-offerings`.

```
aws rds describe-reserved-db-instances-offerings
```

Diese Aktion führt zu folgender oder einer ähnlichen Ausgabe:

```
OFFERING  OfferingId                            Class         Multi-AZ  Duration  Fixed
 Price  Usage Price  Description  Offering Type
OFFERING  438012d3-4052-4cc7-b2e3-8d3372e0e706  db.r3.large   y         1y
 1820.00 USD  0.368 USD    mysql        Partial  Upfront
OFFERING  649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  db.r3.small   n         1y
 227.50 USD  0.046 USD    mysql        Partial  Upfront
OFFERING  123456cd-ab1c-47a0-bfa6-12345667232f  db.r3.small   n         1y
 162.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   0.123   USD       Hourly
OFFERING  123456cd-ab1c-37a0-bfa6-12345667232d  db.r3.large   y         1y
 700.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   1.25    USD       Hourly
OFFERING  123456cd-ab1c-17d0-bfa6-12345667234e  db.r3.xlarge  n         1y
 4242.00 USD   2.42 USD    mysql        No       Upfront
```

Nachdem Sie Informationen über die verfügbaren reservierten DB-Instance-Angebote erhalten haben, können Sie diese Informationen verwenden, um ein Angebot zu erwerben.

Um eine reservierte DB-Instance zu erwerben, verwenden Sie den AWS CLI Befehl [purchase-reserved-db-instances-offering](#)mit den folgenden Parametern:

- `--reserved-db-instances-offering-id` – Die ID des Angebots, das Sie erwerben möchten Siehe das vorhergehende Beispiel, um die Angebots-ID zu erhalten.
- `--reserved-db-instance-id` – Sie können den Reserved DB-Instances, die Sie kaufen, Ihre eigene Kennung zuweisen, um die Übersicht zu behalten.

Example Kauf einer Reserved DB-Instance

Im folgenden Beispiel wird das reservierte DB-Instance-Angebot mit der ID *649fd0c8-cf6d-47a0-bfa6-060f8e75e95f* gekauft und der Identifier von *MyReservation* zugewiesen.

Für LinuxmacOS, oderUnix:

```
aws rds purchase-reserved-db-instances-offering \
    --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
```

```
    --reserved-db-instance-id MyReservation
```

Windows:

```
aws rds purchase-reserved-db-instances-offering ^
    --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
    --reserved-db-instance-id MyReservation
```

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht:

```
RESERVATION  ReservationId      Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State           Description  Offering Type
RESERVATION  MyReservation      db.r3.small  y         2011-12-19T00:30:23.247Z  1y
    455.00 USD   0.092 USD    1      payment-pending  mysql        Partial  Upfront
```

Nachdem Sie Reserved DB-Instances gekauft haben, erhalten Sie Informationen über Ihre Reserved DB-Instances.

Um Informationen über reservierte DB-Instances für Ihr AWS Konto zu erhalten, rufen Sie den AWS CLI Befehl auf `describe-reserved-db-instances`, wie im folgenden Beispiel gezeigt.

Example Erhalten von Reserved DB-Instances

```
aws rds describe-reserved-db-instances
```

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht:

```
RESERVATION  ReservationId     Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State    Description  Offering Type
RESERVATION  MyReservation     db.r3.small  y         2011-12-09T23:37:44.720Z  1y
   455.00 USD   0.092 USD    1      retired  mysql        Partial  Upfront
```

RDS-API

Sie können die RDS-API verwenden, um mit Reserved DB-Instances zu arbeiten:

- Um Informationen zu verfügbaren Reserved DB-Instance-Angeboten zu erhalten, rufen Sie die Amazon RDS-API-Aktion auf `DescribeReservedDBInstancesOfferings`.
- Nachdem Sie Informationen über die verfügbaren reservierten DB-Instance-Angebote erhalten haben, können Sie diese Informationen verwenden, um ein Angebot zu erwerben. Rufen Sie

dazu die RDS-API-Operation `PurchaseReservedDBInstancesOffering` mit den folgenden Parametern auf:

- `--reserved-db-instances-offering-id` – Die ID des Angebots, das Sie erwerben möchten
- `--reserved-db-instance-id` – Sie können den Reserved DB-Instances, die Sie kaufen, Ihre eigene Kennung zuweisen, um die Übersicht zu behalten.

- Nachdem Sie Reserved DB-Instances gekauft haben, erhalten Sie Informationen über Ihre Reserved DB-Instances. Rufen Sie die Aktion `DescribeReservedDBInstances`-RDS-API auf.

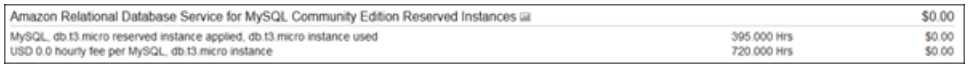
Abrechnung für reservierte DB-Instances für Amazon anzeigen RDS

Sie können die Abrechnung für Ihre reservierten DB-Instances im Abrechnungs-Dashboard im AWS Management Console.

So zeigen Sie Reserved DB-Instance Abrechnung an

1. Melden Sie sich bei der an AWS Management Console.
2. Wählen Sie im Kontomenü oben rechts Abrechnungs-Dashboard aus.
3. Wählen Sie oben rechts im Dashboard Rechnungsdetails aus.
4. Erweitern Sie unter AWS -Servicegebühren den Dienst für relationale Datenbanken.
5. Erweitern Sie den AWS-Region Ort, an dem sich Ihre reservierten DB-Instances befinden, z. B. US West (Oregon).

   Ihre reservierten DB-Instances und deren Stundengebühren für den aktuellen Monat werden unter Amazon Relational Database Service für angezeigt ***Database Engine*** Reservierte Instances.

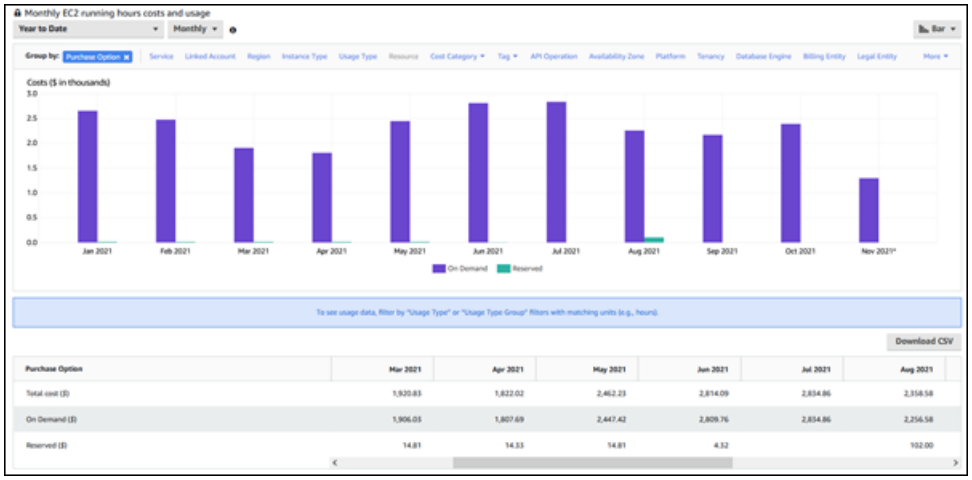
   

   Die reservierte DB-Instance in diesem Beispiel wurde „All Upfront“ gekauft, daher fallen keine Stundengebühren an.
6. Wählen Sie das Cost Explorer-Symbol (Balkendiagramm) neben der Überschrift Reserved Instances.

   Der Cost Explorer zeigt das Diagramm mit den Kosten und der Nutzung der monatlichen EC2 Betriebsstunden an.
7. Deaktivieren Sie den Filter Verwendungstyp-Gruppe rechts neben dem Diagramm.

8. Wählen Sie den Zeitraum und die Zeiteinheit aus, für die Sie die Nutzungskosten untersuchen möchten.

   Das folgende Beispiel zeigt die Nutzungskosten für On-Demand- und reservierte DB-Instances für das bisherige Jahr bis nach Monaten an.



   Die Kosten für reservierte DB-Instance von Januar bis Juni 2021 sind monatliche Gebühren für eine Partial-Upfront-Instance, während die Kosten im August 2021 eine einmalige Gebühr für eine All Upfront-Instance sind.

   Der Rabatt der reservierten Instances für die Partial Upfront-Instance lief im Juni 2021 aus, die DB-Instance wurde jedoch nicht gelöscht. Nach dem Ablaufdatum wurde es einfach zum On-Demand-Tarif berechnet.

Einrichtung Ihrer Amazon RDS-Umgebung

Diese Seite enthält eine umfassende Anleitung zur Einrichtung von Amazon Relational Database Service, einschließlich Kontokonfiguration, Sicherheit und Ressourcenverwaltung. Es führt Sie durch die wichtigsten Schritte, um Ihre Datenbankumgebungen effizient zu erstellen, zu verwalten und zu sichern. Ganz gleich, ob Sie Amazon RDS noch nicht kennen oder sich für spezielle Anforderungen einrichten, diese Abschnitte helfen Ihnen sicherzustellen, dass Ihre Einrichtung optimiert ist und den Best Practices entspricht.

Themen

- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff
- Erteilen programmgesteuerten Zugriffs
- Ermitteln der Anforderungen
- Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe

Wenn Sie bereits über eine verfügen AWS-Konto, Ihre Amazon RDS-Anforderungen kennen und lieber die Standardeinstellungen für IAM- und VPC-Sicherheitsgruppen verwenden möchten, fahren Sie mit fort. Erste Schritte mit Amazon RDS

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Ein Teil des Anmeldevorgangs umfasst den Empfang eines Telefonanrufs oder einer Textnachricht und die Eingabe eines Bestätigungscodes auf der Telefontastatur.

   Wenn Sie sich für eine anmelden AWS-Konto, wird eine Root-Benutzer des AWS-Kontoserstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und

verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Erteilen programmgesteuerten Zugriffs

Benutzer benötigen programmatischen Zugriff, wenn sie mit AWS außerhalb des interagieren möchten. AWS Management Console Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		• Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch AWS SDKs und im Tools-Ref erenzhandbuch.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Folgen Sie den Anweisungen unter Verwenden temporäre r Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeldeinformationen im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch. AWS APIs

Ermitteln der Anforderungen

Die Grundbausteine für Amazon RDS sind Datenbank-Instances. In einer DB-Instance erstellen Sie Ihre Datenbanken. Eine DB-Instance gibt eine Netzwerkadresse, den sogenannten Endpunkt, an. Ihre Anwendungen verwenden diesen Endpunkt, um eine Verbindung mit Ihrer DB-Instance einzurichten. Wenn Sie eine DB-Instance erstellen, geben Sie Details wie Speicher, Arbeitsspeicher, Datenbank-Engine und -Version, Netzwerkkonfiguration, Sicherheit und Wartungszeiträume an. Der Netzwerkzugriff auf eine DB-Instance wird über eine Sicherheitsgruppe kontrolliert.

Bevor Sie eine DB-Instance und eine Sicherheitsgruppe erstellen, müssen Sie Ihre DB-Instance und die Netzwerkanforderungen kennen. Hier einige wichtige Dinge, die Sie berücksichtigen sollten:

- Ressourcenanforderungen – Welche Anforderungen haben Sie an den Arbeitsspeicher und den Prozessor für Ihre Anwendung oder Ihren Service? Sie verwenden diese Einstellungen, um zu bestimmen, welche DB-Instance-Klasse Sie verwenden sollten. Spezifikationen für alle verfügbaren DB-Instance-Klassen finden Sie unter .
- VPC, Subnetz und Sicherheitsgruppe – Ihre DB-Instance befindet sich sehr wahrscheinlich in einer Virtual Private Cloud (VPC). Um eine Verbindung zu Ihrer DB-Instance einzurichten, müssen Sie Sicherheitsgruppenregeln festlegen. Diese Regeln werden abhängig von der Art der VPC und davon, wie Sie diese VPC verwenden, unterschiedlich eingerichtet. Sie können beispielsweise eine Standard-VPC oder eine benutzerdefinierte VPC verwenden.

  Die folgende Liste beschreibt die Regeln für jede VPC-Option:

  - Standard-VPC — Wenn Ihr AWS Konto über eine Standard-VPC in der aktuellen AWS Region verfügt, ist diese VPC für die Unterstützung von DB-Instances konfiguriert. Führen Sie folgende Schritte aus, wenn Sie beim Erstellen der DB-Instance die Standard-VPC angeben:
    - Sie müssen eine VPC-Sicherheitsgruppe anlegen, die Verbindungen von der Anwendung oder dem Service zum Amazon RDS-DB-Instance autorisiert. Verwenden Sie die Option Sicherheitsgruppe auf der VPC-Konsole oder AWS CLI um VPC-Sicherheitsgruppen zu erstellen. Weitere Informationen finden Sie unter Schritt 3: Erstellen einer VPC-Sicherheitsgruppe.
    - Geben Sie die Standard-DB-Subnetzgruppe an. Wenn dies die erste DB-Instance ist, die Sie in dieser AWS Region erstellt haben, erstellt Amazon RDS bei der Erstellung der DB-Instance die Standard-DB-Subnetzgruppe.
  - Benutzerdefinierte VPC – Wenn Sie beim Erstellen einer DB-Instance eine benutzerdefinierte VPC angeben möchten, müssen Sie Folgendes beachten:
    - Sie müssen eine VPC-Sicherheitsgruppe anlegen, die Verbindungen von der Anwendung oder dem Service zum Amazon RDS-DB-Instance autorisiert. Verwenden Sie die Option Sicherheitsgruppe auf der VPC-Konsole oder AWS CLI um VPC-Sicherheitsgruppen zu erstellen. Weitere Informationen finden Sie unter Schritt 3: Erstellen einer VPC-Sicherheitsgruppe.
    - Die VPC muss bestimmte Anforderungen erfüllen, um DB-Instances bereitzustellen, z. B. das Vorhandensein von zwei Subnetzen in jeweils einer separaten Availability Zone. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.

    - Sie müssen eine DB-Subnetzgruppe angeben, die definiert, welche Subnetze in dieser VPC von der DB-Instance genutzt werden können. Weitere Informationen erhalten Sie im Abschnitt "DB-Subnetzgruppen" unter Arbeiten mit einer DB-Instance in einer VPC.
- Hohe Verfügbarkeit: Benötigen Sie Failover-Unterstützung? Auf Amazon RDS erzeugt eine Multi-AZ-Bereitstellung eine primäre DB-Instance und eine sekundäre Standby-DB-Instance in einer anderen Availability Zone, um einen Failover-Anwendungsfall zu unterstützen. Wir empfehlen Multi-AZ-Bereitstellungen, um die hohe Verfügbarkeit von Produktions-Workloads sicherzustellen. Für Entwicklungs- und Testzwecke reicht gewöhnlich eine Bereitstellung ohne Multi-AZ. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.
- IAM-Richtlinien — Verfügt Ihr AWS Konto über Richtlinien, die die für die Durchführung von Amazon RDS-Vorgängen erforderlichen Berechtigungen gewähren? Wenn Sie eine Verbindung AWS mit IAM-Anmeldeinformationen herstellen, muss Ihr IAM-Konto über IAM-Richtlinien verfügen, die die für die Durchführung von Amazon RDS-Vorgängen erforderlichen Berechtigungen gewähren. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.
- Offene Ports — Auf welchem TCP/IP Port wartet Ihre Datenbank? Die Firewalls einiger Unternehmen blockieren möglicherweise Verbindungen zum Standard-Port für Ihre Datenbank-Engine. Wenn die Firewall Ihres Unternehmens den Standardport blockiert, wählen Sie für die neue DB-Instance einen anderen Port. Beachten Sie, dass Sie nach dem Erstellen einer DB-Instance, die einen angegebenen Port abfragt, diesen Port ändern können, indem Sie die DB-Instance modifizieren.
- AWS Region — In welcher AWS Region möchten Sie Ihre Datenbank haben? Indem sich Ihre Datenbank nahe bei der Anwendung oder dem Webdienst befindet, könnten Netzwerklatenzen verringert werden. Weitere Informationen finden Sie unter Regionen, Availability Zones und Local Zones.
- DB-Datenträgersubsystem: Welche Speicheranforderungen haben Sie? Amazon RDS bietet drei Speichertypen:
    - Allzweck (SSD)
    - Bereitgestellte IOPS (PIOPS)
    - Magnetic (auch bekannt als Standardspeicher)

    Weitere Informationen zu Amazon RDS-Speichern finden Sie unter Amazon RDS-DB-Instance-Speicher.

Sobald Ihnen die benötigten Informationen zur Erstellung der Sicherheitsgruppe und der DB-Instance vorliegen, fahren Sie mit dem nächsten Schritt fort.

Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe

VPC-Sicherheitsgruppen bieten Zugriff auf DB-Instances in einer VPC. Sie fungieren als Firewall für die zugeordneten DB-Instances und steuern den ein- und ausgehenden Datenverkehr auf der DB-Instance-Ebene. DB-Instances werden standardmäßig mit einer Firewall und einer Standard-Sicherheitsgruppe erstellt, die DB-Instance schützen.

Bevor Sie eine Verbindung zu Ihrer DB-Instance einrichten können, müssen Sie einer Sicherheitsgruppe Regeln hinzufügen. Verwenden Sie Ihre Netzwerk- und Konfigurationsinformationen, um Regeln für den Zugriff auf Ihre DB-Instance festzulegen.

Nehmen wir beispielsweise an, dass Sie über eine Anwendung verfügen, die auf eine Datenbank in Ihrer DB-Instance in einer VPC zugreift. In diesem Fall müssen Sie eine benutzerdefinierte TCP-Regel hinzufügen, die den Portbereich und IP-Adressen angibt, womit die Anwendung auf die Datenbank zugreift. Wenn Sie eine Anwendung auf einer EC2 Amazon-Instance haben, können Sie die Sicherheitsgruppe verwenden, die Sie für die EC2 Amazon-Instance eingerichtet haben.

Sie können die Konnektivität zwischen einer EC2 Amazon-Instance und einer DB-Instance konfigurieren, wenn Sie die DB-Instance erstellen. Weitere Informationen finden Sie unter Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2 .

> **Tip**
>
> Sie können die Netzwerkkonnektivität zwischen einer EC2 Amazon-Instance und einer DB-Instance automatisch einrichten, wenn Sie die DB-Instance erstellen. Weitere Informationen finden Sie unter Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2 .

Informationen dazu, wie Connect Ressourcen in Amazon Lightsail mit Ihren DB-Instances verbinden, finden Sie unter Lightsail-Ressourcen mit VPC-Peering verbinden. AWS-Services

Informationen zu gängigen Szenarien für den Zugriff auf eine DB-Instance finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.

So erstellen Sie eine VPC-Sicherheitsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc.

   > **Note**
   >
   > Stellen Sie sicher, dass Sie sich in der VPC-Konsole befinden, nicht in der RDS-Konsole.

2. Wählen Sie in der oberen rechten Ecke von die AWS Region aus AWS Management Console, in der Sie Ihre VPC-Sicherheitsgruppe und DB-Instance erstellen möchten. Die Liste der Amazon-VPC-Ressourcen für diese AWS -Region sollte zeigen, dass Sie über mindestens eine VPC und mehrere Subnetze verfügen. Wenn Sie dies nicht tun, haben Sie in dieser AWS Region keine Standard-VPC.
3. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
4. Wählen Sie Create security group (Sicherheitsgruppe erstellen) aus.

   Die Seite Sicherheitsgruppe erstellen wird angezeigt.
5. Geben Sie im Feld Grundlegende Details den Namen der Sicherheitsgruppe und die Beschreibung ein. Wählen Sie unter VPC die VPC aus, in der Sie Ihre DB-Instance erstellen möchten.
6. Wählen Sie in Eingehende Regeln die Option Regel hinzufügen.
   a. Wählen Sie für Type (Typ) die Option Custom TCP (Benutzerdefiniertes TCP) aus.
   b. Geben Sie für Portbereich den Portwert ein, der für Ihre DB-Instance verwendet werden soll.
   c. Wählen Sie für Source (Quelle) den Namen einer Sicherheitsgruppe oder geben Sie den IP-Adressbereich (CIDR-Wert) ein, von dem aus Sie auf die DB-Instance zugreifen. Wenn Sie My IP (Meine IP) auswählen, ermöglicht dies den Zugriff auf die DB-Instance von der in Ihrem Browser erkannten IP-Adresse.
7. Wenn Sie weitere IP-Adressen oder andere Portbereiche hinzufügen müssen, wählen Sie Regel hinzufügen und geben Sie die Informationen für die Regel ein.
8. (Optional) Fügen Sie inRegeln für ausgehenden Datenverkehr Regeln für ausgehenden Datenverkehr hinzu. Standardmäßig ist der gesamte ausgehende Datenverkehr zugelassen.
9. Wählen Sie Sicherheitsgruppe erstellen aus.

Sie können die soeben erstellte VPC-Sicherheitsgruppe als die Sicherheitsgruppe beim Anlegen Ihrer DB-Instance verwenden.

> **Note**
>
> Wenn Sie eine Standard-VPC verwenden, wird eine Standard-Subnetzgruppe für Sie angelegt, die alle Subnetze der VPC umfasst. Wenn Sie eine DB-Instance erstellen, können Sie die Standard-VPC auswählen und default (Standard) für die DB Subnet Group (DB-Subnetzgruppe) verwenden.

Nachdem Sie die erforderliche Einrichtung abgeschlossen haben, können Sie eine DB-Instance unter Verwendung Ihrer Anforderungen und Sicherheitsgruppe erstellen. Befolgen Sie hierzu die Anweisungen unter Erstellen einer Amazon RDS-DB-Instance. Informationen zum Einstieg in die Erstellung einer DB-Instance, die eine bestimmte DB-Engine verwendet, finden Sie in der entsprechenden Dokumentation aus der folgenden Tabelle.

Datenbank-Engine	Dokumentation
MariaDB	Erstellen einer MariaDB-DB-Instance und Herstellen einer Verbindung dazu
Microsoft SQL Server	Erstellen einer DB-Instance von Microsoft SQL Server und Herstellen einer Verbindung
MySQL	Erstellen einer MySQL-DB-Instance und Herstellen einer Verbindung dazu
Oracle	Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung
PostgreSQL	Erstellen einer PostgreSQL-DB-Instance und Herstellen einer Verbindung

> **Note**
>
> Wenn Sie nach dem Erstellen keine Verbindung zu einer DB-Instance herstellen können, finden Sie unter Informationen zur Problembehandlun Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Erste Schritte mit Amazon RDS

In den folgenden Beispielen erfahren Sie, wie mit Amazon Relational Database Service (Amazon RDS) eine DB-Instance erstellt und eine Verbindung zu ihr hergestellt wird. Sie können eine DB-Instance erstellen, die Db2, MariaDB ,MySQL, Microsoft SQL Server, Oracle oder PostgreSQL verwendet.

> **Important**
>
> Sie müssen die Aufgaben im Abschnitt Einrichtung Ihrer Amazon RDS-Umgebung abschließen, um eine DB-Instance erstellen oder eine Verbindung mit einer DB-Instance herstellen zu können.

Das Erstellen einer DB-Instance und ihre Verbindung zu einer Datenbank auf einer DB-Instance funktioniert für jede der DB-Engines etwas anders. Wählen Sie eine der folgenden DB-Engines aus, die Sie verwenden möchten, um detaillierte Informationen zum Erstellen und Verbinden mit der DB-Instance zu erhalten. Nachdem Sie eine DB-Instance erstellt und eine Verbindung mit ihr hergestellt haben, gibt es Anweisungen, mit denen Sie die DB-Instance löschen können.

Themen

- Erstellen einer MariaDB-DB-Instance und Herstellen einer Verbindung dazu
- Erstellen einer DB-Instance von Microsoft SQL Server und Herstellen einer Verbindung
- Erstellen einer MySQL-DB-Instance und Herstellen einer Verbindung dazu
- Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung
- Erstellen einer PostgreSQL-DB-Instance und Herstellen einer Verbindung
- Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance
- Tutorial: Verwenden einer Lambda-Funktion für den Zugriff auf eine Amazon-RDS-Datenbank

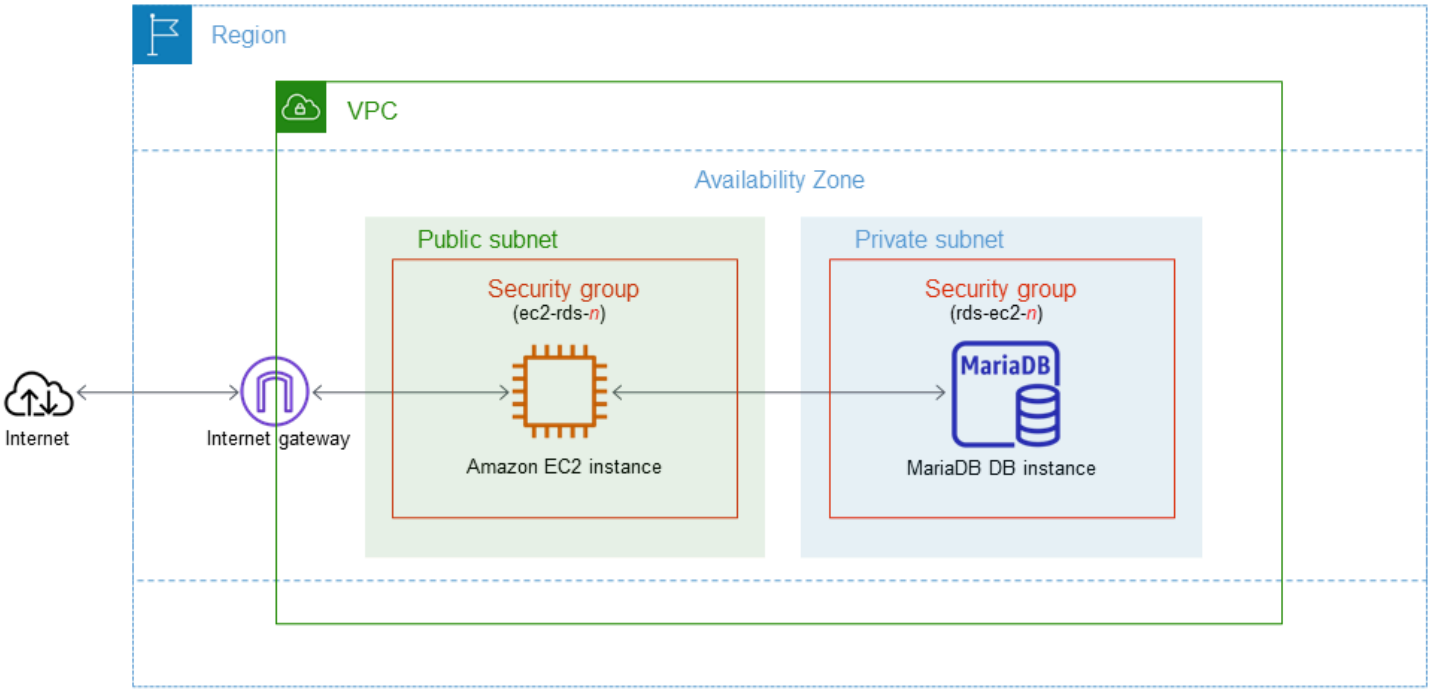
Erstellen einer MariaDB-DB-Instance und Herstellen einer Verbindung dazu

In diesem Tutorial werden eine EC2 Instanz und ein RDS für die MariaDB-DB-Instance erstellt. Das Tutorial zeigt Ihnen, wie Sie mit einem Standard-MySQL-Client von der EC2 Instance aus auf die DB-Instance zugreifen. Als bewährte Methode erstellt dieses Tutorial eine private DB-Instance in einer Virtual Private Cloud (VPC). In den meisten Fällen können andere Ressourcen in derselben VPC, z. B. EC2 Instances, auf die DB-Instance zugreifen, aber Ressourcen außerhalb der VPC können nicht darauf zugreifen.

Nach Abschluss des Tutorials gibt es in jeder Availability Zone im VPC ein öffentliches und ein privates Subnetz. In einer Availability Zone befindet sich die EC2 Instance im öffentlichen Subnetz und die DB-Instance im privaten Subnetz.

> **Important**
>
> Für die Erstellung eines fallen keine Gebühren an. AWS-Konto Bei Durchführung dieses Tutorials können jedoch Kosten für die von Ihnen verwendeten -Ressourcen anfallen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

Das folgende Diagramm zeigt die Konfiguration nach Abschluss des Tutorials.



In diesem Tutorial können Sie Ihre Ressourcen mit einer der folgenden Methoden erstellen:

1. Verwenden Sie das AWS Management Console - Erstellen Sie eine EC2 Instanz und Erstellen Sie eine MariaDB-DB-Instance
2. Verwenden Sie AWS CloudFormation , um die Datenbankinstanz und die EC2 Instanz zu erstellen - (Optional) Erstellen Sie VPC, EC2 Instanz und MariaDB-Instanz mit AWS CloudFormation

Die erste Methode verwendet Easy create, um eine private MariaDB-DB-Instance mit dem zu erstellen. AWS Management Console Hier geben Sie nur den DB-Engine-Typ, die DB-Instance-Größe und die DB-Instance-ID an. Easy Create (Einfache Erstellung) verwendet für die anderen Konfigurationsoptionen die Standardeinstellung.

Wenn Sie stattdessen Standard create verwenden, können Sie beim Erstellen einer DB-Instance weitere Konfigurationsoptionen angeben. Zu diesen Optionen gehören Einstellungen für Verfügbarkeit, Sicherheit, Backups und Wartung. Um eine öffentliche DB-Instance zu erstellen, müssen Sie Standarderstellung verwenden. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Themen

- Voraussetzungen
- Erstellen Sie eine EC2 Instanz

- Erstellen Sie eine MariaDB-DB-Instance
- (Optional) Erstellen Sie VPC, EC2 Instanz und MariaDB-Instanz mit AWS CloudFormation
- Verbindung zu einer MariaDB-DB-Instance
- Löschen Sie die EC2 Instance und die DB-Instance
- (Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation
- (Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

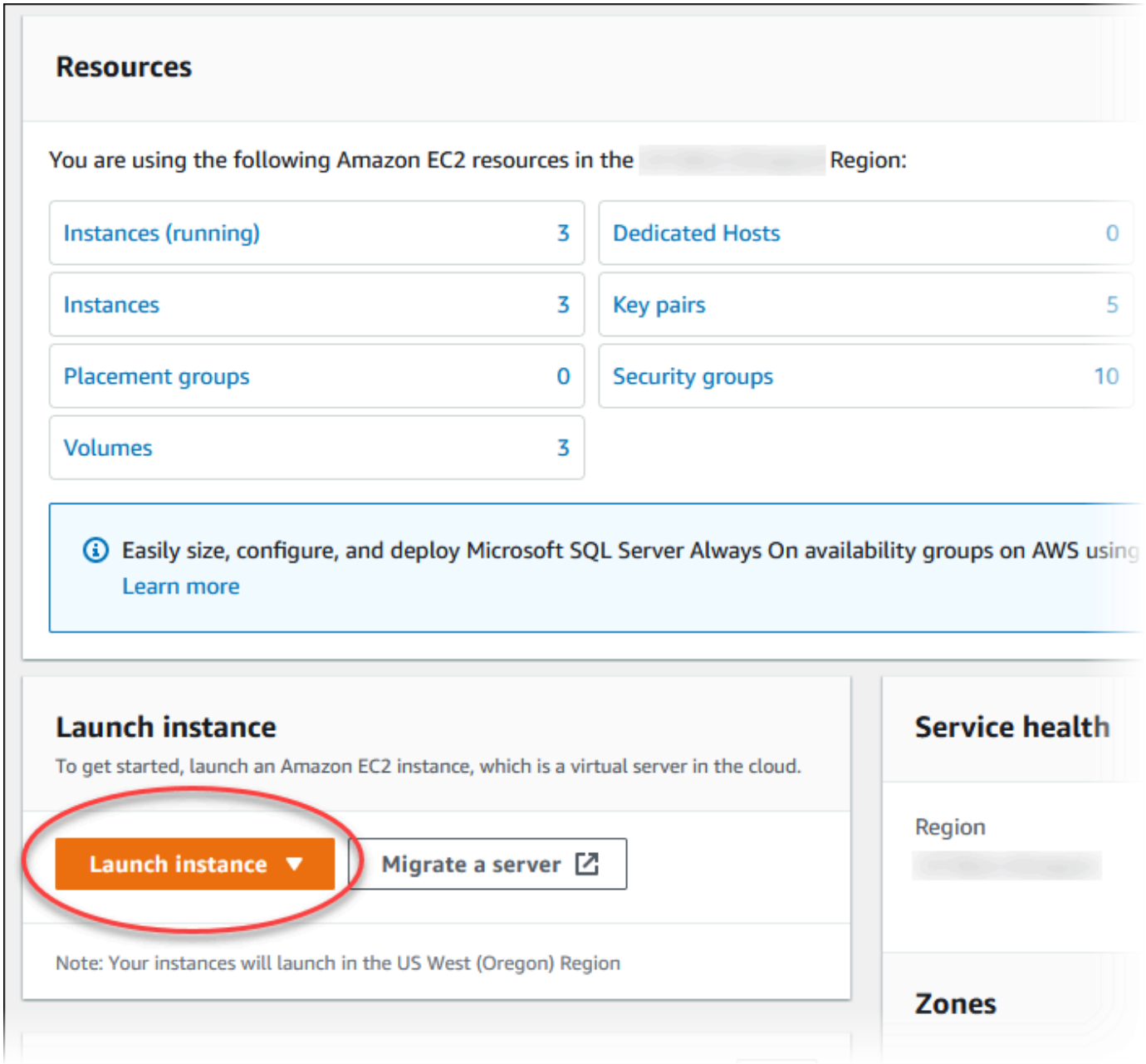
- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Erstellen Sie eine EC2 Instanz

Erstellen Sie eine EC2 Amazon-Instance, mit der Sie eine Verbindung zu Ihrer Datenbank herstellen.

Um eine EC2 Instance zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen rechten Ecke von die aus AWS Management Console, AWS-Region in der Sie die Instance erstellen möchten. EC2
3. Wählen Sie EC2 Dashboard und dann Launch instance aus, wie in der folgenden Abbildung gezeigt.

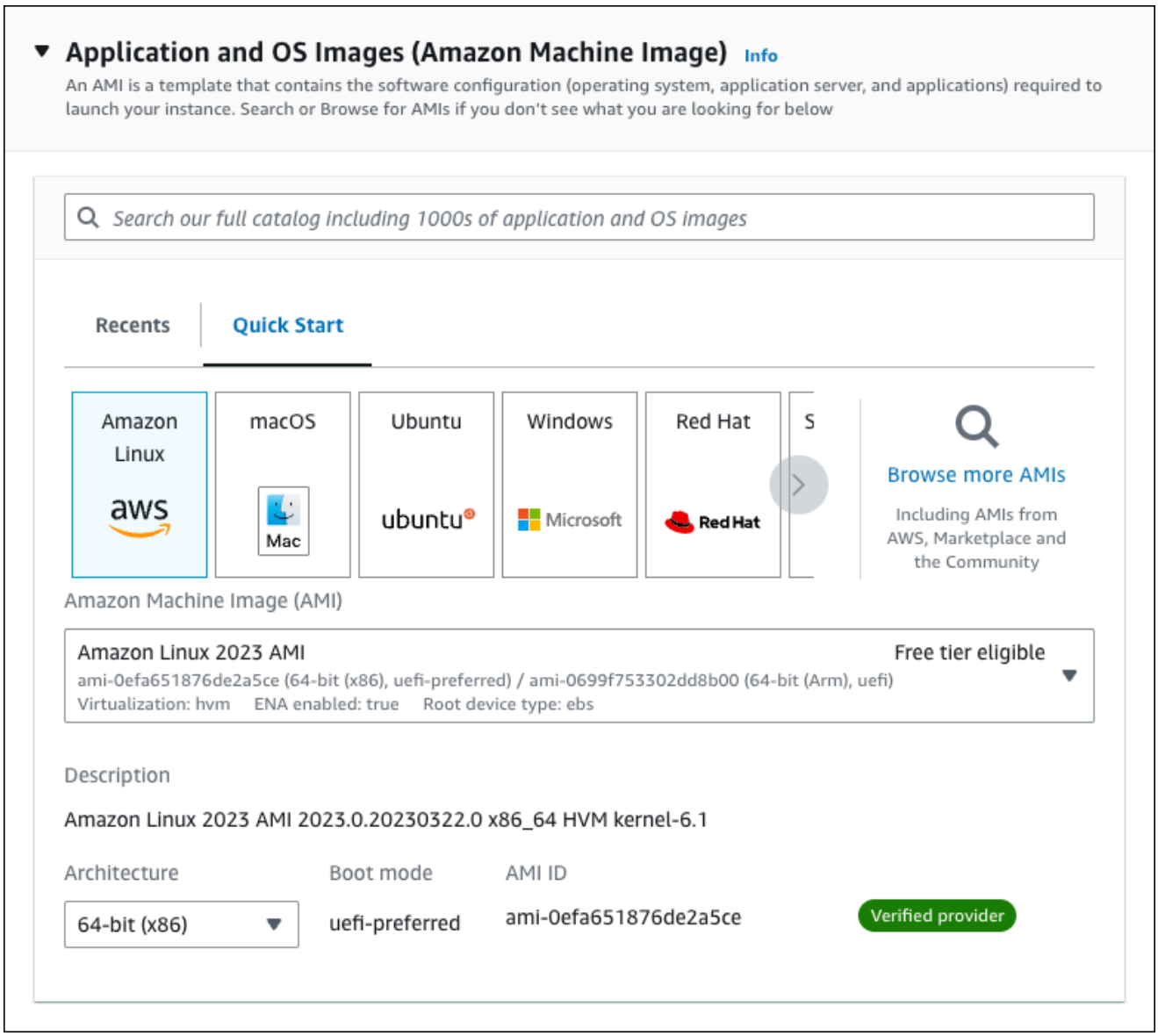


Die Seite Eine Instance starten wird geöffnet.

4. Wählen Sie auf der Seite Eine Instance starten die folgenden Einstellungen aus.

   a. Geben Sie unter Name and tags (Name und Tags) als Name den Namen **`ec2-database-connect`** ein.

   b. Wählen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) die Option Amazon Linux und dann die Registerkarte Amazon Linux 2023 AMI aus. Übernehmen Sie für alle anderen Einstellungen die Standardwerte.



c. Wählen Sie unter Instance type (Instance-Typ) den Wert t2.micro aus.

d. Wählen Sie unter Key pair (login) (Schlüsselpaar (Anmeldung)) einen Key pair name (Schlüsselpaarname), um ein vorhandenes Schlüsselpaar zu verwenden. Um ein neues key pair für die EC2 Amazon-Instance zu erstellen, wählen Sie Neues key pair erstellen und verwenden Sie dann das Fenster key pair erstellen, um es zu erstellen.

   Weitere Informationen zum Erstellen eines neuen key pair finden Sie unter Erstellen eines key pair im EC2 Amazon-Benutzerhandbuch.

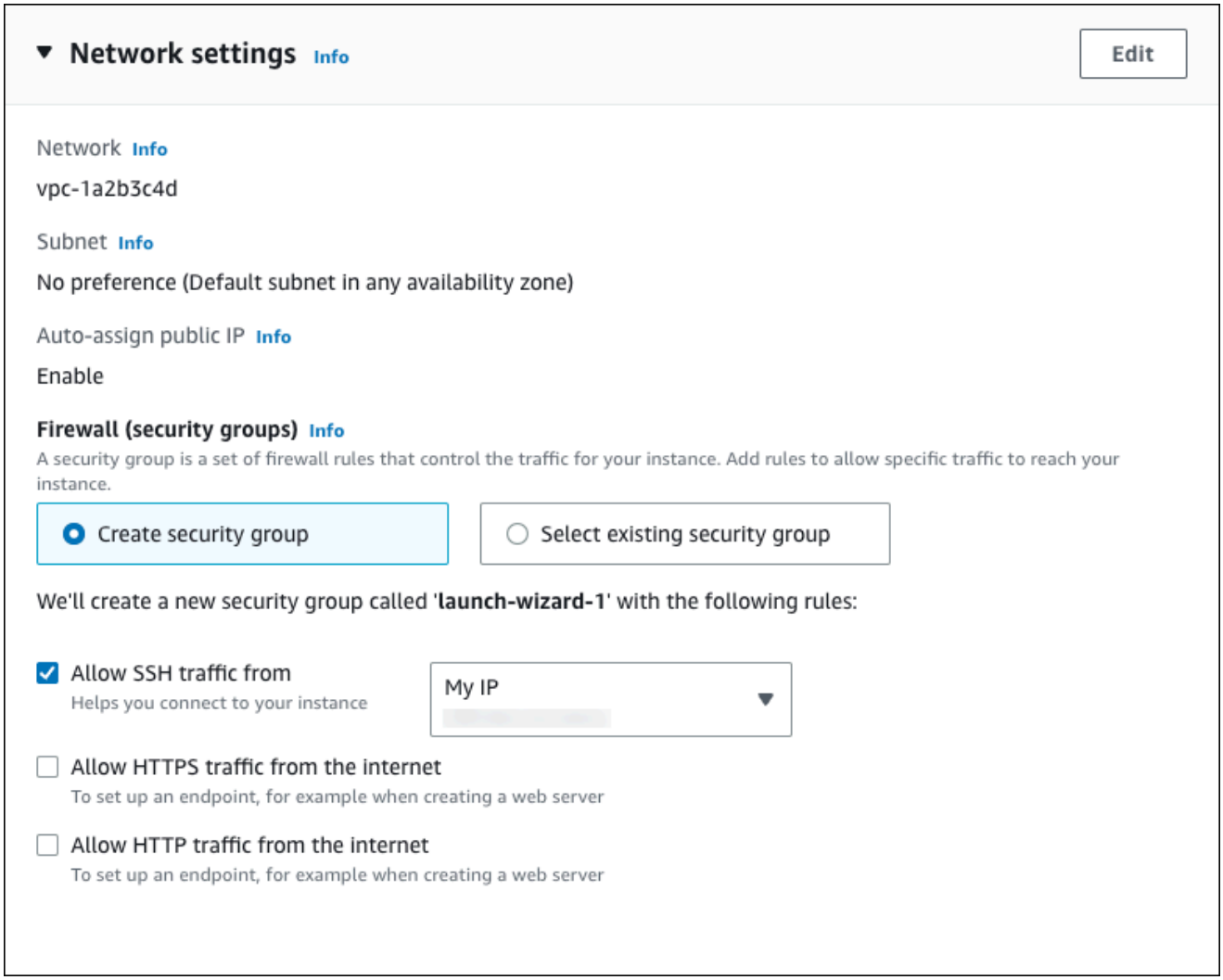
e. Wählen Sie unter „SSH-Verkehr zulassen“ in den Netzwerkeinstellungen die Quelle der SSH-Verbindungen zur Instance aus. EC2

Sie können My IP (Meine IP) auswählen, wenn die angezeigte IP-Adresse für SSH-Verbindungen korrekt ist. Andernfalls können Sie mithilfe von Secure Shell (SSH) die IP-Adresse bestimmen, die für die Verbindung zu EC2 Instances in Ihrer VPC verwendet werden soll. Um Ihre öffentliche IP-Adresse in einem anderen Browserfenster oder einer anderen Registerkarte zu ermitteln, können Sie den Dienst unter verwenden. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Bestimmen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instanzen über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

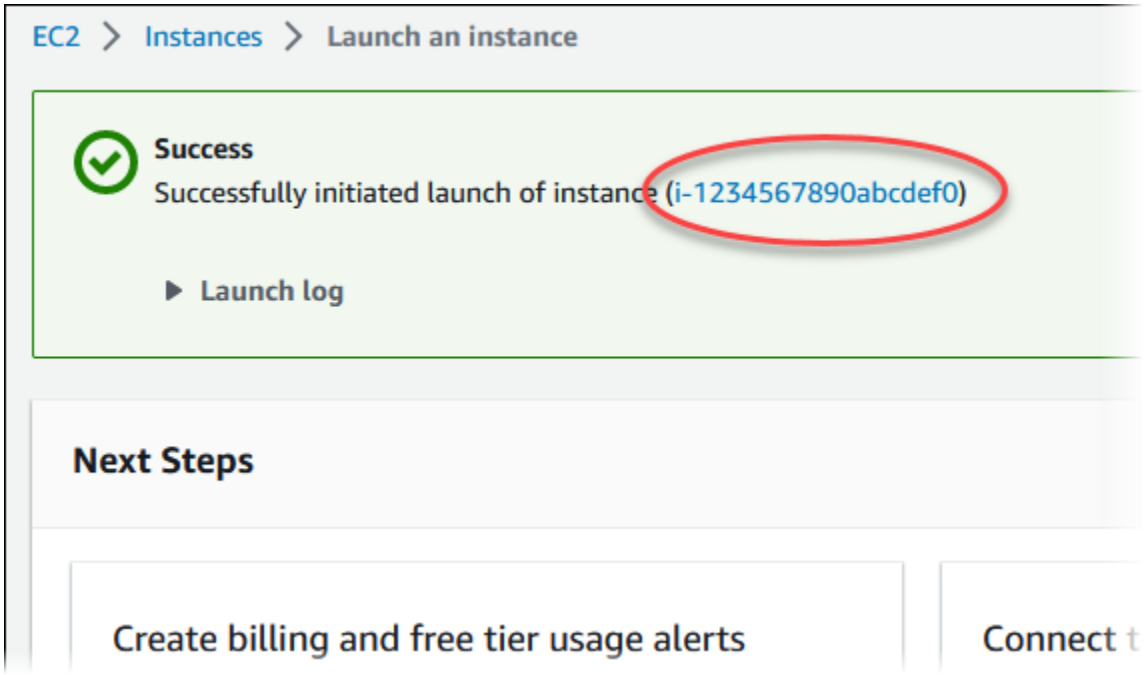
Die folgende Abbildung zeigt ein Beispiel für den Bereich Netzwerkeinstellungen.



f. Übernehmen Sie die Standardwerte für die übrigen Abschnitte.

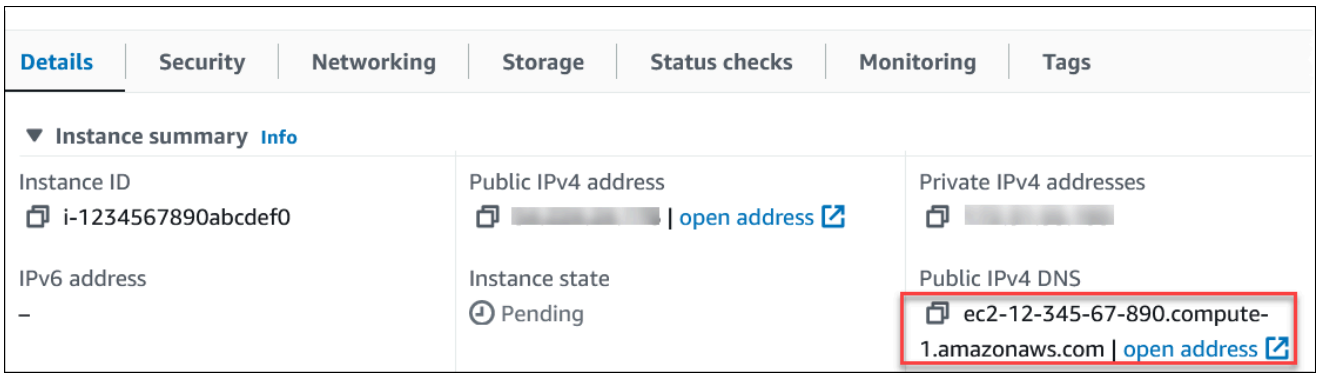
g. Sehen Sie sich im Bereich Zusammenfassung eine Zusammenfassung Ihrer EC2 Instance-Konfiguration an. Wenn Sie bereit sind, wählen Sie Launch instance aus.

5. Notieren Sie sich auf der Seite Startstatus die ID für Ihre neue EC2 Instance, zum Beispiel:`i-1234567890abcdef0`.

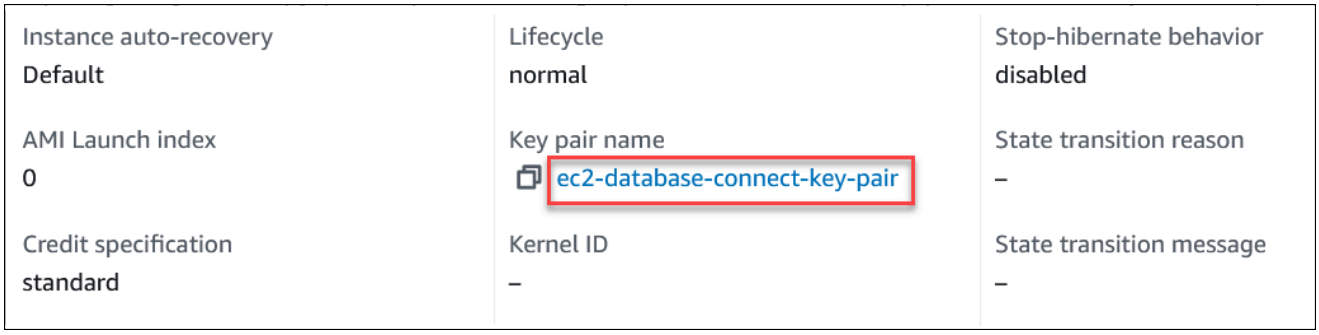


6. Wählen Sie die EC2 Instance-ID, um die Liste der EC2 Instances zu öffnen, und wählen Sie dann Ihre EC2 Instance aus.
7. Notieren Sie sich die folgenden Werte auf der Registerkarte Details. Diese benötigen Sie, wenn Sie eine Verbindung über SSH herstellen:

   a. Notieren Sie sich in der Instanzübersicht den Wert für Public IPv4 DNS.



   b. Notieren Sie sich unter Instance-Details den Wert für Schlüsselpaarname.



8. Warten Sie, bis der Instanzstatus Ihrer EC2 Instance den Status Running hat, bevor Sie fortfahren.

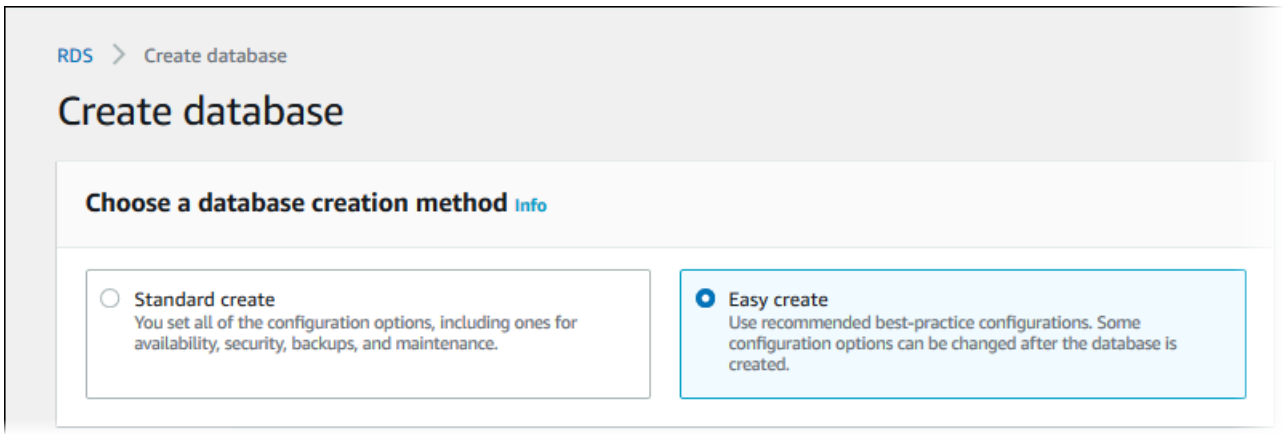
Erstellen Sie eine MariaDB-DB-Instance

Die Grundbausteine für Amazon RDS sind Datenbank-Instances. In dieser Umgebung führen Sie Ihre MariaDB-Datenbanken aus.

In diesem Beispiel verwenden Sie Easy Create, um eine DB-Instance zu erstellen, auf der die MariaDB-Datenbank-Engine mit einer db.t4g.micro DB-Instance-Klasse ausgeführt wird.

So erstellen Sie eine MariaDB-DB-Instance mit einfacher Erstellung:

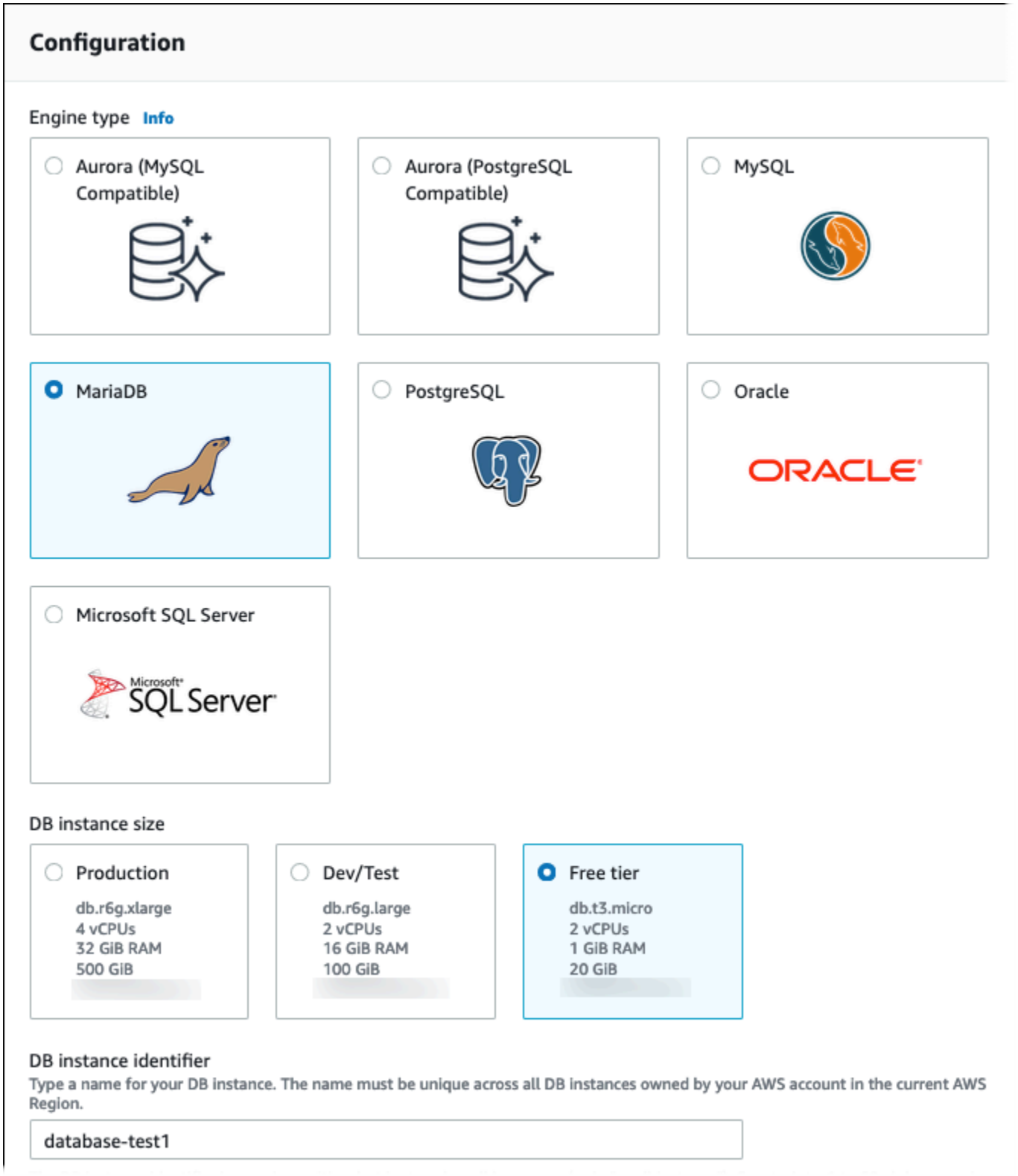
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die aus, AWS-Region in der Sie die DB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Datenbanken aus.
4. Wählen Sie Datenbank erstellen und dann Einfach erstellen aus.



5. Wählen Sie unter Konfiguration die Option MariaDB.

6. Wählen Sie für die DB-Instance-Größe das kostenlose Kontingent oder Sandbox aus. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.
7. Geben Sie als DB-Instance-ID **`database-test1`** ein.
8. Geben Sie unter Hauptbenutzername einen Namen für den Hauptbenutzer ein oder behalten Sie den Standardnamen bei.

   Die Seite Datenbank erstellen sollte ähnlich wie in der folgenden Abbildung gezeigt aussehen. Bei Konten mit kostenlosem Tarif wird das kostenlose Kontingent angezeigt. Bei Konten mit kostenpflichtigem Tarif wird Sandbox angezeigt.

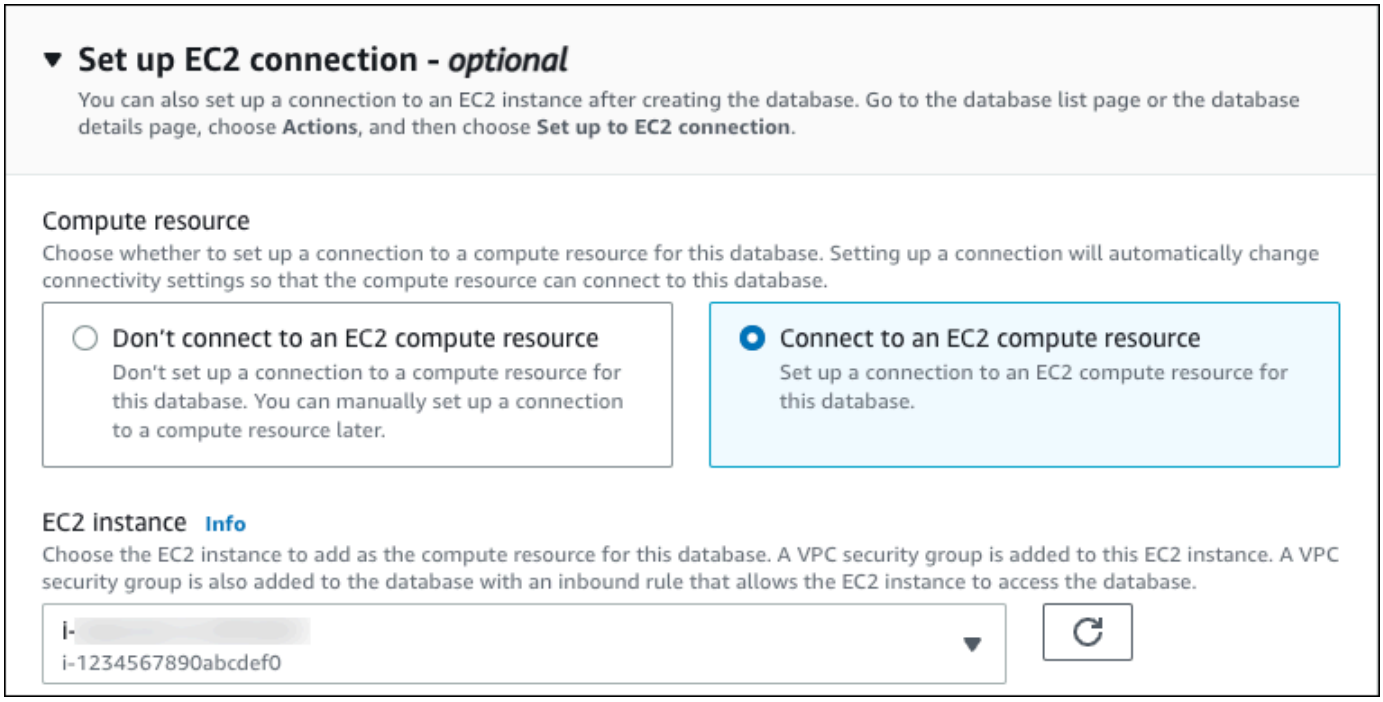


9. Um für die DB-Instance ein automatisch generiertes Hauptpasswort zu verwenden, wählen Sie das Kästchen Passwort automatisch generieren aus.

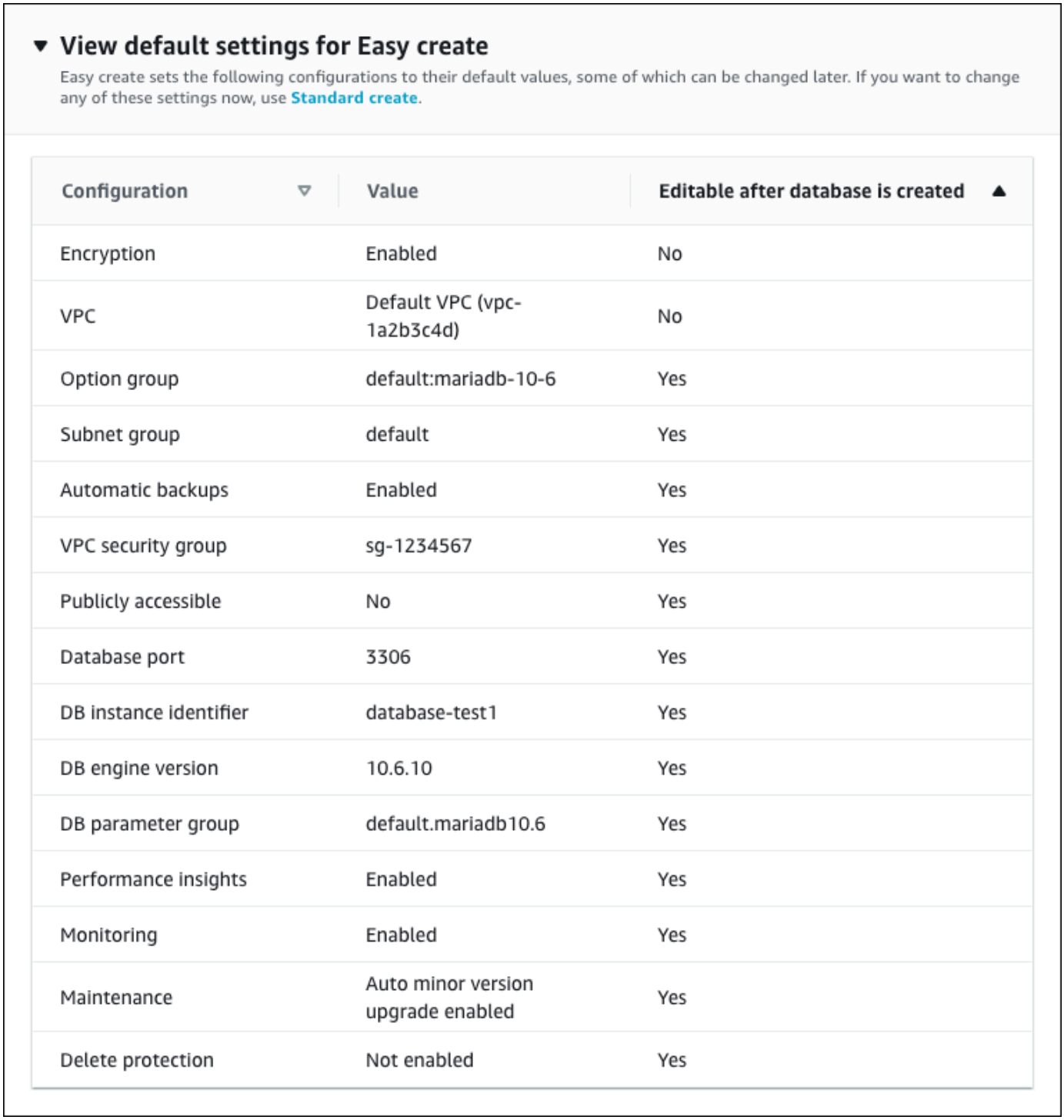
Um Ihr Master-Passwort einzugeben, deaktivieren Sie die Option Passwort automatisch generieren und geben Sie dann dasselbe Passwort in die Felder Master-Passwort und Master-Passwort bestätigen ein.

10. Um eine Verbindung mit der EC2 Instanz einzurichten, die Sie zuvor erstellt haben, erweitern Sie die Option EC2 Verbindung einrichten — optional.

    Wählen Sie Mit einer EC2 Rechenressource verbinden aus. Wählen Sie die EC2 Instanz aus, die Sie zuvor erstellt haben.



11. Erweitern Sie die Option Standardeinstellungen anzeigen für Easy Create.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mariadb-10-6	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	10.6.10	Yes
DB parameter group	default.mariadb10.6	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Sie können die Standardeinstellungen von Einfache Erstellung einsehen. Die Spalte Nach Erstellung der Datenbank editierbar zeigt, welche Optionen Sie nach der Datenbankerstellung ändern können.

- Wenn in einer Einstellung Nein in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie Standarderstellung verwenden, um die DB-Instance zu erstellen.
- Wenn für eine Einstellung Ja in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie entweder Standarderstellung verwenden, oder die DB-Instance nach der Erstellung ändern, um die Einstellung zu ändern.

12. Wählen Sie Datenbank erstellen aus.

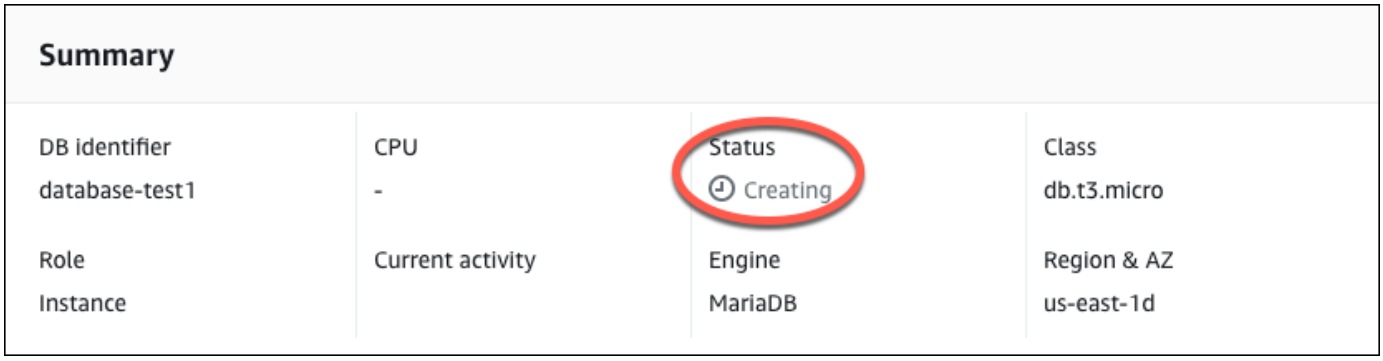
    Um den Masterbenutzernamen und das zugehörige Passwort für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen).

    Sie können den angezeigten Benutzernamen und das angezeigte Passwort verwenden, um als Masterbenutzer eine Verbindung zu DB-Instance herzustellen.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern.
    > Wenn Sie das Passwort für den Hauptbenutzer ändern müssen, nachdem die DB-Instance verfügbar wurde, können Sie die DB-Instance entsprechend ändern. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

13. Wählen Sie in der Liste Datenbanken den Namen der neuen MariaDB-DB-Instance aus.

    Die DB-Instance hat den Status Wird erstellt, bis die DB-Instance bereit für die Verwendung ist.



    Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und vom verfügbaren Speicherplatz kann es bis zu 20 Minuten dauern, bis die neue DB-Instance verfügbar ist.

(Optional) Erstellen Sie VPC, EC2 Instanz und MariaDB-Instanz mit AWS CloudFormation

Anstatt die Konsole zu verwenden, um Ihre VPC, EC2 Instance und MariaDB-Instance AWS CloudFormation zu erstellen, können Sie AWS Ressourcen bereitstellen, indem Sie Infrastruktur als Code behandeln. Um Ihnen zu helfen, Ihre AWS Ressourcen in kleinere und besser verwaltbare Einheiten zu organisieren, können Sie die Nested-Stack-Funktionalität verwenden. AWS CloudFormation Weitere Informationen finden Sie unter Einen Stack auf der AWS CloudFormation Konsole erstellen und Mit verschachtelten Stacks arbeiten.

> **Important**
>
> AWS CloudFormation ist kostenlos, aber die Ressourcen, die CloudFormation erstellt werden, sind live. Es fallen die üblichen Nutzungsgebühren für diese Ressourcen an, bis Sie sie kündigen. Weitere Informationen finden Sie unter Preise für RDS for MariaDB.

- Laden Sie die Vorlage herunter CloudFormation
- Konfigurieren Sie Ihre Ressourcen mit CloudFormation

Laden Sie die CloudFormation Vorlage herunter

Eine CloudFormation Vorlage ist eine JSON- oder YAML-Textdatei, die die Konfigurationsinformationen zu den Ressourcen enthält, die Sie im Stack erstellen möchten. Diese Vorlage erstellt zusammen mit der RDS-Instanz auch eine VPC und einen Bastion-Host für Sie.

Um die Vorlagendatei herunterzuladen, öffnen Sie den folgenden Link: CloudFormation MariaDB-Vorlage.

Klicken Sie auf der Github-Seite auf die Schaltfläche Rohdatei herunterladen, um die YAML-Vorlagendatei zu speichern.

Konfigurieren Sie Ihre Ressourcen mit CloudFormation

> **Note**
>
> Bevor Sie diesen Vorgang starten, stellen Sie sicher, dass Sie ein Schlüsselpaar für eine EC2 Instance in Ihrem haben AWS-Konto. Weitere Informationen finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

Wenn Sie die AWS CloudFormation Vorlage verwenden, müssen Sie die richtigen Parameter auswählen, um sicherzustellen, dass Ihre Ressourcen ordnungsgemäß erstellt werden. Führen Sie die folgenden Schritte aus:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie Stapel erstellen aus.
3. Wählen Sie im Abschnitt Vorlage angeben die Option Eine Vorlagendatei von Ihrem Computer hochladen und dann Weiter aus.
4. Legen Sie auf der Seite „Stack-Details angeben“ die folgenden Parameter fest:
   a. Setzen Sie den Stack-Namen auf Maria DBTest Stack.
   b. Legen Sie unter Parameter Availability Zones fest, indem Sie drei Availability Zones auswählen.
   c. Wählen Sie unter Linux Bastion Host configuration für Key Name ein key pair aus, um sich bei Ihrer EC2 Instance anzumelden.
   d. Stellen Sie in den Linux Bastion Host-Konfigurationseinstellungen den Bereich Zulässiger IP auf Ihre IP-Adresse ein. Um mithilfe von Secure Shell (SSH) eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen, ermitteln Sie Ihre öffentliche IP-Adresse mithilfe des Dienstes unter. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instances über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

e. Stellen Sie unter Allgemeine Datenbankkonfiguration die Datenbankinstanzklasse auf db.t3.micro ein.

f. Setzen Sie den Datenbanknamen auf. **`database-test1`**

g. Geben Sie unter Datenbank-Master-Benutzername einen Namen für den Masterbenutzer ein.

h. Stellen Sie `false` für dieses Tutorial das DB-Master-Benutzerpasswort mit Secrets Manager verwalten auf ein.

i. Geben Sie für das Datenbankkennwort ein Passwort Ihrer Wahl ein. Merken Sie sich dieses Passwort für weitere Schritte im Tutorial.

j. Stellen Sie unter Datenbankspeicherkonfiguration den Datenbankspeichertyp auf gp2 ein.

k. Stellen Sie unter Konfiguration der Datenbanküberwachung die Option Enable RDS Performance Insights auf false ein.

l. Behalten Sie für alle anderen Einstellungen die Standardwerte bei. Klicken Sie auf Weiter, um fortzufahren.

5. Wählen Sie auf der Seite „Stack überprüfen“ die Option Senden aus, nachdem Sie die Datenbank- und Linux-Bastion-Host-Optionen überprüft haben.

Sehen Sie sich nach Abschluss der Stack-Erstellung die Stacks mit Namen BastionStackund RDSNS an, um die Informationen zu notieren, die Sie für die Verbindung mit der Datenbank benötigen. Weitere Informationen finden Sie unter AWS CloudFormation Stack-Daten und Ressourcen anzeigen auf der. AWS Management Console

Verbindung zu einer MariaDB-DB-Instance

Sie können für die Verbindung zur DB-Instance eine beliebige Standard-SQL-Client-Anwendung verwenden. In diesem Beispiel stellen Sie eine Verbindung mit einer MariaDB-DB-Instance mithilfe des mysql-Befehlszeilen-Tools her.

So stellen Sie eine Verbindung zu einer MariaDB-DB-Instance her

1. Suchen Sie nach dem Endpunkt (DNS-Name) und der Portnummer für Ihre DB-Instance.

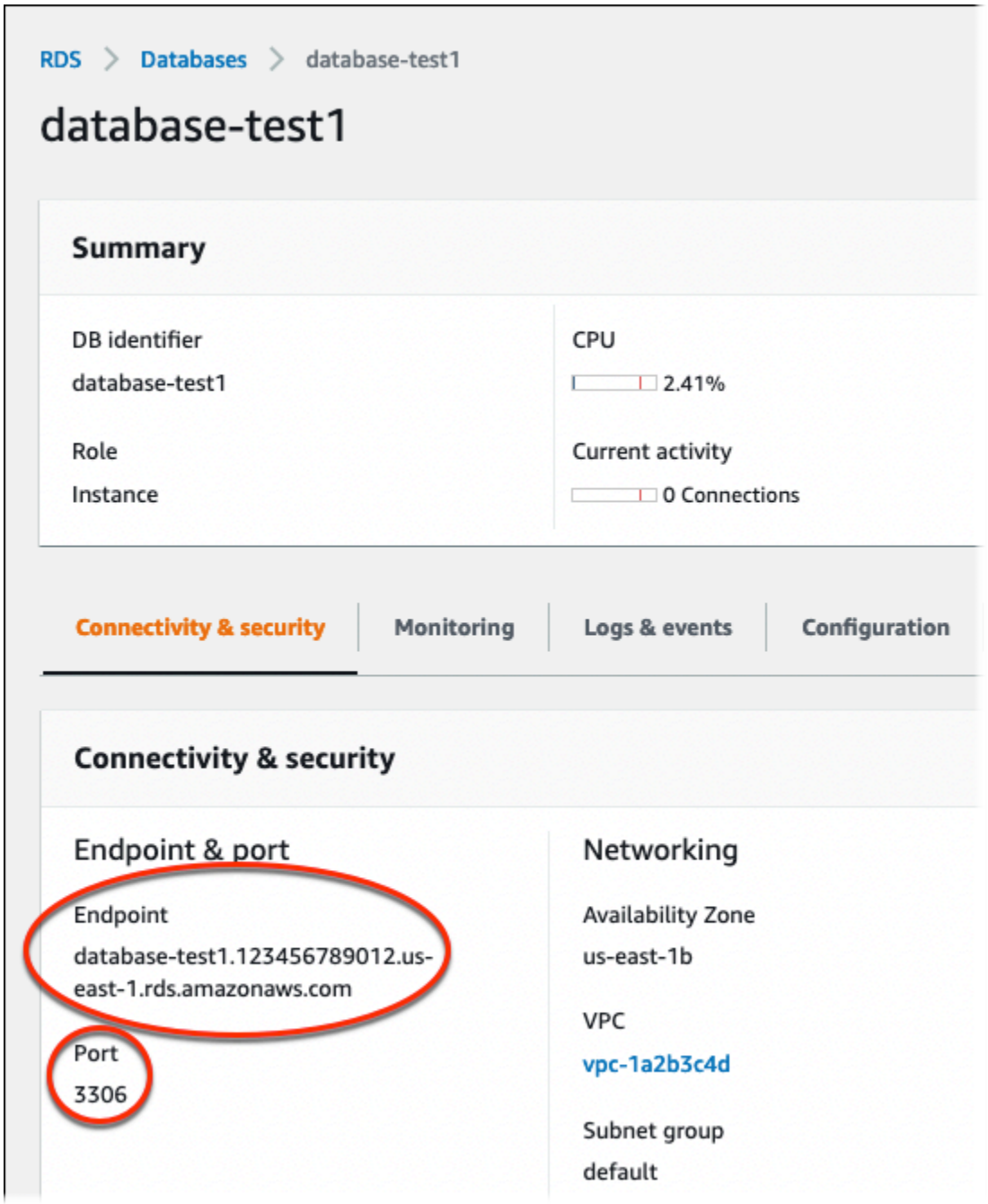
   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

   b. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die AWS-Region für die DB-Instance aus.

   c. Wählen Sie im Navigationsbereich Datenbanken aus.

d. Wählen Sie den Namen der MariaDB-DB-Instance, um deren Details anzuzeigen.

e. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



2. Stellen Sie eine Verbindung zu der EC2 Instance her, die Sie zuvor erstellt haben, indem Sie den Schritten unter Connect to your Linux Instance im EC2 Amazon-Benutzerhandbuch folgen.

Wir empfehlen, dass Sie über SSH eine Verbindung zu Ihrer EC2 Instance herstellen. Wenn das SSH-Client-Dienstprogramm unter Windows, Linux oder Mac installiert ist, können Sie mit dem folgenden Befehlsformat eine Verbindung mit der Instance herstellen:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Gehen Sie beispielsweise davon aus, dass dies `/dir1` unter Linux gespeichert `ec2-database-connect-key-pair.pem` ist und das öffentliche IPv4 DNS für Ihre EC2 Instance bereits. `ec2-12-345-678-90.compute-1.amazonaws.com` Ihr SSH-Befehl würde wie folgt aussehen:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Holen Sie sich die neuesten Bugfixes und Sicherheitsupdates, indem Sie die Software auf Ihrer EC2 Instance aktualisieren. Verwenden Sie dazu den folgenden Befehl.

   > **Note**
   >
   > Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Um Updates vor der Installation zu überprüfen, lassen Sie diese Option aus.

   ```
   sudo dnf update -y
   ```

4. Installieren Sie den MySQL-Befehlszeilenclient von MariaDB.

   Führen Sie den folgenden Befehl aus, um den Befehlszeilen-Client von MariaDB auf Amazon Linux 2023 zu installieren:

   ```
   sudo dnf install mariadb105
   ```

5. Stellen Sie eine Verbindung mit der DB-Instance her. Geben Sie z. B. den folgenden Befehl ein. Mit dieser Aktion können Sie eine Verbindung mit der MariaDB-DB-Instance mithilfe des MySQL-Clients herstellen.

Ersetzen Sie den DB-Instance-Endpunkt (DNS-Name) für *endpoint* und den Hauptbenutzernamen, den Sie für *admin* verwendet haben. Geben Sie das Master-Passwort ein, das Sie bei der Aufforderung zur Eingabe eines Passworts verwendet haben.

```
mysql -h endpoint -P 3306 -u admin -p
```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, sollte eine Ausgabe wie die folgende angezeigt werden.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 156
Server version: 10.6.10-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Weitere Informationen zum Herstellen einer Verbindung zur MariaDB-DB-Instance finden Sie unter Verbindung zu Ihrer MariaDB-DB-Instance herstellen. Wenn Sie sich nicht mit Ihrer DB-Instance verbinden können, erhalten Sie unter Hilf Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Aus Sicherheitsgründen empfiehlt es sich, verschlüsselte Verbindungen zu verwenden. Verwenden Sie eine unverschlüsselte MariaDB-Verbindung nur, wenn sich Client und Server in derselben VPC befinden und das Netzwerk vertrauenswürdig ist. Weitere IInformationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Herstellen einer Verbindung zu Ihrer MariaDB-DB-Instance auf Amazon RDS SSL mit/über den TLS SQL Befehlszeilenclient My (verschlüsselt).

6. SQL-Befehle ausführen

   Der folgende SQL-Befehl zeigt z B. das aktuelle Datum und die aktuelle Zeit an:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Löschen Sie die EC2 Instance und die DB-Instance

Nachdem Sie sich mit der EC2 Beispiel-Instance und DB-Instance, die Sie erstellt haben, verbunden und diese erkundet haben, löschen Sie sie, damit Ihnen keine Gebühren mehr berechnet werden.

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, überspringen Sie diesen Schritt und fahren Sie mit dem nächsten Schritt fort.

Um die EC2 Instanz zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die EC2 Instance aus und wählen Sie dann Instance-Status, Instanz beenden aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

Weitere Informationen zum Löschen einer EC2 Instance finden Sie unter Kündigen Ihrer Instance im EC2 Amazon-Benutzerhandbuch.

So löschen Sie die DB-Instance ohne finalen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die zu löschende DB-Instance aus.
4. Klicken Sie bei Actions auf Delete.
5. Löschen Sie Abschließenden Snapshot erstellen? und Automatische Backups aufbewahren.
6. Bestätigen Sie, und wählen Sie Löschen.

(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, löschen Sie den CloudFormation Stack, nachdem Sie sich mit der EC2 Beispiel-Instance und der DB-Instance verbunden und diese erkundet haben, sodass Ihnen diese nicht mehr in Rechnung gestellt werden.

Um die CloudFormation Ressourcen zu löschen

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite Stacks in der CloudFormation Konsole den Root-Stack aus (den Stack ohne Namen VPCStack BastionStack oder RDSNS).
3. Wählen Sie Löschen aus.
4. Wählen Sie Stack löschen aus, wenn Sie zur Bestätigung aufgefordert werden.

Weitere Informationen zum Löschen eines Stacks in CloudFormation finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks auf der AWS CloudFormation Konsole.

(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Sie können Ihre DB-Instance von RDS für MariaDB auch mit einer Lambda-Serverless-Rechenressource verbinden. Mit Lambda-Funktionen können Sie Code ausführen, ohne die Infrastruktur bereitstellen oder verwalten zu müssen. Eine Lambda-Funktion ermöglicht es Ihnen auch, automatisch auf Codeausführungsanfragen jeder Größenordnung zu reagieren, von einem Dutzend Ereignissen pro Tag bis hin zu Hunderten von Ereignissen pro Sekunde. Weitere Informationen finden Sie unter Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance.

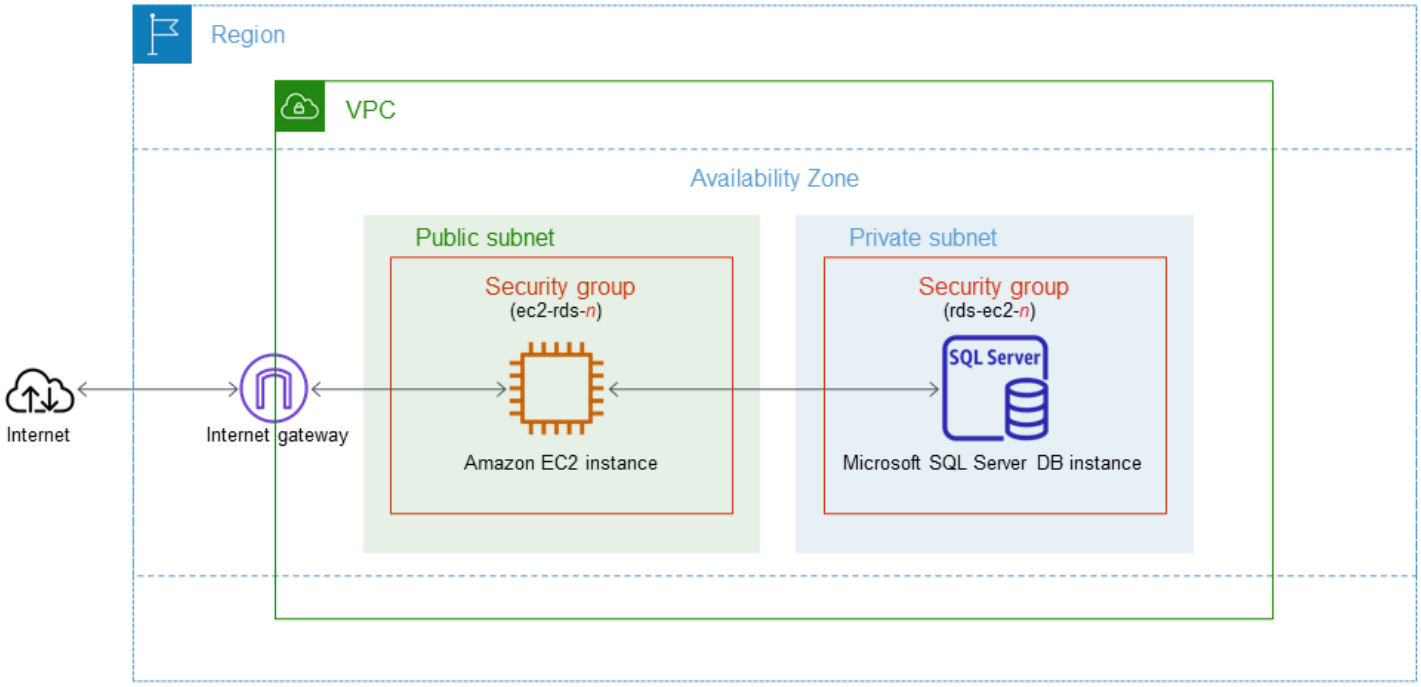
Erstellen einer DB-Instance von Microsoft SQL Server und Herstellen einer Verbindung

In diesem Tutorial werden eine EC2 Instance und eine RDS für Microsoft SQL Server-DB-Instance erstellt. Das Tutorial zeigt Ihnen, wie Sie mit dem Microsoft SQL Server Management Studio-Client von der EC2 Instance aus auf die DB-Instance zugreifen. Als bewährte Methode erstellt dieses Tutorial eine private DB-Instance in einer Virtual Private Cloud (VPC). In den meisten Fällen können andere Ressourcen in derselben VPC, z. B. EC2 Instances, auf die DB-Instance zugreifen, aber Ressourcen außerhalb der VPC können nicht darauf zugreifen.

Nach Abschluss des Tutorials gibt es in jeder Availability Zone im VPC ein öffentliches und ein privates Subnetz. In einer Availability Zone befindet sich die EC2 Instance im öffentlichen Subnetz und die DB-Instance im privaten Subnetz.

> **Important**
>
> Für die Erstellung eines Kontos fallen keine Gebühren an AWS . Wenn Sie dieses Tutorial abschließen, können Ihnen jedoch Kosten für die von Ihnen verwendeten AWS Ressourcen entstehen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

Das folgende Diagramm zeigt die Konfiguration nach Abschluss des Tutorials.



In diesem Tutorial können Sie Ihre Ressourcen mithilfe einer der folgenden Methoden erstellen:

1. Verwenden Sie das AWS Management Console - Erstellen Sie eine SQL Server-DB-Instance und Erstellen einer EC2-Instance
2. Verwenden Sie AWS CloudFormation , um die Datenbankinstanz und die EC2 Instanz zu erstellen - (Optional) Erstellen Sie VPC, EC2 Instanz und SQL Server-Instanz mit AWS CloudFormation

Die erste Methode verwendet Easy Create, um eine private SQL Server-DB-Instance mit dem zu erstellen AWS Management Console. Hier geben Sie nur den DB-Engine-Typ, die DB-Instance-Größe und die DB-Instance-ID an. Easy Create (Einfache Erstellung) verwendet für die anderen Konfigurationsoptionen die Standardeinstellung.

Wenn Sie stattdessen Standard create verwenden, können Sie beim Erstellen einer DB-Instance weitere Konfigurationsoptionen angeben. Zu diesen Optionen gehören Einstellungen für Verfügbarkeit, Sicherheit, Backups und Wartung. Um eine öffentliche DB-Instance zu erstellen, müssen Sie Standarderstellung verwenden. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Themen

- Voraussetzungen
- Erstellen einer EC2-Instance

- Erstellen Sie eine SQL Server-DB-Instance
- (Optional) Erstellen Sie VPC, EC2 Instanz und SQL Server-Instanz mit AWS CloudFormation
- Connect zu Ihrer SQL Server-DB-Instance her
- Erkunden Sie Ihre Beispiel-SQL Server-DB-Instance
- Löschen Sie die EC2 Instanz und die DB-Instance
- (Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation
- (Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

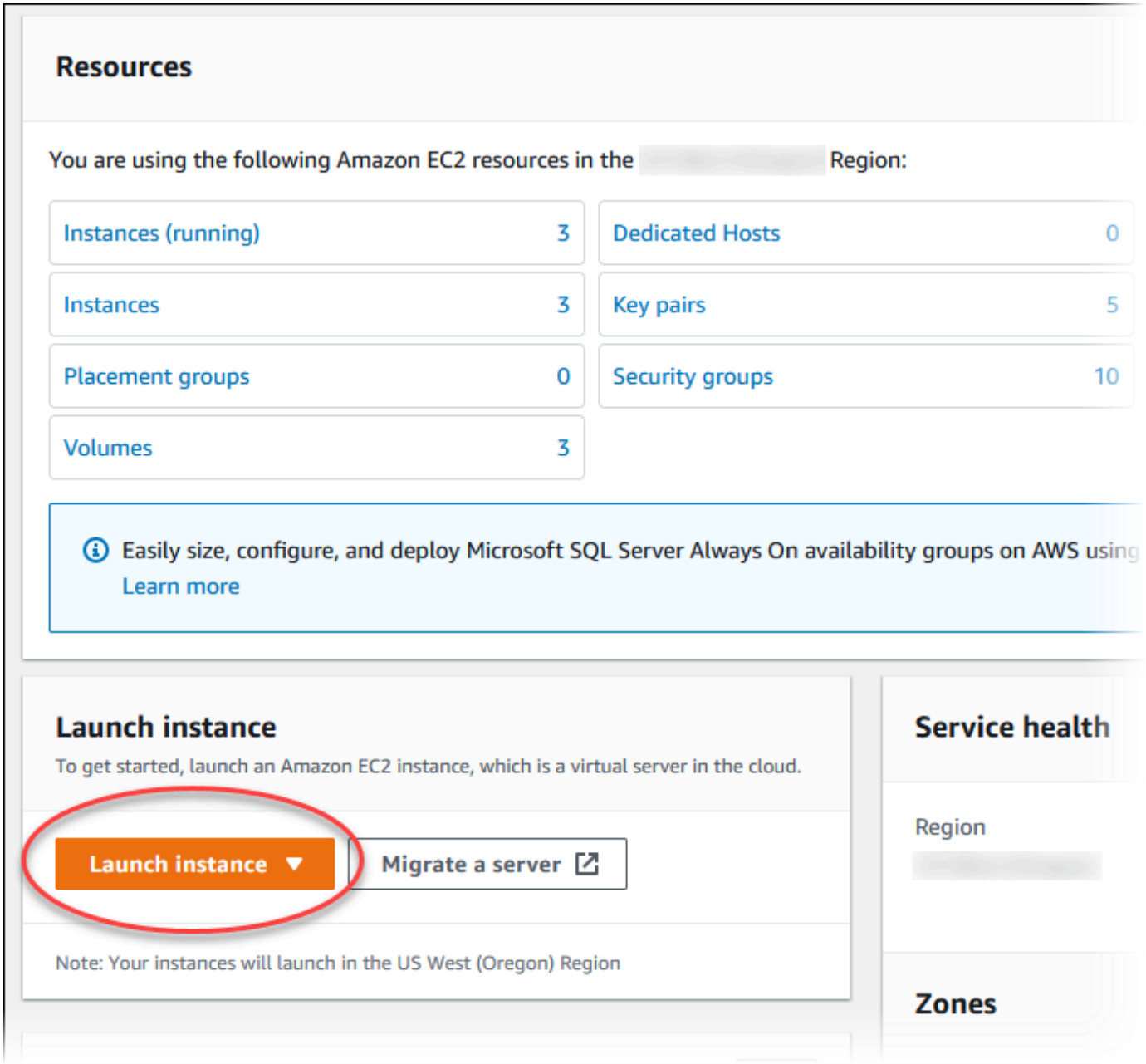
- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Erstellen einer EC2-Instance

Erstellen Sie eine EC2 Amazon-Instance, mit der Sie eine Verbindung zu Ihrer Datenbank herstellen.

Um eine EC2 Instance zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen rechten Ecke von die aus AWS Management Console, die AWS-Region Sie zuvor für die Datenbank verwendet haben.
3. Wählen Sie EC2 Dashboard und dann Launch instance aus, wie in der folgenden Abbildung gezeigt.

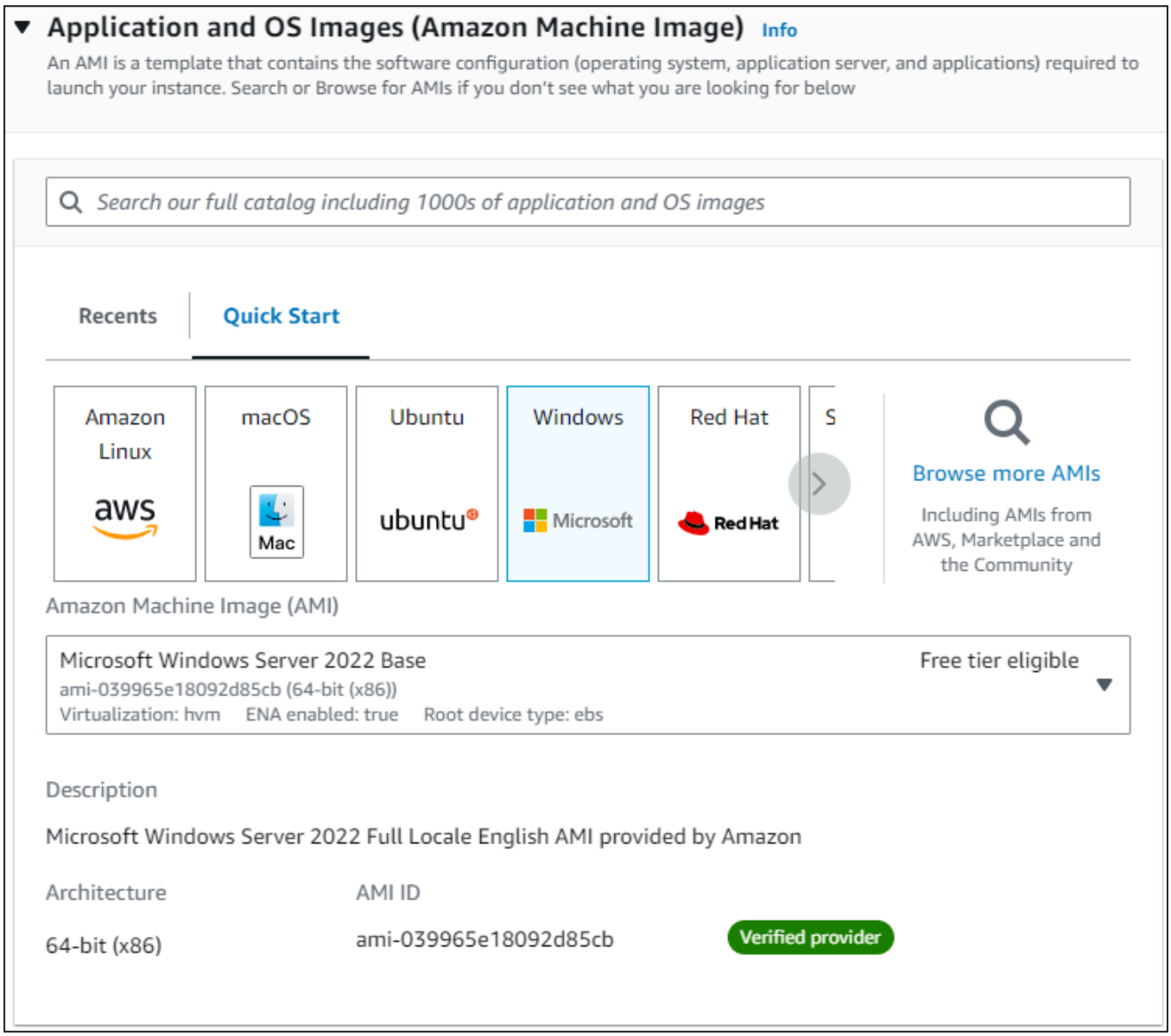


Die Seite Eine Instance starten wird geöffnet.

4. Wählen Sie auf der Seite Eine Instance starten die folgenden Einstellungen aus.

   a. Geben Sie unter Name and tags (Name und Tags) als Name den Namen **`ec2-database-connect`** ein.

   b. Wählen Sie unter Anwendungs- und Betriebssystemabbilder (Amazon Machine Image) die Option Windows und dann Microsoft Windows Server 2022 Base aus. Übernehmen Sie für alle anderen Einstellungen die Standardwerte.



c. Wählen Sie unter Instance type (Instance-Typ) den Wert t2.micro aus.

d. Wählen Sie unter Key pair (login) (Schlüsselpaar (Anmeldung)) einen Key pair name (Schlüsselpaarname), um ein vorhandenes Schlüsselpaar zu verwenden. Um ein neues key pair für die EC2 Amazon-Instance zu erstellen, wählen Sie Neues key pair erstellen und verwenden Sie dann das Fenster key pair erstellen, um es zu erstellen.

   Weitere Informationen zum Erstellen eines neuen Schlüsselpaars finden Sie unter Create a key pair im EC2 Amazon-Benutzerhandbuch für Windows-Instances.

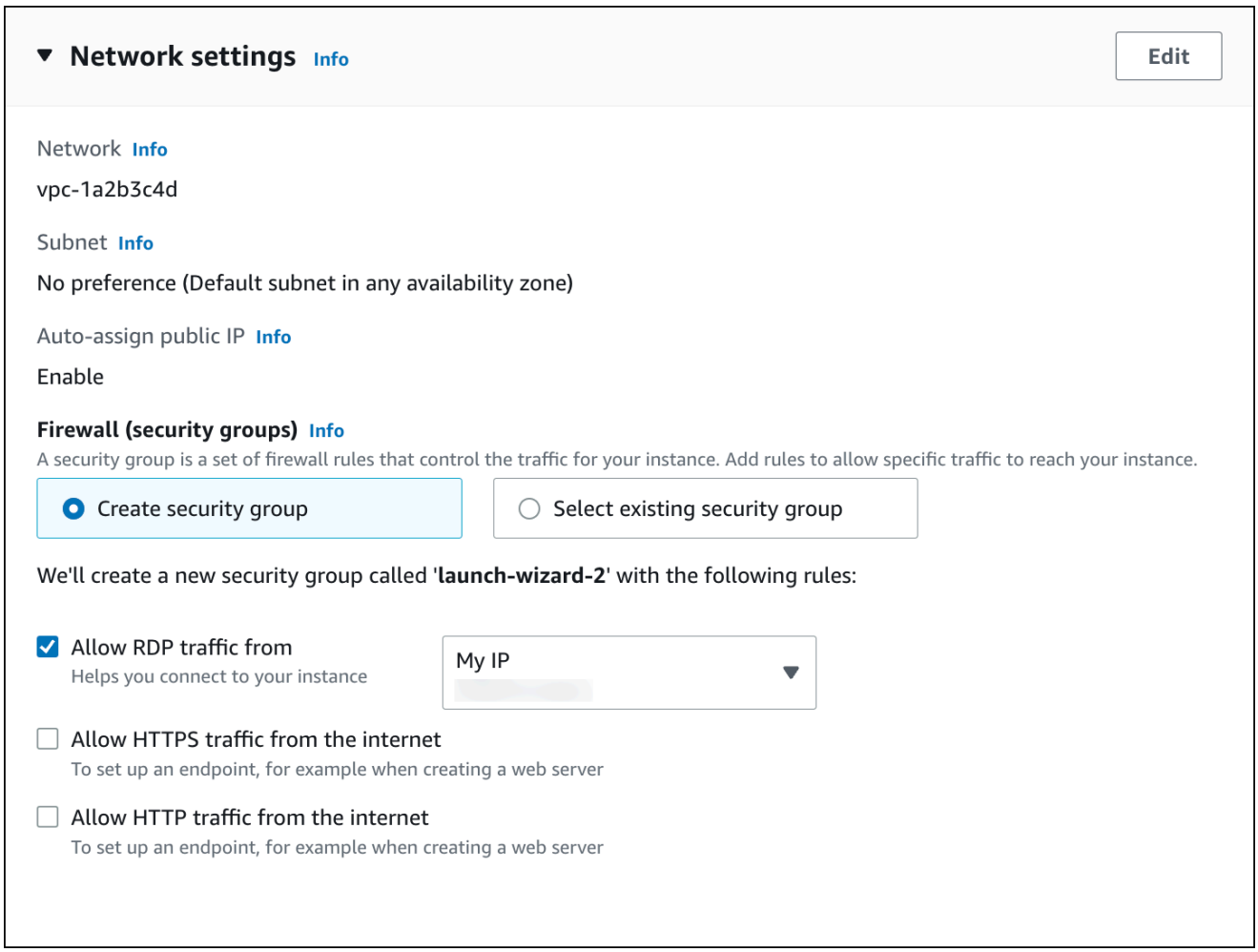
e. Wählen Sie unter Firewall (Sicherheitsgruppen) in den Netzwerkeinstellungen die Option RDP-Verkehr zulassen von, um eine Verbindung zur EC2 Instance herzustellen.

Sie können Mein IP auswählen, wenn die angezeigte IP-Adresse für RDP-Verbindungen korrekt ist. Andernfalls können Sie die IP-Adresse bestimmen, die verwendet werden soll, um mithilfe von RDP eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen. Um Ihre öffentliche IP-Adresse in einem anderen Browserfenster oder einer anderen Registerkarte zu ermitteln, können Sie den Dienst unter verwenden. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Bestimmen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den RDP-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instanzen über RDP. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktion nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instanzen mithilfe von RDP.

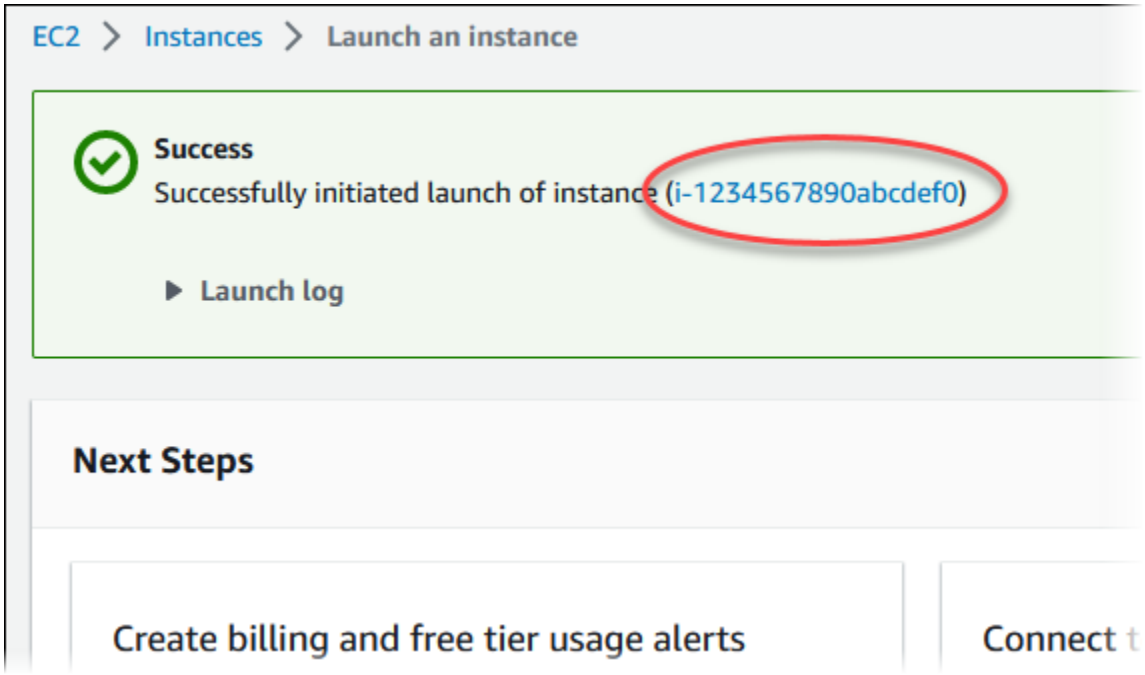
Die folgende Abbildung zeigt ein Beispiel für den Bereich Netzwerkeinstellungen.



f. Übernehmen Sie die Standardwerte für die übrigen Abschnitte.

g. Sehen Sie sich im Bereich Zusammenfassung eine Zusammenfassung Ihrer EC2 Instance-Konfiguration an. Wenn Sie bereit sind, wählen Sie Launch instance aus.

5. Notieren Sie sich auf der Seite Startstatus die ID für Ihre neue EC2 Instance, zum Beispiel:`i-1234567890abcdef0`.



6. Wählen Sie die EC2 Instance-ID, um die Liste der EC2 Instances zu öffnen.
7. Warten Sie, bis der Instanzstatus Ihrer EC2 Instance den Status Running hat, bevor Sie fortfahren.

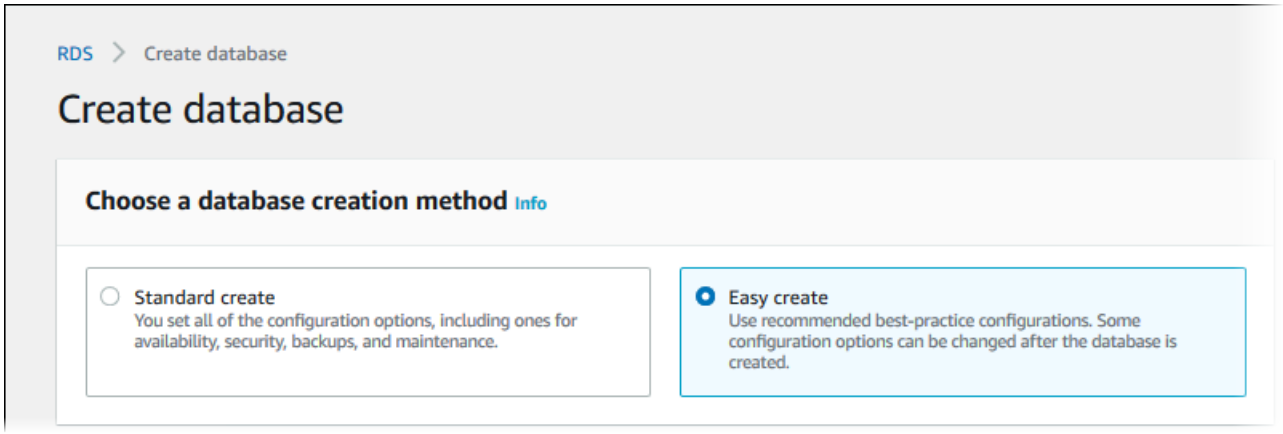
Erstellen Sie eine SQL Server-DB-Instance

Die Grundbausteine für Amazon RDS sind Datenbank-Instances. In dieser Umgebung führen Sie Ihre SQL-Server-Datenbanken aus.

Für dieses Beispiel verwenden Sie Einfache Erstellung, um eine DB-Instance zu erstellen, die die Datenbank-Engine von SQL Server mit einer DB-Instance-Klasse des Typs „db.t2.micro“ ausführt.

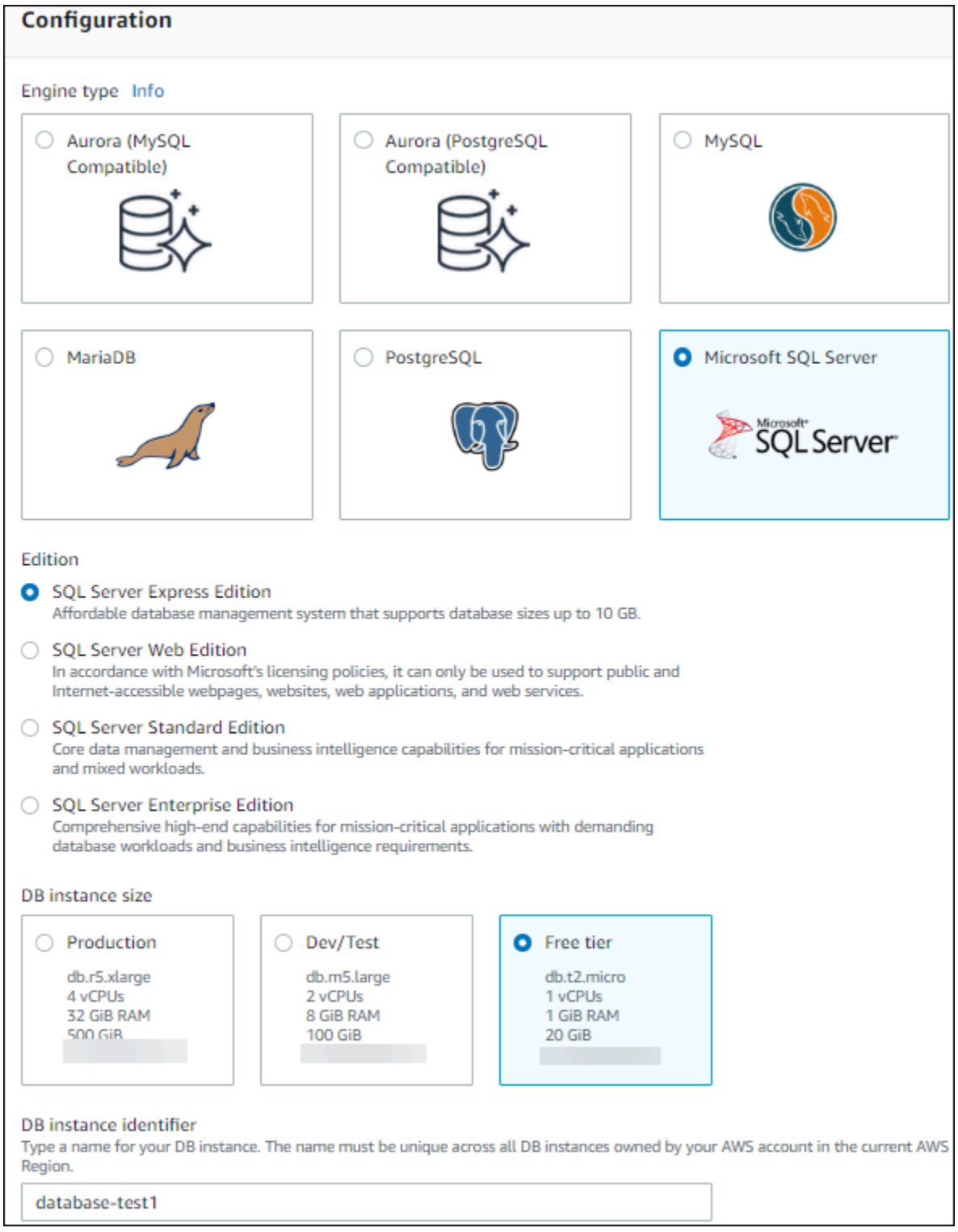
So erstellen Sie eine Microsoft SQL Server DB-Instance mit Easy Create (Einfache Erstellung)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die aus, AWS-Region in der Sie die DB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Datenbank erstellen aus und vergewissern Sie sich, dass Einfache Erstellung ausgewählt ist.



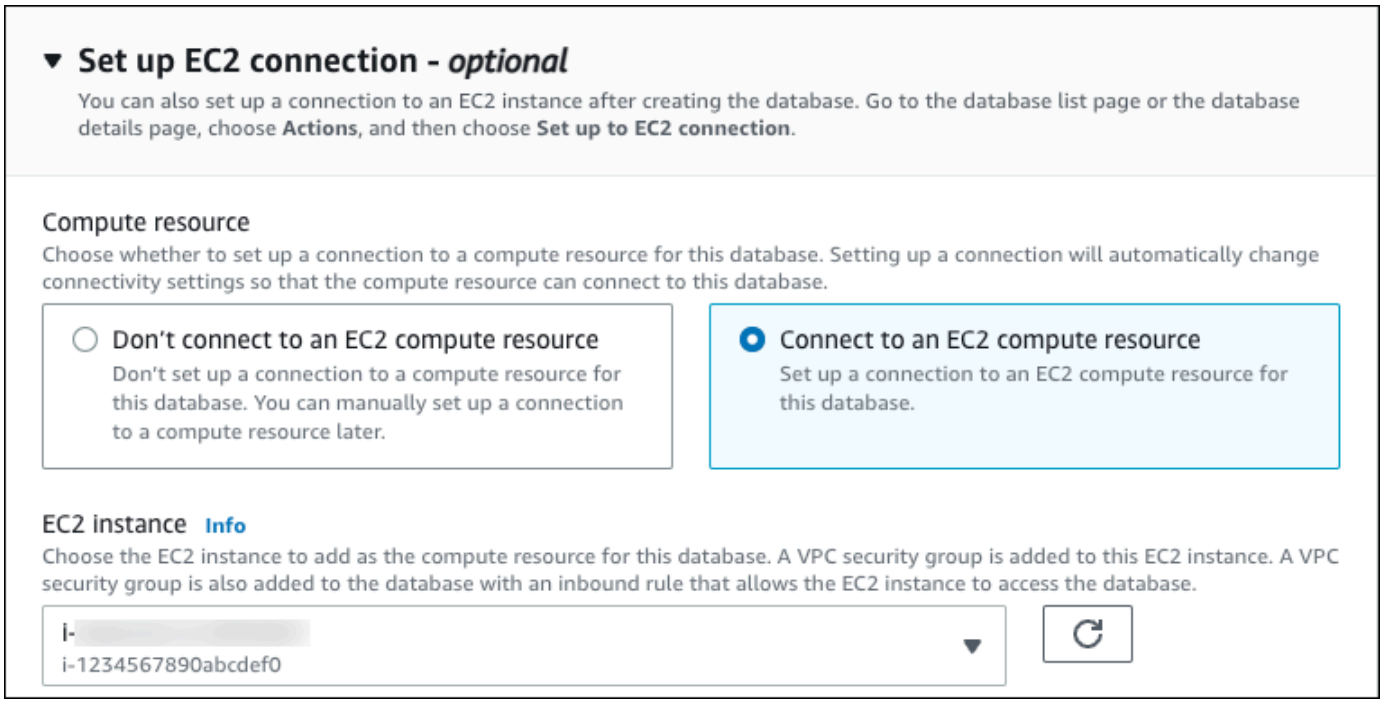
5. Wählen Sie unter Configuration (Konfiguration), die Option Microsoft SQL Server.
6. Wählen Sie unter Edition die Option SQL Server Express Edition aus.
7. Wählen Sie in DB-Instance-Größe die Option Kostenloses Kontingent aus. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.
8. Geben Sie als DB-Instance-ID **`database-test1`** ein.

   Die Seite Datenbank erstellen sollte ähnlich wie in der folgenden Abbildung gezeigt aussehen. Bei Konten mit kostenlosem Tarif wird das kostenlose Kontingent angezeigt. Bei Konten mit kostenpflichtigem Tarif wird Sandbox angezeigt.



9. Geben Sie unter Hauptbenutzername einen Namen für den Hauptbenutzer ein oder behalten Sie den Standardnamen bei.
10. Um eine Verbindung mit der EC2 Instanz einzurichten, die Sie zuvor erstellt haben, öffnen Sie EC2 Verbindung einrichten — optional.

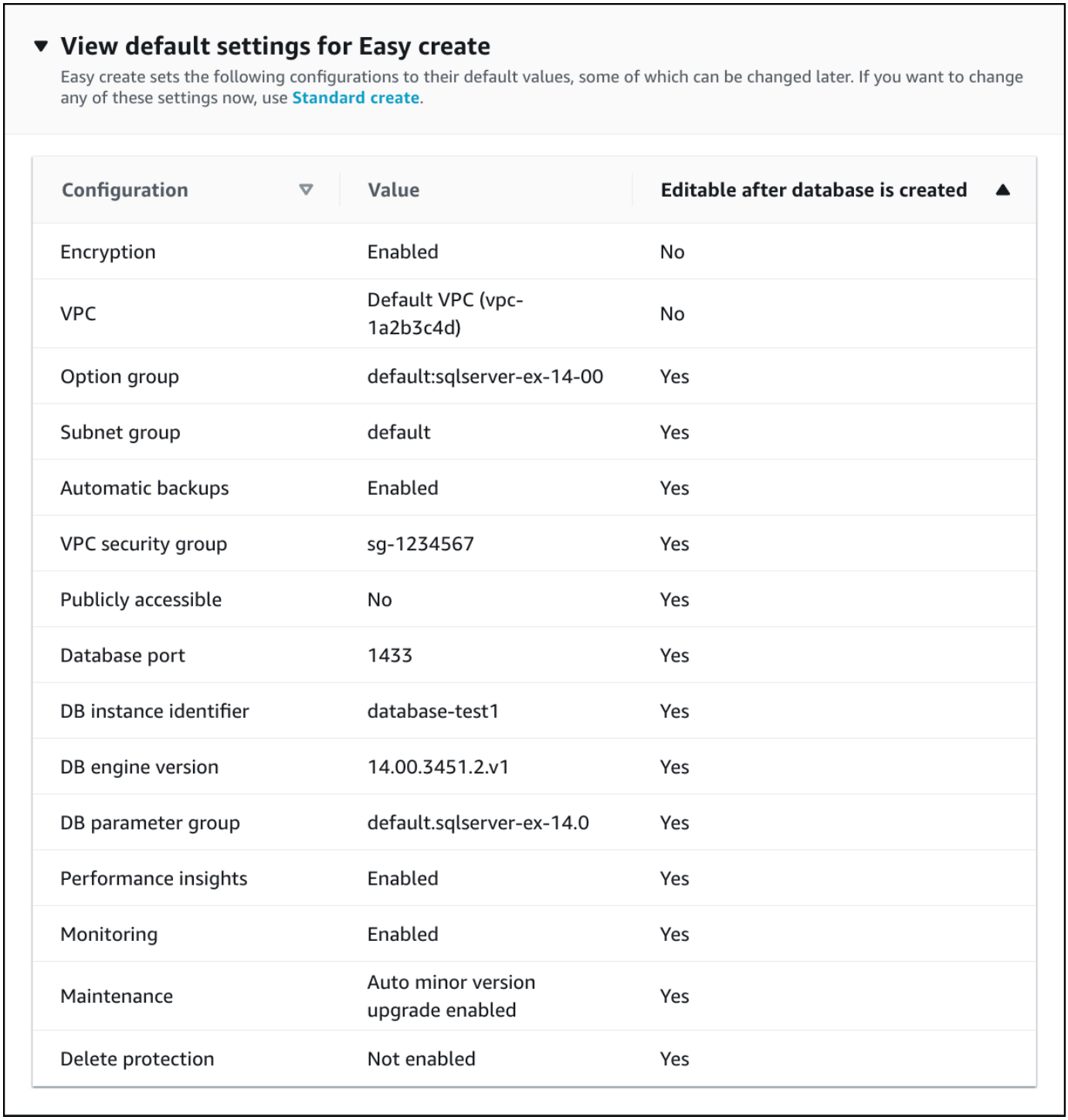
Wählen Sie Mit einer EC2 Rechenressource verbinden aus. Wählen Sie die EC2 Instanz aus, die Sie zuvor erstellt haben.



11. Um für die DB-Instance ein automatisch generiertes Hauptpasswort zu verwenden, wählen Sie das Kästchen Auto generate a password (Passwort automatisch generieren) aus.

    Um das Hauptpasswort einzugeben, deaktivieren Sie das Kästchen Auto generate a password (Passwort automatisch generieren) und geben Sie anschließend dasselbe Passwort in Master password (Masterpasswort) und Confirm password (Passwort bestätigen) ein.

12. (Optional) Öffnen Sie Anzeigen von Standardeinstellungen für eine einfache Erstellung.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:sqlserver-ex-14-00	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	1433	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.00.3451.2.v1	Yes
DB parameter group	default.sqlserver-ex-14.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Sie können die Standardeinstellungen von Einfache Erstellung einsehen. Die Spalte Nach Erstellung der Datenbank editierbar zeigt, welche Optionen Sie nach der Datenbankerstellung ändern können.

- Wenn in einer Einstellung Nein in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie Standarderstellung verwenden, um die DB-Instance zu erstellen.

- Wenn für eine Einstellung Ja in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie entweder Standarderstellung verwenden, oder die DB-Instance nach der Erstellung ändern, um die Einstellung zu ändern.

13. Wählen Sie Datenbank erstellen aus.

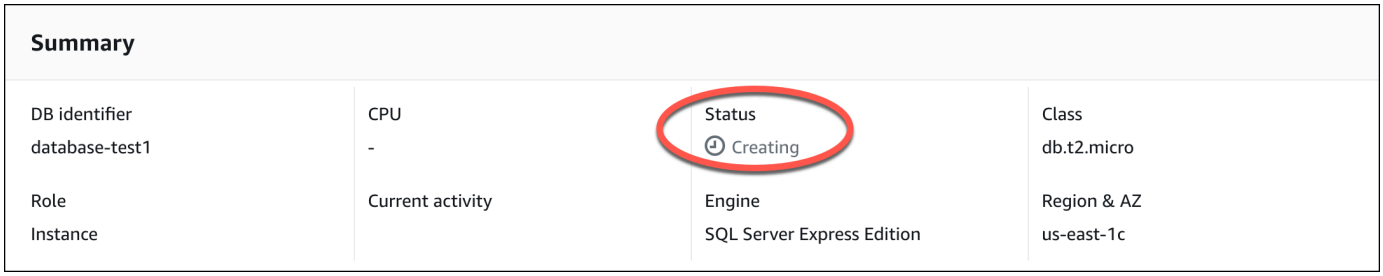
    Um den Masterbenutzernamen und das zugehörige Passwort für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen).

    Sie können den angezeigten Benutzernamen und das angezeigte Passwort verwenden, um als Masterbenutzer eine Verbindung zu DB-Instance herzustellen.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern.
    > Wenn Sie das Passwort für den Hauptbenutzer ändern müssen, nachdem die DB-Instance verfügbar wurde, können Sie die DB-Instance entsprechend ändern. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

14. Wählen Sie in der Liste Datenbanken den Namen der neuen DB-Instance von SQL Server aus, um deren Details anzuzeigen.

    Die DB-Instance hat den Status Wird erstellt, bis die DB-Instance bereit für die Verwendung ist.



    Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und vom verfügbaren Speicherplatz kann es bis zu 20 Minuten dauern, bis die neue DB-Instance verfügbar ist.

(Optional) Erstellen Sie VPC, EC2 Instanz und SQL Server-Instanz mit AWS CloudFormation

Anstatt die Konsole zum Erstellen Ihrer VPC, EC2 Instanz und SQL Server-Instanz AWS CloudFormation zu verwenden, können Sie AWS Ressourcen bereitstellen, indem Sie Infrastruktur als Code behandeln. Um Ihnen zu helfen, Ihre AWS Ressourcen in kleinere und besser verwaltbare Einheiten zu organisieren, können Sie die AWS CloudFormation Nested-Stack-Funktionalität verwenden. Weitere Informationen finden Sie unter Einen Stack auf der AWS CloudFormation Konsole erstellen und Mit verschachtelten Stacks arbeiten.

> **Important**
>
> AWS CloudFormation ist kostenlos, aber die Ressourcen, die CloudFormation erstellt werden, sind live. Es fallen die üblichen Nutzungsgebühren für diese Ressourcen an, bis Sie sie kündigen. Weitere Informationen finden Sie unter Preise für RDS for SQL Server.

Gehen Sie wie folgt vor, um Ihre Ressourcen mithilfe der AWS CloudFormation Konsole zu erstellen:

- Laden Sie die CloudFormation Vorlage herunter
- Konfigurieren Sie Ihre Ressourcen mit CloudFormation

Laden Sie die CloudFormation Vorlage herunter

Eine CloudFormation Vorlage ist eine JSON- oder YAML-Textdatei, die die Konfigurationsinformationen zu den Ressourcen enthält, die Sie im Stack erstellen möchten. Diese Vorlage erstellt zusammen mit der RDS-Instanz auch eine VPC und einen Bastion-Host für Sie.

Um die Vorlagendatei herunterzuladen, öffnen Sie den folgenden Link: SQL CloudFormation Server-Vorlage.

Klicken Sie auf der Github-Seite auf die Schaltfläche Rohdatei herunterladen, um die YAML-Vorlagendatei zu speichern.

Konfigurieren Sie Ihre Ressourcen mit CloudFormation

> **Note**
>
> Bevor Sie diesen Vorgang starten, stellen Sie sicher, dass Sie ein Schlüsselpaar für eine EC2 Instance in Ihrem haben AWS-Konto. Weitere Informationen finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

Wenn Sie die AWS CloudFormation Vorlage verwenden, müssen Sie die richtigen Parameter auswählen, um sicherzustellen, dass Ihre Ressourcen ordnungsgemäß erstellt werden. Führen Sie die folgenden Schritte aus:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie Stapel erstellen aus.
3. Wählen Sie im Abschnitt Vorlage angeben die Option Eine Vorlagendatei von Ihrem Computer hochladen und dann Weiter aus.
4. Legen Sie auf der Seite „Stack-Details angeben“ die folgenden Parameter fest:
   a. Setzen Sie den Stacknamen auf SQLServerTestStack.
   b. Legen Sie unter Parameter Availability Zones fest, indem Sie drei Availability Zones auswählen.
   c. Wählen Sie unter Linux Bastion Host configuration für Key Name ein key pair aus, um sich bei Ihrer EC2 Instance anzumelden.
   d. Stellen Sie in den Linux Bastion Host-Konfigurationseinstellungen den Bereich Zulässiger IP auf Ihre IP-Adresse ein. Um mithilfe von Secure Shell (SSH) eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen, ermitteln Sie Ihre öffentliche IP-Adresse mithilfe des Dienstes unter. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instances über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktion nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

e. Stellen Sie unter Allgemeine Datenbankkonfiguration die Datenbankinstanzklasse auf db.t3.micro ein.

f. Setzen Sie den Datenbanknamen auf. **`database-test1`**

g. Geben Sie unter Datenbank-Master-Benutzername einen Namen für den Masterbenutzer ein.

h. Stellen Sie `false` für dieses Tutorial das DB-Master-Benutzerpasswort mit Secrets Manager verwalten auf ein.

i. Geben Sie für das Datenbankkennwort ein Passwort Ihrer Wahl ein. Merken Sie sich dieses Passwort für weitere Schritte im Tutorial.

j. Stellen Sie unter Datenbankspeicherkonfiguration den Datenbankspeichertyp auf gp2 ein.

k. Stellen Sie unter Konfiguration der Datenbanküberwachung die Option Enable RDS Performance Insights auf false ein.

l. Behalten Sie für alle anderen Einstellungen die Standardwerte bei. Klicken Sie auf Weiter, um fortzufahren.

5. Behalten Sie auf der Seite „Stack-Optionen konfigurieren“ alle Standardoptionen bei. Klicken Sie auf Weiter, um fortzufahren.
6. Wählen Sie auf der Seite „Stack überprüfen“ die Option Senden aus, nachdem Sie die Datenbank- und Linux-Bastion-Host-Optionen überprüft haben.

Sehen Sie sich nach Abschluss der Stack-Erstellung die Stacks mit Namen BastionStackund RDSNS an, um die Informationen zu notieren, die Sie für die Verbindung mit der Datenbank benötigen. Weitere Informationen finden Sie unter AWS CloudFormation Stack-Daten und Ressourcen anzeigen auf der. AWS Management Console

Connect zu Ihrer SQL Server-DB-Instance her

Im folgenden Verfahren stellen Sie mithilfe von Microsoft SQL Server Management Studio (SSMS) eine Verbindung mit Ihrer DB-Instance her.

So stellen Sie mithilfe von SSMS eine Verbindung mit einer DB-Instance von RDS für SQL Server her

1. Suchen Sie nach dem Endpunkt (DNS-Name) und der Portnummer für Ihre DB-Instance.

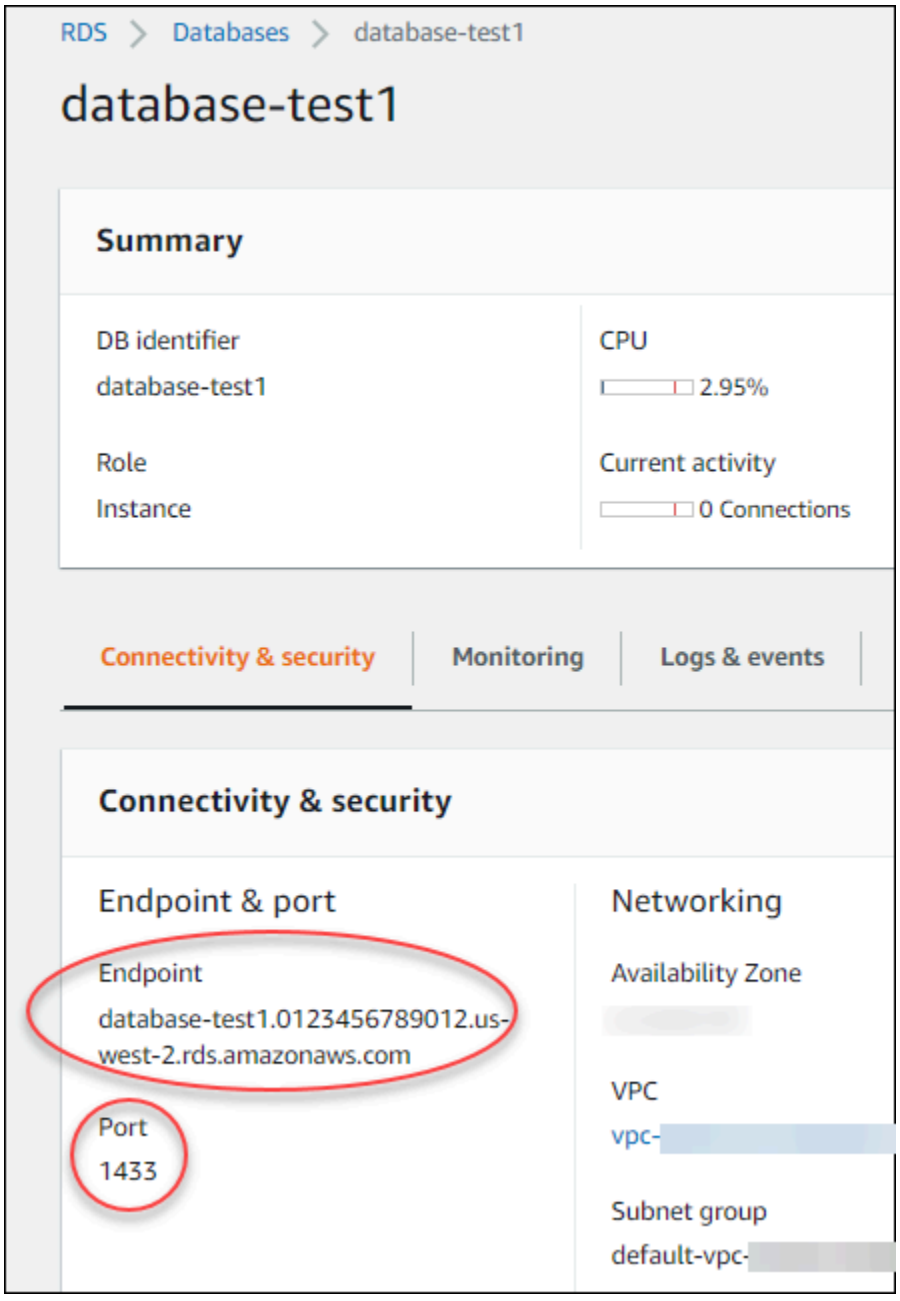
   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

   b. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die AWS-Region für die DB-Instance aus.

c. Wählen Sie im Navigationsbereich Datenbanken aus.

d. Wählen Sie den Namen der SQL Server-DB-Instance, um deren Details anzuzeigen.

e. Kopieren Sie auf der Registerkarte Anbindung den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



2. Stellen Sie eine Verbindung zu der EC2 Instance her, die Sie zuvor erstellt haben, indem Sie den Schritten unter Connect to your Microsoft Windows Instance im EC2 Amazon-Benutzerhandbuch für Windows-Instances folgen.

3. Installieren Sie den SQL Server Management Studio (SSMS) Client von Microsoft aus.

   Informationen zum Herunterladen einer eigenständigen Version von SSMS auf Ihre EC2 Instanz finden Sie in der Microsoft-Dokumentation unter Herunterladen von SQL Server Management Studio (SSMS).

   a. Öffnen Sie Internet Explorer über das Menü „Start“.

   b. Verwenden Sie Internet Explorer, um eine eigenständige Version von SSMS herunterzuladen und zu installieren. Wenn Sie aufgefordert werden, zu bestätigen, dass die Website nicht vertrauenswürdig ist, fügen Sie die Website der Liste der vertrauenswürdigen Websites hinzu.

4. Starten Sie SQL Server Management Studio (SSMS).

   Das Dialogfeld Connect to Server (Mit Server verbinden) erscheint.

5. Geben Sie die folgenden Informationen für Ihre Beispiel-DB-Instance an:

   a. Wählen Sie für Servertyp die Option Datenbank-Engine aus.

   b. Geben Sie für Server name (Servername) den DNS-Namen, gefolgt von einem Komma und der Portnummer (der Standard-Port ist 1433) ein. Ihr Servername sollte z. B. wie folgt aussehen:

   ```
   database-test1.0123456789012.us-west-2.rds.amazonaws.com,1433
   ```

   c. Wählen Sie für Authentifizierung die Option SQL Server-Authentifizierung aus.

   d. Geben Sie unter Anmeldung den Benutzernamen ein, den Sie für Ihre Beispiel-DB-Instance gewählt haben. Dieser wird auch als Hauptbenutzername bezeichnet.

   e. Geben Sie für Password (Passwort) das Passwort ein, das Sie vorher für Ihre Beispiel-DB-Instance ausgewählt haben. Dies wird auch als Master-Benutzerpasswort bezeichnet.

6. Wählen Sie Connect (Verbinden) aus.

   Nach wenigen Augenblicken stellt SSMS die Verbindung zur DB-Instance her. Aus Sicherheitsgründen empfiehlt es sich, verschlüsselte Verbindungen zu verwenden. Verwenden Sie eine unverschlüsselte SQL-Server-Verbindung nur, wenn sich Client und Server in derselben VPC befinden und das Netzwerk vertrauenswürdig ist. Weitere Informationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Verwendung SSL mit einer Microsoft SQL Server-DB-Instance.

Weitere Informationen zum Herstellen einer Verbindung mit einer Microsoft SQL Server DB-Instance finden Sie unter Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen.

Weitere Informationen zu Verbindungsproblemen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Erkunden Sie Ihre Beispiel-SQL Server-DB-Instance

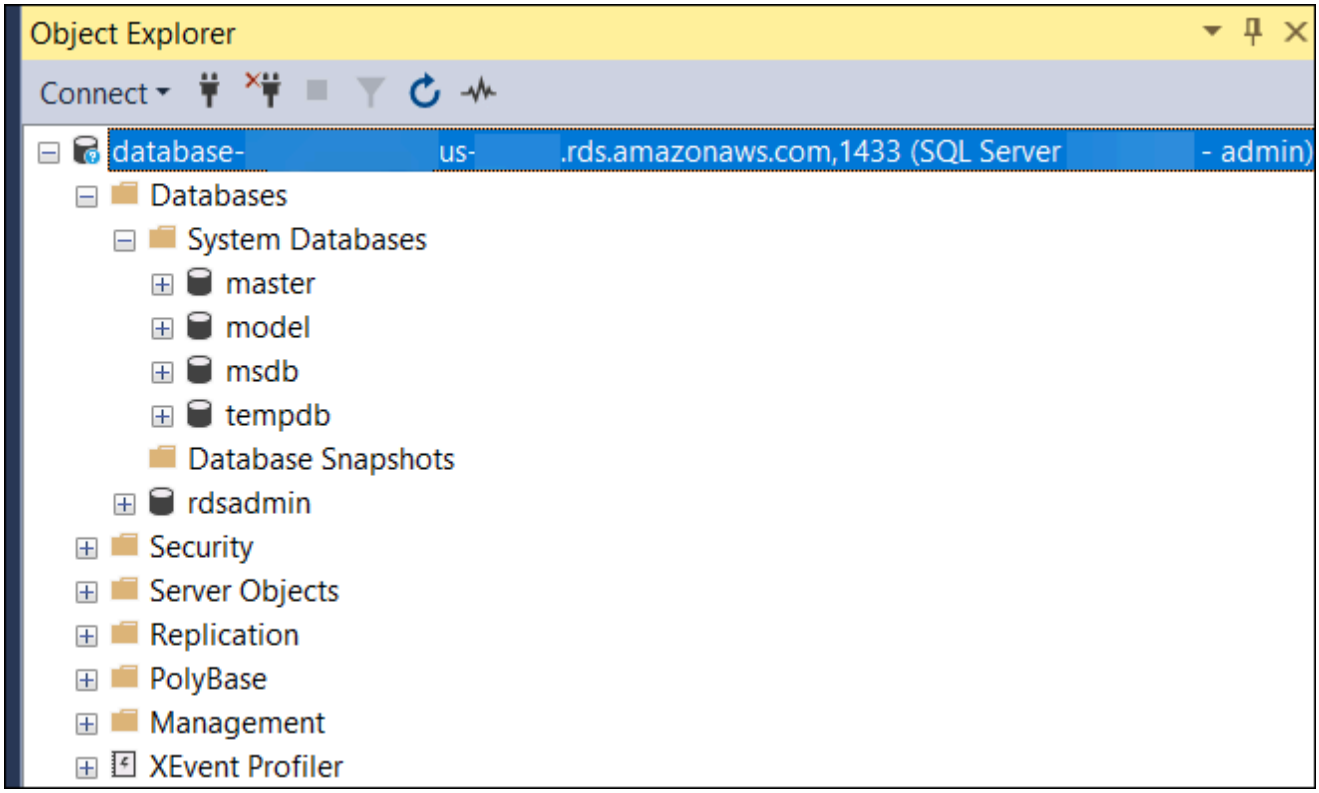
Sie können Ihre Beispiel-DB-Instance mithilfe von Microsoft SQL Server Management Studio (SSMS) erkunden.

So können Sie mit einer DB-Instance mithilfe von SSMS arbeiten

1. Ihre SQL Server-DB-Instance verfügt über integrierte Standard-Systemdatenbanken von SQL Server (master, model, msdb und tempdb). Führen Sie folgende Schritte aus, um die Systemdatenbanken zu durchforschen:

   a. Wählen Sie in SSMS im Menü Ansicht die Option Objekt-Explorer aus.

   b. Erweitern Sie die DB-Instance, Datenbanken und anschließend Systemdatenbanken wie gezeigt.



Ihre SQL-Server-DB-Instance kommt auch mit einer Datenbank namens `rdsadmin`. Amazon RDS verwendet diese Datenbank, um die Objekte für die Datenbankverwaltung zu speichern. Die Datenbank `rdsadmin` beinhaltet auch gespeicherte Prozeduren, die Sie ausführen können, um erweiterte Aufgaben durchzuführen.

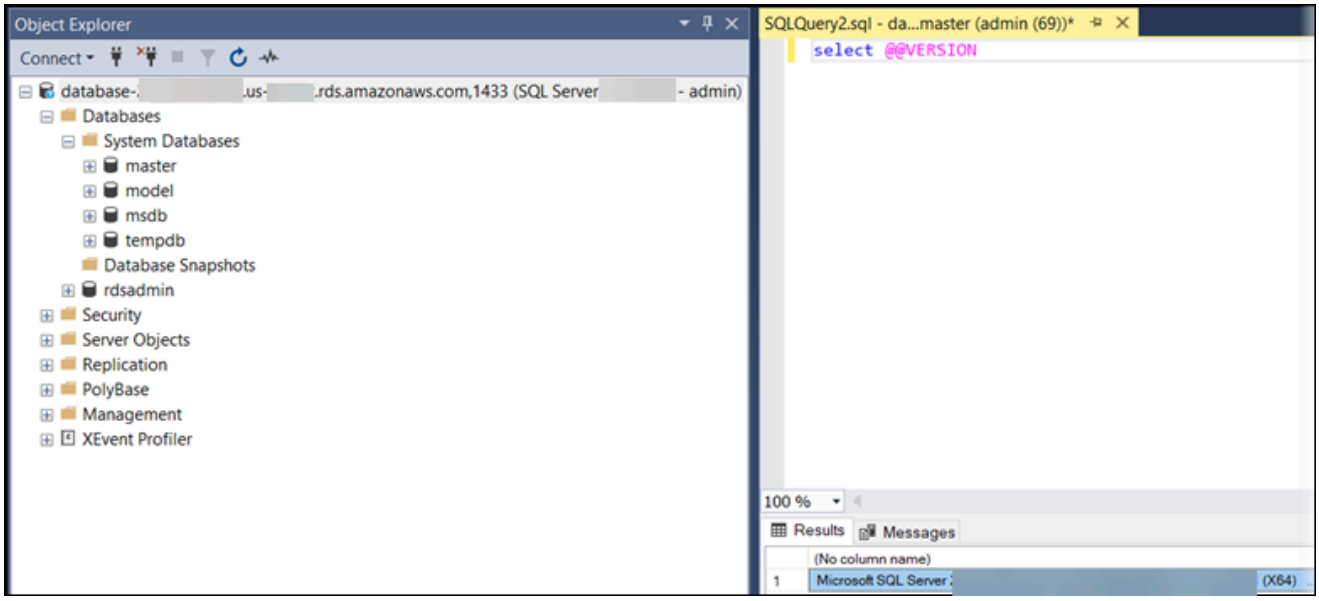
2. Beginnen Sie nun wie üblich, Ihre eigenen Datenbanken zu erstellen und Abfragen gegen Ihre DB-Instance und Datenbanken auszuführen. Führen Sie Folgendes aus, um eine Testabfrage in Ihrer Beispiel-DB-Instance auszuführen:

   a. Wählen Sie in SSMS im Menü Datei die Option Neu und anschließend Abfrage mit bestehender Verbindung aus.

   b. Geben Sie die folgende SQL-Abfrage ein:

   ```
   select @@VERSION
   ```

   c. Führen Sie die Abfrage aus. SSMS gibt die SQL Server-Version Ihrer Amazon RDS-DB-Instance zurück.



Löschen Sie die EC2 Instanz und die DB-Instance

Nachdem Sie sich mit der EC2 Beispiel-Instance und DB-Instance, die Sie erstellt haben, verbunden und diese erkundet haben, löschen Sie sie, damit Ihnen keine Gebühren mehr berechnet werden.

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, überspringen Sie diesen Schritt und fahren Sie mit dem nächsten Schritt fort.

Um die EC2 Instanz zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die EC2 Instance aus und wählen Sie dann Instance-Status, Instanz beenden aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

Weitere Informationen zum Löschen einer EC2 Instanz finden Sie unter Kündigen Ihrer Instanz im Benutzerhandbuch für Windows-Instances.

So löschen Sie die DB-Instance ohne finalen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Löschen Sie Abschließenden Snapshot erstellen? und Automatische Backups aufbewahren.
6. Bestätigen Sie, und wählen Sie Löschen.

(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, löschen Sie den CloudFormation Stack, nachdem Sie sich mit der EC2 Beispiel-Instance und der DB-Instance verbunden und diese erkundet haben, sodass Ihnen diese nicht mehr in Rechnung gestellt werden.

Um die CloudFormation Ressourcen zu löschen

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite Stacks in den den CloudFormationconsole Root-Stack aus (den Stack ohne Namen VPCStack BastionStack oder RDSNS).

3. Wählen Sie Löschen aus.
4. Wählen Sie Stack löschen, wenn Sie zur Bestätigung aufgefordert werden.

Weitere Informationen zum Löschen eines Stacks in CloudFormation finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks auf der AWS CloudFormation Konsole.

(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Sie können Ihre DB-Instance von RDS für SQL Server auch mit einer Lambda-Serverless-Rechenressource verbinden. Mit Lambda-Funktionen können Sie Code ausführen, ohne die Infrastruktur bereitstellen oder verwalten zu müssen. Eine Lambda-Funktion ermöglicht es Ihnen auch, automatisch auf Codeausführungsanfragen jeder Größenordnung zu reagieren, von einem Dutzend Ereignissen pro Tag bis hin zu Hunderten von Ereignissen pro Sekunde. Weitere Informationen finden Sie unter Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance.

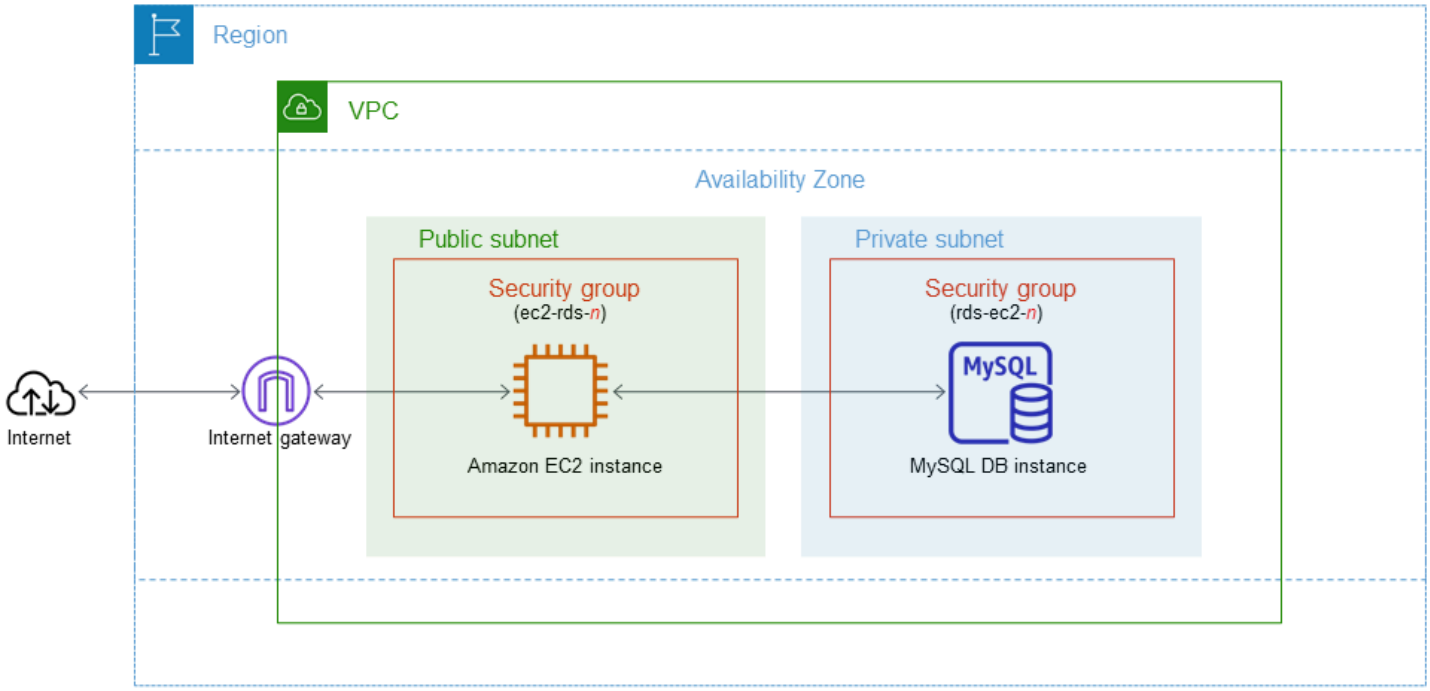
Erstellen einer MySQL-DB-Instance und Herstellen einer Verbindung dazu

In diesem Tutorial werden eine EC2 Instance und eine RDS für MySQL-DB-Instance erstellt. Das Tutorial zeigt Ihnen, wie Sie mit einem Standard-MySQL-Client von der EC2 Instance aus auf die DB-Instance zugreifen. Als bewährte Methode erstellt dieses Tutorial eine private DB-Instance in einer Virtual Private Cloud (VPC). In den meisten Fällen können andere Ressourcen in derselben VPC, z. B. EC2 Instances, auf die DB-Instance zugreifen, aber Ressourcen außerhalb der VPC können nicht darauf zugreifen.

Nach Abschluss des Tutorials gibt es in jeder Availability Zone im VPC ein öffentliches und ein privates Subnetz. In einer Availability Zone befindet sich die EC2 Instance im öffentlichen Subnetz und die DB-Instance im privaten Subnetz.

> **Important**
>
> Für die Erstellung eines Kontos fallen keine Gebühren an AWS . Wenn Sie dieses Tutorial abschließen, können Ihnen jedoch Kosten für die von Ihnen verwendeten AWS Ressourcen entstehen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

Das folgende Diagramm zeigt die Konfiguration nach Abschluss des Tutorials.



In diesem Tutorial können Sie Ihre Ressourcen mithilfe einer der folgenden Methoden erstellen:

1. Verwenden Sie das AWS Management Console - Erstellen Sie eine MySQL-DB-Instance und Erstellen einer EC2-Instance
2. Verwenden Sie AWS CloudFormation , um die Datenbankinstanz und die EC2 Instanz zu erstellen - (Optional) Erstellen Sie VPC, EC2 Instanz und MySQL-Instanz mit AWS CloudFormation

Die erste Methode verwendet Easy Create, um eine private MySQL-DB-Instance mit dem zu erstellen AWS Management Console. Hier geben Sie nur den DB-Engine-Typ, die DB-Instance-Größe und die DB-Instance-ID an. Easy Create (Einfache Erstellung) verwendet für die anderen Konfigurationsoptionen die Standardeinstellung.

Wenn Sie stattdessen Standard create verwenden, können Sie beim Erstellen einer DB-Instance weitere Konfigurationsoptionen angeben. Zu diesen Optionen gehören Einstellungen für Verfügbarkeit, Sicherheit, Backups und Wartung. Um eine öffentliche DB-Instance zu erstellen, müssen Sie Standarderstellung verwenden. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Themen

- Voraussetzungen
- Erstellen einer EC2-Instance

- Erstellen Sie eine MySQL-DB-Instance
- (Optional) Erstellen Sie VPC, EC2 Instanz und MySQL-Instanz mit AWS CloudFormation
- Verbindung zu einer MySQL-DB-Instance
- Löschen Sie die EC2 Instance und die DB-Instance
- (Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation
- (Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

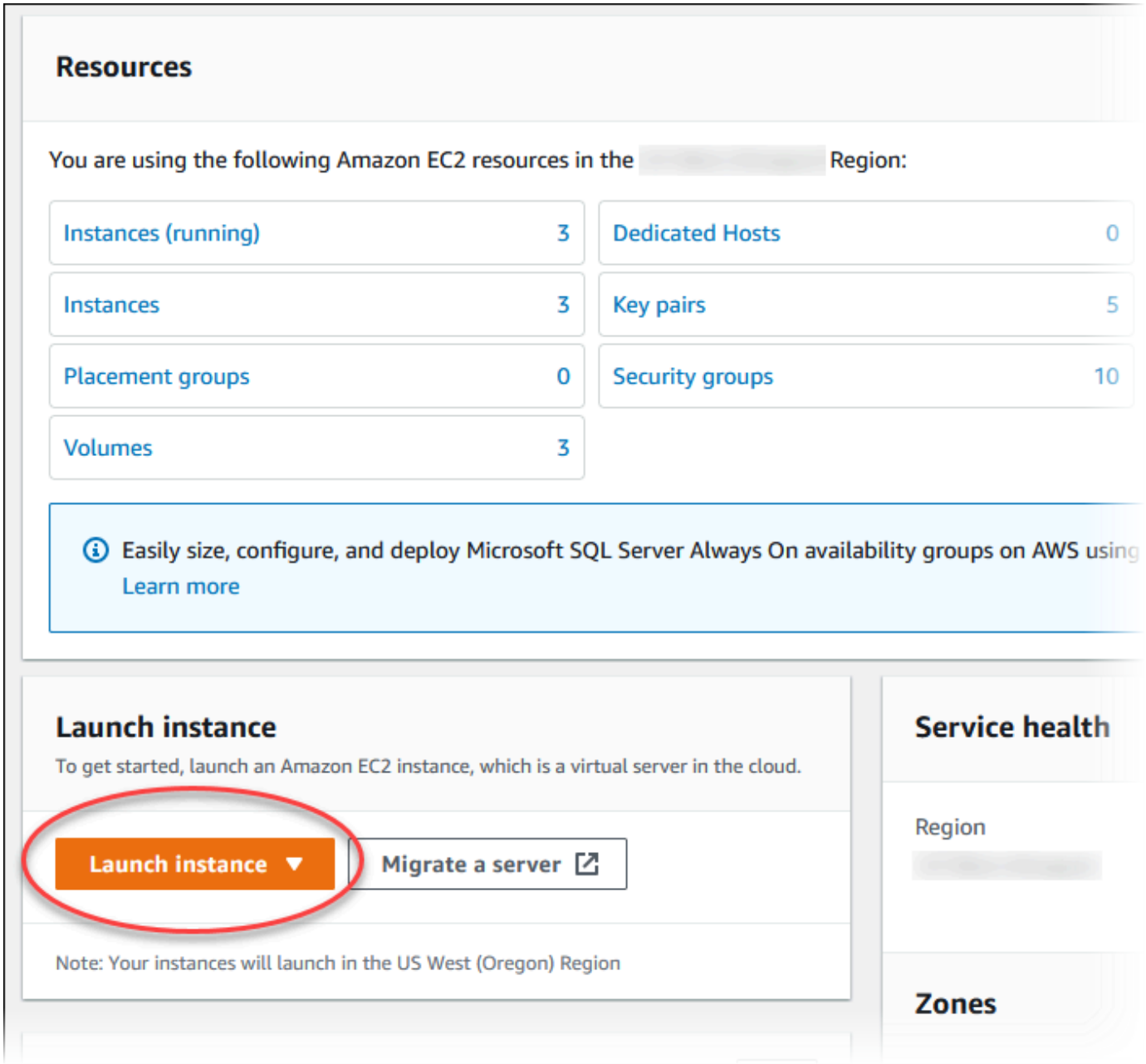
- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Erstellen einer EC2-Instance

Erstellen Sie eine EC2 Amazon-Instance, mit der Sie eine Verbindung zu Ihrer Datenbank herstellen.

Um eine EC2 Instance zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen rechten Ecke von die aus AWS Management Console, AWS-Region in der Sie die Instance erstellen möchten. EC2
3. Wählen Sie EC2 Dashboard und dann Launch instance aus, wie in der folgenden Abbildung gezeigt.

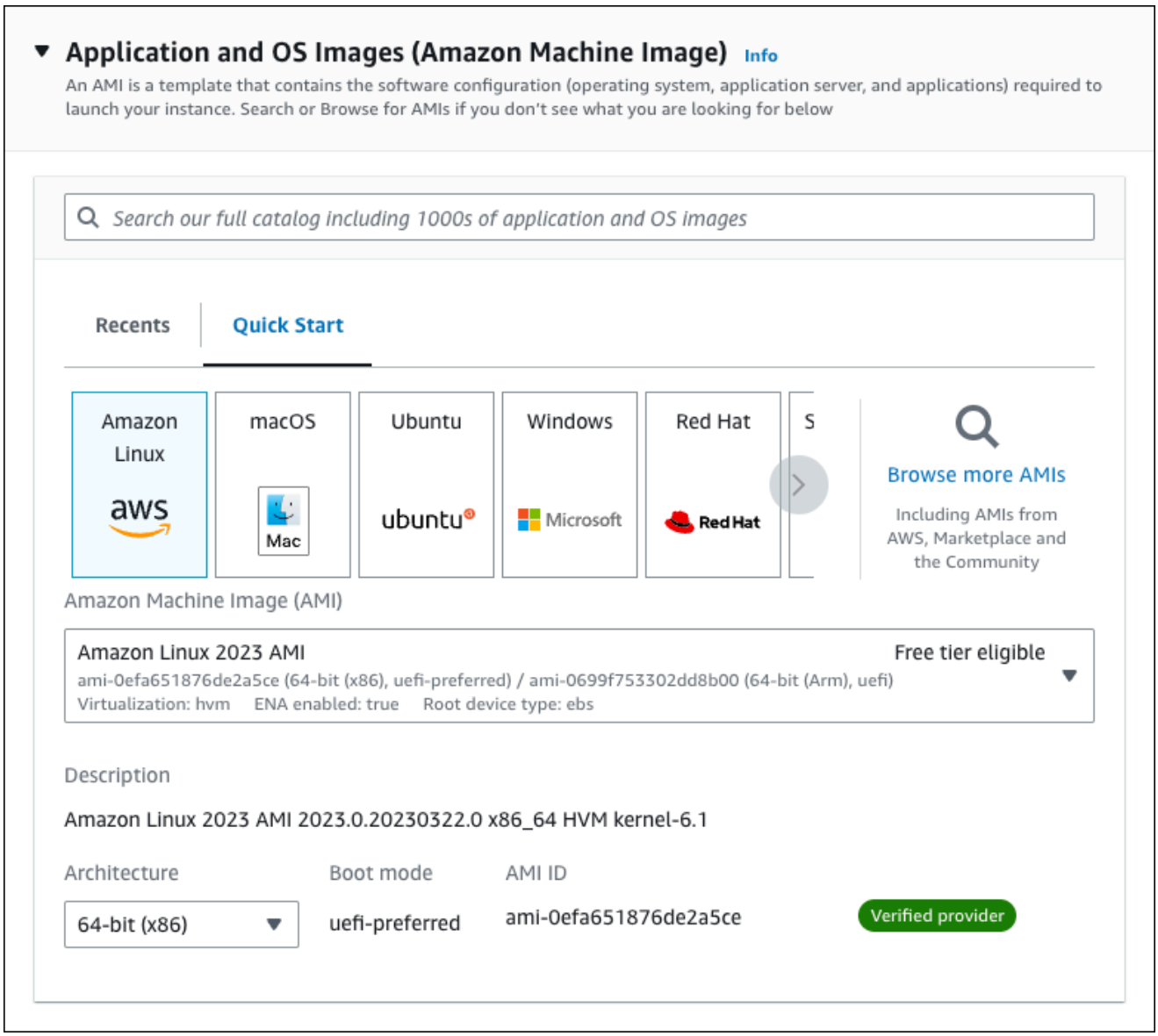


Die Seite Eine Instance starten wird geöffnet.

4. Wählen Sie auf der Seite Eine Instance starten die folgenden Einstellungen aus.

   a. Geben Sie unter Name and tags (Name und Tags) als Name den Namen **`ec2-database-connect`** ein.

   b. Wählen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) die Option Amazon Linux und dann die Registerkarte Amazon Linux 2023 AMI aus. Übernehmen Sie für alle anderen Einstellungen die Standardwerte.



c. Wählen Sie unter Instance type (Instance-Typ) den Wert t2.micro aus.

d. Wählen Sie unter Key pair (login) (Schlüsselpaar (Anmeldung)) einen Key pair name (Schlüsselpaarname), um ein vorhandenes Schlüsselpaar zu verwenden. Um ein neues key pair für die EC2 Amazon-Instance zu erstellen, wählen Sie Neues key pair erstellen und verwenden Sie dann das Fenster key pair erstellen, um es zu erstellen.

   Weitere Informationen zum Erstellen eines neuen key pair finden Sie unter Erstellen eines key pair im EC2 Amazon-Benutzerhandbuch.

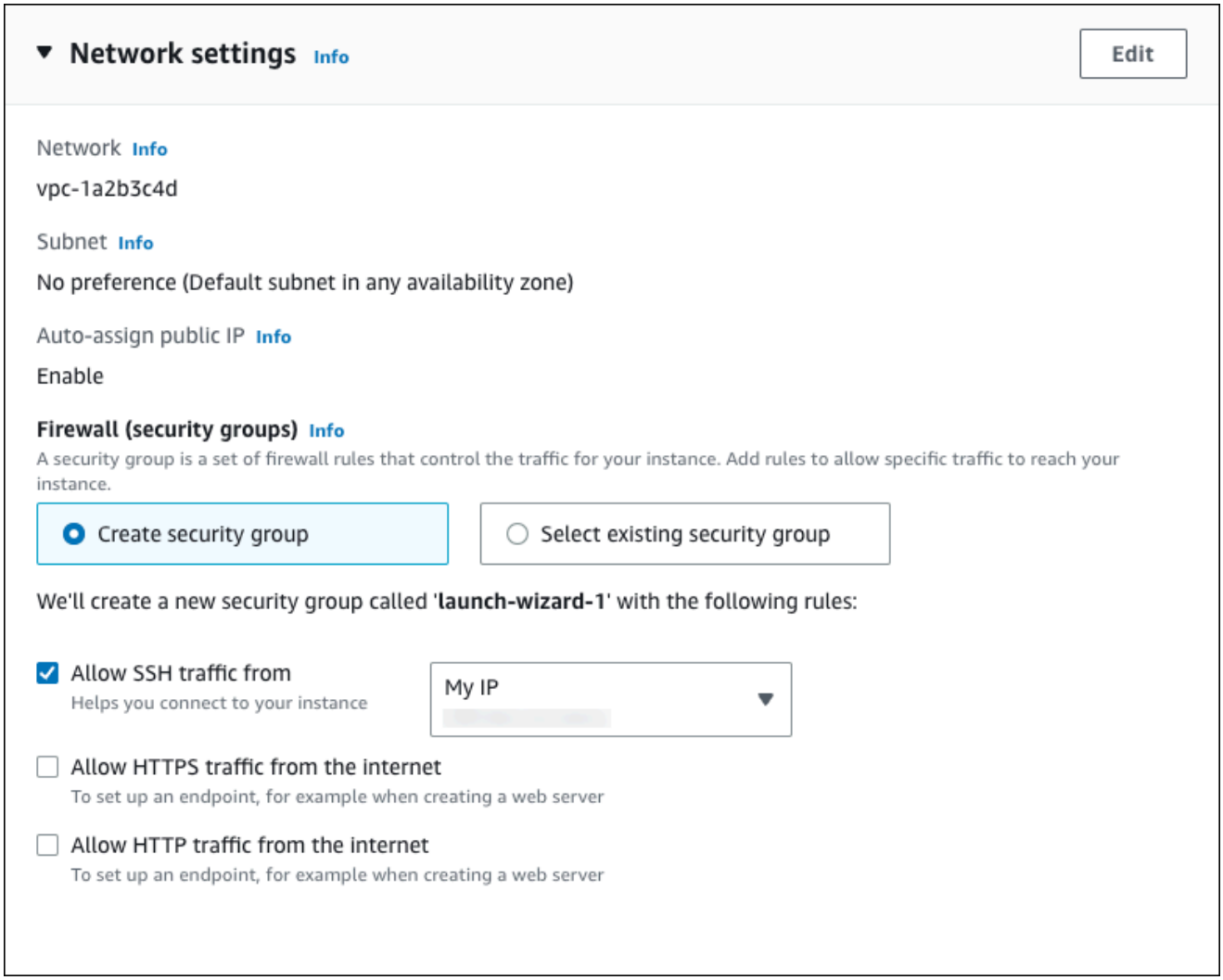
e. Wählen Sie unter „SSH-Verkehr zulassen“ in den Netzwerkeinstellungen die Quelle der SSH-Verbindungen zur Instance aus. EC2

Sie können My IP (Meine IP) auswählen, wenn die angezeigte IP-Adresse für SSH-Verbindungen korrekt ist. Andernfalls können Sie mithilfe von Secure Shell (SSH) die IP-Adresse bestimmen, die für die Verbindung zu EC2 Instances in Ihrer VPC verwendet werden soll. Um Ihre öffentliche IP-Adresse in einem anderen Browserfenster oder einer anderen Registerkarte zu ermitteln, können Sie den Dienst unter verwenden. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Bestimmen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instanzen über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

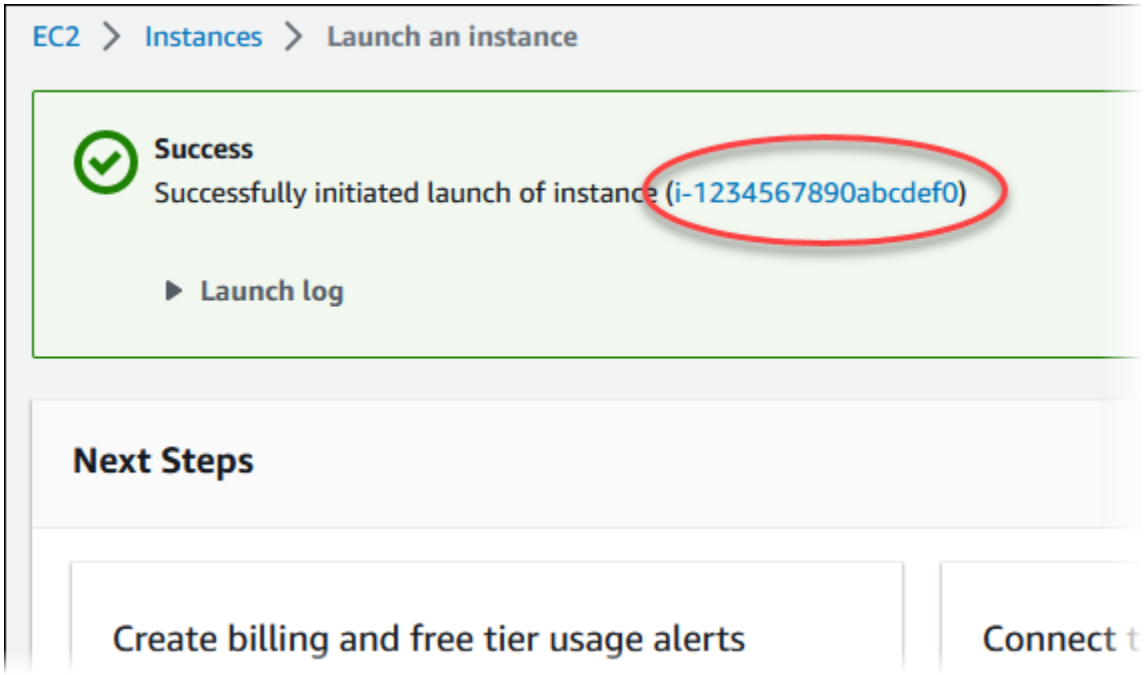
Die folgende Abbildung zeigt ein Beispiel für den Bereich Netzwerkeinstellungen.



f. Übernehmen Sie die Standardwerte für die übrigen Abschnitte.

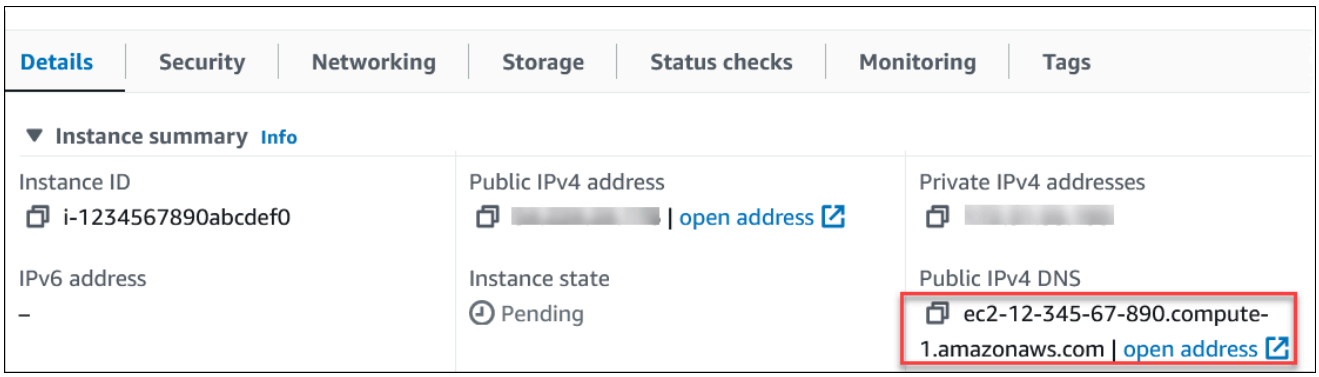
g. Sehen Sie sich im Bereich Zusammenfassung eine Zusammenfassung Ihrer EC2 Instance-Konfiguration an. Wenn Sie bereit sind, wählen Sie Launch instance aus.

5. Notieren Sie sich auf der Seite Startstatus die ID für Ihre neue EC2 Instance, zum Beispiel:`i-1234567890abcdef0`.

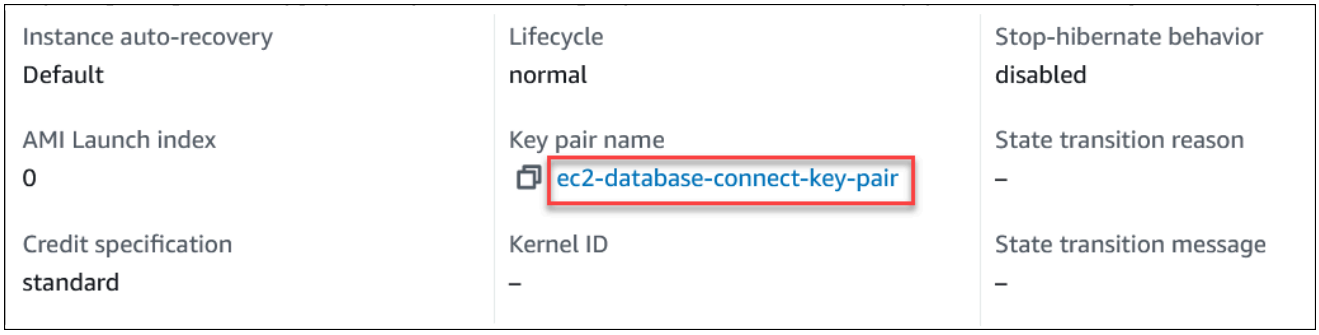


6. Wählen Sie die EC2 Instance-ID, um die Liste der EC2 Instances zu öffnen, und wählen Sie dann Ihre EC2 Instance aus.
7. Notieren Sie sich die folgenden Werte auf der Registerkarte Details. Diese benötigen Sie, wenn Sie eine Verbindung über SSH herstellen:

   a. Notieren Sie sich in der Instanzübersicht den Wert für Public IPv4 DNS.



   b. Notieren Sie sich unter Instance-Details den Wert für Schlüsselpaarname.



8. Warten Sie, bis der Instanzstatus Ihrer EC2 Instance den Status Running hat, bevor Sie fortfahren.

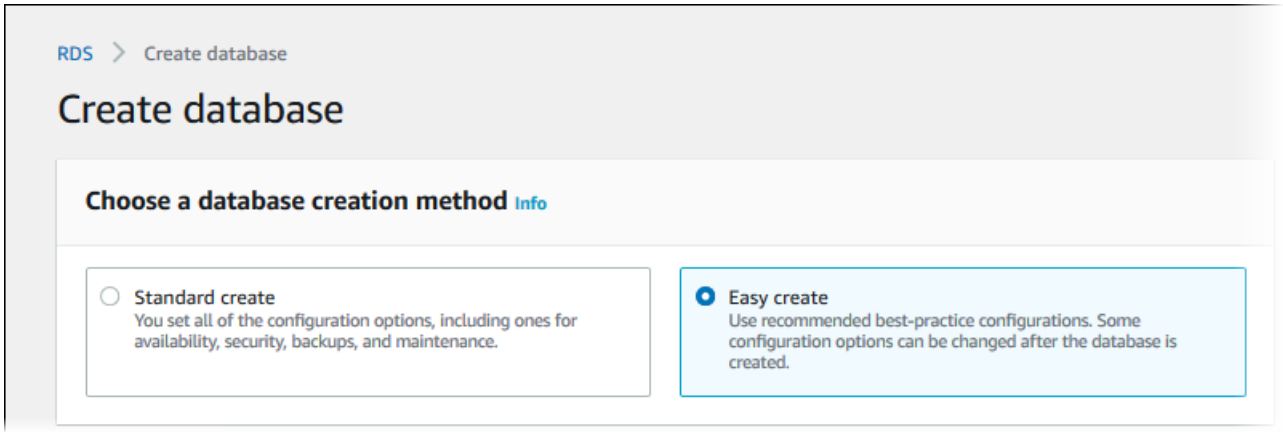
Erstellen Sie eine MySQL-DB-Instance

Die Grundbausteine für Amazon RDS sind Datenbank-Instances. In dieser Umgebung führen Sie Ihre MySQL-Datenbanken aus.

In diesem Beispiel verwenden Sie Einfache Erstellung, um eine DB-Instance zu erstellen, die die MySQL-Datenbank-Engine mit einer db.t3.micro DB-Instance-Klasse ausführt.

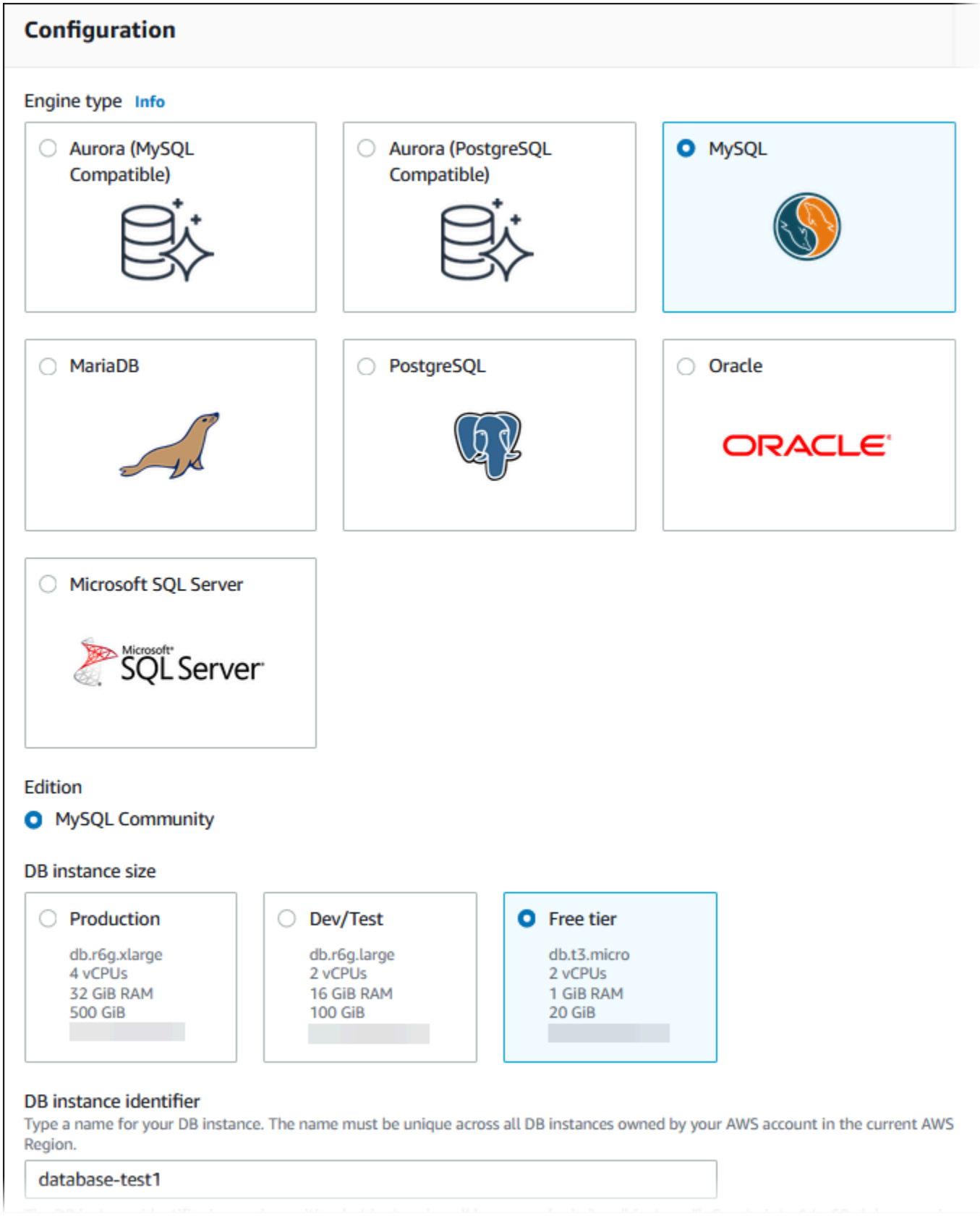
So erstellen Sie eine MySQL-DB-Instance mit aktiviertem Easy Create (Einfache Erstellung):

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die aus, die AWS-Region Sie zuvor für die EC2 Instance verwendet haben.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Datenbank erstellen aus und vergewissern Sie sich, dass Einfache Erstellung ausgewählt ist.



5. Wählen Sie unter Configuration (Konfiguration), die Option MySQL.
6. Wählen Sie in DB instance size (DB-Instance-Größe) die Option Free tier (Kostenloses Kontingent) aus. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.
7. Geben Sie als DB-Instance-ID **`database-test1`** ein.
8. Geben Sie unter Hauptbenutzername einen Namen für den Hauptbenutzer ein oder behalten Sie den Standardnamen bei.

   Die Seite Datenbank erstellen sollte ähnlich wie in der folgenden Abbildung gezeigt aussehen. Bei Konten mit kostenlosem Tarif wird das kostenlose Kontingent angezeigt. Bei Konten mit kostenpflichtigem Tarif wird Sandbox angezeigt.

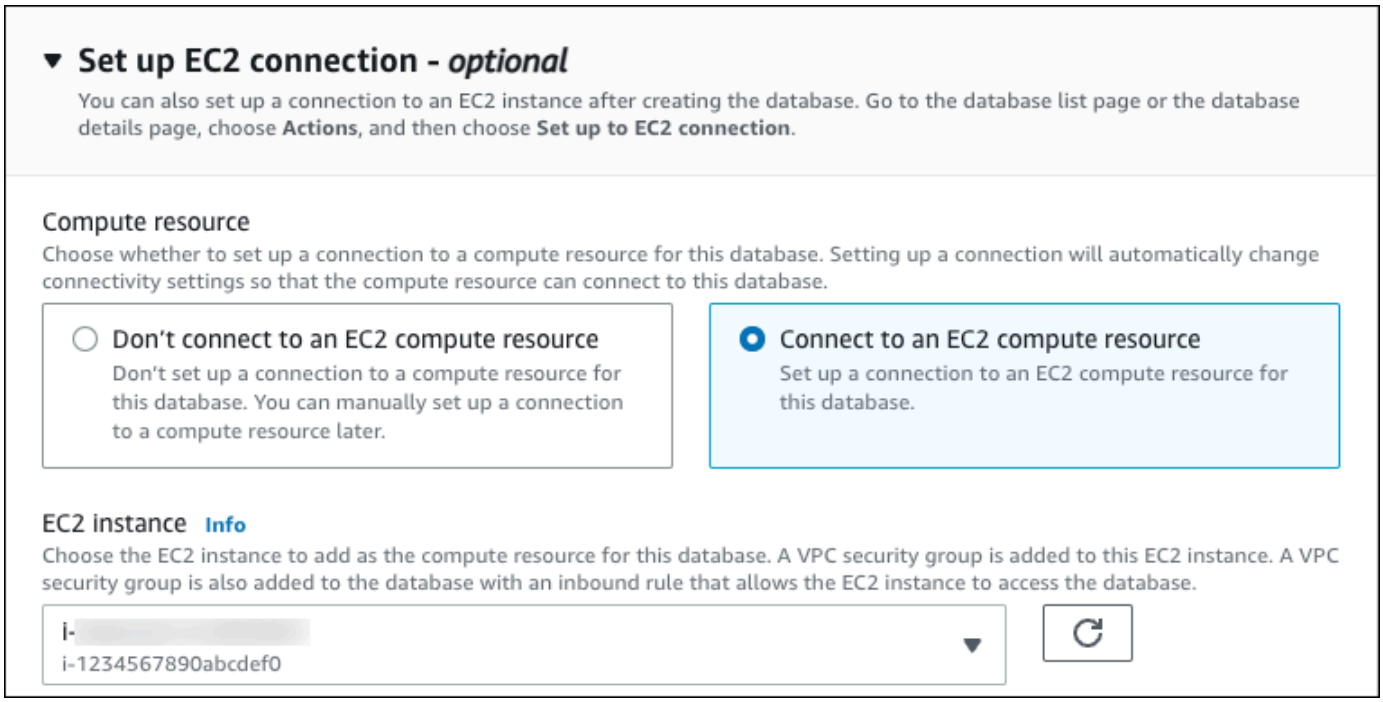

Configuration
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
Edition
MySQL Community
DB instance size
Production
db.r6g.xlarge
4 vCPUs
32 GiB RAM
500 GiB
Dev/Test
db.r6g.large
2 vCPUs
16 GiB RAM
100 GiB
Free tier
db.t3.micro
2 vCPUs
1 GiB RAM
20 GiB
DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-test1

9. Um für die DB-Instance ein automatisch generiertes Hauptpasswort zu verwenden, wählen Sie das Kästchen Passwort automatisch generieren aus.

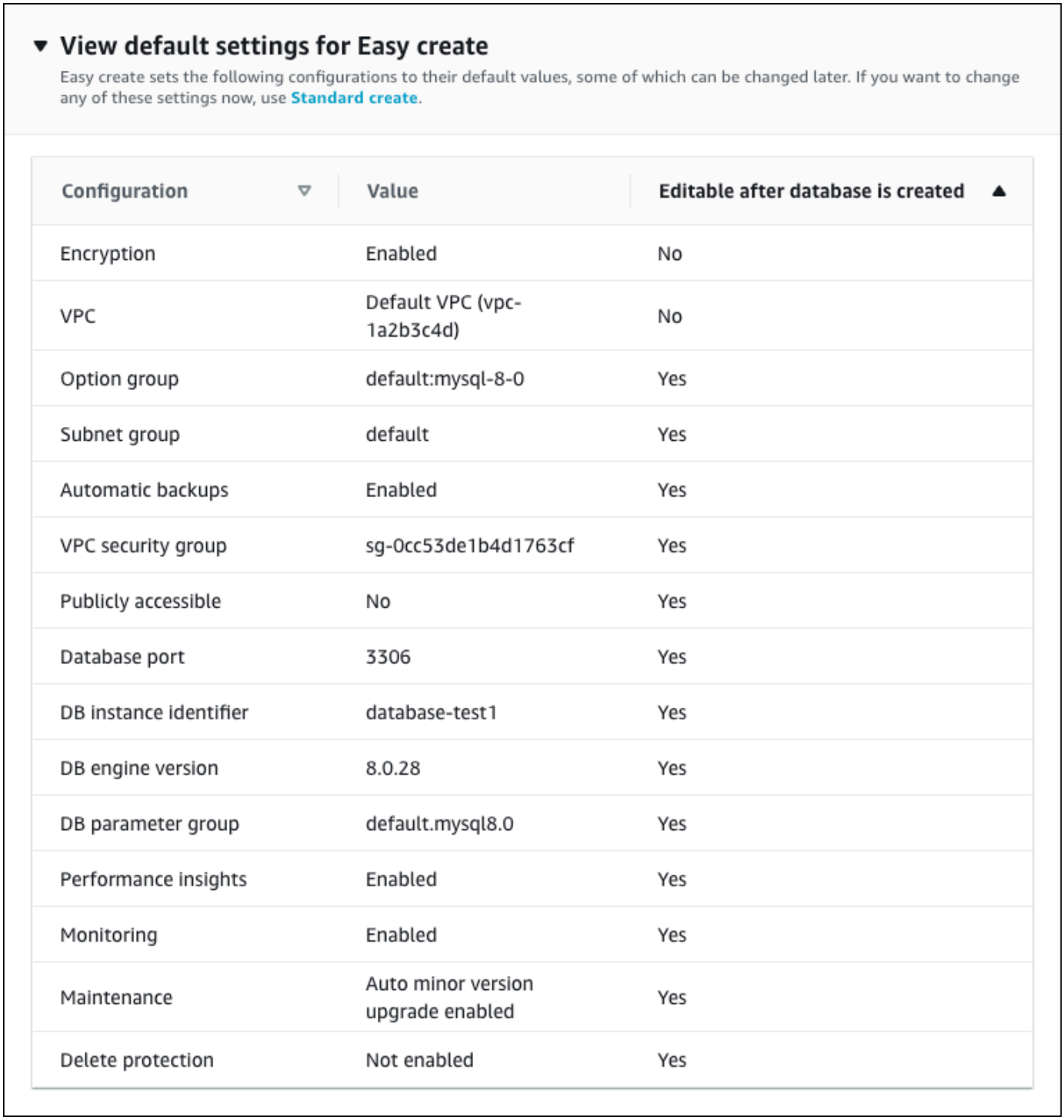
   Um das Hauptpasswort einzugeben, deaktivieren Sie das Kästchen Passwort automatisch generieren und geben Sie anschließend dasselbe Passwort in Hauptpasswort und Passwort bestätigen ein.

10. Um eine Verbindung mit der EC2 Instanz einzurichten, die Sie zuvor erstellt haben, öffnen Sie EC2 Verbindung einrichten — optional.

    Wählen Sie Mit einer EC2 Rechenressource verbinden aus. Wählen Sie die EC2 Instanz aus, die Sie zuvor erstellt haben.



11. (Optional) Öffnen Sie View default settings for Easy create (Standardeinstellungen für die einfache Erstellung anzeigen).



Sie können die Standardeinstellungen von Einfache Erstellung einsehen. Die Spalte Nach Erstellung der Datenbank editierbar zeigt, welche Optionen Sie nach der Datenbankerstellung ändern können.

- Wenn in einer Einstellung Nein in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie Standarderstellung verwenden, um die DB-Instance zu erstellen.
- Wenn für eine Einstellung Ja in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie entweder Standarderstellung verwenden, oder die DB-Instance nach der Erstellung ändern, um die Einstellung zu ändern.

12. Wählen Sie Datenbank erstellen aus.

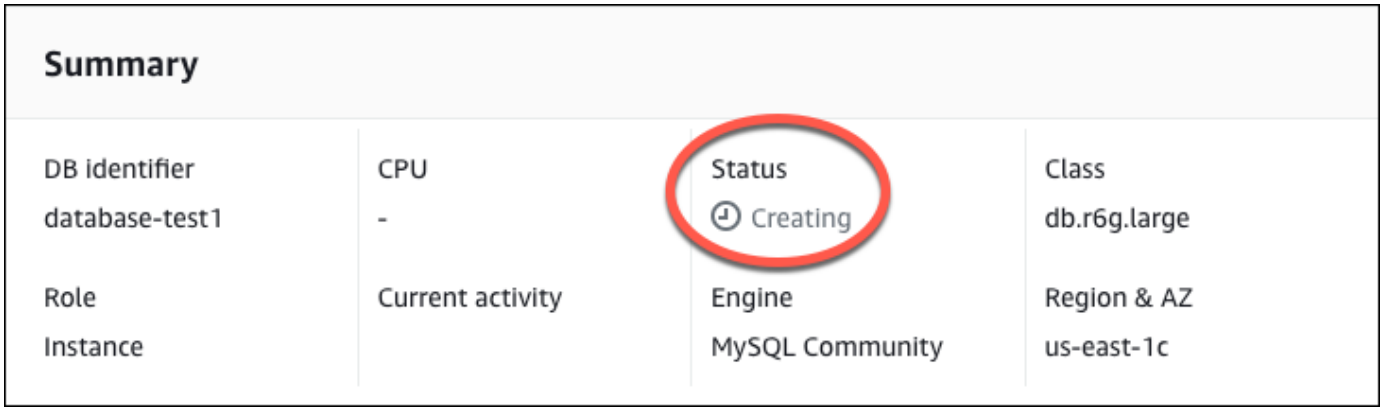
    Um den Masterbenutzernamen und das zugehörige Passwort für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen).

    Sie können den angezeigten Benutzernamen und das angezeigte Passwort verwenden, um als Masterbenutzer eine Verbindung zu DB-Instance herzustellen.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern.
    > Wenn Sie das Passwort für den Hauptbenutzer ändern müssen, nachdem die DB-Instance verfügbar wurde, können Sie die DB-Instance entsprechend ändern. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

13. Wählen Sie in der Liste Datenbanken den Namen der neuen MySQL-DB-Instance aus.

    Die DB-Instance hat den Status Wird erstellt, bis die DB-Instance bereit für die Verwendung ist.



Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und vom verfügbaren Speicherplatz kann es bis zu 20 Minuten dauern, bis die neue DB-Instance verfügbar ist.

(Optional) Erstellen Sie VPC, EC2 Instanz und MySQL-Instanz mit AWS CloudFormation

Anstatt die Konsole zum Erstellen Ihrer VPC, EC2 Instance und MySQL-Instance AWS CloudFormation zu verwenden, können Sie AWS Ressourcen bereitstellen, indem Sie Infrastruktur als Code behandeln. Um Ihnen zu helfen, Ihre AWS Ressourcen in kleinere und besser verwaltbare Einheiten zu organisieren, können Sie die AWS CloudFormation Nested-Stack-Funktionalität verwenden. Weitere Informationen finden Sie unter Einen Stack auf der AWS CloudFormation Konsole erstellen und Mit verschachtelten Stacks arbeiten.

> **Important**
>
> AWS CloudFormation ist kostenlos, aber die Ressourcen, die CloudFormation erstellt werden, sind live. Es fallen die üblichen Nutzungsgebühren für diese Ressourcen an, bis Sie sie kündigen. Weitere Informationen finden Sie unter Preise für RDS for MySQL.

Gehen Sie wie folgt vor, um Ihre Ressourcen mithilfe der AWS CloudFormation Konsole zu erstellen:

- Laden Sie die CloudFormation Vorlage herunter
- Konfigurieren Sie Ihre Ressourcen mit CloudFormation

Laden Sie die CloudFormation Vorlage herunter

Eine CloudFormation Vorlage ist eine JSON- oder YAML-Textdatei, die die Konfigurationsinformationen zu den Ressourcen enthält, die Sie im Stack erstellen möchten. Diese Vorlage erstellt zusammen mit der RDS-Instanz auch eine VPC und einen Bastion-Host für Sie.

Um die Vorlagendatei herunterzuladen, öffnen Sie den folgenden Link, CloudFormation MySQL-Vorlage.

Klicken Sie auf der Github-Seite auf die Schaltfläche Rohdatei herunterladen, um die YAML-Vorlagendatei zu speichern.

Konfigurieren Sie Ihre Ressourcen mit CloudFormation

> **Note**
>
> Bevor Sie diesen Vorgang starten, stellen Sie sicher, dass Sie ein Schlüsselpaar für eine EC2 Instance in Ihrem haben AWS-Konto. Weitere Informationen finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

Wenn Sie die AWS CloudFormation Vorlage verwenden, müssen Sie die richtigen Parameter auswählen, um sicherzustellen, dass Ihre Ressourcen ordnungsgemäß erstellt werden. Führen Sie die folgenden Schritte aus:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie Stapel erstellen aus.
3. Wählen Sie im Abschnitt Vorlage angeben die Option Eine Vorlagendatei von Ihrem Computer hochladen und dann Weiter aus.
4. Legen Sie auf der Seite „Stack-Details angeben“ die folgenden Parameter fest:
   a. Stellen Sie den Stacknamen auf Mein SQLTest Stack ein.
   b. Legen Sie unter Parameter Availability Zones fest, indem Sie drei Availability Zones auswählen.
   c. Wählen Sie unter Linux Bastion Host configuration für Key Name ein key pair aus, um sich bei Ihrer EC2 Instance anzumelden.
   d. Stellen Sie in den Linux Bastion Host-Konfigurationseinstellungen den Bereich Zulässiger IP auf Ihre IP-Adresse ein. Um mithilfe von Secure Shell (SSH) eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen, ermitteln Sie Ihre öffentliche IP-Adresse mithilfe des Dienstes unter. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instances über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

e. Stellen Sie unter Allgemeine Datenbankkonfiguration die Datenbankinstanzklasse auf db.t3.micro ein.

f. Setzen Sie den Datenbanknamen auf. **`database-test1`**

g. Geben Sie unter Datenbank-Master-Benutzername einen Namen für den Masterbenutzer ein.

h. Stellen Sie `false` für dieses Tutorial das DB-Master-Benutzerpasswort mit Secrets Manager verwalten auf ein.

i. Geben Sie für das Datenbankkennwort ein Passwort Ihrer Wahl ein. Merken Sie sich dieses Passwort für weitere Schritte im Tutorial.

j. Stellen Sie unter Datenbankspeicherkonfiguration den Datenbankspeichertyp auf gp2 ein.

k. Stellen Sie unter Konfiguration der Datenbanküberwachung die Option Enable RDS Performance Insights auf false ein.

l. Behalten Sie für alle anderen Einstellungen die Standardwerte bei. Klicken Sie auf Weiter, um fortzufahren.

5. Behalten Sie auf der Seite „Stack-Optionen konfigurieren“ alle Standardoptionen bei. Klicken Sie auf Weiter, um fortzufahren.
6. Wählen Sie auf der Seite „Stack überprüfen“ die Option Senden aus, nachdem Sie die Datenbank- und Linux-Bastion-Host-Optionen überprüft haben.

Sehen Sie sich nach Abschluss der Stack-Erstellung die Stacks mit Namen BastionStackund RDSNS an, um die Informationen zu notieren, die Sie für die Verbindung mit der Datenbank benötigen. Weitere Informationen finden Sie unter AWS CloudFormation Stack-Daten und Ressourcen anzeigen auf der. AWS Management Console

Verbindung zu einer MySQL-DB-Instance

Sie können für die Verbindung zur DB-Instance eine beliebige Standard-SQL-Client-Anwendung verwenden. In diesem Beispiel stellen Sie eine Verbindung mit einer MySQL-DB-Instance mithilfe des mysql-Befehlszeilen-Tools her.

Verbindung zu einer MySQL-DB-Instance

1. Suchen Sie nach dem Endpunkt (DNS-Name) und der Portnummer für Ihre DB-Instance.

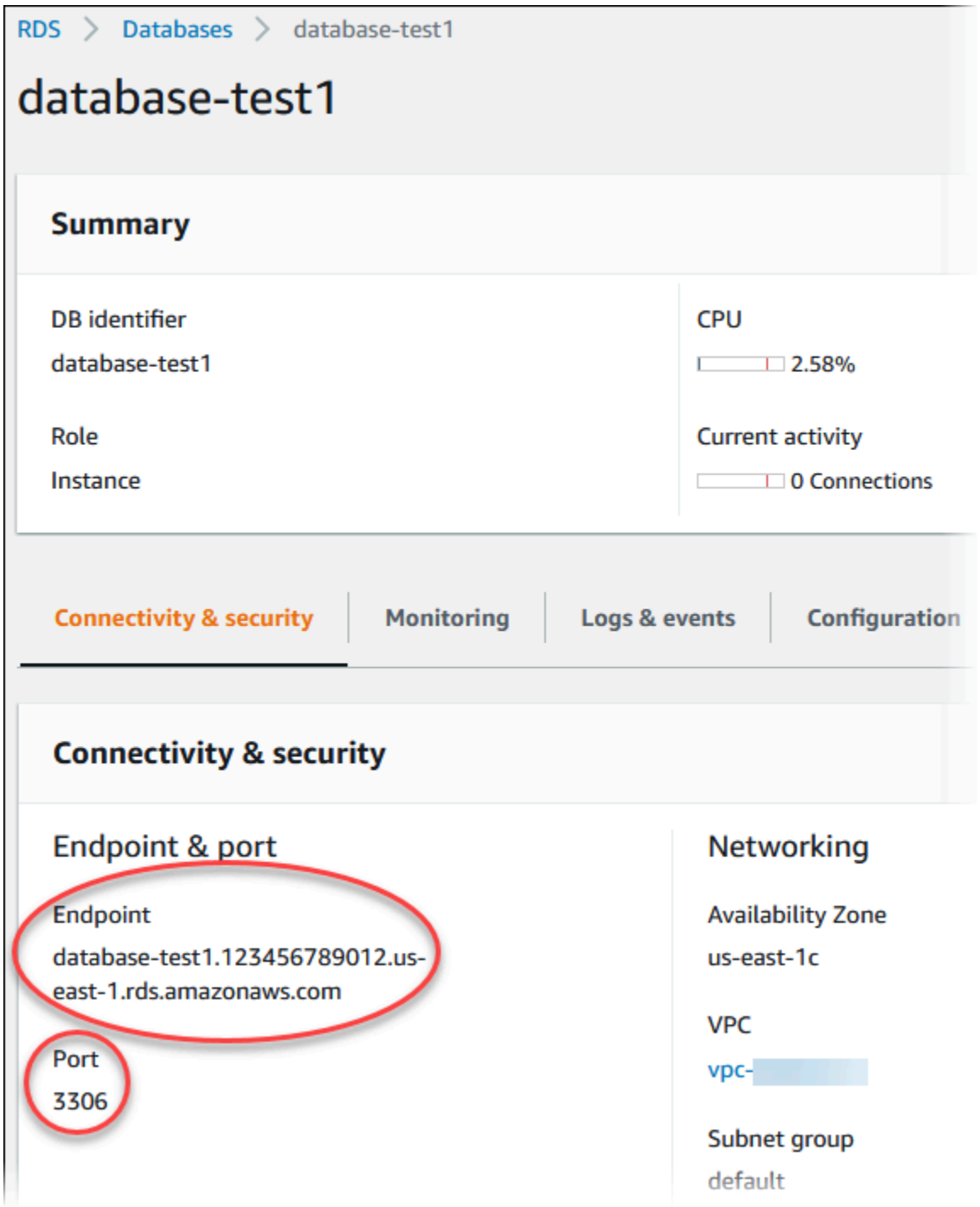
   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

b. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die AWS-Region für die DB-Instance aus.

c. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.

d. Wählen Sie den Namen der MySQL DB-Instance, um deren Details anzuzeigen.

e. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



2. Stellen Sie eine Verbindung zu der EC2 Instance her, die Sie zuvor erstellt haben, indem Sie den Schritten unter Connect to your Linux Instance im EC2 Amazon-Benutzerhandbuch folgen.

   Wir empfehlen, dass Sie über SSH eine Verbindung zu Ihrer EC2 Instance herstellen. Wenn das SSH-Client-Dienstprogramm unter Windows, Linux oder Mac installiert ist, können Sie mit dem folgenden Befehlsformat eine Verbindung mit der Instance herstellen:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Gehen Sie beispielsweise davon aus, dass dies `/dir1` unter Linux gespeichert `ec2-database-connect-key-pair.pem` ist und das öffentliche IPv4 DNS für Ihre EC2 Instance bereits. `ec2-12-345-678-90.compute-1.amazonaws.com` Ihr SSH-Befehl würde wie folgt aussehen:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Holen Sie sich die neuesten Bugfixes und Sicherheitsupdates, indem Sie die Software auf Ihrer EC2 Instance aktualisieren. Verwenden Sie dazu den folgenden Befehl.

   > **Note**
   >
   > Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Um Updates vor der Installation zu überprüfen, lassen Sie diese Option aus.

   ```
   sudo dnf update -y
   ```

4. Führen Sie den folgenden Befehl aus, um den mysql-Befehlszeilen-Client von MariaDB auf Amazon Linux 2023 zu installieren:

   ```
   sudo dnf install mariadb105
   ```

5. Stellen Sie eine Verbindung mit der MySQL-DB-Instance her. Geben Sie z. B. den folgenden Befehl ein. Mit dieser Aktion können Sie eine Verbindung mit der MySQL-DB-Instance mithilfe des MySQL-Clients herstellen.

   Ersetzen Sie den DB-Instance-Endpunkt (DNS-Name) für *`endpoint`* und den Hauptbenutzernamen, den Sie für *`admin`* verwendet haben. Geben Sie das Master-Passwort ein, das Sie bei der Aufforderung zur Eingabe eines Passworts verwendet haben.

   ```
   mysql -h endpoint -P 3306 -u admin -p
   ```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, sollte eine Ausgabe wie die folgende angezeigt werden.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 3082
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

Weitere Informationen zum Herstellen einer Verbindung zur MySQL-DB-Instance finden Sie unter Verbindung zu Ihrer MySQL-DB-Instance herstellen. Wenn Sie sich nicht mit Ihrer DB-Instance verbinden können, erhalten Sie unter Hilf Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Aus Sicherheitsgründen empfiehlt es sich, verschlüsselte Verbindungen zu verwenden. Verwenden Sie eine unverschlüsselte MySQL Verbindung nur, wenn sich Client und Server in derselben VPC befinden und das Netzwerk vertrauenswürdig ist. Weitere IInformationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Herstellen einer Verbindung zu Ihrer My SQL DB-Instance auf Amazon RDS SSL TLS mit/über den SQL My-Befehlszeilenclient (verschlüsselt).

6. SQL-Befehle ausführen

   Der folgende SQL-Befehl zeigt z B. das aktuelle Datum und die aktuelle Zeit an:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Löschen Sie die EC2 Instance und die DB-Instance

Nachdem Sie sich mit der EC2 Beispiel-Instance und DB-Instance, die Sie erstellt haben, verbunden und diese erkundet haben, löschen Sie sie, damit Ihnen keine Gebühren mehr berechnet werden.

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, überspringen Sie diesen Schritt und fahren Sie mit dem nächsten Schritt fort.

Um die EC2 Instanz zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die EC2 Instance aus und wählen Sie dann Instance-Status, Instanz beenden aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

Weitere Informationen zum Löschen einer EC2 Instance finden Sie unter Kündigen Ihrer Instance im EC2 Amazon-Benutzerhandbuch.

So löschen Sie die DB-Instance ohne finalen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Löschen Sie Abschließenden Snapshot erstellen? und Automatische Backups aufbewahren.
6. Bestätigen Sie, und wählen Sie Löschen.

(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, löschen Sie den CloudFormation Stack, nachdem Sie sich mit der EC2 Beispiel-Instance und der DB-Instance verbunden und diese erkundet haben, sodass Ihnen diese nicht mehr in Rechnung gestellt werden.

Um die CloudFormation Ressourcen zu löschen

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite Stacks in den den CloudFormationconsole Root-Stack aus (den Stack ohne Namen VPCStack BastionStack oder RDSNS).
3. Wählen Sie Löschen aus.
4. Wählen Sie Stack löschen, wenn Sie zur Bestätigung aufgefordert werden.

Weitere Informationen zum Löschen eines Stacks in CloudFormation finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks auf der AWS CloudFormation Konsole.

(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Sie können Ihre DB-Instance von RDS für MySQL auch mit einer Lambda-Serverless-Rechenressource verbinden. Mit Lambda-Funktionen können Sie Code ausführen, ohne die Infrastruktur bereitstellen oder verwalten zu müssen. Eine Lambda-Funktion ermöglicht es Ihnen auch, automatisch auf Codeausführungsanfragen jeder Größenordnung zu reagieren, von einem Dutzend Ereignissen pro Tag bis hin zu Hunderten von Ereignissen pro Sekunde. Weitere Informationen finden Sie unter Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance.

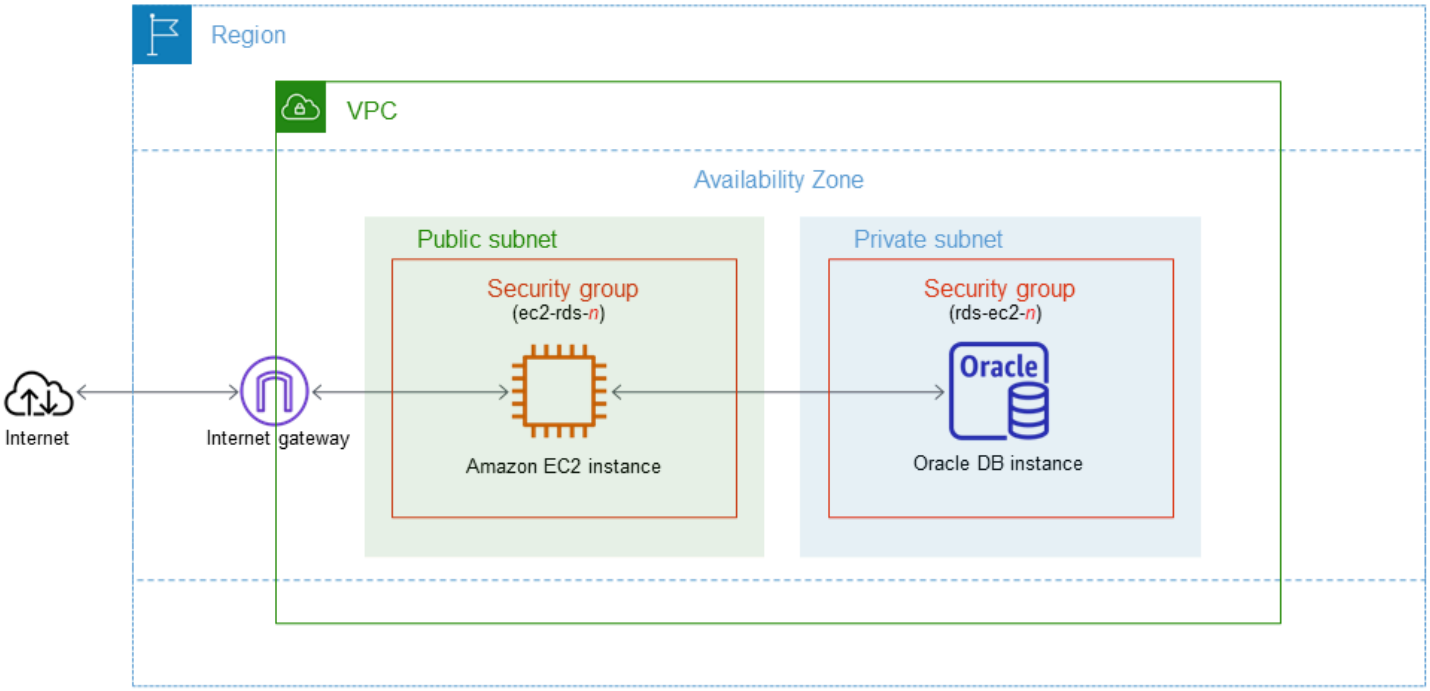
Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung

In diesem Tutorial werden eine EC2 Instance und eine RDS für Oracle-DB-Instance erstellt. Das Tutorial zeigt Ihnen, wie Sie mit einem Standard-Oracle-Client von der EC2 Instance aus auf die DB-Instance zugreifen. Als bewährte Methode erstellt dieses Tutorial eine private DB-Instance in einer Virtual Private Cloud (VPC). In den meisten Fällen können andere Ressourcen in derselben VPC, z. B. EC2 Instances, auf die DB-Instance zugreifen, aber Ressourcen außerhalb der VPC können nicht darauf zugreifen.

Nach Abschluss des Tutorials gibt es in jeder Availability Zone im VPC ein öffentliches und ein privates Subnetz. In einer Availability Zone befindet sich die EC2 Instance im öffentlichen Subnetz und die DB-Instance im privaten Subnetz.

> **Important**
>
> Für die Erstellung eines Kontos fallen keine Gebühren an AWS . Wenn Sie dieses Tutorial abschließen, können Ihnen jedoch Kosten für die von Ihnen verwendeten AWS Ressourcen entstehen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

Das folgende Diagramm zeigt die Konfiguration nach Abschluss des Tutorials.



In diesem Tutorial können Sie Ihre Ressourcen mithilfe einer der folgenden Methoden erstellen:

1. Verwenden Sie das AWS Management Console - Schritt 2: Erstellen einer Oracle-DB-Instance und Schritt 1: Erstellen Sie eine EC2 Instanz
2. Verwenden Sie AWS CloudFormation , um die Datenbankinstanz und die EC2 Instanz zu erstellen - (Optional) Erstellen Sie VPC, EC2 Instance und Oracle DB-Instance mit AWS CloudFormation

Die erste Methode verwendet Easy Create, um eine private Oracle-DB-Instance mit dem zu erstellen AWS Management Console. Hier geben Sie nur den DB-Engine-Typ, die DB-Instance-Größe und die DB-Instance-ID an. Easy Create (Einfache Erstellung) verwendet für die anderen Konfigurationsoptionen die Standardeinstellung.

Wenn Sie stattdessen Standard create verwenden, können Sie beim Erstellen einer DB-Instance weitere Konfigurationsoptionen angeben. Zu diesen Optionen gehören Einstellungen für Verfügbarkeit, Sicherheit, Backups und Wartung. Um eine öffentliche DB-Instance zu erstellen, müssen Sie Standarderstellung verwenden. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Themen

- Voraussetzungen
- Schritt 1: Erstellen Sie eine EC2 Instanz

- Schritt 2: Erstellen einer Oracle-DB-Instance
- (Optional) Erstellen Sie VPC, EC2 Instance und Oracle DB-Instance mit AWS CloudFormation
- Schritt 3: Verbinden Ihres SQL-Clients mit einer Oracle-DB-Instance
- Schritt 4: Löschen Sie die Instance und die DB-Instance EC2
- (Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation
- (Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

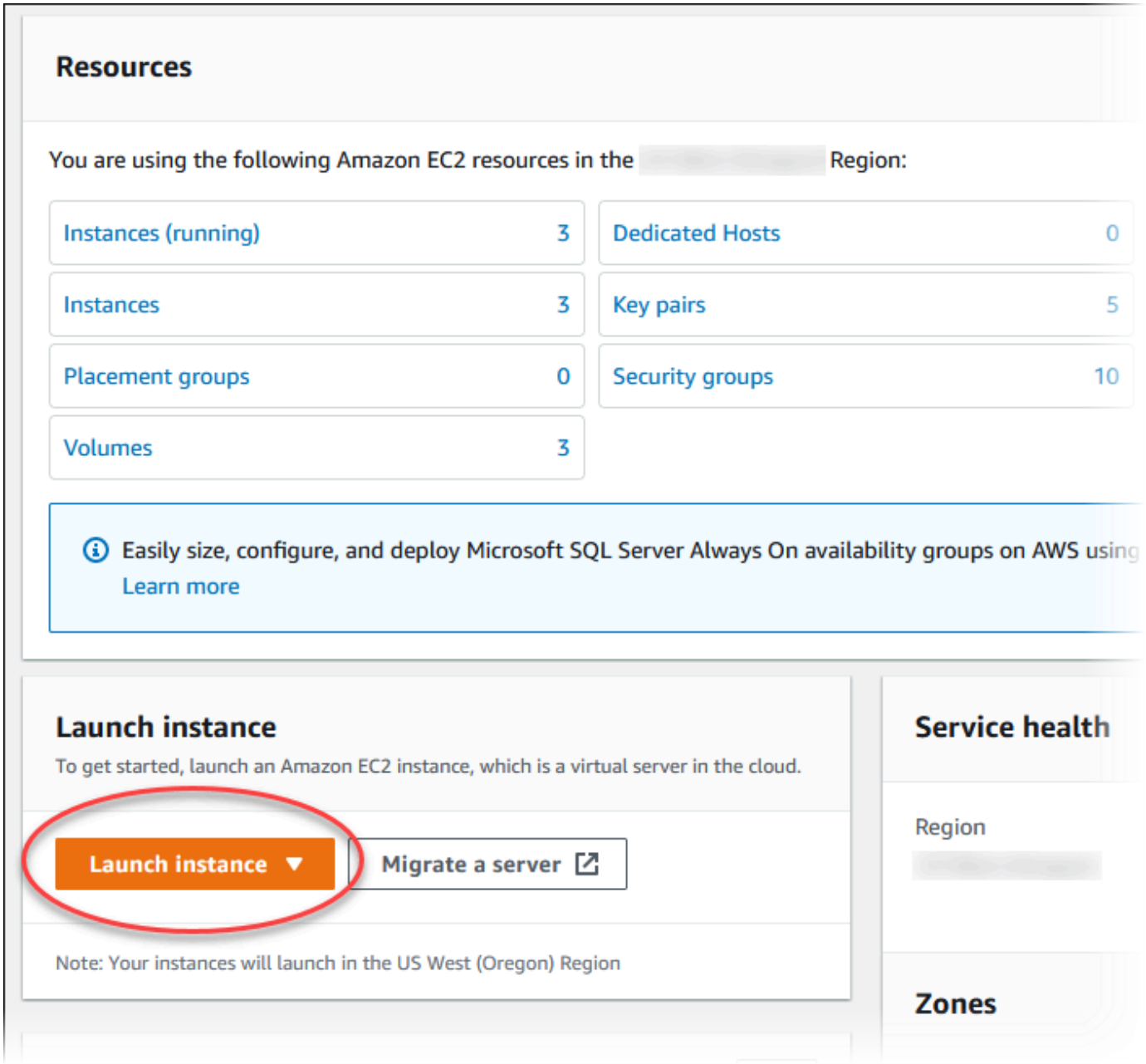
- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Schritt 1: Erstellen Sie eine EC2 Instanz

Erstellen Sie eine EC2 Amazon-Instance, mit der Sie eine Verbindung zu Ihrer Datenbank herstellen.

Um eine EC2 Instanz zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen rechten Ecke von die aus AWS Management Console, AWS-Region in der Sie die Instance erstellen möchten. EC2
3. Wählen Sie EC2 Dashboard und dann Launch instance aus, wie in der folgenden Abbildung gezeigt.

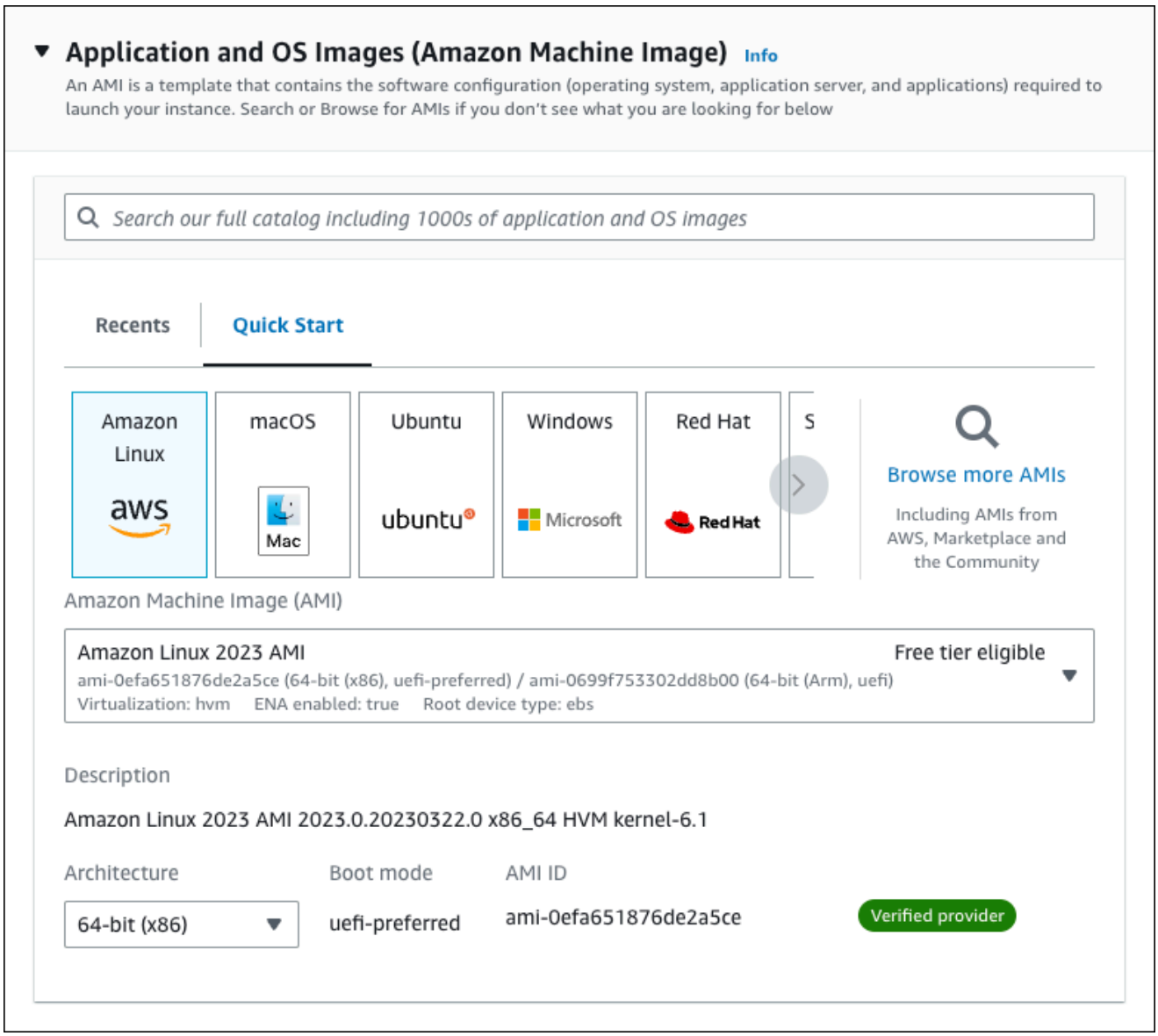


Die Seite Eine Instance starten wird geöffnet.

4. Wählen Sie auf der Seite Eine Instance starten die folgenden Einstellungen aus.

   a. Geben Sie unter Name and tags (Name und Tags) als Name den Namen **`ec2-database-connect`** ein.

   b. Wählen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) die Option Amazon Linux und dann die Registerkarte Amazon Linux 2023 AMI aus. Übernehmen Sie für alle anderen Einstellungen die Standardwerte.



c. Wählen Sie unter Instance type (Instance-Typ) den Wert t2.micro aus.

d. Wählen Sie unter Key pair (login) (Schlüsselpaar (Anmeldung)) einen Key pair name (Schlüsselpaarname), um ein vorhandenes Schlüsselpaar zu verwenden. Um ein neues key pair für die EC2 Amazon-Instance zu erstellen, wählen Sie Neues key pair erstellen und verwenden Sie dann das Fenster key pair erstellen, um es zu erstellen.

   Weitere Informationen zum Erstellen eines neuen key pair finden Sie unter Erstellen eines key pair im EC2 Amazon-Benutzerhandbuch.

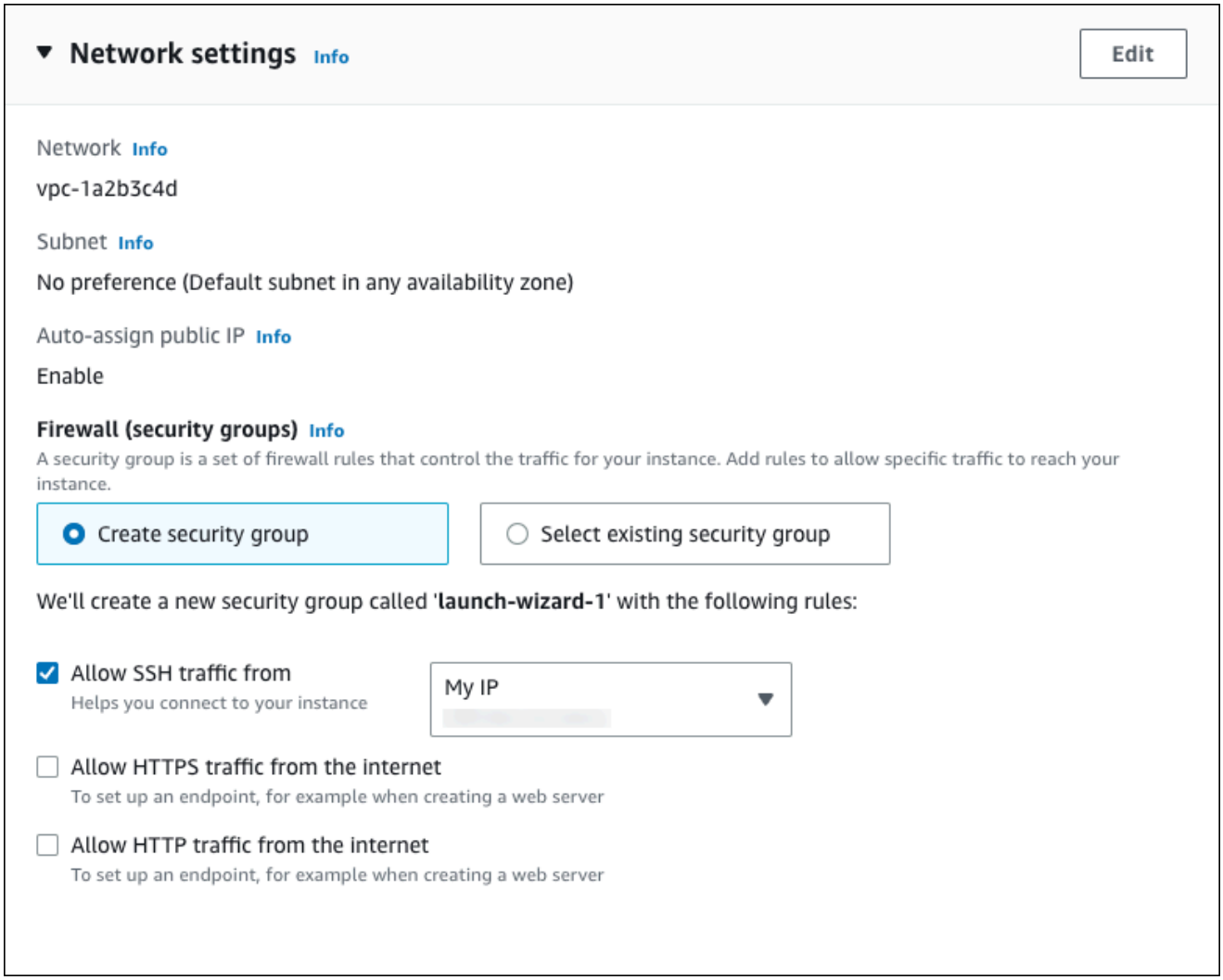
e. Wählen Sie unter „SSH-Verkehr zulassen“ in den Netzwerkeinstellungen die Quelle der SSH-Verbindungen zur Instance aus. EC2

Sie können My IP (Meine IP) auswählen, wenn die angezeigte IP-Adresse für SSH-Verbindungen korrekt ist. Andernfalls können Sie mithilfe von Secure Shell (SSH) die IP-Adresse bestimmen, die für die Verbindung zu EC2 Instances in Ihrer VPC verwendet werden soll. Um Ihre öffentliche IP-Adresse in einem anderen Browserfenster oder einer anderen Registerkarte zu ermitteln, können Sie den Dienst unter verwenden. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Bestimmen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instanzen über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktion nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

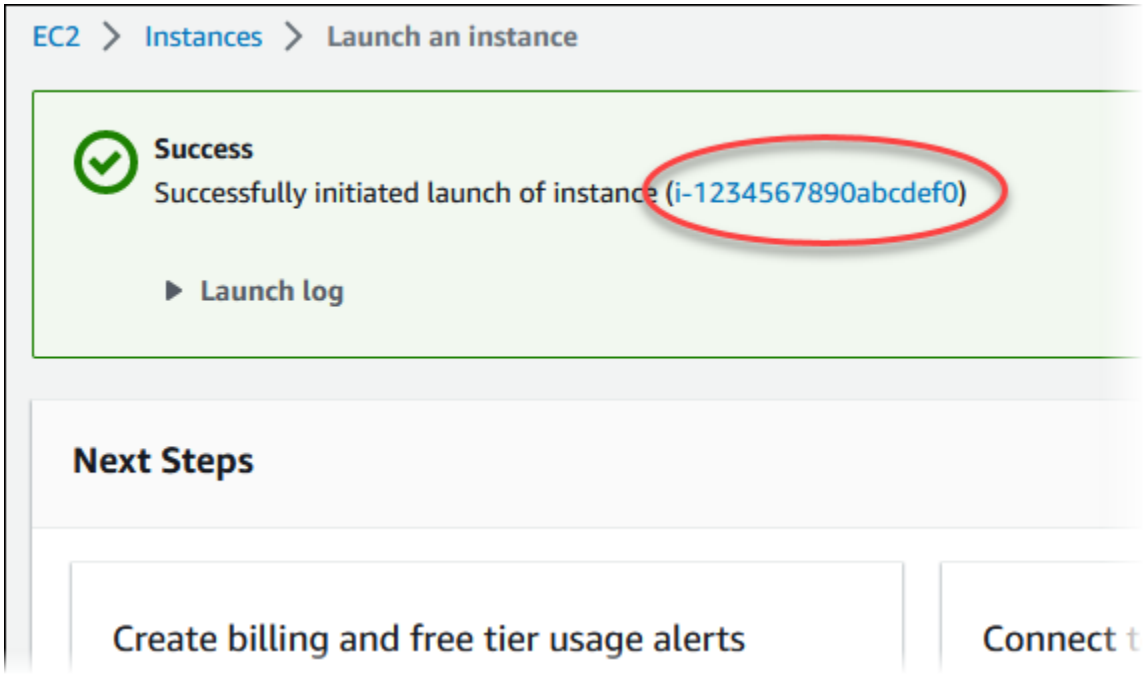
Die folgende Abbildung zeigt ein Beispiel für den Bereich Netzwerkeinstellungen.



f. Übernehmen Sie die Standardwerte für die übrigen Abschnitte.

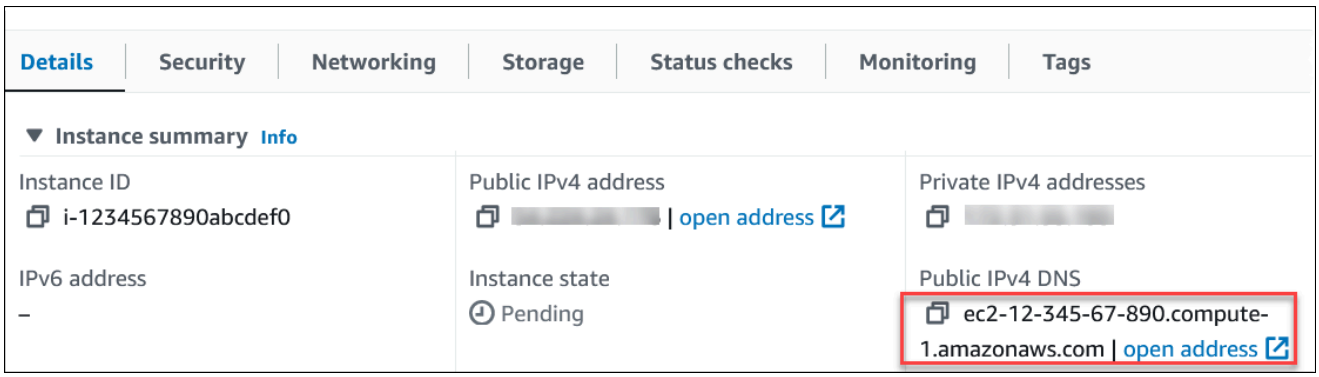
g. Sehen Sie sich im Bereich Zusammenfassung eine Zusammenfassung Ihrer EC2 Instance-Konfiguration an. Wenn Sie bereit sind, wählen Sie Launch instance aus.

5. Notieren Sie sich auf der Seite Startstatus die ID für Ihre neue EC2 Instance, zum Beispiel:`i-1234567890abcdef0`.

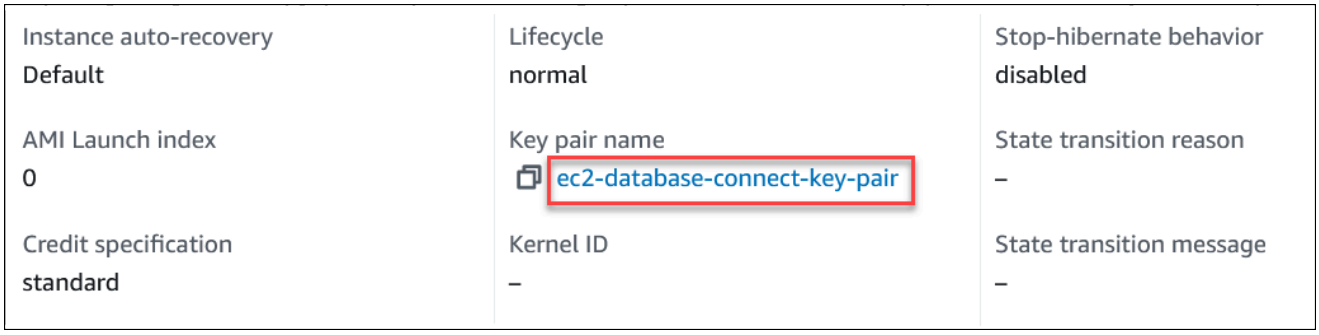


6. Wählen Sie die EC2 Instance-ID, um die Liste der EC2 Instances zu öffnen, und wählen Sie dann Ihre EC2 Instance aus.
7. Notieren Sie sich die folgenden Werte auf der Registerkarte Details. Diese benötigen Sie, wenn Sie eine Verbindung über SSH herstellen:

   a. Notieren Sie sich in der Instanzübersicht den Wert für Public IPv4 DNS.



   b. Notieren Sie sich unter Instance-Details den Wert für Schlüsselpaarname.



8. Warten Sie, bis der Instanzstatus Ihrer EC2 Instance den Status Running hat, bevor Sie fortfahren.

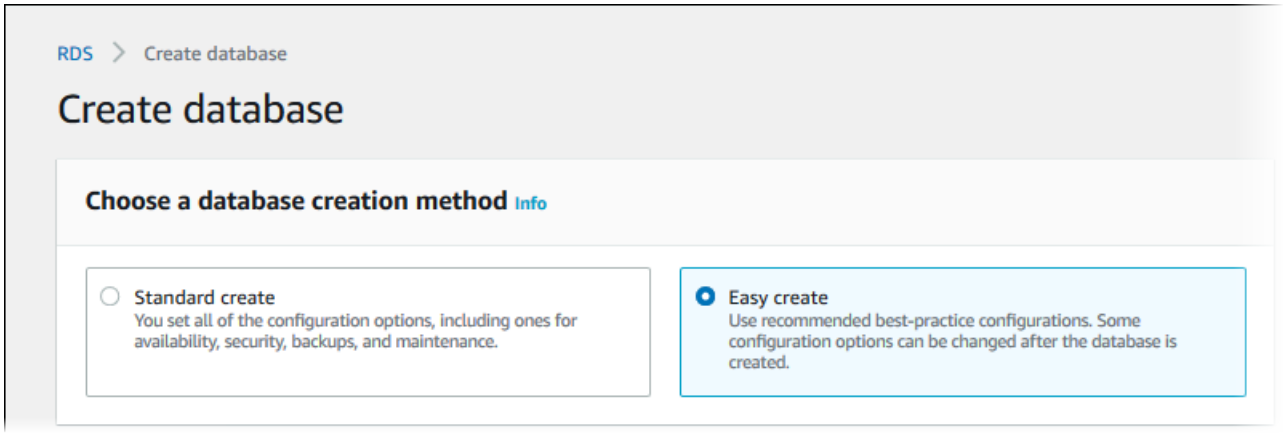
Schritt 2: Erstellen einer Oracle-DB-Instance

Die Grundbausteine für Amazon RDS sind Datenbank-Instances. In dieser Umgebung führen Sie Ihre Oracle-Datenbanken aus.

In diesem Beispiel verwenden Sie Einfache Erstellung, um eine DB-Instance zu erstellen, die die Oracle-Datenbank-Engine mit einer DB-Instance-Klasse des Typs „db.m5.large“ ausführt.

So erstellen Sie eine Oracle-DB-Instance mit der Option „Einfache Erstellung“

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die aus, AWS-Region in der Sie die DB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Datenbank erstellen aus und vergewissern Sie sich, dass Einfache Erstellung ausgewählt ist.



5. Wählen Sie unter Configuration (Konfiguration), die Option Oracle.
6. Wählen Sie in DB instance size (DB-Instance-Größe) die Option Dev/Test (Entwicklung/Testen) aus.
7. Geben Sie als DB-Instance-ID **`database-test1`** ein.
8. Geben Sie unter Hauptbenutzername einen Namen für den Hauptbenutzer ein oder behalten Sie den Standardnamen bei.

   Die Seite Datenbank erstellen sollte ähnlich wie in der folgenden Abbildung gezeigt aussehen.

Configuration

Engine type Info

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

Edition

- Oracle Enterprise Edition
  Affordable and full-featured database management system supporting up to 16 vCPUs.
- Oracle Standard Edition Two
  Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

DB instance size

- Production
  db.r5.large
  2 vCPUs
  16 GiB RAM
  500 GiB
- Dev/Test
  db.m5.large
  2 vCPUs
  8 GiB RAM
  100 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-test1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info

Type a login ID for the master user of your DB instance.

admin

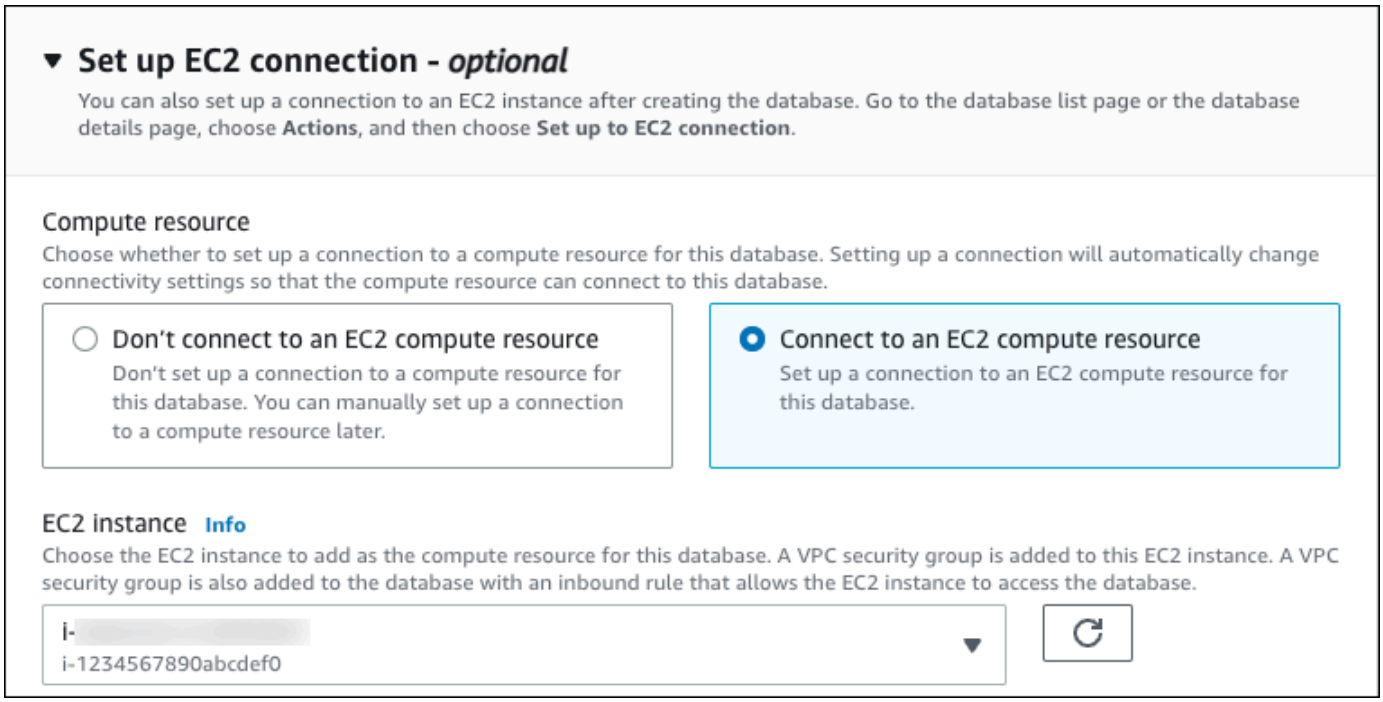
1 to 16 alphanumeric characters. First character must be a letter.

9. Um für die DB-Instance ein automatisch generiertes Hauptpasswort zu verwenden, wählen Sie das Kästchen Passwort automatisch generieren aus.

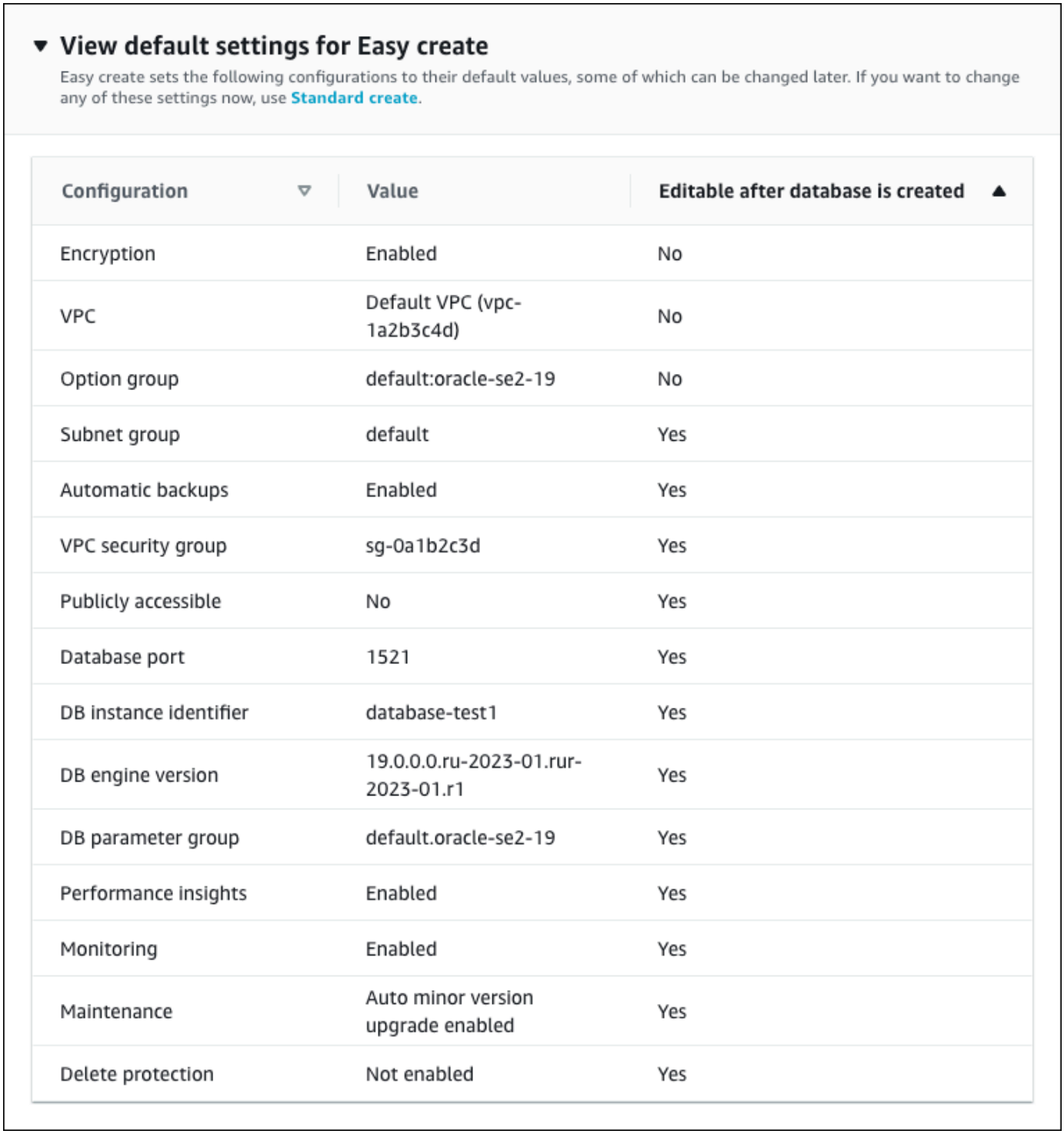
   Um das Hauptpasswort einzugeben, deaktivieren Sie das Kästchen Passwort automatisch generieren und geben Sie anschließend dasselbe Passwort in Hauptpasswort und Passwort bestätigen ein.

10. Um eine Verbindung mit der EC2 Instance einzurichten, die Sie zuvor erstellt haben, öffnen Sie EC2 Verbindung einrichten — optional.

    Wählen Sie Mit einer EC2 Rechenressource verbinden aus. Wählen Sie die EC2 Instanz aus, die Sie zuvor erstellt haben.



11. (Optional) Öffnen Sie Anzeigen von Standardeinstellungen für eine einfache Erstellung.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:oracle-se2-19	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0a1b2c3d	Yes
Publicly accessible	No	Yes
Database port	1521	Yes
DB instance identifier	database-test1	Yes
DB engine version	19.0.0.0.ru-2023-01.rur-2023-01.r1	Yes
DB parameter group	default.oracle-se2-19	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Sie können die Standardeinstellungen von Einfache Erstellung einsehen. Die Spalte Nach Erstellung der Datenbank editierbar zeigt, welche Optionen Sie nach der Datenbankerstellung ändern können.

- Wenn in einer Einstellung Nein in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie Standarderstellung verwenden, um die DB-Instance zu erstellen.
- Wenn für eine Einstellung Ja in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie entweder Standarderstellung verwenden, oder die DB-Instance nach der Erstellung ändern, um die Einstellung zu ändern.

12. Wählen Sie Datenbank erstellen aus.

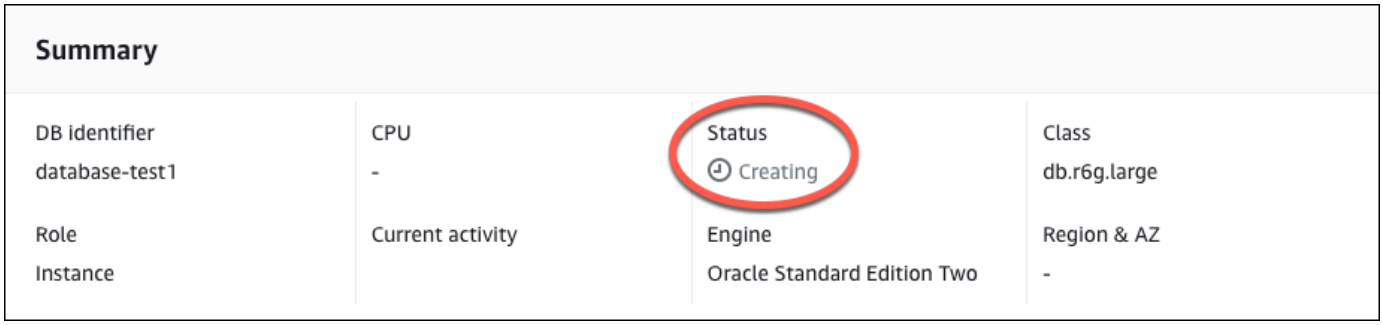
    Um den Masterbenutzernamen und das zugehörige Passwort für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen).

    Sie können den angezeigten Benutzernamen und das angezeigte Passwort verwenden, um als Masterbenutzer eine Verbindung zu DB-Instance herzustellen.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern.
    > Wenn Sie das Passwort für den Hauptbenutzer ändern müssen, nachdem die DB-Instance verfügbar wurde, können Sie die DB-Instance entsprechend ändern. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

13. Wählen Sie in der Liste Datenbanken den Namen der neuen Oracle-DB-Instance aus, um deren Details anzuzeigen.

    Die DB-Instance hat den Status Wird erstellt, bis die DB-Instance bereit für die Verwendung ist.



    Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und vom verfügbaren Speicherplatz kann es bis zu 20 Minuten dauern, bis die neue DB-Instance verfügbar ist.

Während die DB-Instance erstellt wird, können Sie mit dem nächsten Schritt fortfahren und eine EC2 Instance erstellen.

(Optional) Erstellen Sie VPC, EC2 Instance und Oracle DB-Instance mit AWS CloudFormation

Anstatt die Konsole zum Erstellen Ihrer VPC, EC2 Instance und Oracle-DB-Instance AWS CloudFormation zu verwenden, können Sie AWS Ressourcen bereitstellen, indem Sie Infrastruktur als Code behandeln. Um Ihnen zu helfen, Ihre AWS Ressourcen in kleinere und besser verwaltbare Einheiten zu organisieren, können Sie die AWS CloudFormation Nested-Stack-Funktionalität verwenden. Weitere Informationen finden Sie unter Einen Stack auf der AWS CloudFormation Konsole erstellen und Mit verschachtelten Stacks arbeiten.

> **Important**
>
> AWS CloudFormation ist kostenlos, aber die Ressourcen, die CloudFormation erstellt werden, sind live. Es fallen die üblichen Nutzungsgebühren für diese Ressourcen an, bis Sie sie kündigen. Weitere Informationen finden Sie unter Preise für RDS for Oracle.

Gehen Sie wie folgt vor, um Ihre Ressourcen mithilfe der AWS CloudFormation Konsole zu erstellen:

- Schritt 1: Laden Sie die CloudFormation Vorlage herunter
- Schritt 2: Konfigurieren Sie Ihre Ressourcen mit CloudFormation

Laden Sie die CloudFormation Vorlage herunter

Eine CloudFormation Vorlage ist eine JSON- oder YAML-Textdatei, die die Konfigurationsinformationen zu den Ressourcen enthält, die Sie im Stack erstellen möchten. Diese Vorlage erstellt zusammen mit der RDS-Instanz auch eine VPC und einen Bastion-Host für Sie.

Um die Vorlagendatei herunterzuladen, öffnen Sie den folgenden Link: CloudFormation Oracle-Vorlage.

Klicken Sie auf der Github-Seite auf die Schaltfläche Rohdatei herunterladen, um die YAML-Vorlagendatei zu speichern.

Konfigurieren Sie Ihre Ressourcen mit CloudFormation

> **Note**
>
> Bevor Sie diesen Vorgang starten, stellen Sie sicher, dass Sie ein Schlüsselpaar für eine EC2 Instance in Ihrem haben AWS-Konto. Weitere Informationen finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

Wenn Sie die AWS CloudFormation Vorlage verwenden, müssen Sie die richtigen Parameter auswählen, um sicherzustellen, dass Ihre Ressourcen ordnungsgemäß erstellt werden. Führen Sie die folgenden Schritte aus:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie Stapel erstellen aus.
3. Wählen Sie im Abschnitt Vorlage angeben die Option Eine Vorlagendatei von Ihrem Computer hochladen und dann Weiter aus.
4. Legen Sie auf der Seite „Stack-Details angeben“ die folgenden Parameter fest:
   a. Setzen Sie den Stacknamen auf OracleTestStack.
   b. Legen Sie unter Parameter Availability Zones fest, indem Sie drei Availability Zones auswählen.
   c. Wählen Sie unter Linux Bastion Host configuration für Key Name ein key pair aus, um sich bei Ihrer EC2 Instance anzumelden.
   d. Stellen Sie in den Linux Bastion Host-Konfigurationseinstellungen den Bereich Zulässiger IP auf Ihre IP-Adresse ein. Um mithilfe von Secure Shell (SSH) eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen, ermitteln Sie Ihre öffentliche IP-Adresse mithilfe des Dienstes unter. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instances über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktion nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

e. Stellen Sie unter Allgemeine Datenbankkonfiguration die Datenbankinstanzklasse auf db.t3.micro ein.

f. Setzen Sie den Datenbanknamen auf. **`database-test1`**

g. Geben Sie unter Datenbank-Master-Benutzername einen Namen für den Masterbenutzer ein.

h. Stellen Sie `false` für dieses Tutorial das DB-Master-Benutzerpasswort mit Secrets Manager verwalten auf ein.

i. Geben Sie für das Datenbankkennwort ein Passwort Ihrer Wahl ein. Merken Sie sich dieses Passwort für weitere Schritte im Tutorial.

j. Stellen Sie unter Datenbankspeicherkonfiguration den Datenbankspeichertyp auf gp2 ein.

k. Stellen Sie unter Konfiguration der Datenbanküberwachung die Option Enable RDS Performance Insights auf false ein.

l. Behalten Sie für alle anderen Einstellungen die Standardwerte bei. Klicken Sie auf Weiter, um fortzufahren.

5. Behalten Sie auf der Seite „Stack-Optionen konfigurieren“ alle Standardoptionen bei. Klicken Sie auf Weiter, um fortzufahren.
6. Wählen Sie auf der Seite „Stack überprüfen“ die Option Senden aus, nachdem Sie die Datenbank- und Linux-Bastion-Host-Optionen überprüft haben.

Sehen Sie sich nach Abschluss der Stack-Erstellung die Stacks mit Namen BastionStackund RDSNS an, um die Informationen zu notieren, die Sie für die Verbindung mit der Datenbank benötigen. Weitere Informationen finden Sie unter AWS CloudFormation Stack-Daten und Ressourcen anzeigen auf der. AWS Management Console

Schritt 3: Verbinden Ihres SQL-Clients mit einer Oracle-DB-Instance

Sie können für die Verbindung mit der DB-Instance eine beliebige SQL-Client-Standardanwendung verwenden. In diesem Beispiel stellen Sie eine Verbindung mit einer Oracle-DB-Instance mithilfe des Oracle-Befehlszeilen-Clients her.

So stellen Sie eine Verbindung mit einer Oracle-DB-Instance her

1. Suchen Sie nach dem Endpunkt (DNS-Name) und der Portnummer für Ihre DB-Instance.

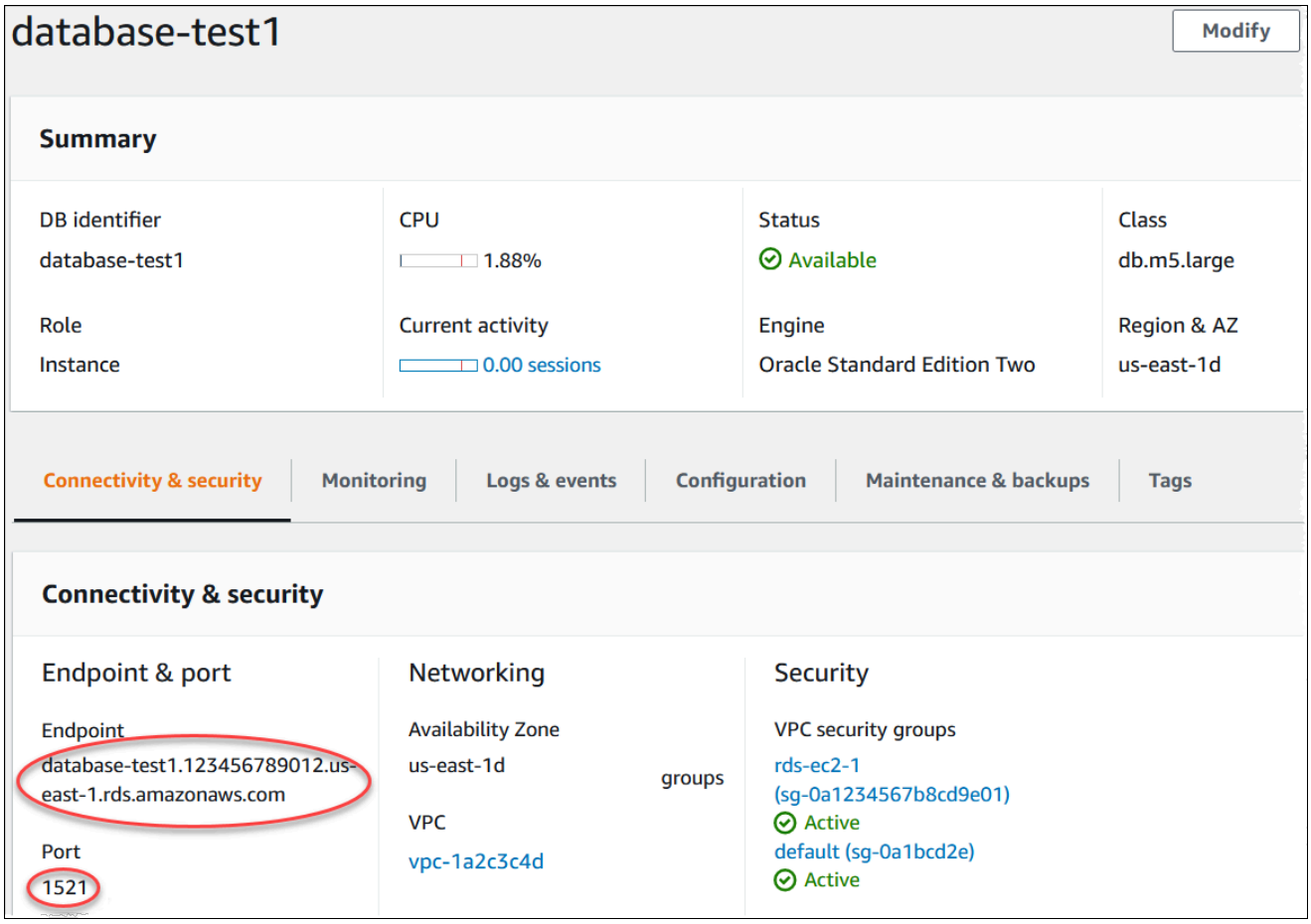
   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

   b. Wählen Sie oben rechts in der Amazon-RDS-Konsole die AWS-Region für die DB-Instance.

c. Wählen Sie im Navigationsbereich Datenbanken aus.

d. Wählen Sie den Namen der Oracle DB-Instance aus, um deren Details anzuzeigen.

e. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



2. Stellen Sie eine Verbindung zu der EC2 Instance her, die Sie zuvor erstellt haben, indem Sie den Schritten unter Connect to your Linux Instance im EC2 Amazon-Benutzerhandbuch folgen.

   Wir empfehlen, dass Sie über SSH eine Verbindung zu Ihrer EC2 Instance herstellen. Wenn das SSH-Client-Dienstprogramm unter Windows, Linux oder Mac installiert ist, können Sie mit dem folgenden Befehlsformat eine Verbindung mit der Instance herstellen:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Gehen Sie beispielsweise davon aus, dass dies `/dir1` unter Linux gespeichert `ec2-database-connect-key-pair.pem` ist und das öffentliche IPv4 DNS für Ihre EC2 Instance

bereits. `ec2-12-345-678-90.compute-1.amazonaws.com` Ihr SSH-Befehl würde wie folgt aussehen:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Holen Sie sich die neuesten Bugfixes und Sicherheitsupdates, indem Sie die Software auf Ihrer EC2 Instance aktualisieren. Führen Sie dazu den folgenden Befehl aus.

> **Note**
>
> Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Um Updates vor der Installation zu überprüfen, lassen Sie diese Option aus.

```
sudo dnf update -y
```

4. Gehen Sie in einem Webbrowser zu https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
5. Für die neueste Datenbankversion, die auf der Webseite angezeigt wird, kopieren Sie die .rpm-Links (nicht die .zip-Links) für das Instant Client Basic Package und das SQL*Plus-Paket. Die folgenden Links beziehen sich beispielsweise auf Oracle Database Version 21.9:
   - https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
   - https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
6. Führen Sie in Ihrer SSH-Sitzung den Befehl `wget` aus, um die .rpm-Dateien von den Links herunterzuladen, die Sie im vorherigen Schritt erhalten haben. Das folgende Beispiel lädt die .rpm-Dateien für Oracle Database Version 21.9 herunter:

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

7. Installieren Sie die Pakete, indem Sie den Befehl `dnf` wie folgt ausführen:

```
sudo dnf install oracle-instantclient-*.rpm
```

8. Starten SQL*Plus und stellen Sie eine Verbindung mit der Oracle-DB-Instance her. Geben Sie z. B. den folgenden Befehl ein.

   Ersetzen Sie den DB-Instance-Endpunkt (DNS-Name) für *oracle-db-instance-endpoint* und den Hauptbenutzernamen, den Sie für *admin* verwendet haben. Wenn Sie Einfache Erstellung für Oracle verwenden, lautet der Datenbankname DATABASE. Geben Sie das Master-Passwort ein, das Sie bei der Aufforderung zur Eingabe eines Passworts verwendet haben.

   ```
   sqlplus admin@oracle-db-instance-endpoint:1521/DATABASE
   ```

   Nachdem Sie das Passwort für den Benutzer eingegeben haben, sollte eine Ausgabe wie die folgende angezeigt werden.

   ```
   SQL*Plus: Release 21.0.0.0.0 - Production on Wed Mar 1 16:41:28 2023
   Version 21.9.0.0.0

   Copyright (c) 1982, 2022, Oracle.  All rights reserved.

   Enter password:
   Last Successful login time: Wed Mar 01 2023 16:30:52 +00:00

   Connected to:
   Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
   Version 19.18.0.0.0

   SQL>
   ```

   Weitere Informationen zum Herstellen einer Verbindung mit einer DB-Instance von RDS für Oracle finden Sie unter Herstellen der Verbindung zu Ihrer Oracle-DB-Instance. Wenn Sie sich nicht mit Ihrer DB-Instance verbinden können, erhalten Sie unter Hilf Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

   Aus Sicherheitsgründen empfiehlt es sich, verschlüsselte Verbindungen zu verwenden. Verwenden Sie eine unverschlüsselte Oracle-Verbindung nur, wenn sich Client und Server in derselben VPC befinden und das Netzwerk vertrauenswürdig ist. Weitere IInformationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Sichern von Verbindungen von Oracle DB-Instances.

9. SQL-Befehle ausführen

   Der folgende SQL-Befehl zeigt z B. das aktuelle Datum an:

```
SELECT SYSDATE FROM DUAL;
```

Schritt 4: Löschen Sie die Instance und die DB-Instance EC2

Nachdem Sie eine Verbindung zu der von Ihnen erstellten EC2 Beispiel-Instance und DB-Instance hergestellt und diese erkundet haben, löschen Sie sie, damit Ihnen keine Gebühren mehr berechnet werden.

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, überspringen Sie diesen Schritt und fahren Sie mit dem nächsten Schritt fort.

Um die EC2 Instanz zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die EC2 Instance aus und wählen Sie dann Instance-Status, Instanz beenden aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

Weitere Informationen zum Löschen einer EC2 Instance finden Sie unter Kündigen Ihrer Instance im EC2 Amazon-Benutzerhandbuch.

So löschen Sie die DB-Instance ohne finalen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Löschen Sie Abschließenden Snapshot erstellen? und Automatische Backups aufbewahren.
6. Bestätigen Sie, und wählen Sie Löschen.

(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, löschen Sie den CloudFormation Stack, nachdem Sie sich mit der EC2 Beispiel-Instance und der DB-Instance verbunden und diese erkundet haben, sodass Ihnen diese nicht mehr in Rechnung gestellt werden.

Um die CloudFormation Ressourcen zu löschen

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite Stacks in den den CloudFormationconsole Root-Stack aus (den Stack ohne Namen VPCStack BastionStack oder RDSNS).
3. Wählen Sie Löschen aus.
4. Wählen Sie Stack löschen, wenn Sie zur Bestätigung aufgefordert werden.

Weitere Informationen zum Löschen eines Stacks in CloudFormation finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks auf der AWS CloudFormation Konsole.

(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Sie können Ihre DB-Instance von RDS für Oracle auch mit einer Lambda-Serverless-Rechenressource verbinden. Mit Lambda-Funktionen können Sie Code ausführen, ohne die Infrastruktur bereitstellen oder verwalten zu müssen. Eine Lambda-Funktion ermöglicht es Ihnen auch, automatisch auf Codeausführungsanfragen jeder Größenordnung zu reagieren, von einem Dutzend Ereignissen pro Tag bis hin zu Hunderten von Ereignissen pro Sekunde. Weitere Informationen finden Sie unter Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance.

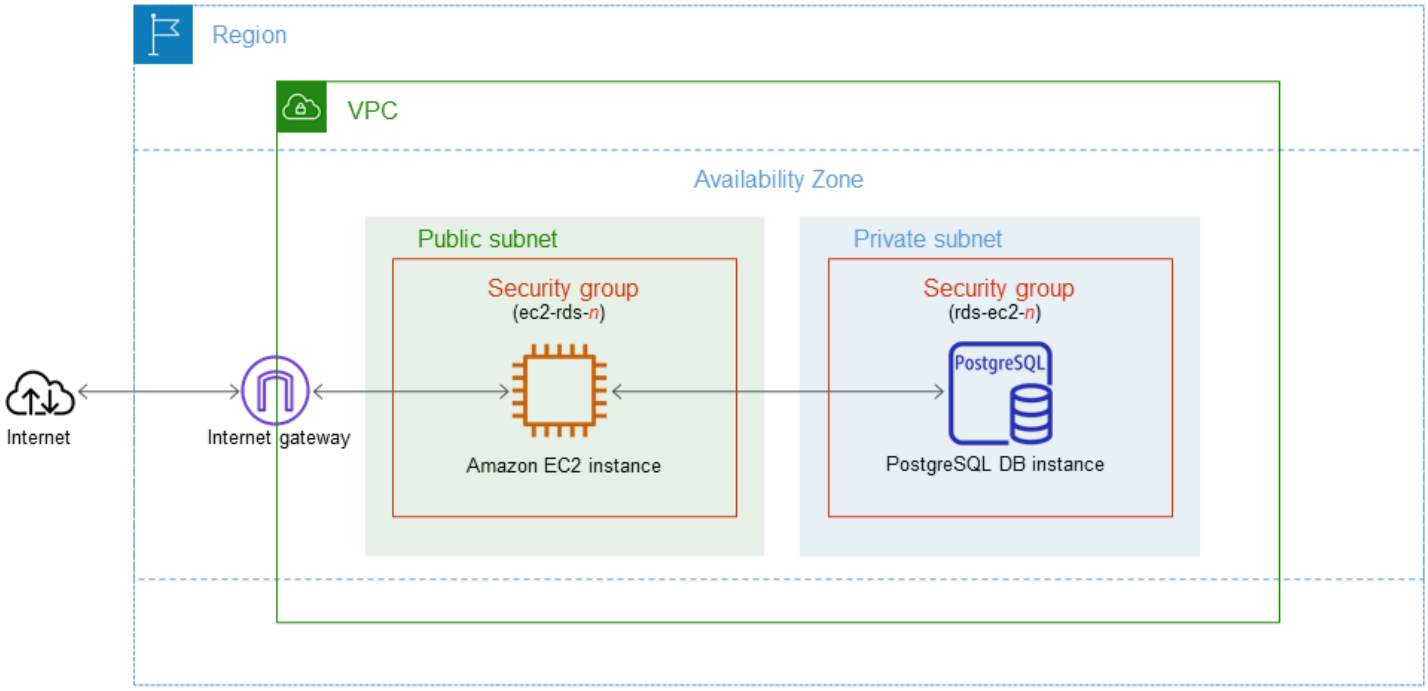
Erstellen einer PostgreSQL-DB-Instance und Herstellen einer Verbindung

In diesem Tutorial werden eine EC2 Instanz und eine RDS für PostgreSQL-DB-Instance erstellt. Das Tutorial zeigt Ihnen, wie Sie mit einem Standard-PostgreSQL-Client von der EC2 Instance aus auf die DB-Instance zugreifen. Als bewährte Methode erstellt dieses Tutorial eine private DB-Instance in einer Virtual Private Cloud (VPC). In den meisten Fällen können andere Ressourcen in derselben VPC, z. B. EC2 Instances, auf die DB-Instance zugreifen, aber Ressourcen außerhalb der VPC können nicht darauf zugreifen.

Nach Abschluss des Tutorials gibt es in jeder Availability Zone im VPC ein öffentliches und ein privates Subnetz. In einer Availability Zone befindet sich die EC2 Instance im öffentlichen Subnetz und die DB-Instance im privaten Subnetz.

> **Important**
>
> Für die Erstellung eines Kontos fallen keine Gebühren an AWS . Wenn Sie dieses Tutorial abschließen, können Ihnen jedoch Kosten für die von Ihnen verwendeten AWS Ressourcen entstehen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

Das folgende Diagramm zeigt die Konfiguration nach Abschluss des Tutorials.



In diesem Tutorial können Sie Ihre Ressourcen mithilfe einer der folgenden Methoden erstellen:

1. Verwenden Sie das AWS Management Console - Erstellen Sie eine EC2 Instanz und Erstellen Sie eine PostgreSQL-DB-Instance
2. Verwenden Sie AWS CloudFormation , um die Datenbankinstanz und die EC2 Instanz zu erstellen - (Optional) Erstellen Sie VPC, EC2 Instanz und PostgreSQL-Instanz mit AWS CloudFormation

Die erste Methode verwendet Easy create, um eine private PostgreSQL-DB-Instance mit dem zu erstellen. AWS Management Console Hier geben Sie nur den DB-Engine-Typ, die DB-Instance-Größe und die DB-Instance-ID an. Easy Create (Einfache Erstellung) verwendet für die anderen Konfigurationsoptionen die Standardeinstellung.

Wenn Sie stattdessen Standard create verwenden, können Sie beim Erstellen einer DB-Instance weitere Konfigurationsoptionen angeben. Zu diesen Optionen gehören Einstellungen für Verfügbarkeit, Sicherheit, Backups und Wartung. Um eine öffentliche DB-Instance zu erstellen, müssen Sie Standarderstellung verwenden. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Themen

- Voraussetzungen
- Erstellen Sie eine EC2 Instanz

- Erstellen Sie eine PostgreSQL-DB-Instance
- (Optional) Erstellen Sie VPC, EC2 Instanz und PostgreSQL-Instanz mit AWS CloudFormation
- Herstellen einer Verbindung zu einer PostgreSQL-DB-Instance
- Löschen Sie die EC2 Instance und die DB-Instance
- (Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation
- (Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

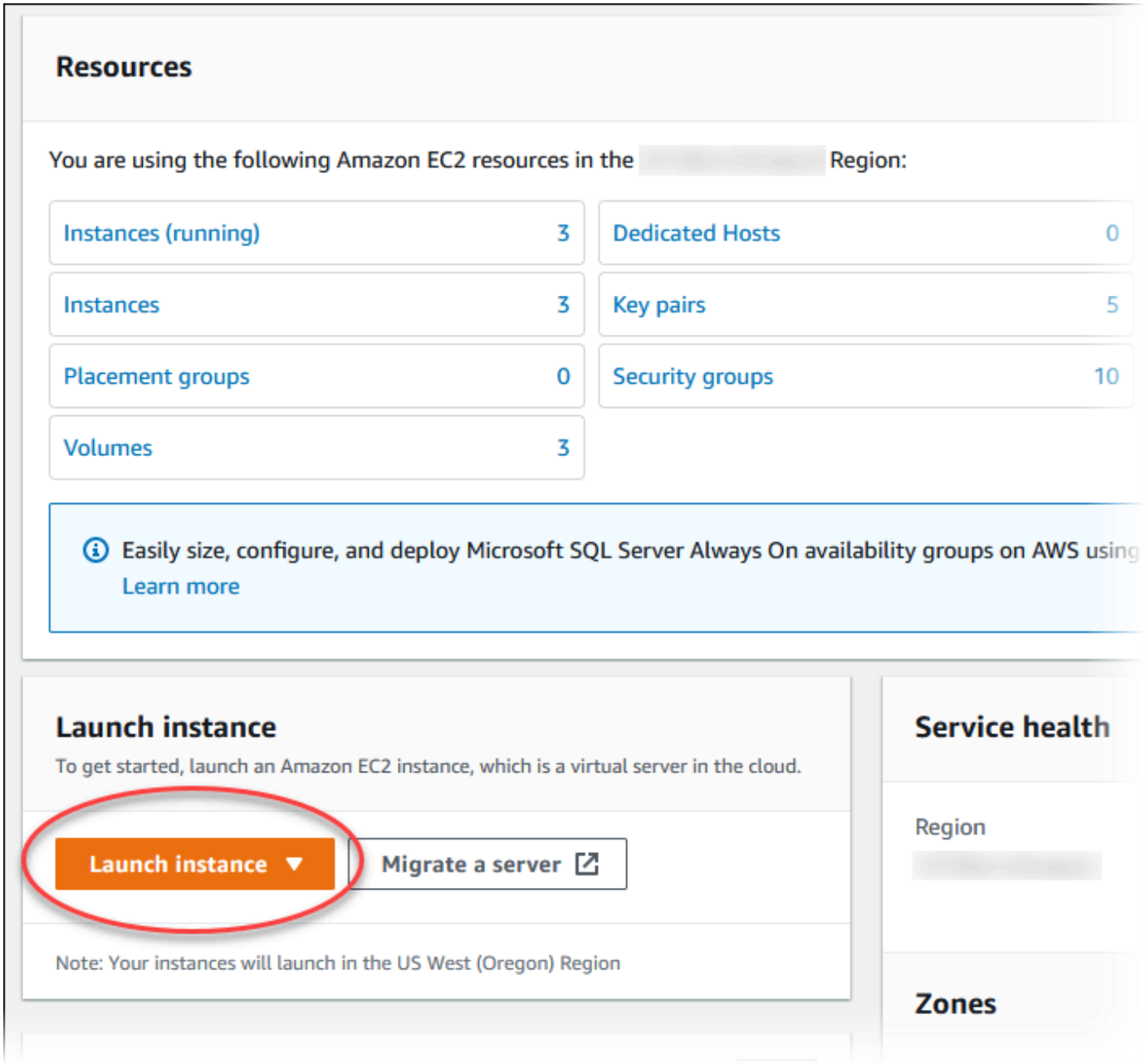
- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Erstellen Sie eine EC2 Instanz

Erstellen Sie eine EC2 Amazon-Instance, mit der Sie eine Verbindung zu Ihrer Datenbank herstellen.

Um eine EC2 Instance zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen rechten Ecke von die aus AWS Management Console, AWS-Region in der Sie die Instance erstellen möchten. EC2
3. Wählen Sie EC2 Dashboard und dann Launch instance aus, wie in der folgenden Abbildung gezeigt.

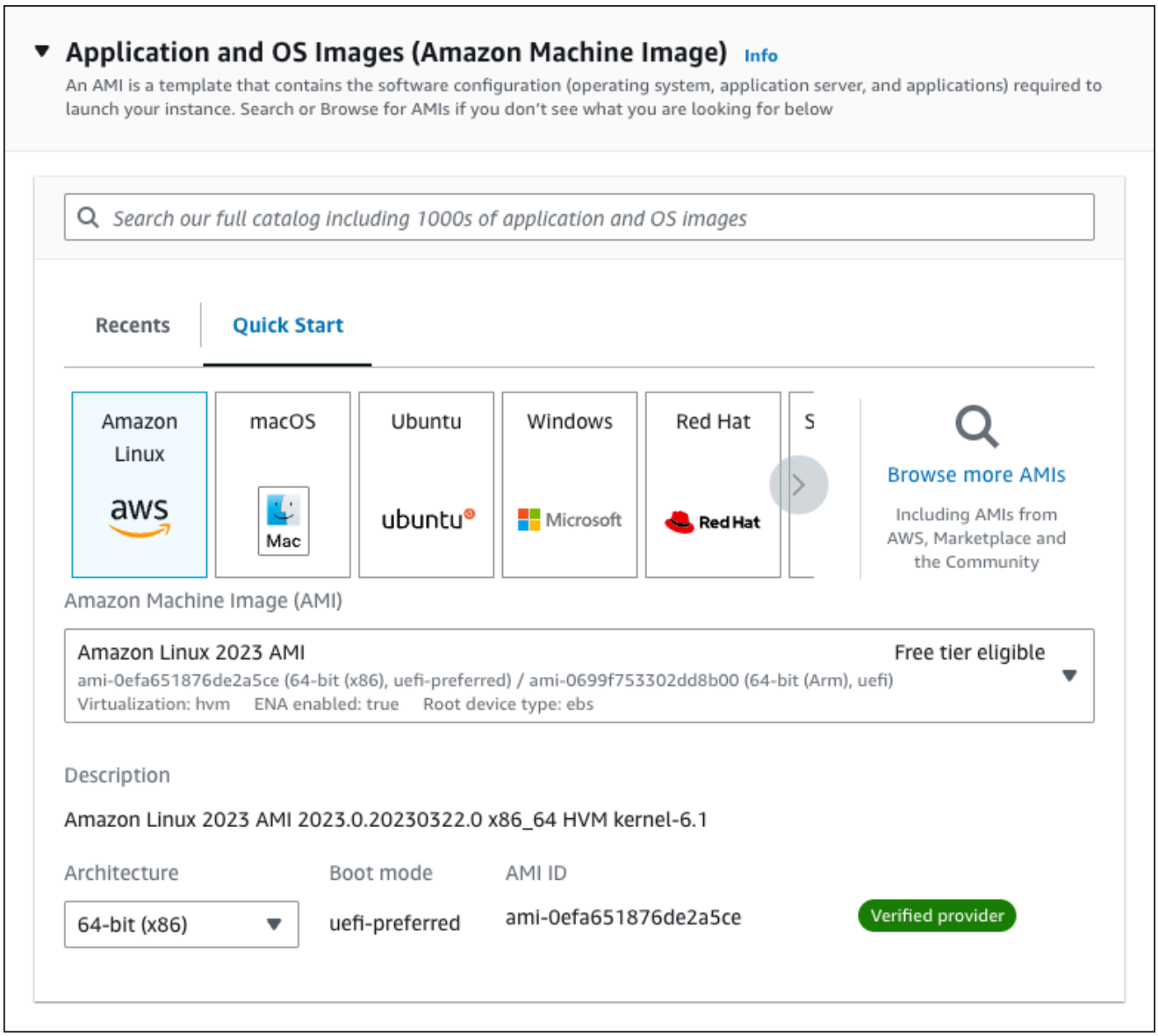


Die Seite Eine Instance starten wird geöffnet.

4. Wählen Sie auf der Seite Eine Instance starten die folgenden Einstellungen aus.

   a. Geben Sie unter Name and tags (Name und Tags) als Name den Namen **`ec2-database-connect`** ein.

   b. Wählen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) die Option Amazon Linux und dann die Registerkarte Amazon Linux 2023 AMI aus. Übernehmen Sie für alle anderen Einstellungen die Standardwerte.



c. Wählen Sie unter Instance type (Instance-Typ) den Wert t2.micro aus.

d. Wählen Sie unter Key pair (login) (Schlüsselpaar (Anmeldung)) einen Key pair name (Schlüsselpaarname), um ein vorhandenes Schlüsselpaar zu verwenden. Um ein neues key pair für die EC2 Amazon-Instance zu erstellen, wählen Sie Neues key pair erstellen und verwenden Sie dann das Fenster key pair erstellen, um es zu erstellen.

   Weitere Informationen zum Erstellen eines neuen key pair finden Sie unter Erstellen eines key pair im EC2 Amazon-Benutzerhandbuch.

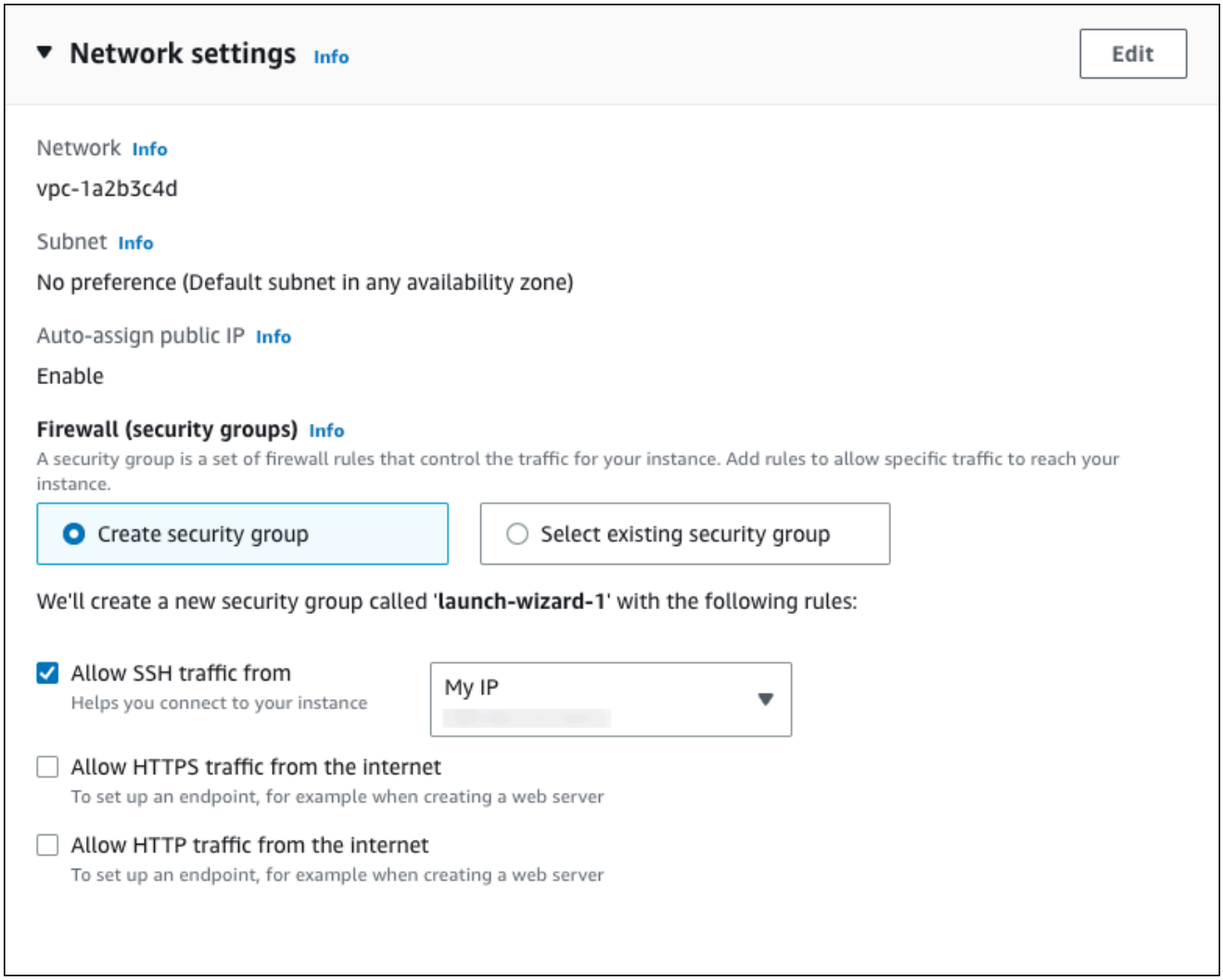
e. Wählen Sie unter „SSH-Verkehr zulassen“ in den Netzwerkeinstellungen die Quelle der SSH-Verbindungen zur Instance aus. EC2

Sie können My IP (Meine IP) auswählen, wenn die angezeigte IP-Adresse für SSH-Verbindungen korrekt ist. Andernfalls können Sie mithilfe von Secure Shell (SSH) die IP-Adresse bestimmen, die für die Verbindung zu EC2 Instances in Ihrer VPC verwendet werden soll. Um Ihre öffentliche IP-Adresse in einem anderen Browserfenster oder einer anderen Registerkarte zu ermitteln, können Sie den Dienst unter verwenden. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Bestimmen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instanzen über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

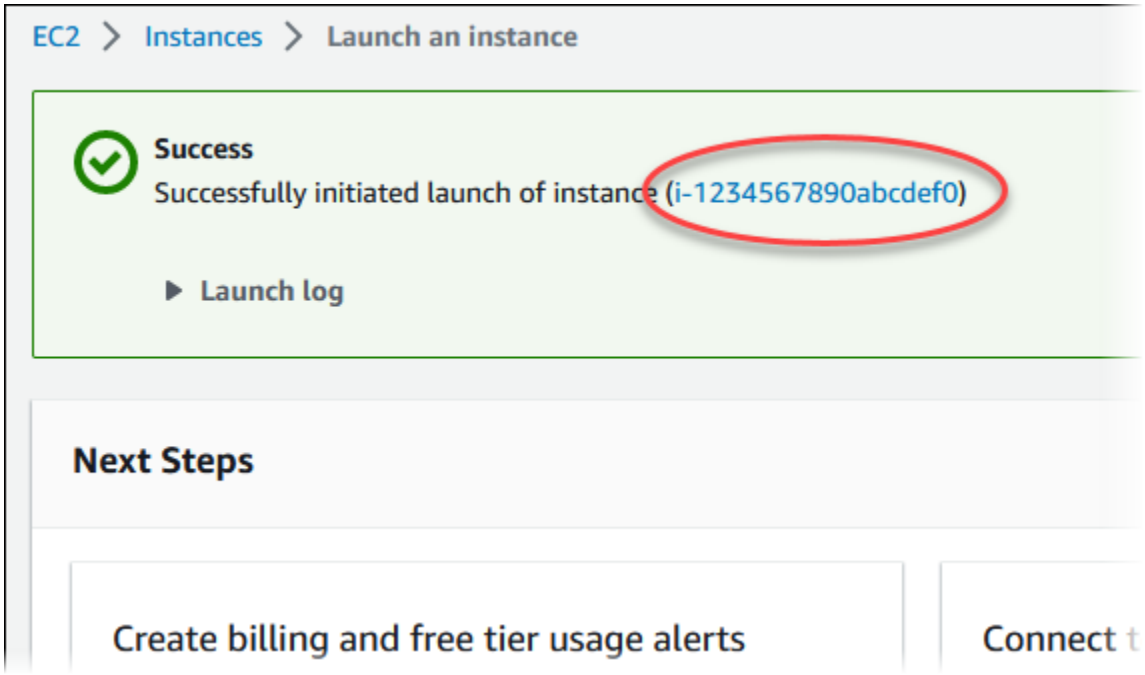
Die folgende Abbildung zeigt ein Beispiel für den Bereich Netzwerkeinstellungen.



f. Übernehmen Sie die Standardwerte für die übrigen Abschnitte.

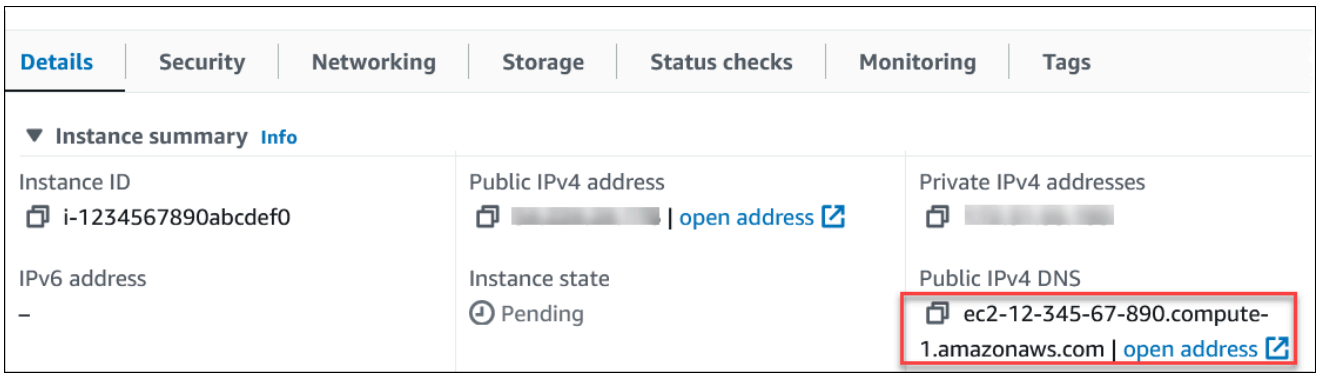
g. Sehen Sie sich im Bereich Zusammenfassung eine Zusammenfassung Ihrer EC2 Instance-Konfiguration an. Wenn Sie bereit sind, wählen Sie Launch instance aus.

5. Notieren Sie sich auf der Seite Startstatus die ID für Ihre neue EC2 Instance, zum Beispiel:`i-1234567890abcdef0`.

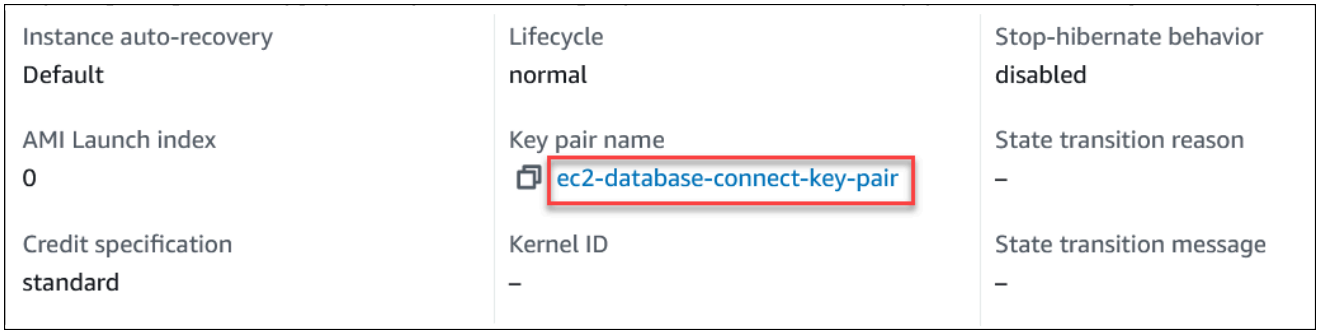


6. Wählen Sie die EC2 Instance-ID, um die Liste der EC2 Instances zu öffnen, und wählen Sie dann Ihre EC2 Instance aus.
7. Notieren Sie sich die folgenden Werte auf der Registerkarte Details. Diese benötigen Sie, wenn Sie eine Verbindung über SSH herstellen:

   a. Notieren Sie sich in der Instanzübersicht den Wert für Public IPv4 DNS.



   b. Notieren Sie sich unter Instance-Details den Wert für Schlüsselpaarname.



8. Warten Sie, bis der Instanzstatus Ihrer EC2 Instance den Status Running hat, bevor Sie fortfahren.

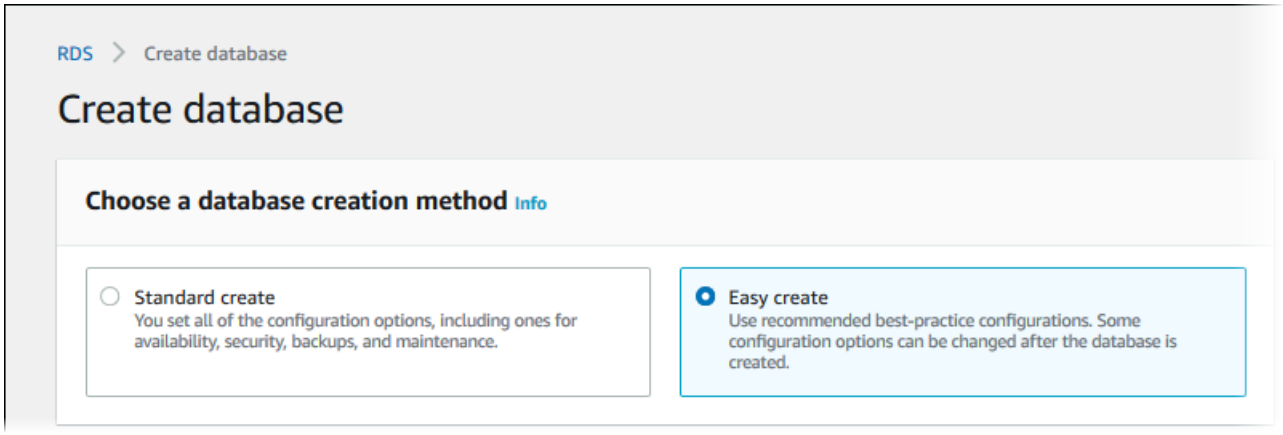
Erstellen Sie eine PostgreSQL-DB-Instance

Die Grundbausteine für Amazon RDS sind Datenbank-Instances. In dieser Umgebung führen Sie Ihre PostgreSQL-Datenbanken aus.

In diesem Beispiel verwenden Sie die Option Einfache Erstellung, um eine DB-Instance zu erstellen, die die PostgreSQL-Datenbank-Engine mit einer DB-Instance-Klasse des Typs „db.t3.micro“ ausführt.

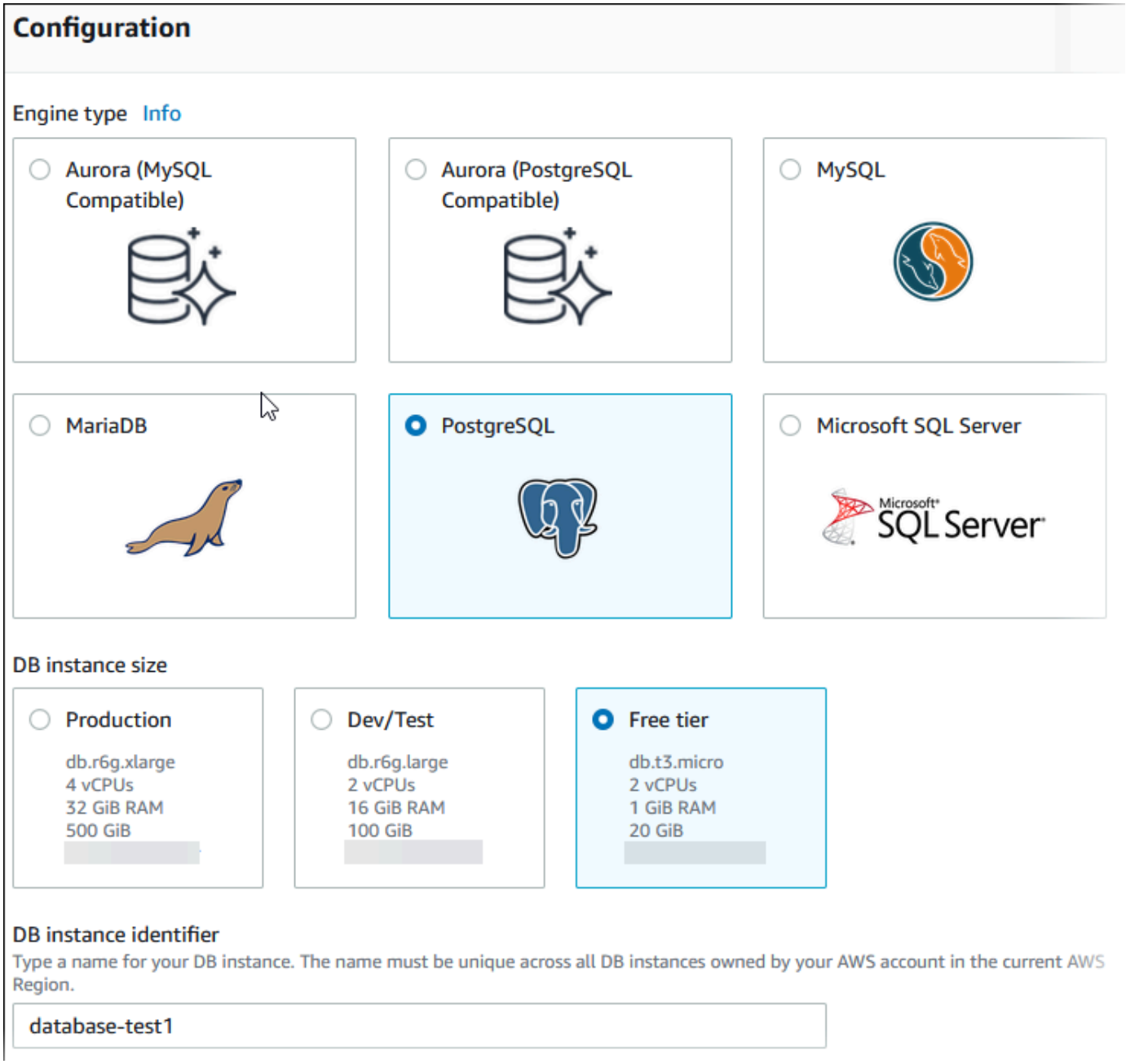
So erstellen Sie eine PostgreSQL DB-Instance mit Easy Create (Einfache Erstellung):

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die AWS Region aus, in der Sie die DB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Datenbank erstellen aus und vergewissern Sie sich, dass Einfache Erstellung ausgewählt ist.



5. Wählen Sie unter Configuration (Konfiguration), die Option PostgreSQL.
6. Wählen Sie in DB instance size (DB-Instance-Größe) die Option Free tier (Kostenloses Kontingent) aus. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.
7. Geben Sie als DB-Instance-ID **`database-test1`** ein.
8. Geben Sie unter Hauptbenutzername einen Namen für den Hauptbenutzer ein oder behalten Sie den Standardnamen bei (**`postgres`**).

   Die Seite Datenbank erstellen sollte ähnlich wie in der folgenden Abbildung gezeigt aussehen. Bei Konten mit kostenlosem Tarif wird das kostenlose Kontingent angezeigt. Bei Konten mit kostenpflichtigem Tarif wird Sandbox angezeigt.

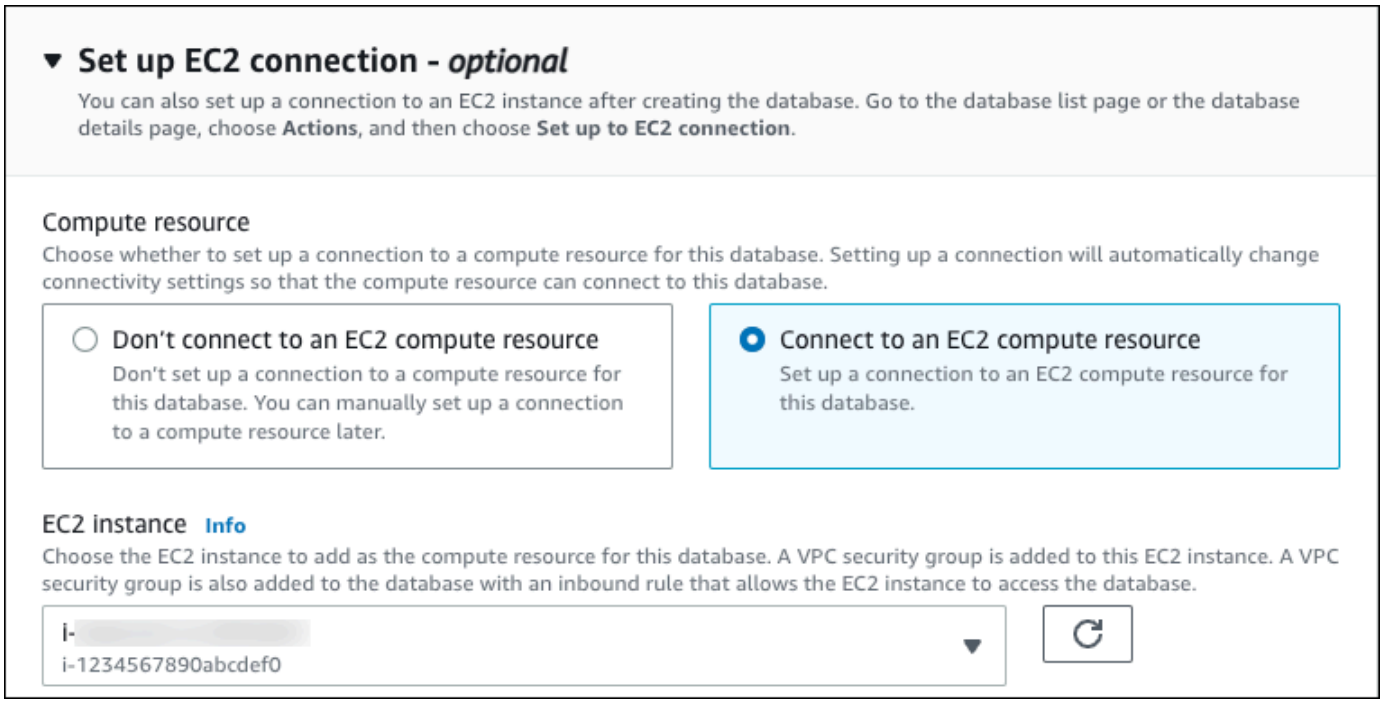


9. Um für die DB-Instance ein automatisch generiertes Hauptpasswort zu verwenden, wählen Sie das Kästchen Passwort automatisch generieren aus.

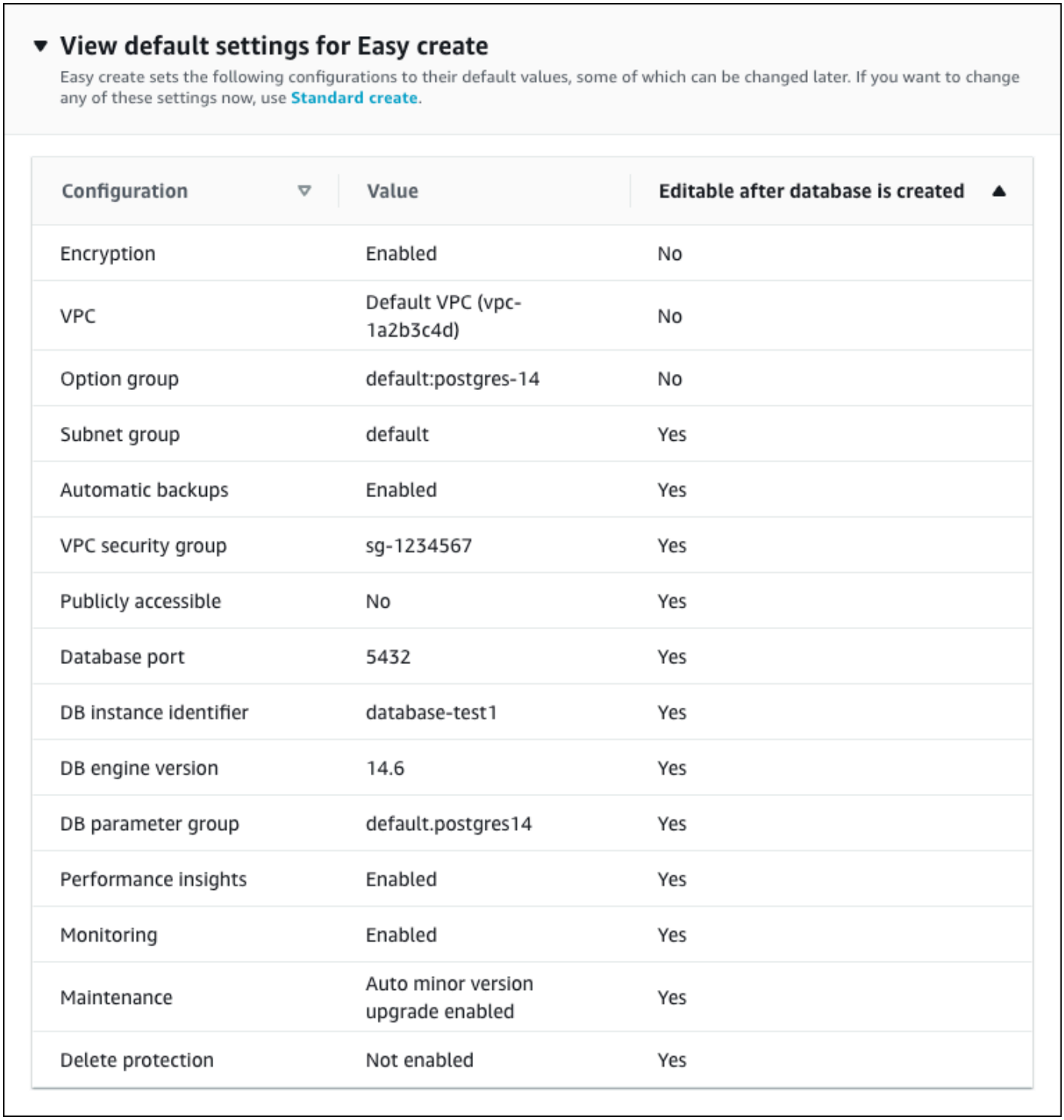
   Um das Hauptpasswort einzugeben, deaktivieren Sie das Kästchen Passwort automatisch generieren und geben Sie anschließend dasselbe Passwort in Hauptpasswort und Passwort bestätigen ein.

10. Um eine Verbindung mit der EC2 Instanz einzurichten, die Sie zuvor erstellt haben, öffnen Sie EC2 Verbindung einrichten — optional.

Wählen Sie Mit einer EC2 Rechenressource verbinden aus. Wählen Sie die EC2 Instanz aus, die Sie zuvor erstellt haben.



11. (Optional) Öffnen Sie Anzeigen von Standardeinstellungen für eine einfache Erstellung.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:postgres-14	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.6	Yes
DB parameter group	default.postgres14	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Sie können die Standardeinstellungen von Einfache Erstellung einsehen. Die Spalte Nach Erstellung der Datenbank editierbar zeigt, welche Optionen Sie nach der Datenbankerstellung ändern können.

- Wenn in einer Einstellung Nein in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie Standarderstellung verwenden, um die DB-Instance zu erstellen.
- Wenn für eine Einstellung Ja in dieser Spalte steht und Sie eine andere Einstellung wünschen, können Sie entweder Standarderstellung verwenden, oder die DB-Instance nach der Erstellung ändern, um die Einstellung zu ändern.

12. Wählen Sie Datenbank erstellen aus.

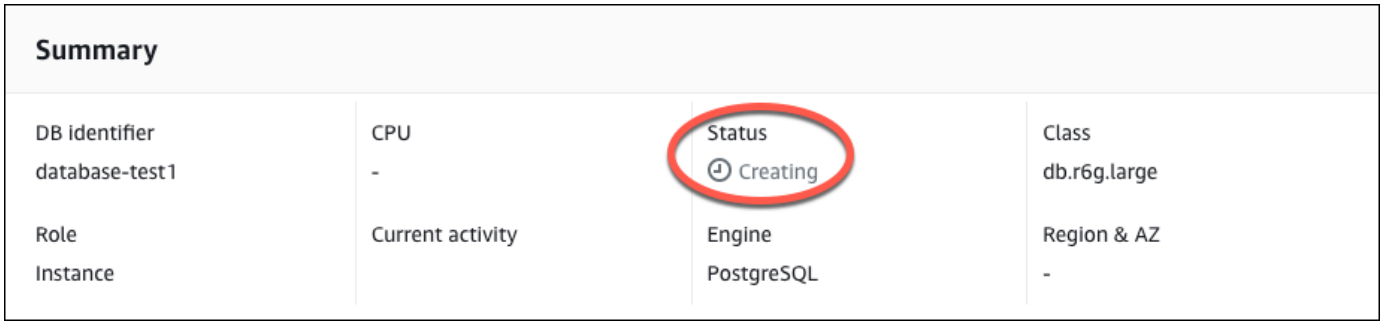
    Um den Masterbenutzernamen und das zugehörige Passwort für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen).

    Sie können den angezeigten Benutzernamen und das angezeigte Passwort verwenden, um als Masterbenutzer eine Verbindung zu DB-Instance herzustellen.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern.
    > Wenn Sie das Passwort für den Hauptbenutzer ändern müssen, nachdem die DB-Instance verfügbar wurde, können Sie die DB-Instance entsprechend ändern. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

13. Wählen Sie in der Liste Datenbanken den Namen der neuen PostgreSQL-DB-Instance aus, um deren Details anzuzeigen.

    Die DB-Instance hat den Status Wird erstellt, bis die DB-Instance bereit für die Verwendung ist.



    Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und vom verfügbaren Speicherplatz kann es bis zu 20 Minuten dauern, bis die neue DB-Instance verfügbar ist.

(Optional) Erstellen Sie VPC, EC2 Instanz und PostgreSQL-Instanz mit AWS CloudFormation

Anstatt die Konsole zum Erstellen Ihrer VPC, EC2 Instance und PostgreSQL-Instance AWS CloudFormation zu verwenden, können Sie AWS Ressourcen bereitstellen, indem Sie Infrastruktur als Code behandeln. Um Ihnen zu helfen, Ihre AWS Ressourcen in kleinere und besser verwaltbare Einheiten zu organisieren, können Sie die Nested-Stack-Funktionalität verwenden. AWS CloudFormation Weitere Informationen finden Sie unter Einen Stack auf der AWS CloudFormation Konsole erstellen und Mit verschachtelten Stacks arbeiten.

> **Important**
>
> AWS CloudFormation ist kostenlos, aber die Ressourcen, die CloudFormation erstellt werden, sind live. Es fallen die üblichen Nutzungsgebühren für diese Ressourcen an, bis Sie sie kündigen. Weitere Informationen finden Sie unter Preise für RDS for PostgreSQL.

Gehen Sie wie folgt vor, um Ihre Ressourcen mithilfe der AWS CloudFormation Konsole zu erstellen:

- Laden Sie die CloudFormation Vorlage herunter
- Konfigurieren Sie Ihre Ressourcen mit CloudFormation

Laden Sie die CloudFormation Vorlage herunter

Eine CloudFormation Vorlage ist eine JSON- oder YAML-Textdatei, die die Konfigurationsinformationen zu den Ressourcen enthält, die Sie im Stack erstellen möchten. Diese Vorlage erstellt zusammen mit der RDS-Instanz auch eine VPC und einen Bastion-Host für Sie.

Um die Vorlagendatei herunterzuladen, öffnen Sie den folgenden Link: CloudFormation PostgreSQL-Vorlage.

Klicken Sie auf der Github-Seite auf die Schaltfläche Rohdatei herunterladen, um die YAML-Vorlagendatei zu speichern.

Konfigurieren Sie Ihre Ressourcen mit CloudFormation

> **Note**
>
> Bevor Sie diesen Vorgang starten, stellen Sie sicher, dass Sie ein Schlüsselpaar für eine EC2 Instance in Ihrem haben AWS-Konto. Weitere Informationen finden Sie unter EC2Amazon-Schlüsselpaare und Linux-Instances.

Wenn Sie die AWS CloudFormation Vorlage verwenden, müssen Sie die richtigen Parameter auswählen, um sicherzustellen, dass Ihre Ressourcen ordnungsgemäß erstellt werden. Führen Sie die folgenden Schritte aus:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Wählen Sie Stapel erstellen aus.
3. Wählen Sie im Abschnitt Vorlage angeben die Option Eine Vorlagendatei von Ihrem Computer hochladen und dann Weiter aus.
4. Legen Sie auf der Seite „Stack-Details angeben“ die folgenden Parameter fest:
   a. Stellen Sie den Stacknamen auf Postgre SQLTest Stack ein.
   b. Legen Sie unter Parameter Availability Zones fest, indem Sie drei Availability Zones auswählen.
   c. Wählen Sie unter Linux Bastion Host configuration für Key Name ein key pair aus, um sich bei Ihrer EC2 Instance anzumelden.
   d. Stellen Sie in den Linux Bastion Host-Konfigurationseinstellungen den Bereich Zulässiger IP auf Ihre IP-Adresse ein. Um mithilfe von Secure Shell (SSH) eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen, ermitteln Sie Ihre öffentliche IP-Adresse mithilfe des Dienstes unter. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist 192.0.2.1/32.

> **Warning**
>
> Wenn Sie `0.0.0.0/0` den SSH-Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen EC2 Instances über SSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre EC2 Instances über SSH.

e. Stellen Sie unter Allgemeine Datenbankkonfiguration die Datenbankinstanzklasse auf db.t3.micro ein.

f. Setzen Sie den Datenbanknamen auf. **`database-test1`**

g. Geben Sie unter Datenbank-Master-Benutzername einen Namen für den Masterbenutzer ein.

h. Stellen Sie `false` für dieses Tutorial das DB-Master-Benutzerpasswort mit Secrets Manager verwalten auf ein.

i. Geben Sie für das Datenbankkennwort ein Passwort Ihrer Wahl ein. Merken Sie sich dieses Passwort für weitere Schritte im Tutorial.

j. Stellen Sie unter Datenbankspeicherkonfiguration den Datenbankspeichertyp auf gp2 ein.

k. Stellen Sie unter Konfiguration der Datenbanküberwachung die Option Enable RDS Performance Insights auf false ein.

l. Behalten Sie für alle anderen Einstellungen die Standardwerte bei. Klicken Sie auf Weiter, um fortzufahren.

5. Behalten Sie auf der Seite „Stack-Optionen konfigurieren“ alle Standardoptionen bei. Klicken Sie auf Weiter, um fortzufahren.
6. Wählen Sie auf der Seite „Stack überprüfen“ die Option Senden aus, nachdem Sie die Datenbank- und Linux-Bastion-Host-Optionen überprüft haben.

Sehen Sie sich nach Abschluss der Stack-Erstellung die Stacks mit Namen BastionStackund RDSNS an, um die Informationen zu notieren, die Sie für die Verbindung mit der Datenbank benötigen. Weitere Informationen finden Sie unter AWS CloudFormation Stack-Daten und Ressourcen anzeigen auf der. AWS Management Console

Herstellen einer Verbindung zu einer PostgreSQL-DB-Instance

Sie können mit pgadmin oder psql eine Verbindung mit der DB-Instance herstellen. In diesem Beispiel wird erklärt, wie Sie mit dem psql-Befehlszeilen-Client eine Verbindung mit einer PostgreSQL-DB-Instance herstellen.

So stellen Sie eine Verbindung mit einer PostgreSQL-DB-Instance mit psql her

1. Suchen Sie nach dem Endpunkt (DNS-Name) und der Portnummer für Ihre DB-Instance.

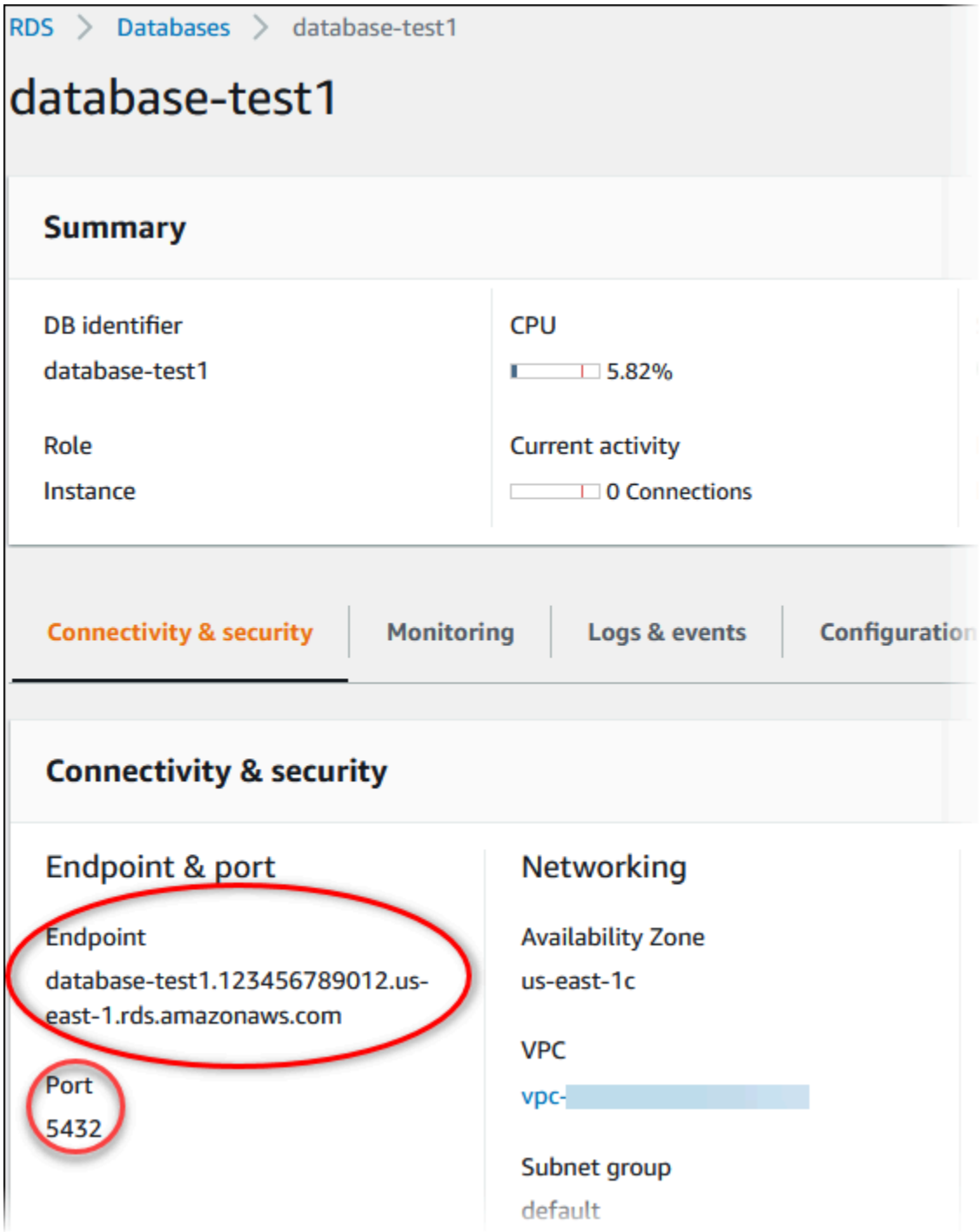
   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

   b. Wählen Sie oben rechts in der Amazon-RDS-Konsole die AWS-Region für die DB-Instance.

c. Wählen Sie im Navigationsbereich Datenbanken aus.

d. Wählen Sie den Namen der PostgreSQL-DB-Instance, um deren Details anzuzeigen.

e. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



2. Stellen Sie eine Verbindung zu der EC2 Instance her, die Sie zuvor erstellt haben, indem Sie den Schritten unter Connect to your Linux Instance im EC2 Amazon-Benutzerhandbuch folgen.

   Wir empfehlen, dass Sie über SSH eine Verbindung zu Ihrer EC2 Instance herstellen. Wenn das SSH-Client-Dienstprogramm unter Windows, Linux oder Mac installiert ist, können Sie mit dem folgenden Befehlsformat eine Verbindung mit der Instance herstellen:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Gehen Sie beispielsweise davon aus, dass dies `/dir1` unter Linux gespeichert `ec2-database-connect-key-pair.pem` ist und das öffentliche IPv4 DNS für Ihre EC2 Instance bereits. `ec2-12-345-678-90.compute-1.amazonaws.com` Ihr SSH-Befehl würde wie folgt aussehen:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

3. Holen Sie sich die neuesten Bugfixes und Sicherheitsupdates, indem Sie die Software auf Ihrer EC2 Instance aktualisieren. Verwenden Sie dazu den folgenden Befehl.

   > **Note**
   >
   > Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Um Updates vor der Installation zu überprüfen, lassen Sie diese Option aus.

   ```
   sudo dnf update -y
   ```

4. Führen Sie den folgenden Befehl aus, um den psql-Befehlszeilen-Client von PostgreSQL auf Amazon Linux 2023 zu installieren:

   ```
   sudo dnf install postgresql15
   ```

5. Stellen Sie eine Verbindung mit der PostgreSQL-DB-Instance her. Geben Sie beispielsweise den folgenden Befehl an einer Eingabeaufforderung auf einem Clientcomputer ein. Mit dieser Aktion können Sie eine Verbindung mit der PostgreSQL-DB-Instance mithilfe des psql-Clients herstellen.

Ersetzen Sie den Endpunkt der DB-Instance (DNS-Name) für *endpoint*, den Namen der Datenbank `--dbname`, mit der Sie eine Verbindung für *postgres* herstellen möchten, und den Hauptbenutzernamen, den Sie für *postgres* verwendet haben. Geben Sie das Master-Passwort ein, das Sie bei der Aufforderung zur Eingabe eines Passworts verwendet haben.

```
psql --host=endpoint --port=5432 --dbname=postgres --username=postgres
```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, sollte eine Ausgabe wie die folgende angezeigt werden:

```
psql (14.3, server 14.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
 compression: off)
Type "help" for help.

postgres=>
```

Weitere Informationen zum Herstellen einer Verbindung mit einer PostgreSQL-DB-Instance finden Sie unter Herstellen einer Verbindung zu einer DB-Instance, in der die PostgreSQL-Datenbank-Engine ausgeführt wird. Wenn Sie sich nicht mit Ihrer DB-Instance verbinden können, erhalten Sie unter Hilf Fehlerbehebung bei Verbindungen mit Ihrer RDS für PostgreSQL-Instance.

Aus Sicherheitsgründen empfiehlt es sich, verschlüsselte Verbindungen zu verwenden. Verwenden Sie eine unverschlüsselte PostgreSQL-Verbindung nur, wenn sich Client und Server in derselben VPC befinden und das Netzwerk vertrauenswürdig ist. Weitere IInformationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Herstellen einer Verbindung mit einer PostgreSQL-DB-Instance über SSL.

6. SQL-Befehle ausführen

   Der folgende SQL-Befehl zeigt z B. das aktuelle Datum und die aktuelle Zeit an:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Löschen Sie die EC2 Instance und die DB-Instance

Nachdem Sie sich mit der EC2 Beispiel-Instance und DB-Instance, die Sie erstellt haben, verbunden und diese erkundet haben, löschen Sie sie, damit Ihnen keine Gebühren mehr berechnet werden.

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, überspringen Sie diesen Schritt und fahren Sie mit dem nächsten Schritt fort.

Um die EC2 Instanz zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die EC2 Instance aus und wählen Sie dann Instance-Status, Instanz beenden aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Beenden aus.

Weitere Informationen zum Löschen einer EC2 Instance finden Sie unter Kündigen Ihrer Instance im EC2 Amazon-Benutzerhandbuch.

So löschen Sie eine DB-Instance ohne finalen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Löschen Sie Abschließenden Snapshot erstellen? und Automatische Backups aufbewahren.
6. Bestätigen Sie, und wählen Sie Löschen.

(Optional) Löschen Sie die EC2 Instance und die DB-Instance, die mit erstellt wurden CloudFormation

Wenn Sie früher AWS CloudFormation Ressourcen erstellt haben, löschen Sie den CloudFormation Stack, nachdem Sie sich mit der EC2 Beispiel-Instance und der DB-Instance verbunden und diese erkundet haben, sodass Ihnen diese nicht mehr in Rechnung gestellt werden.

Um die CloudFormation Ressourcen zu löschen

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie auf der Seite Stacks in den den CloudFormationconsole Root-Stack aus (den Stack ohne Namen VPCStack BastionStack oder RDSNS).
3. Wählen Sie Löschen aus.
4. Wählen Sie Stack löschen, wenn Sie zur Bestätigung aufgefordert werden.

Weitere Informationen zum Löschen eines Stacks in CloudFormation finden Sie im AWS CloudFormation Benutzerhandbuch unter Löschen eines Stacks auf der AWS CloudFormation Konsole.

(Optional) Verbinden Sie Ihre DB-Instance mit einer Lambda-Funktion.

Sie können Ihre DB-Instance von RDS für PostgreSQL auch mit einer Lambda-Serverless-Rechenressource verbinden. Mit Lambda-Funktionen können Sie Code ausführen, ohne die Infrastruktur bereitstellen oder verwalten zu müssen. Eine Lambda-Funktion ermöglicht es Ihnen auch, automatisch auf Codeausführungsanfragen jeder Größenordnung zu reagieren, von einem Dutzend Ereignissen pro Tag bis hin zu Hunderten von Ereignissen pro Sekunde. Weitere Informationen finden Sie unter Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance.

Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance

Dieses Tutorial veranschaulicht, wie Sie einen Apache-Webserver mit PHP installieren und eine MariaDB-, MySQL- oder PostgreSQL-Datenbank erstellen. Der Webserver wird auf einer Amazon-EC2-Instance unter Verwendung von Amazon Linux 2023 ausgeführt und Sie können zwischen einer MySQL- oder PostgreSQL-DB-Instance wählen. Die Amazon EC2-Instance und die -DB-Instance/der werden beide in einer virtuellen privaten Cloud (VPC) auf der Basis des Amazon VPC-Service ausgeführt.

Important

Die Einrichtung eines AWS-Kontos ist kostenlos. Bei Durchführung dieses Tutorials können jedoch Kosten für die von Ihnen verwendeten AWS-Ressourcen anfallen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

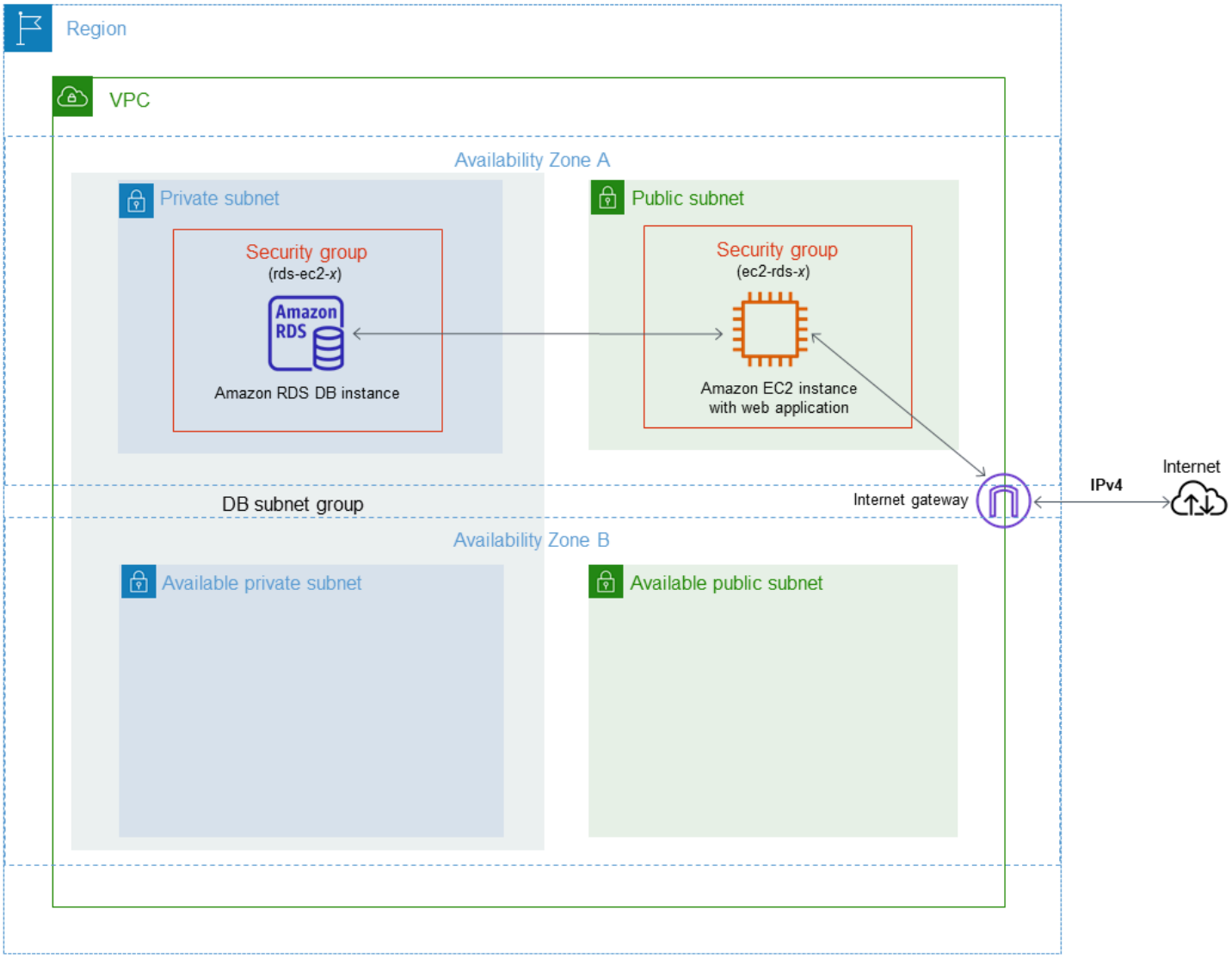
Note

Dieses Tutorial funktioniert mit Amazon Linux 2023 und möglicherweise nicht mit anderen Linux-Versionen.

Im folgenden Tutorial erstellen Sie eine EC2-Instance, die die Standard-VPC, Subnetze und die Sicherheitsgruppe für Ihr AWS-Konto verwendet. In diesem Tutorial erhalten Sie Informationen zum Erstellen der DB-Instance und richten die Verbindung mit der EC2-Instance, die Sie erstellt haben, automatisch ein. Das Tutorial zeigt Ihnen dann, wie Sie den Webserver auf der EC2-Instance installieren. Sie verbinden Ihren Webserver in der VPC mit Ihrer DB-Instance in der VPC mithilfe des DB-Instance-Endpunkts.

1. Starten Sie eine EC2 Instance, um eine Verbindung mit Ihrer herzustellen
2. Erstellen einer DB-Instance von Amazon RDS
3. Installieren eines Webservers auf Ihrer EC2-Instance

Das folgende Diagramm zeigt die Konfiguration nach Abschluss des Tutorials.



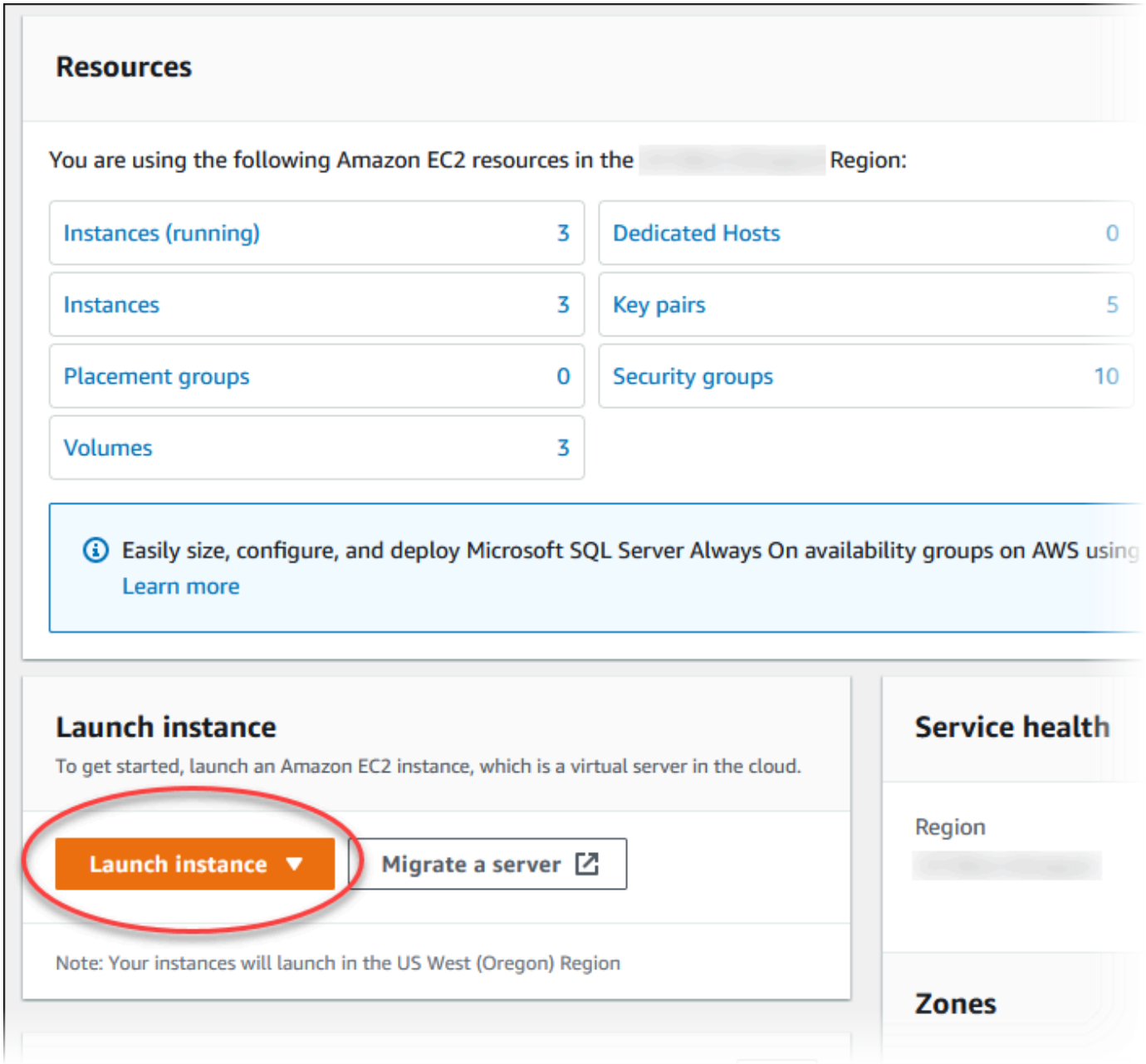
> **Note**
>
> Nach Abschluss des Tutorials gibt es in jeder Availability Zone im VPC ein öffentliches und ein privates Subnetz. Dieses Tutorial verwendet die Standard-VPC für Ihr AWS-Kontound richtet automatisch die Verbindung zwischen Ihrer EC2-Instance und der DB-Instance ein. Wenn Sie stattdessen lieber eine neue VPC für dieses Szenario konfigurieren möchten, führen Sie die Aufgaben in Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur) aus.

Starten Sie eine EC2 Instance, um eine Verbindung mit Ihrer herzustellen

Erstellen Sie eine EC2 Amazon-Instance im öffentlichen Subnetz IhresVPC.

Starten Sie eine EC2-Instance wie folgt:

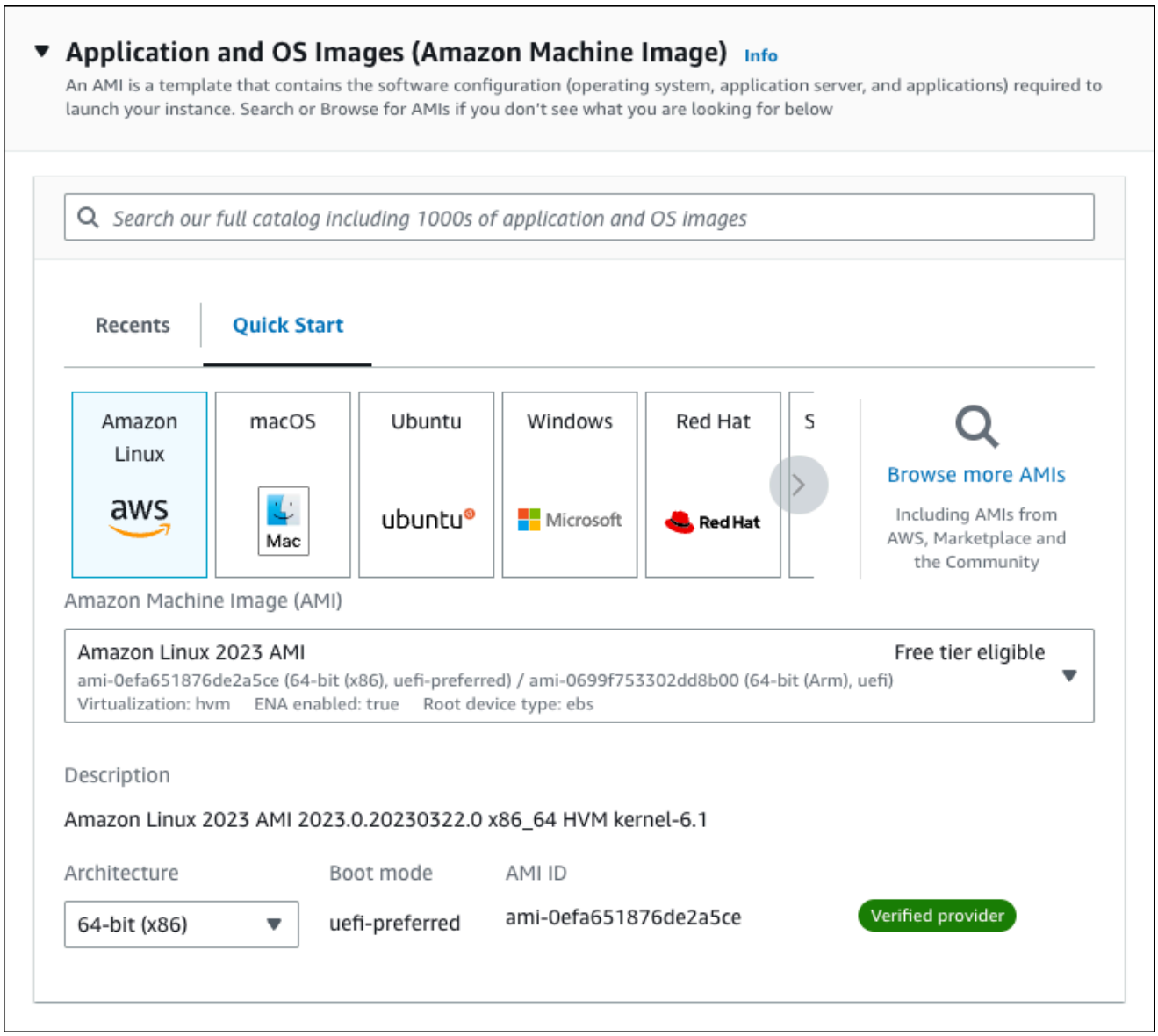
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie in der oberen rechten Ecke von den aus AWS Management Console, AWS-Region wo Sie die Instance erstellen möchten. EC2
3. Wählen Sie EC2Dashboard und dann Launch instance aus, wie im Folgenden gezeigt.



4. Wählen Sie auf der Seite Eine Instance starten die folgenden Einstellungen aus.

   a. Geben Sie unter Name and tags (Name und Tags) als Name den Namen **`tutorial-ec2-instance-web-server`** ein.

   b. Wählen Sie unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image) Amazon Linux und dann Amazon Linux 2023 ausAMI. Übernehmen Sie für alle anderen Einstellungen die Standardwerte.



c. Wählen Sie unter Instance type (Instance-Typ) den Wert t2.micro aus.

d. Wählen Sie unter Key pair (login) (Schlüsselpaar (Anmeldung)) einen Key pair name (Schlüsselpaarname), um ein vorhandenes Schlüsselpaar zu verwenden. Um ein neues key pair für die EC2 Amazon-Instance zu erstellen, wählen Sie Neues key pair erstellen und verwenden Sie dann das Fenster key pair erstellen, um es zu erstellen.

   Weitere Informationen zum Erstellen eines neuen key pair finden Sie unter Erstellen eines key pair im EC2Amazon-Benutzerhandbuch.

e. Legen Sie unter Network settings (Netzwerkeinstellungen) die folgenden Werte fest und übernehmen Sie für die anderen Einstellungen die Standardwerte:

- Wählen Sie für Allow SSH traffic from die Quelle der SSH Verbindungen zur EC2 Instance aus.

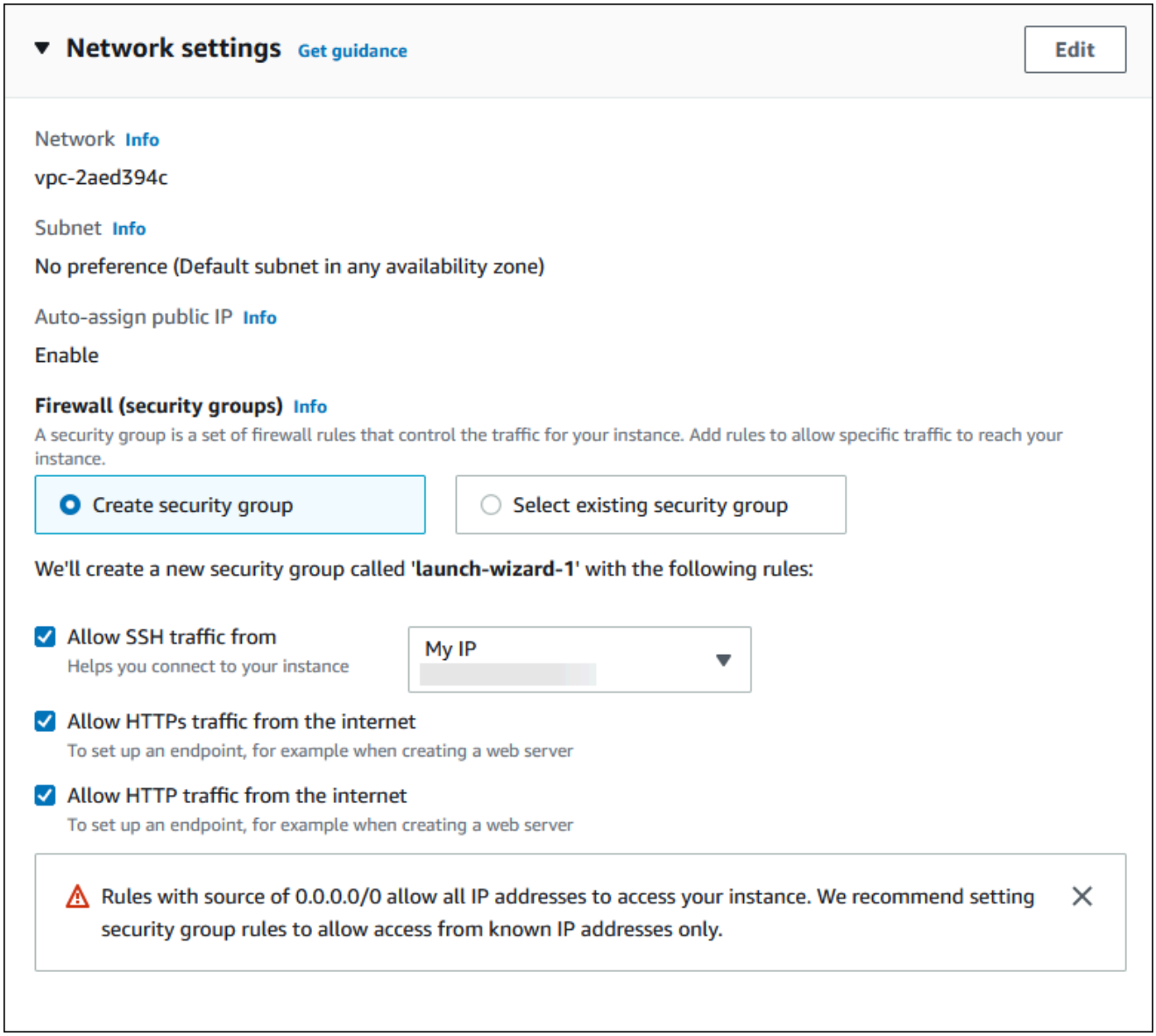
  Sie können Meine IP wählen, wenn die angezeigte IP-Adresse für SSH Verbindungen korrekt ist.

  Andernfalls können Sie mithilfe von Secure Shell (SSH) die IP-Adresse ermitteln, die für die Verbindung zu Ihren EC2 Instances VPC verwendet werden soll. Um Ihre öffentliche IP-Adresse zu ermitteln, können Sie in einem anderen Browserfenster oder einer anderen Registerkarte den Service unter https://checkip.amazonaws.com verwenden. Ein Beispiel für eine IP-Adresse ist `203.0.113.25/32`.

  In vielen Fällen stellen Sie möglicherweise eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse her. Bestimmen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

  > **Warning**
  >
  > Wenn Sie `0.0.0.0/0` die Option für SSH den Zugriff verwenden, ermöglichen Sie allen IP-Adressen den Zugriff auf Ihre öffentlichen Instances überSSH. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in der Produktionsumgebung nur eine bestimmte IP-Adresse oder einen bestimmten Adressbereich für den Zugriff auf Ihre InstancesSSH.

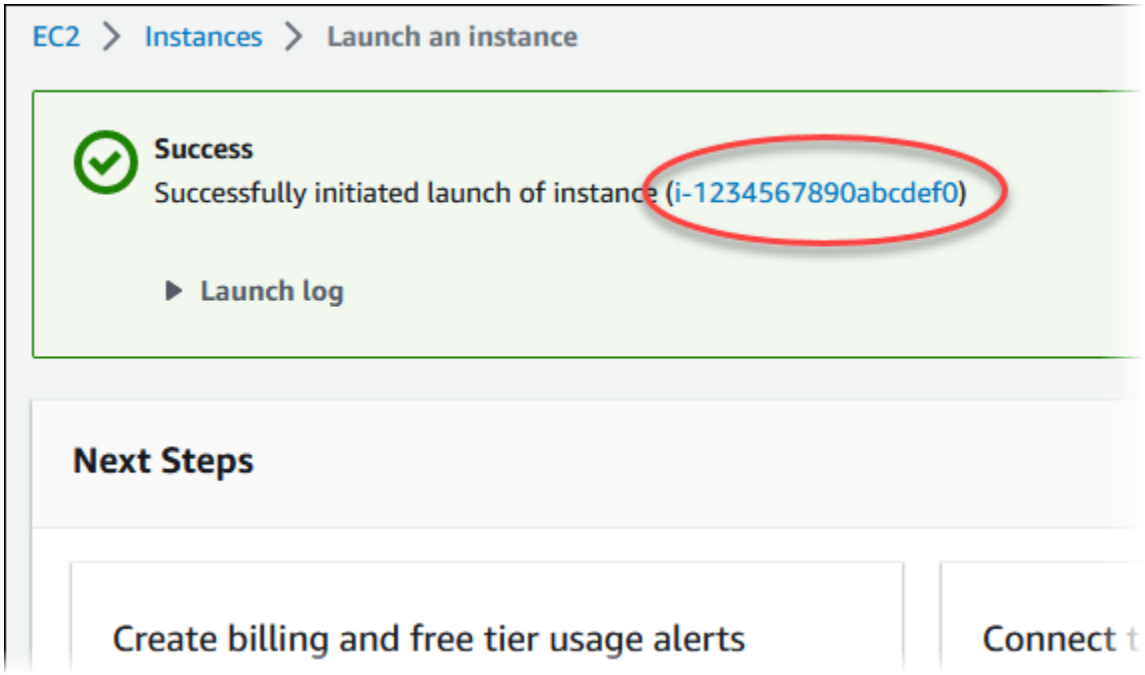
- Aktivieren Sie die Option „HTTPsDatenverkehr aus dem Internet zulassen“.
- Aktivieren Sie die Option HTTP Datenverkehr aus dem Internet zulassen.



f. Übernehmen Sie die Standardwerte für die übrigen Abschnitte.

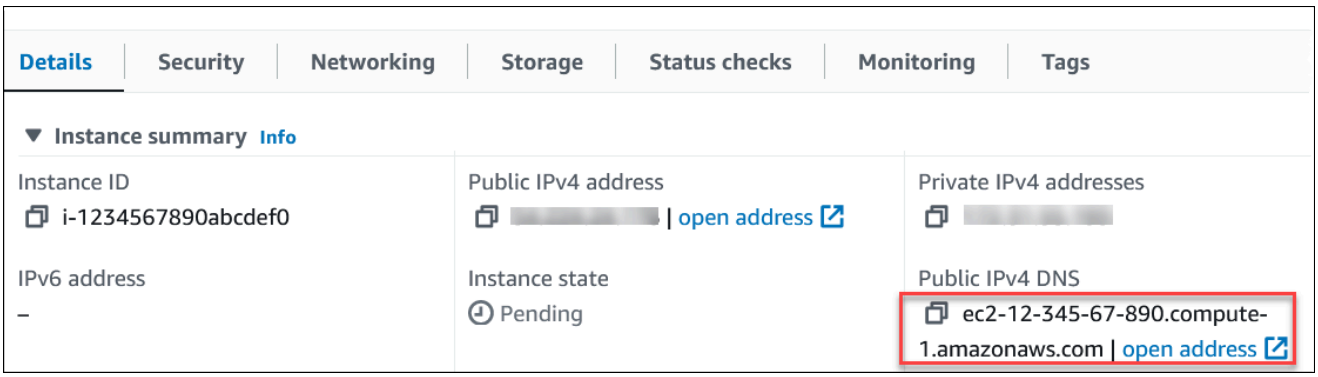
g. Überprüfen Sie Ihre Instance-Konfiguration im Bereich Summary (Übersicht). Wenn alles in Ordnung ist, klicken Sie auf Launch instance (Instance starten).

5. Notieren Sie sich auf der Seite Startstatus die ID für Ihre neue EC2 Instance, zum Beispiel:`i-1234567890abcdef0`.

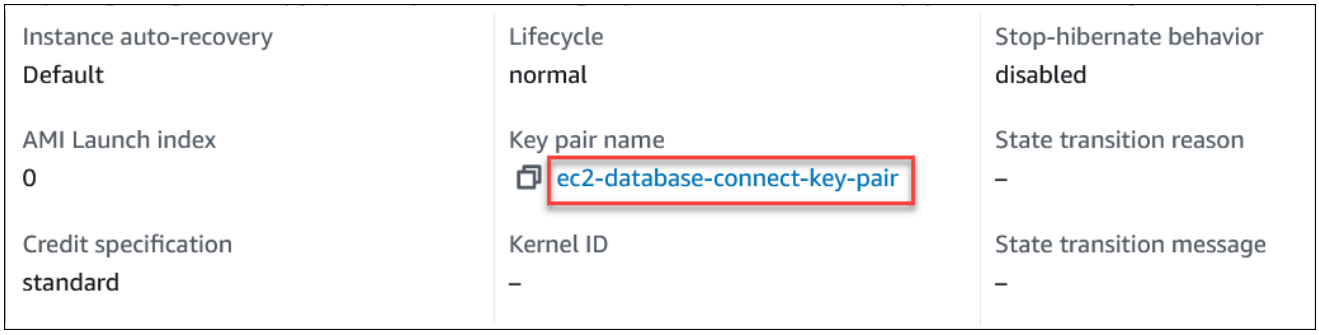


6. Wählen Sie die EC2 Instance-ID, um die Liste der EC2 Instances zu öffnen, und wählen Sie dann Ihre EC2 Instance aus.
7. Notieren Sie sich auf der Registerkarte „Details“ die folgenden Werte, die Sie benötigen, wenn Sie eine Verbindung herstellen mitSSH:

   a. Notieren Sie sich unter Instanzübersicht den Wert für Öffentlich IPv4 DNS.



   b. Notieren Sie sich unter Instance-Details den Wert für Schlüsselpaarname.



8. Warten Sie, bis der Instance state (Instance-Status) für Ihre Instance als Running (Wird ausgeführt) angezeigt wird, bevor Sie fortfahren.
9. Schließen Sie Erstellen einer DB-Instance von Amazon RDS ab.

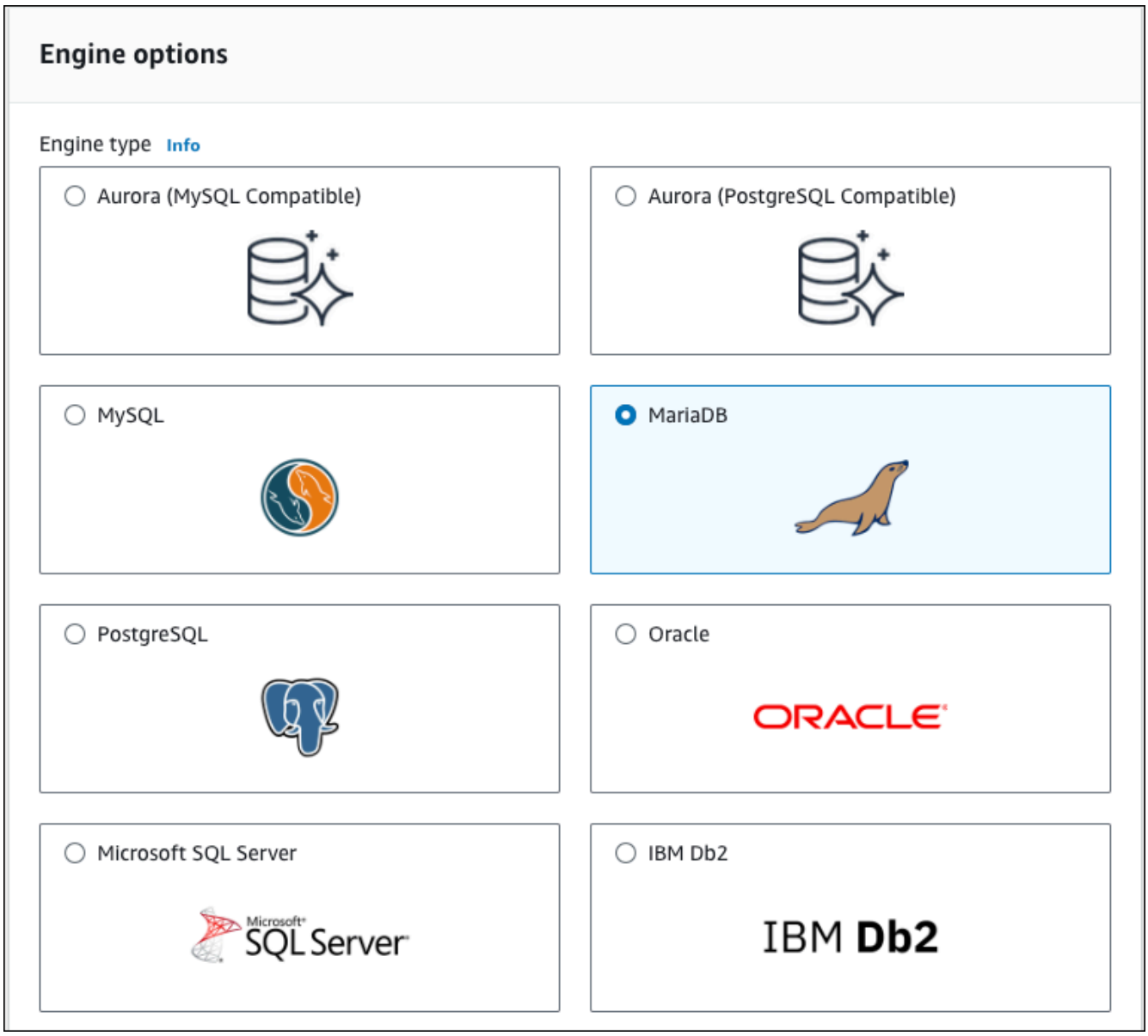
Erstellen einer DB-Instance von Amazon RDS

Erstellen Sie eine DB-Instance von RDS für MariaDB, RDS für MySQL oder RDS für PostgreSQL, die die von einer Webanwendung verwendeten Daten enthält.

RDS for MariaDB

So erstellen Sie eine MariaDB-Instance

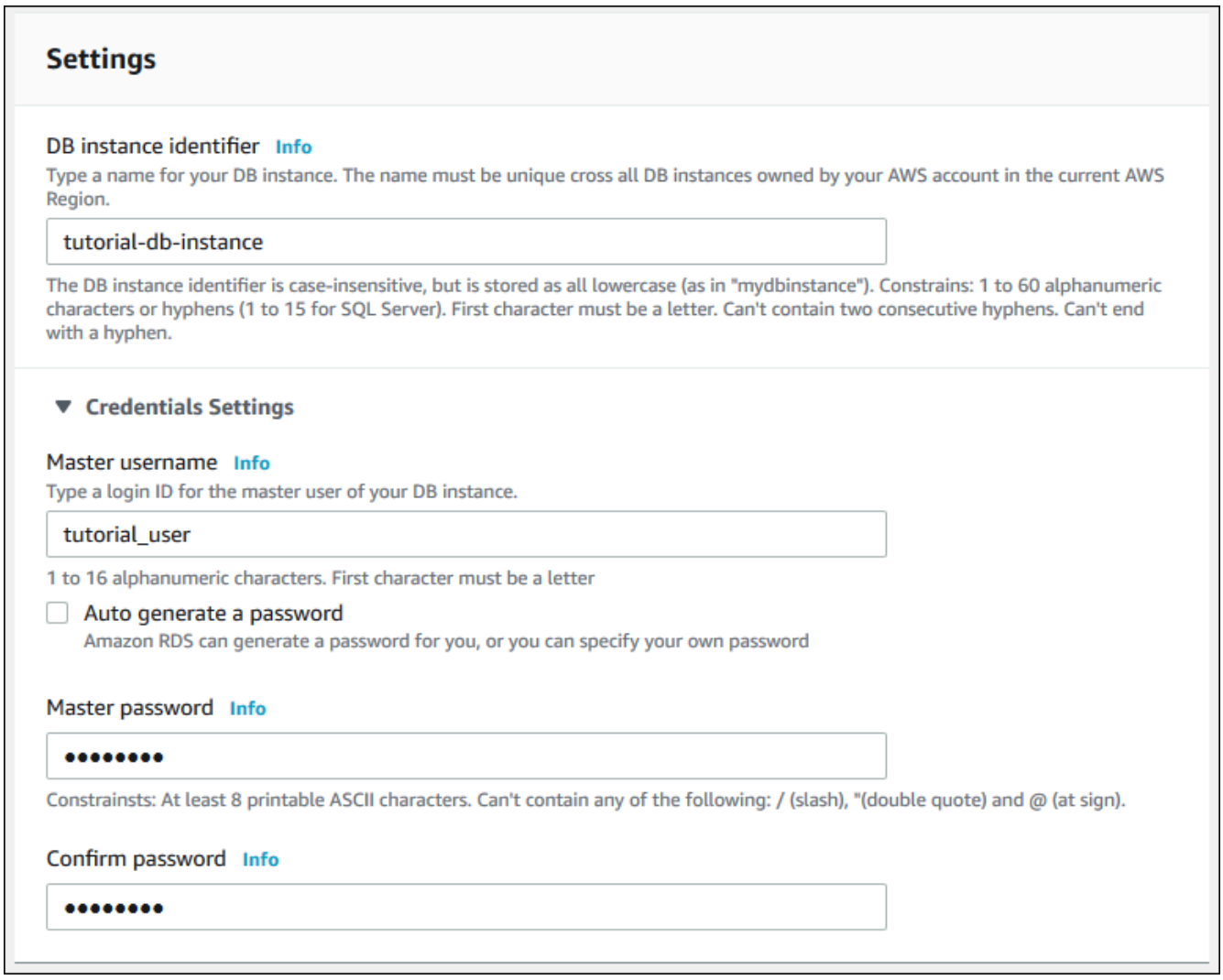
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie in der oberen rechten Ecke von auf AWS Management Console. AWS-Region Es sollte mit dem identisch sein, in dem Sie Ihre EC2 Instanz erstellt haben.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.
5. Wählen Sie auf der Seite Datenbank erstellen die Option Standarderstellung aus.
6. Wählen Sie für Engine options (Engine-Optionen) die Option MariaDB aus.



7. Wählen Sie für Vorlagen das kostenlose Kontingent oder Sandbox. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.

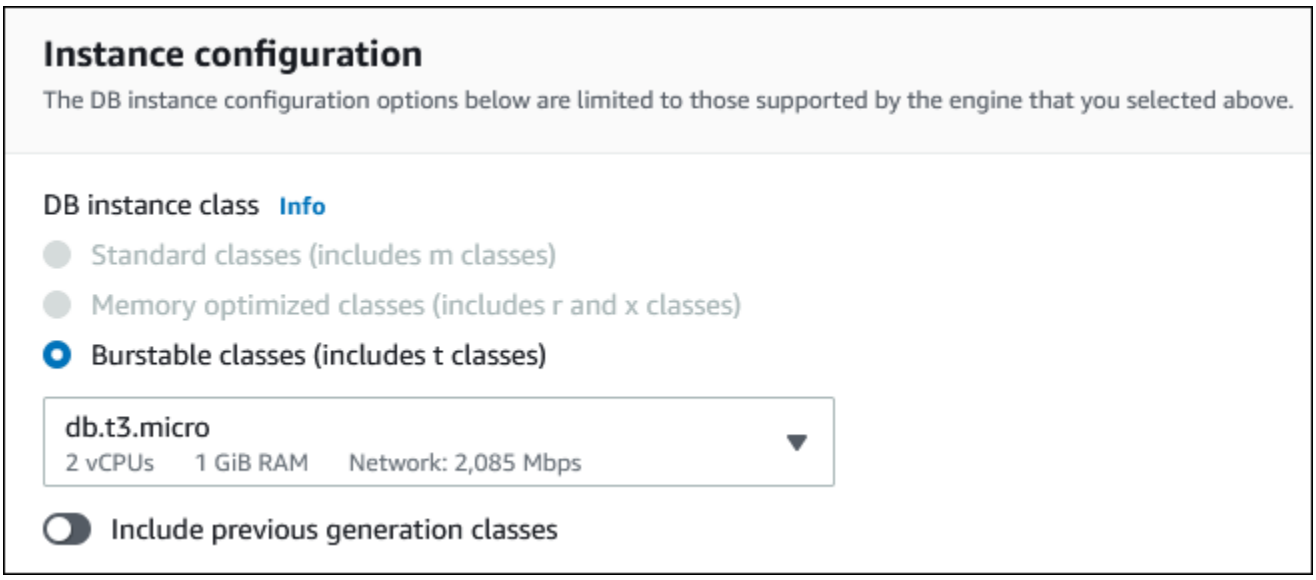


8. Verwenden Sie im Abschnitt Availability & durability (Verfügbarkeit und Stabilität) die Standardwerte.
9. Legen Sie im Abschnitt Settings (Einstellungen) die folgenden Werte fest:
   - DB Instance Identifier (DB-Instance-Kennung – Typ **`tutorial-db-instance`**.
   - Master username (Masterbenutzername) – Typ **`tutorial_user`**.
   - Automatisch ein Passwort generieren - Lassen Sie die Option ausgeschaltet.
   - Master-Passwort - Geben Sie ein Passwort ein.
   - Confirm password (Passwort bestätigen): Geben Sie das Passwort erneut ein.



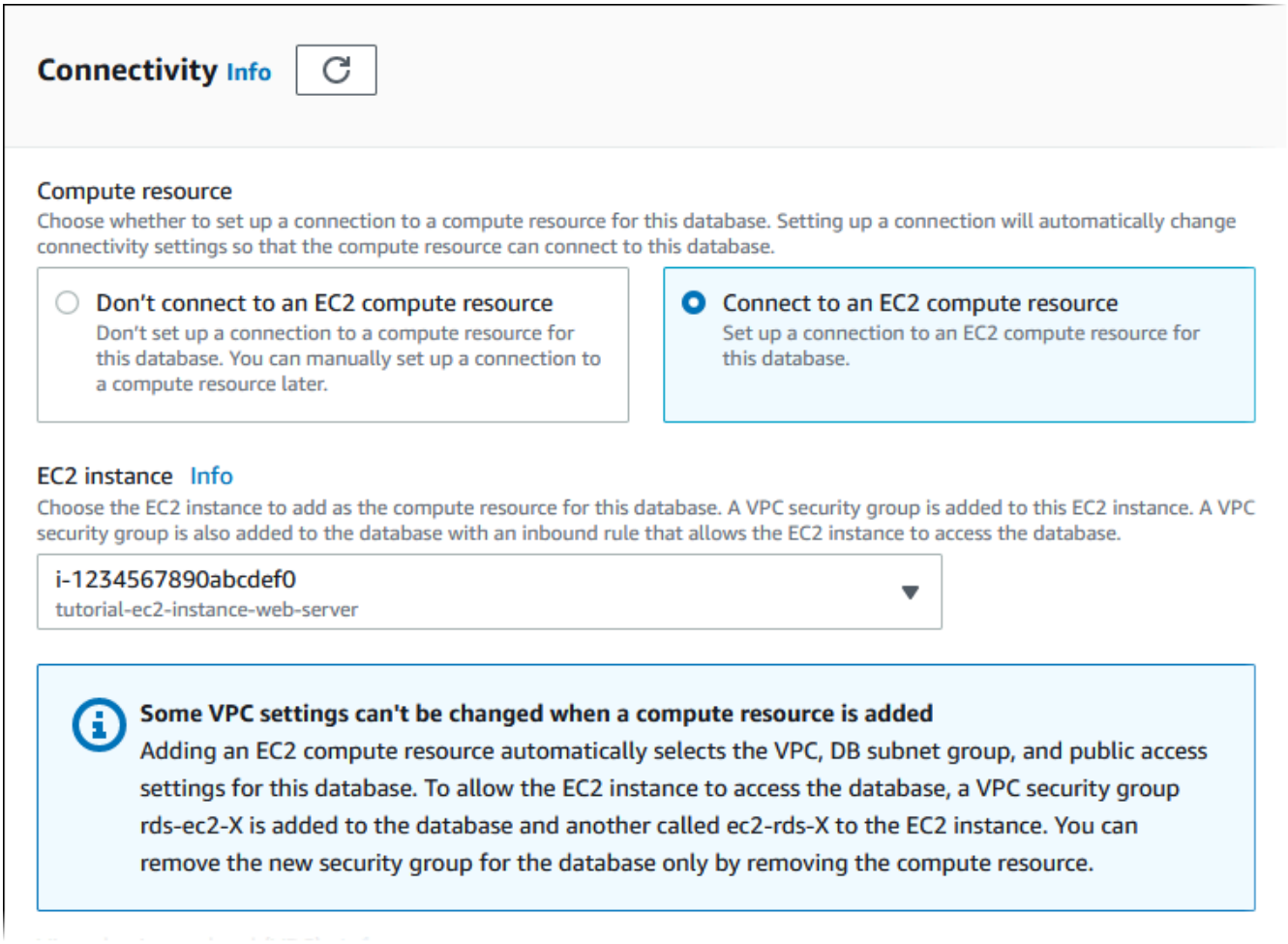
10. Legen Sie im Abschnitt Instance Configuration folgende Werte fest:

    - Burst-fähige Klassen (einschließlich t-Klassen)
    - db.t3.micro



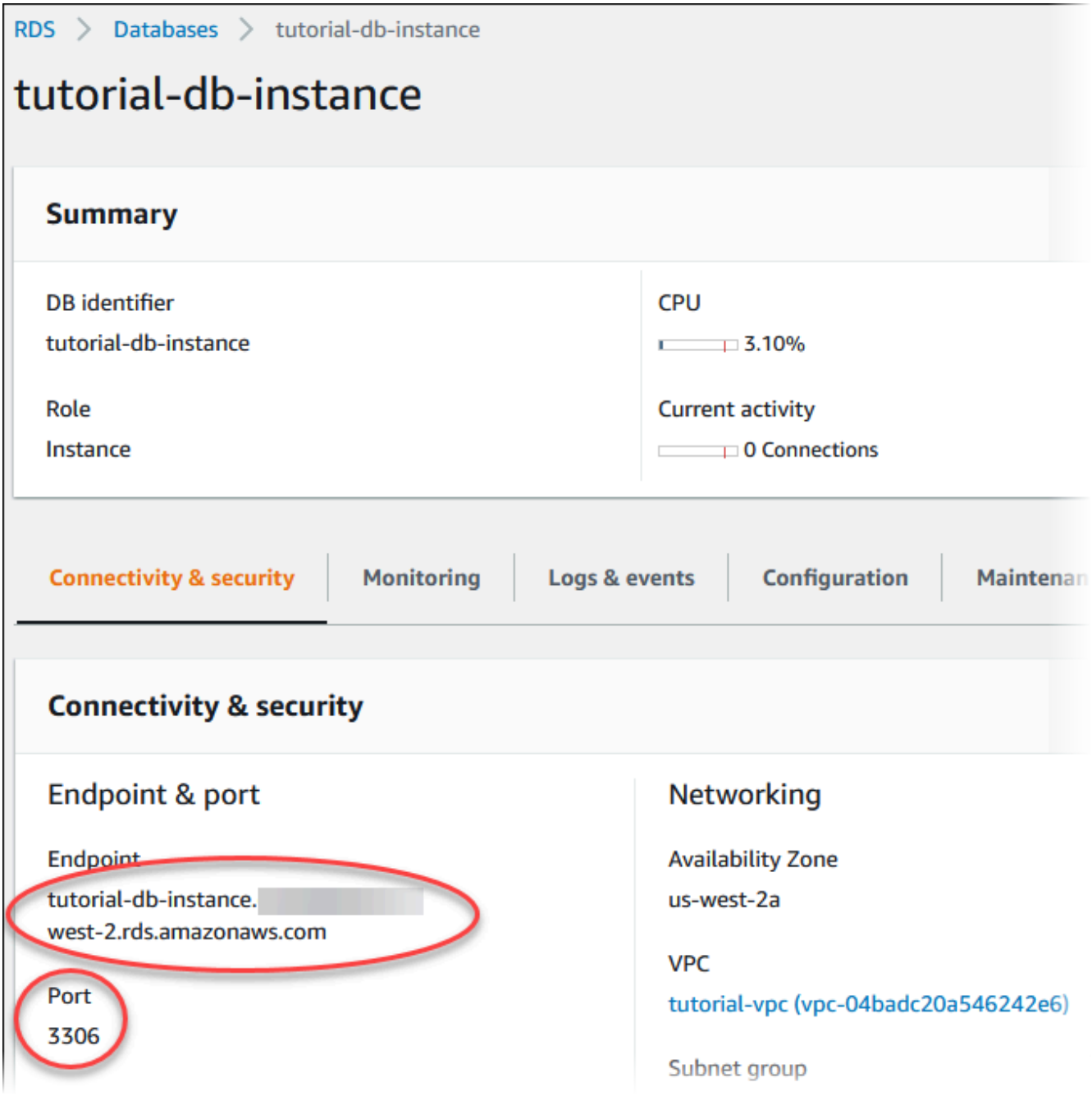
11. Im Abschnitt Speicher behalten Sie die Standardwerte eingestellt.
12. Im Abschnitt Konnektivität legen Sie die folgenden Werte fest und behalten Sie die Standardwerte für die anderen Werte bei:

    - Wählen Sie für Rechenressource die Option Mit einer EC2 Rechenressource Connect aus.
    - Wählen Sie beispielsweise die EC2 Instanz aus, die EC2 Sie zuvor erstellt haben, z. B. tutorial-ec2 -. instance-web-server



13. Stellen Sie im Abschnitt Datenbankauthentifizierung sicher, dass Passwortauthentifizierung ausgewählt ist.
14. Öffnen Sie den Abschnitt Additional configuration (Zusätzliche Konfiguration) und geben Sie **`sample`** für Initial database name (Erster Datenbankname) ein. Behalten Sie für die anderen Optionen die Standardeinstellungen bei.
15. Um Ihre MariaDB-Instance zu erstellen, wählen Sie Create database (Datenbank erstellen) aus.

    Ihre neue DB-Instance wird in der Liste Databases (Datenbanken) mit dem Status Creating (Wird erstellt) angezeigt.
16. Warten Sie, bis der Status Ihrer neuen DB-Instance als Available (Verfügbar) angezeigt wird. Wählen Sie dann den Namen der DB-Instance aus, um deren Details anzuzeigen.
17. Zeigen Sie im Abschnitt Connectivity & security (Anbindung und Sicherheit) den Endpoint (Endpunkt) und den Port der DB-Instance an.



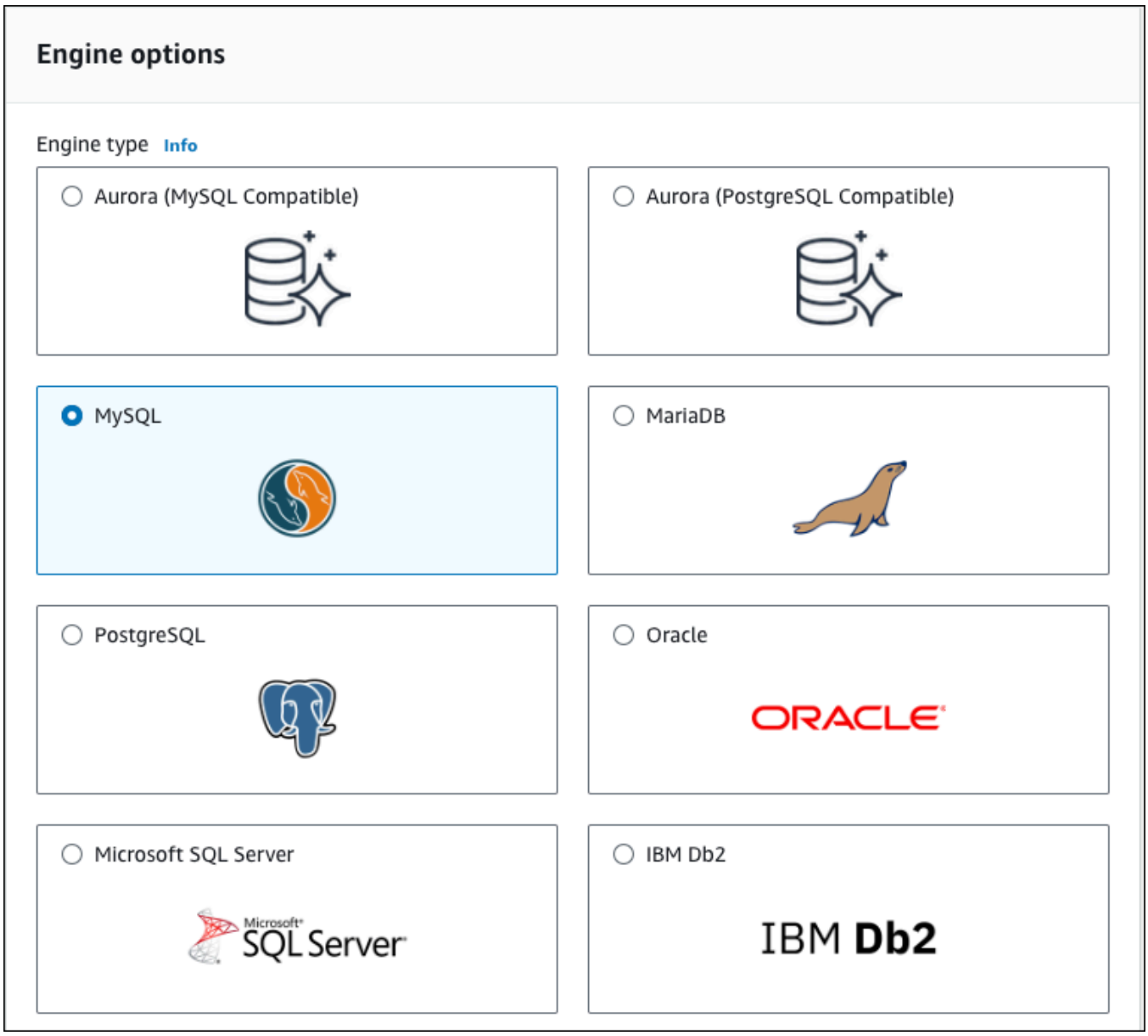
Notieren Sie den Endpunkt und den Port Ihrer DB-Instance. Sie verwenden diese Informationen, um Ihren Webserver mit Ihrer DB-Instance zu verbinden.

18. Schließen Sie Installieren eines Webservers auf Ihrer EC2-Instance ab.

RDS for MySQL

So erstellen Sie eine MySQL-DB-Instance

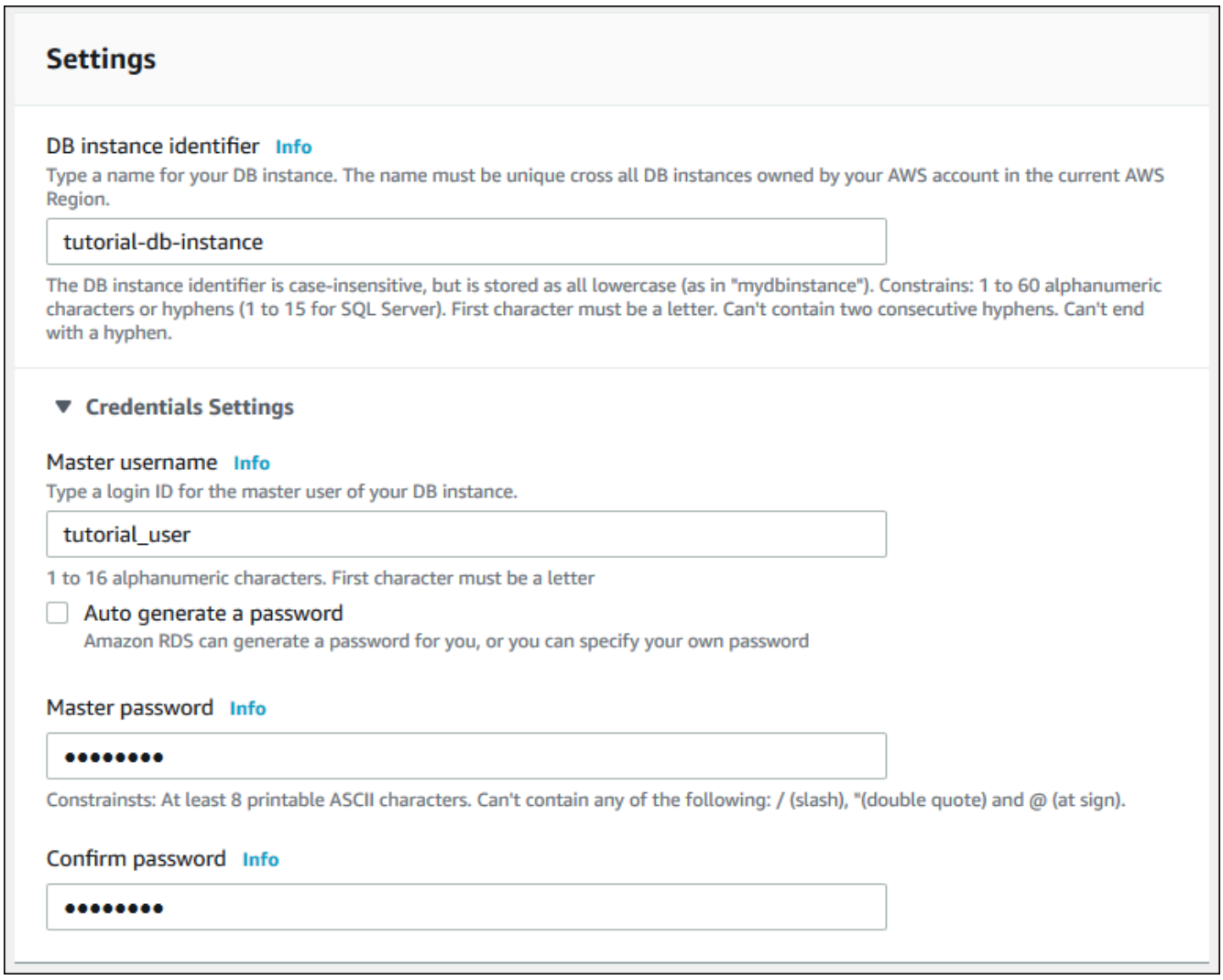
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie in der oberen rechten Ecke von auf AWS Management Console. AWS-Region Es sollte mit dem identisch sein, in dem Sie Ihre EC2 Instanz erstellt haben.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.
5. Wählen Sie auf der Seite Datenbank erstellen die Option Standarderstellung aus.
6. Wählen Sie unter Engine-Optionen die Option MySQL aus.



7. Wählen Sie für Vorlagen das kostenlose Kontingent oder Sandbox. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.

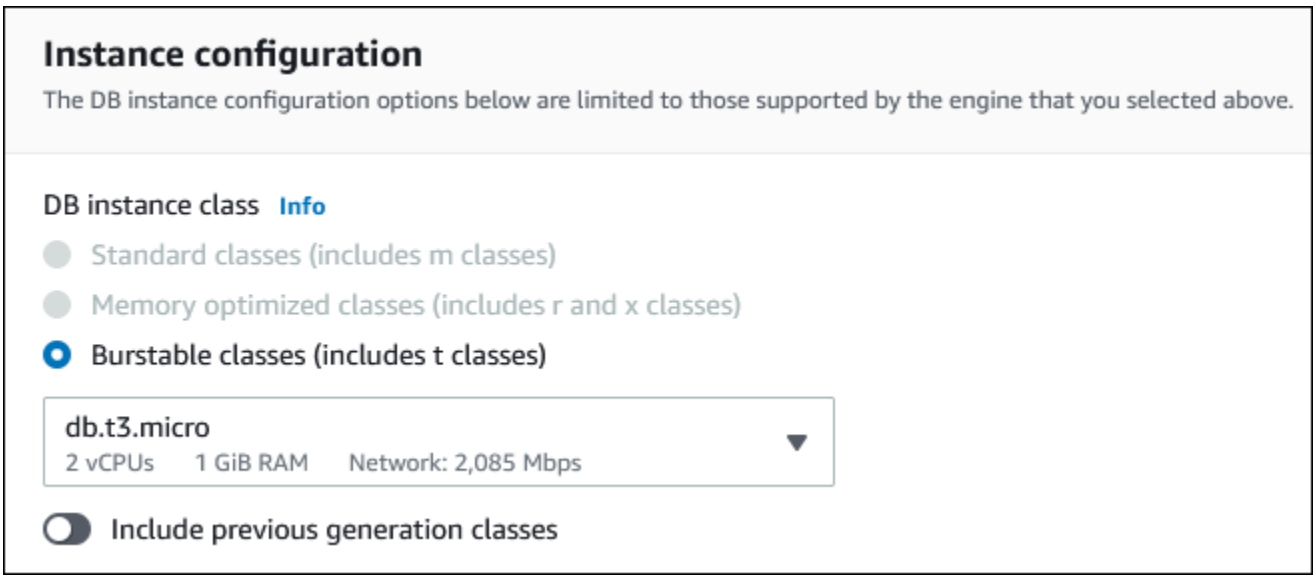


8. Verwenden Sie im Abschnitt Availability & durability (Verfügbarkeit und Stabilität) die Standardwerte.
9. Legen Sie im Abschnitt Settings (Einstellungen) die folgenden Werte fest:
   - DB Instance Identifier (DB-Instance-Kennung – Typ **`tutorial-db-instance`**.
   - Master username (Masterbenutzername) – Typ **`tutorial_user`**.
   - Automatisch ein Passwort generieren - Lassen Sie die Option ausgeschaltet.
   - Master-Passwort - Geben Sie ein Passwort ein.
   - Confirm password (Passwort bestätigen): Geben Sie das Passwort erneut ein.



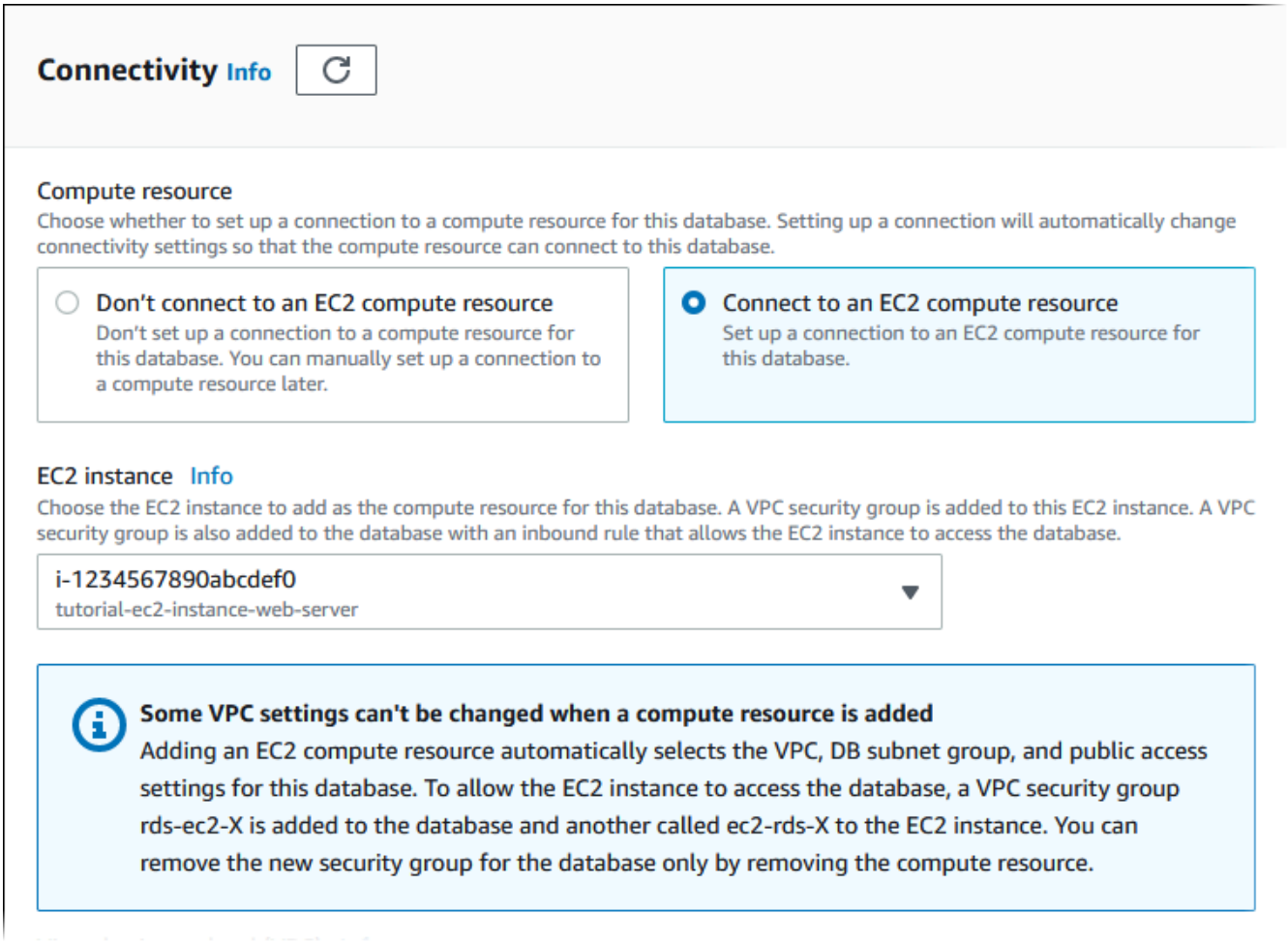
10. Legen Sie im Abschnitt Instance Configuration folgende Werte fest:

    - Burst-fähige Klassen (einschließlich t-Klassen)
    - db.t3.micro



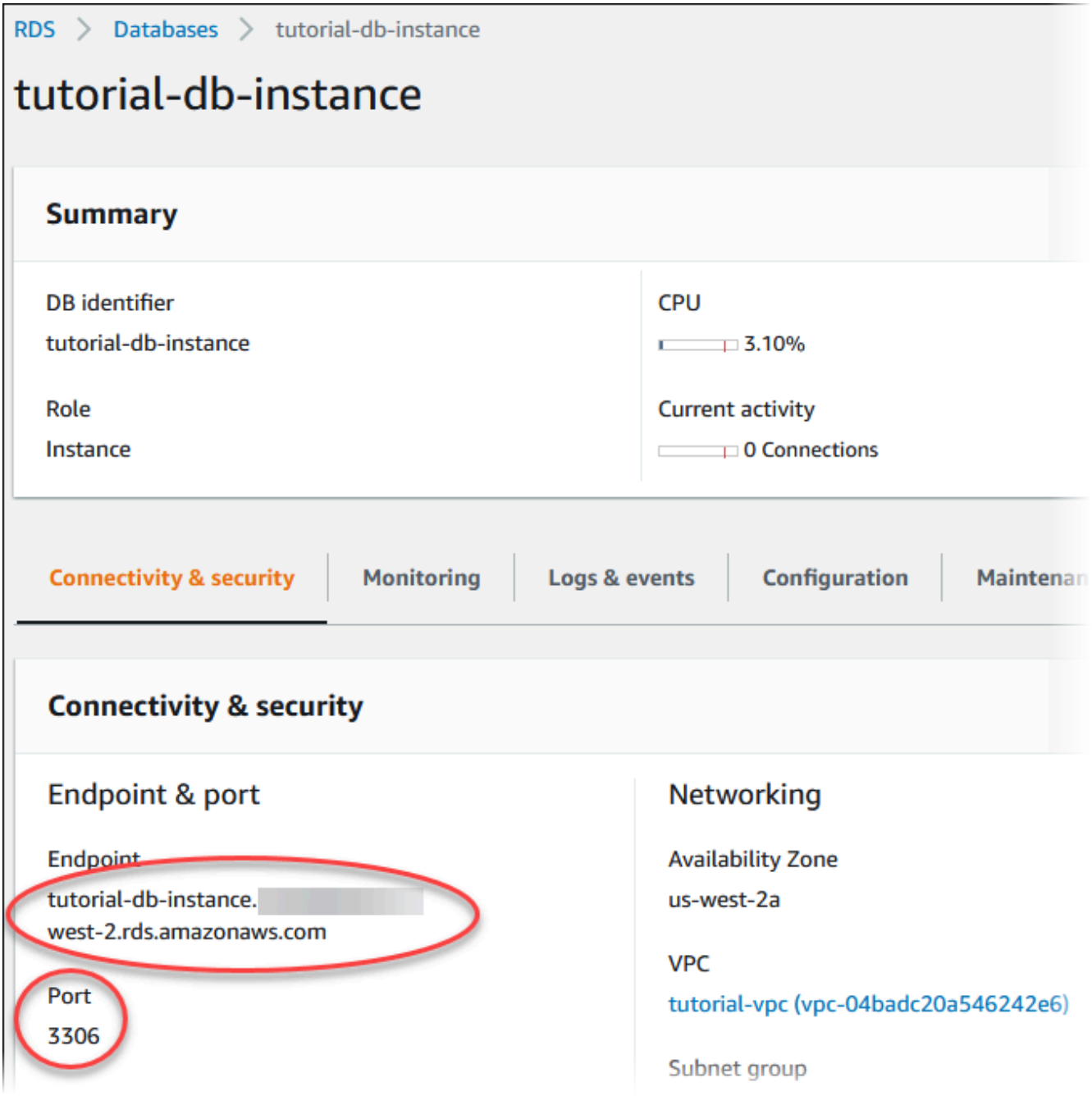
11. Im Abschnitt Speicher behalten Sie die Standardwerte eingestellt.
12. Im Abschnitt Konnektivität legen Sie die folgenden Werte fest und behalten Sie die Standardwerte für die anderen Werte bei:

    - Wählen Sie für Rechenressource die Option Mit einer EC2 Rechenressource Connect aus.
    - Wählen Sie beispielsweise die EC2 Instanz aus, die EC2 Sie zuvor erstellt haben, z. B. tutorial-ec2 -. instance-web-server



13. Stellen Sie im Abschnitt Datenbankauthentifizierung sicher, dass Passwortauthentifizierung ausgewählt ist.
14. Öffnen Sie den Abschnitt Additional configuration (Zusätzliche Konfiguration) und geben Sie **`sample`** für Initial database name (Erster Datenbankname) ein. Behalten Sie für die anderen Optionen die Standardeinstellungen bei.
15. Um Ihre MySQL-DB-Instance zu erstellen, wählen Sie Create database (Datenbank erstellen) aus.

    Ihre neue DB-Instance wird in der Liste Databases (Datenbanken) mit dem Status Creating (Wird erstellt) angezeigt.
16. Warten Sie, bis der Status Ihrer neuen DB-Instance als Available (Verfügbar) angezeigt wird. Wählen Sie dann den Namen der DB-Instance aus, um deren Details anzuzeigen.
17. Zeigen Sie im Abschnitt Connectivity & security (Anbindung und Sicherheit) den Endpoint (Endpunkt) und den Port der DB-Instance an.



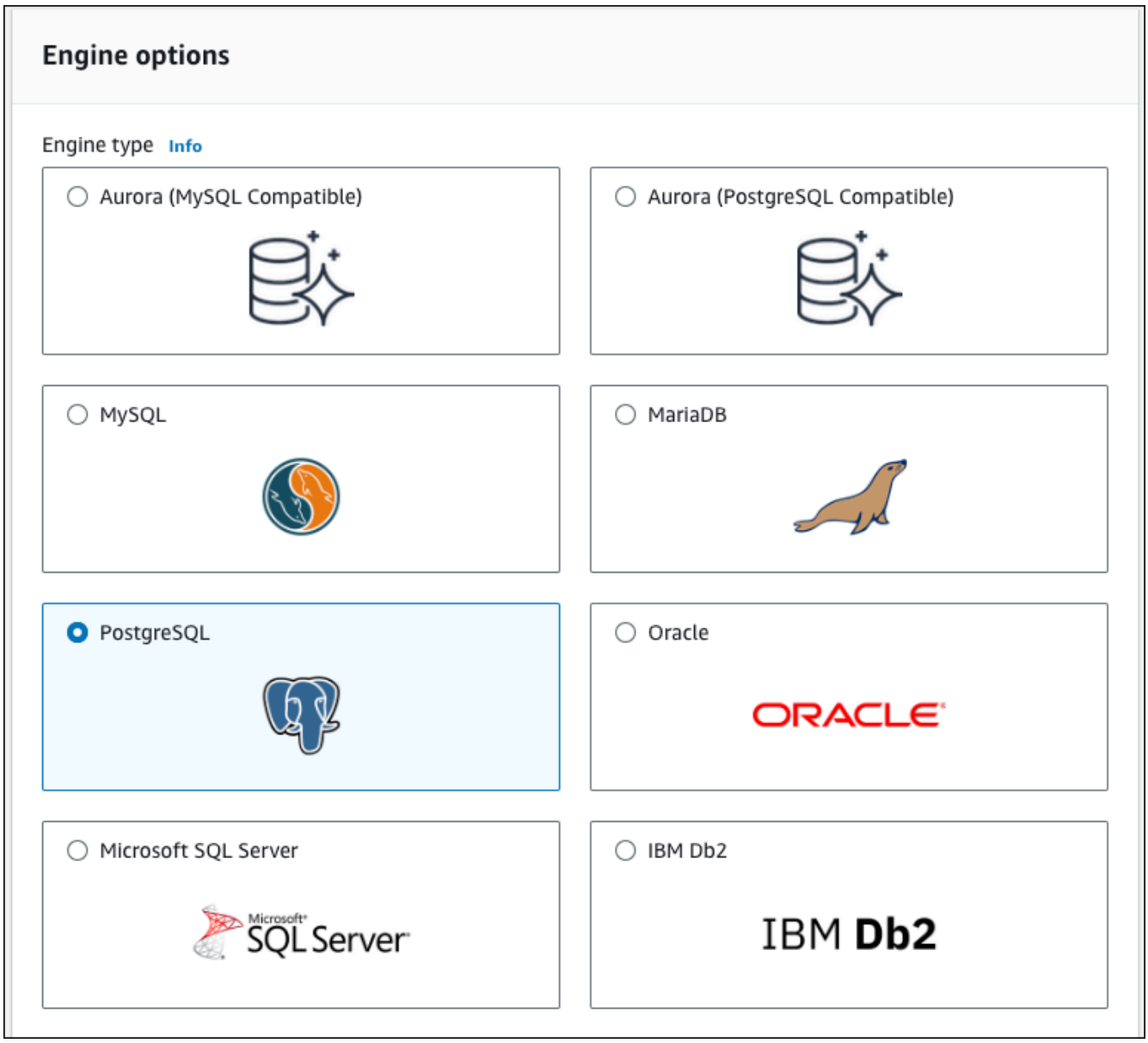
Notieren Sie den Endpunkt und den Port Ihrer DB-Instance. Sie verwenden diese Informationen, um Ihren Webserver mit Ihrer DB-Instance zu verbinden.

18. Schließen Sie Installieren eines Webservers auf Ihrer EC2-Instance ab.

RDS for PostgreSQL

Erstellen einer PostgreSQL-DB-Instance

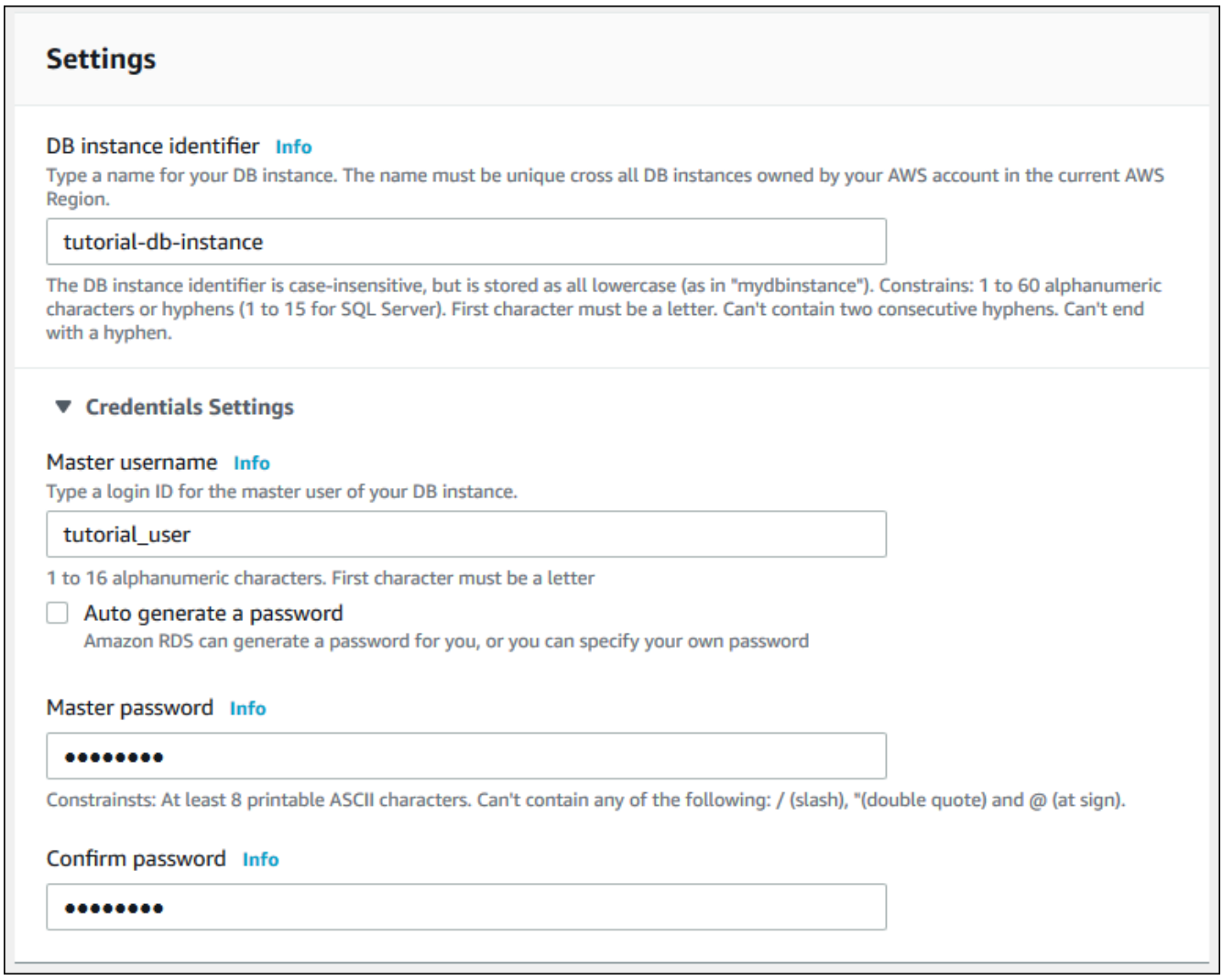
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie in der oberen rechten Ecke von auf AWS Management Console. AWS-Region Es sollte mit dem identisch sein, in dem Sie Ihre EC2 Instanz erstellt haben.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.
5. Wählen Sie auf der Seite Datenbank erstellen die Option Standarderstellung aus.
6. Wählen Sie unter Engine-Optionen die Option PostgreSQL.



7. Wählen Sie für Vorlagen das kostenlose Kontingent oder Sandbox. Das kostenlose Kontingent wird für Konten mit kostenlosem Tarif angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.

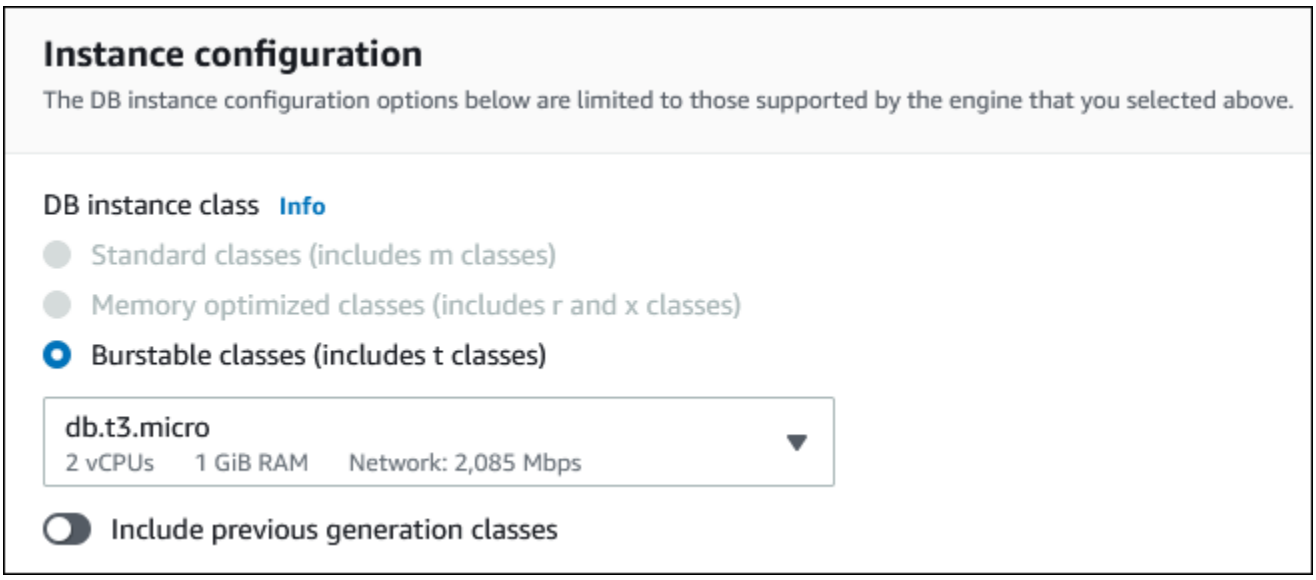


8. Verwenden Sie im Abschnitt Availability & durability (Verfügbarkeit und Stabilität) die Standardwerte.
9. Legen Sie im Abschnitt Settings (Einstellungen) die folgenden Werte fest:
   - DB Instance Identifier (DB-Instance-Kennung – Typ **`tutorial-db-instance`**.
   - Master username (Masterbenutzername) – Typ **`tutorial_user`**.
   - Automatisch ein Passwort generieren - Lassen Sie die Option ausgeschaltet.
   - Master-Passwort - Geben Sie ein Passwort ein.
   - Confirm password (Passwort bestätigen): Geben Sie das Passwort erneut ein.



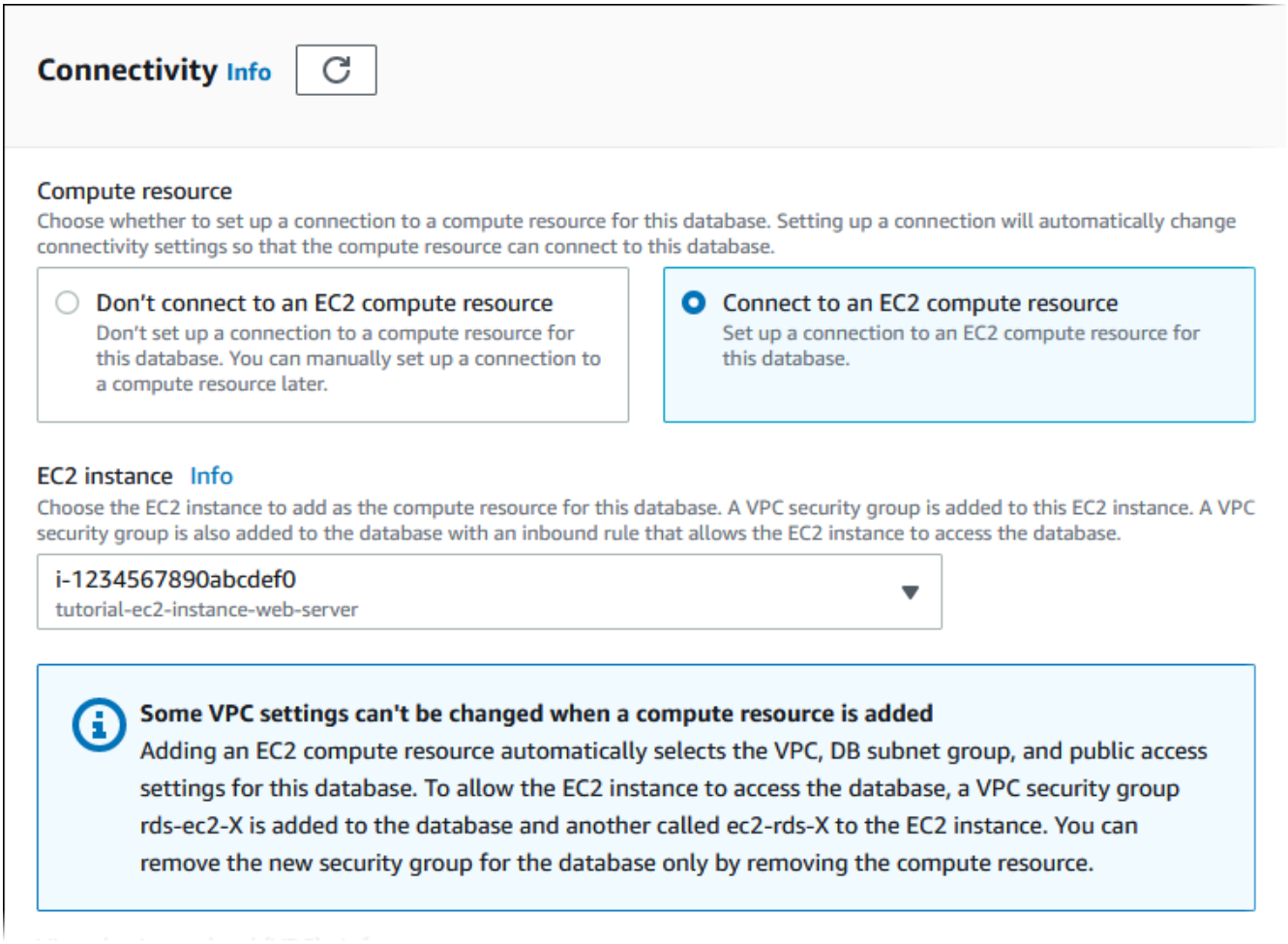
10. Legen Sie im Abschnitt Instance Configuration folgende Werte fest:

    - Burst-fähige Klassen (einschließlich t-Klassen)
    - db.t3.micro



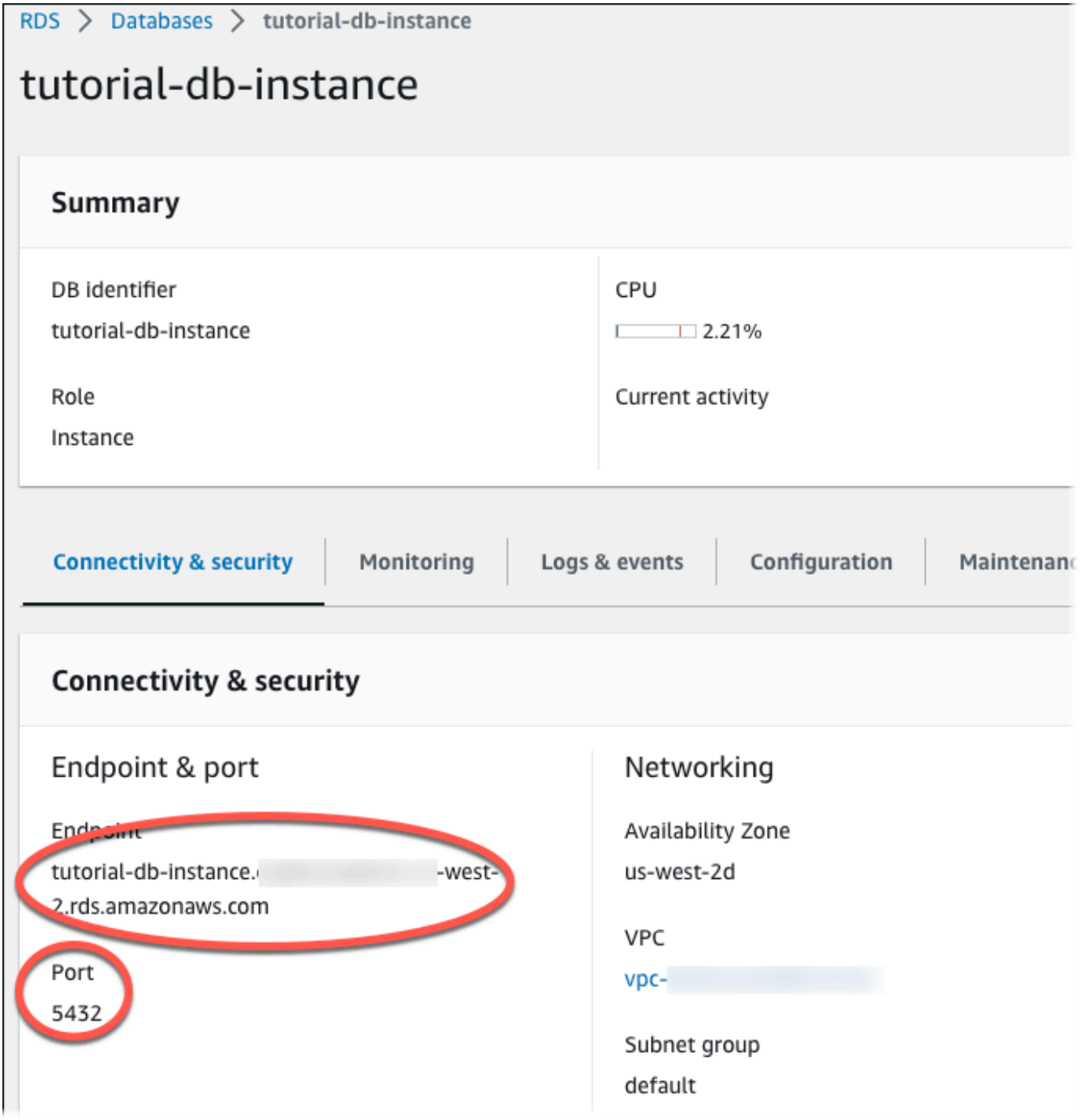
11. Im Abschnitt Speicher behalten Sie die Standardwerte eingestellt.
12. Im Abschnitt Konnektivität legen Sie die folgenden Werte fest und behalten Sie die Standardwerte für die anderen Werte bei:

    - Wählen Sie für Rechenressource die Option Mit einer EC2 Rechenressource Connect aus.
    - Wählen Sie beispielsweise die EC2 Instanz aus, die EC2 Sie zuvor erstellt haben, z. B. tutorial-ec2 -. instance-web-server



13. Stellen Sie im Abschnitt Datenbankauthentifizierung sicher, dass Passwortauthentifizierung ausgewählt ist.
14. Öffnen Sie den Abschnitt Additional configuration (Zusätzliche Konfiguration) und geben Sie **`sample`** für Initial database name (Erster Datenbankname) ein. Behalten Sie für die anderen Optionen die Standardeinstellungen bei.
15. Um Ihre PostgreSQL-DB-Instance zu erstellen, wählen Sie Datenbank erstellen aus.

    Ihre neue DB-Instance wird in der Liste Databases (Datenbanken) mit dem Status Creating (Wird erstellt) angezeigt.
16. Warten Sie, bis der Status Ihrer neuen DB-Instance als Available (Verfügbar) angezeigt wird. Wählen Sie dann den Namen der DB-Instance aus, um deren Details anzuzeigen.
17. Zeigen Sie im Abschnitt Connectivity & security (Anbindung und Sicherheit) den Endpoint (Endpunkt) und den Port der DB-Instance an.



Notieren Sie den Endpunkt und den Port Ihrer DB-Instance. Sie verwenden diese Informationen, um Ihren Webserver mit Ihrer DB-Instance zu verbinden.

18. Schließen Sie Installieren eines Webservers auf Ihrer EC2-Instance ab.

Installieren eines Webservers auf Ihrer EC2-Instance

Installieren Sie einen Webserver auf der EC2-Instance, die Sie in Starten Sie eine EC2 Instance, um eine Verbindung mit Ihrer herzustellen erstellt haben. Der Webserver stellt eine Verbindung mit der DB-Instance von Amazon RDS her, die Sie in Erstellen einer DB-Instance von Amazon RDS erstellt haben.

Installieren eines Apache-Webservers mit PHP und MariaDB

Stellen Sie eine Verbindung mit Ihrer EC2-Instance her und installieren Sie den Webserver.

So stellen Sie eine Verbindung mit Ihrer EC2-Instance her und installieren den Apache-Webserver mit PHP

1. Stellen Sie eine Verbindung zu der EC2-Instance her, die Sie zuvor erstellt haben, indem Sie den Schritten unter Connect to your Linux Instance im Amazon EC2 EC2-Benutzerhandbuch folgen.

   Wir empfehlen, dass Sie eine Verbindung mit Ihrer EC2-Instance mithilfe von SSH herstellen. Wenn das SSH-Client-Dienstprogramm unter Windows, Linux oder Mac installiert ist, können Sie mit dem folgenden Befehlsformat eine Verbindung mit der Instance herstellen:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Nehmen wir zum Beispiel an, das `ec2-database-connect-key-pair.pem` in `/dir1` unter Linux gespeichert und das öffentliche IPv4-DNS für Ihre EC2-Instance `ec2-12-345-678-90.compute-1.amazonaws.com` ist. Ihr SSH-Befehl würde wie folgt aussehen:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

2. Installieren Sie die neuesten Fehlerbehebungen und Sicherheitsupdates, indem Sie die Software auf Ihrer EC2-Instance aktualisieren. Verwenden Sie dazu den folgenden Befehl.

   > **Note**
   >
   > Mit der Option `-y` werden die Updates installiert, ohne um Bestätigung zu bitten. Um Updates vor der Installation zu überprüfen, lassen Sie diese Option aus.

```
sudo dnf update -y
```

3. Nachdem die Aktualisierungen abgeschlossen sind, installieren Sie die Software von Apache-Webserver, PHP und MariaDB mit den folgenden Befehlen. Mit diesem Befehl werden gleichzeitig mehrere Softwarepakete und zugehörige Abhängigkeiten installiert.

   MariaDB & MySQL

   ```
   sudo dnf install -y httpd php php-mysqli mariadb105
   ```

   PostgreSQL

   ```
   sudo dnf install -y httpd php php-pgsql postgresql15
   ```

   Wenn Sie eine Fehlermeldung erhalten, wurde Ihre Instance wahrscheinlich nicht mit einem Amazon-Linux-2023-AMI gestartet. Stattdessen verwenden Sie möglicherweise das Amazon-Linux-2-AMI. Sie können Ihre Version von Amazon Linux mit dem folgenden Befehl anzeigen.

   ```
   cat /etc/system-release
   ```

   Weitere Informationen finden Sie unter Aktualisieren der Software einer Instance.

4. Starten Sie den Webserver mithilfe des folgenden Befehls.

   ```
   sudo systemctl start httpd
   ```

   Sie können testen, ob Ihr Webserver richtig installiert und gestartet ist. Geben Sie dazu den öffentlichen DNS-Namen (Domain Name System) Ihrer EC2-Instance in die Adressleiste eines Webbrowsers ein, zum Beispiel: `http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com`. Wenn Ihr Webserver ausgeführt wird, wird Ihnen die Apache-Testseite angezeigt.

   Wenn Sie die Apache-Testseite nicht sehen, überprüfen Sie die Regeln für eingehenden Datenverkehr für die VPC-Sicherheitsgruppe, die Sie in erstellt habe Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur). Stellen Sie sicher, dass die Regeln für eingehenden

Datenverkehr eine Regel enthalten, die den HTTP-Zugriff (Port 80) für die IP-Adresse ermöglicht, um eine Verbindung mit dem Webserver herzustellen.

> **Note**
>
> Die Apache-Testseite wird nur angezeigt, wenn im Dokumentstammverzeichnis `/var/www/html` keine Inhalte vorhanden sind. Wenn Sie dem Dokumentstammverzeichnis Inhalte hinzugefügt haben, werden unter der öffentlichen DNS-Adresse Ihrer EC2-Instance Ihre Inhalte angezeigt. Bis zu diesem Zeitpunkt erscheinen sie auf der Apache-Testseite.

5. Konfigurieren Sie den Webserver so, dass er mit jedem Systemstart gestartet wird. Verwenden Sie hierzu den Befehl `systemctl`.

```
sudo systemctl enable httpd
```

Um für `ec2-user` die Verwaltung von Dateien im Standardstammverzeichnis Ihres Apache-Webservers zuzulassen, ändern Sie die Eigentümerschaft und die Berechtigungen für das Verzeichnis `/var/www`. Es gibt viele Möglichkeiten, um diese Aufgabe zu erfüllen. In diesem Tutorial fügen Sie `ec2-user` zur `apache`-Gruppe hinzu, geben der `apache`-Gruppe Eigentümerschaft über das Verzeichnis `/var/www` und weisen der Gruppe Schreibberechtigungen zu.

So legen Sie Dateiberechtigungen für den Apache-Webserver fest

1. Fügen Sie den Benutzer `ec2-user` zur Gruppe `apache` hinzu.

```
sudo usermod -a -G apache ec2-user
```

2. Melden Sie sich ab, um Ihre Berechtigungen zu aktualisieren und die neue Gruppe `apache` einzufügen.

```
exit
```

3. Melden Sie sich wieder an und überprüfen Sie mit dem Befehl `apache`, ob die Gruppe `groups` vorhanden ist.

```
groups
```

Die Ausgabe sieht folgendermaßen oder ähnlich aus:

```
ec2-user adm wheel apache systemd-journal
```

4. Ändern Sie die Besitzergruppe für das Verzeichnis `/var/www` und dessen Inhalte in die Gruppe `apache`.

```
sudo chown -R ec2-user:apache /var/www
```

5. Ändern Sie die Verzeichnisberechtigungen von `/var/www` und dessen Unterverzeichnissen, indem Sie Schreibberechtigungen für die Gruppe hinzufügen und die Gruppen-ID für zukünftige Unterverzeichnisse einrichten.

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. Ändern Sie die Dateiberechtigungen für Dateien im Verzeichnis `/var/www` und dessen Unterverzeichnissen rekursiv, um Schreibberechtigungen für die Gruppe hinzuzufügen.

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Jetzt kann `ec2-user` (und jedes künftige Mitglied der `apache`-Gruppe) im Dokumentstammverzeichnis von Apache Dateien hinzufügen, löschen und bearbeiten. Damit haben Sie die Möglichkeit, Inhalte hinzuzufügen, z. B. eine statische Website oder eine PHP-Anwendung.

> **Note**
>
> Ein Webserver, auf dem HTTP ausgeführt wird, bietet keine Transportsicherheit für die gesendeten oder empfangenen Daten. Wenn Sie über einen Webbrowser eine Verbindung mit einem HTTP-Server herstellen, sind viele Informationen überall auf dem Netzwerkpfad für Lauscher zugänglich. Zu diesen Informationen gehören die URLs, die Sie aufrufen, die Inhalte von Webseiten, die Sie empfangen, und die Inhalte (einschließlich Passwörtern) von HTML-Formularen.
> Die bewährte Methode, Ihren Webserver abzusichern, besteht darin, Unterstützung für HTTPS (HTTP Secure) zu installieren. Dieses Protokoll schützt Ihre Daten mit SSL-/TLS-Verschlüsselung. Weitere Informationen finden Sie unter Tutorial: SSL/TLS mit dem Amazon Linux AMI konfigurieren im Amazon EC2-Benutzerhandbuch .

Herstellen der Verbindung zwischen Ihrem Apache-Webserver und Ihrer DB-Instance

Als Nächstes fügen Sie Ihrem Apache-Server, der mit Ihrer Amazon RDS-DB-Instance/dem verbunden ist, Inhalte hinzu.

So fügen Sie dem Apache-Server, der mit Ihrer -DB-Instance/dem verbunden ist, Inhalte hinzu

1. Während die Verbindung mit Ihrer EC2-Instance besteht, ändern Sie das Verzeichnis in `/var/www` und erstellen Sie ein neues Unterverzeichnis namens `inc`.

   ```
   cd /var/www
   mkdir inc
   cd inc
   ```

2. Erstellen Sie im Verzeichnis `inc` eine neue Datei namens `dbinfo.inc` und bearbeiten Sie anschließend die Datei, indem Sie Nano (oder einen Editor Ihrer Wahl) aufrufen.

   ```
   >dbinfo.inc
   nano dbinfo.inc
   ```

3. Fügen Sie der Datei `dbinfo.inc` die folgenden Inhalte hinzu. In diesem Fall ist *`db_instance_endpoint`* Ihr DB-Instance-Endpunkt ohne den Port für Ihre DB-Instance.

   > **Note**
   >
   > Es wird empfohlen, die Informationen zu Benutzername und Passwort in einem Ordner abzulegen, der nicht Teil des Dokumentstammverzeichnisses für Ihren Webserver ist. Auf diese Weise wird die Möglichkeit verringert, dass Ihre Sicherheitsinformationen offengelegt werden.
   > Stellen Sie sicher, dass Sie `master password` in Ihrer Anwendung in ein geeignetes Passwort ändern.

   ```
   <?php

   define('DB_SERVER', 'db_instance_endpoint');
   define('DB_USERNAME', 'tutorial_user');
   define('DB_PASSWORD', 'master password');
   define('DB_DATABASE', 'sample');
   ```

```
?>
```

4. Speichern und schließen Sie die Datei `dbinfo.inc`. Wenn Sie Nano verwenden, speichern und schließen Sie die Datei mit Strg+S und Strg+X.
5. Ändern Sie das Verzeichnis in `/var/www/html`.

```
cd /var/www/html
```

6. Erstellen Sie im Verzeichnis `html` eine neue Datei namens `SamplePage.php` und bearbeiten Sie anschließend die Datei, indem Sie Nano (oder einen Editor Ihrer Wahl) aufrufen.

```
>SamplePage.php
nano SamplePage.php
```

7. Fügen Sie der Datei `SamplePage.php` die folgenden Inhalte hinzu:

   MariaDB & MySQL

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
<?php

  /* Connect to MySQL and select the database. */
  $connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

  if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " .
 mysqli_connect_error();

  $database = mysqli_select_db($connection, DB_DATABASE);

  /* Ensure that the EMPLOYEES table exists. */
  VerifyEmployeesTable($connection, DB_DATABASE);

  /* If input fields are populated, add a row to the EMPLOYEES table. */
  $employee_name = htmlentities($_POST['NAME']);
  $employee_address = htmlentities($_POST['ADDRESS']);

  if (strlen($employee_name) || strlen($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
  }
```

```
?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
        <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
        <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
        <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = mysqli_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = mysqli_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>
```

```
</table>

<!-- Clean up. -->
<?php

  mysqli_free_result($result);
  mysqli_close($connection);

?>

</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = mysqli_real_escape_string($connection, $name);
   $a = mysqli_real_escape_string($connection, $address);

   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</
p>");
  }
}

/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
```

```
  $t = mysqli_real_escape_string($connection, $tableName);
  $d = mysqli_real_escape_string($connection, $dbName);

  $checktable = mysqli_query($connection,
      "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t'
 AND TABLE_SCHEMA = '$d'");

  if(mysqli_num_rows($checktable) > 0) return true;

  return false;
}
?>
```

PostgreSQL

```
<?php include "../inc/dbinfo.inc"; ?>

<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to PostgreSQL and select the database. */
$constring = "host=" . DB_SERVER . " dbname=" . DB_DATABASE . " user=" .
 DB_USERNAME . " password=" . DB_PASSWORD ;
$connection = pg_connect($constring);

if (!$connection){
 echo "Failed to connect to PostgreSQL";
 exit;
}

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);

if (strlen($employee_name) || strlen($employee_address)) {
  AddEmployee($connection, $employee_name, $employee_address);
}
```

```
?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
    <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
    <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
    <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>
<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = pg_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = pg_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>
</table>
```

```
<!-- Clean up. -->
<?php

  pg_free_result($result);
  pg_close($connection);
?>
</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = pg_escape_string($name);
   $a = pg_escape_string($address);
   echo "Forming Query";
   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!pg_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID serial PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!pg_query($connection, $query)) echo("<p>Error creating table.</p>");
  }
}
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = strtolower(pg_escape_string($tableName)); //table name is case sensitive
  $d = pg_escape_string($dbName); //schema is 'public' instead of 'sample' db
 name so not using that
```

```
  $query = "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME =
 '$t';";
  $checktable = pg_query($connection, $query);

  if (pg_num_rows($checktable) >0) return true;
  return false;

}
?>
```

8. Speichern und schließen Sie die Datei `SamplePage.php`.
9. Überprüfen Sie, ob Ihr Webserver erfolgreich eine Verbindung mit Ihrer DB-Instance herstellen kann, indem Sie einen Webbrowser öffnen und zu `http://`*`EC2 instance endpoint`*`/SamplePage.php` navigieren, beispielsweise: `http://ec2-12-345-67-890.us-west-2.compute.amazonaws.com/SamplePage.php`.

Sie können `SamplePage.php` verwenden, um Ihrer DB-Instance Daten hinzuzufügen. Die von Ihnen hinzugefügten Daten werden anschließend auf der Seite angezeigt. Wenn Sie überprüfen möchten, ob die Daten in die Tabelle eingefügt wurden, installieren Sie den MySQL-Client auf der Amazon-EC2-Instance. Stellen Sie dann eine Verbindung mit der DB-Instance her und führen Sie eine Abfrage der Tabelle aus.

Informationen zum Installieren des MySQL-Clients und zum Herstellen einer Verbindung mit einer DB-Instance finden Sie unter Verbindung zu Ihrer MySQL-DB-Instance herstellen.

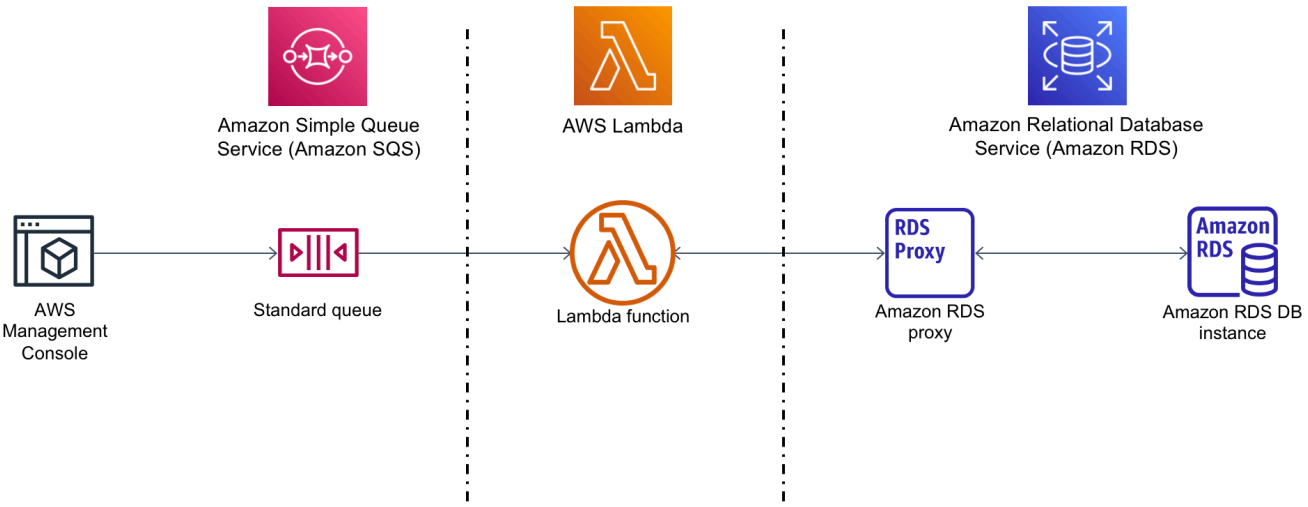
Um sicherzustellen, dass Ihre DB-Instance so sicher wie möglich ist, müssen Sie sich vergewissern, dass Quellen außerhalb der VPC keine Verbindungen mit Ihrer DB-Instance herstellen können.

Nachdem Sie Ihren Webserver und Ihre Datenbank getestet haben, sollten Sie Ihre DB-Instance und Ihre Amazon EC2-Instance löschen.

- Um eine DB-Instance zu löschen, folgen Sie den Anweisungen in Löschen einer DB-Instance. Sie müssen keinen abschließenden Snapshot erstellen.
- Um eine Amazon EC2-Instance zu beenden, folgen Sie den Anweisungen in Beenden Ihrer Instance im Amazon EC2-Benutzerhandbuch.

Tutorial: Verwenden einer Lambda-Funktion für den Zugriff auf eine Amazon-RDS-Datenbank

In diesem Tutorial verwenden Sie eine Lambda-Funktion, um Daten über RDS Proxy in eine Amazon Relational Database Service (Amazon RDS)-Datenbank zu schreiben. Ihre Lambda-Funktion liest Datensätze aus einer Amazon Simple Queue Service (Amazon SQS)-Warteschlange und schreibt ein neues Element in eine Tabelle in Ihrer Datenbank, sobald eine Nachricht hinzugefügt wird. In diesem Beispiel verwenden Sie die AWS Management Console , um Ihrer Warteschlange manuell Nachrichten hinzuzufügen. Das folgende Diagramm zeigt die AWS Ressourcen, die Sie zur Durchführung des Tutorials verwenden.



Mit Amazon RDS können Sie eine verwaltete relationale Datenbank in der Cloud mithilfe gängiger Datenbankprodukte wie Microsoft SQL Server, MariaDB, MySQL, Oracle Database und PostgreSQL ausführen. Wenn Sie Lambda für den Zugriff auf Ihre Datenbank verwenden, können Sie Daten als Reaktion auf Ereignisse lesen und schreiben, z. B. wenn sich ein neuer Kunde auf Ihrer Website registriert. Ihre Funktion, Datenbank-Instance und der Proxy skalieren automatisch, um Zeiten hoher Nachfrage gerecht zu werden.

Führen Sie für dieses Tutorial die folgenden Aufgaben aus:

1. Starten Sie eine RDS for MySQL-Datenbank-Instance und einen Proxy in Ihrer AWS-Konto Standard-VPC.

2. Erstellen und testen Sie eine Lambda-Funktion, die eine neue Tabelle in Ihrer Datenbank erstellt und Daten in diese schreibt.
3. Erstellen Sie eine Amazon-SQS-Warteschlange und konfigurieren Sie sie so, dass Ihre Lambda-Funktion aufgerufen wird, wenn eine neue Nachricht hinzugefügt wird.
4. Testen Sie das komplette Setup, indem Sie Nachrichten mithilfe von zu Ihrer Warteschlange hinzufügen AWS Management Console und die Ergebnisse mithilfe von CloudWatch Logs überwachen.

Wenn Sie diese Schritte ausführen, lernen Sie Folgendes:

- Verwenden von Amazon RDS, um eine Datenbank-Instance und einen Proxy zu erstellen und eine Lambda-Funktion mit dem Proxy zu verbinden.
- Verwenden von Lambda, um Erstellungs- und Lesevorgänge in einer Amazon-RDS-Datenbank durchzuführen
- Verwenden von Amazon SQS zum Aufrufen einer Lambda-Funktion

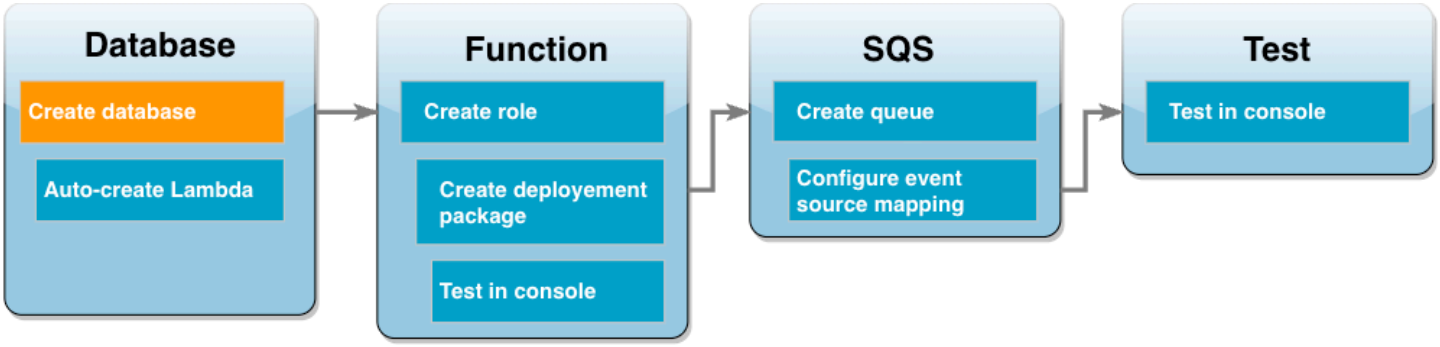
Sie können dieses Tutorial mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) abschließen.

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Erstellen einer DB-Instance von Amazon RDS



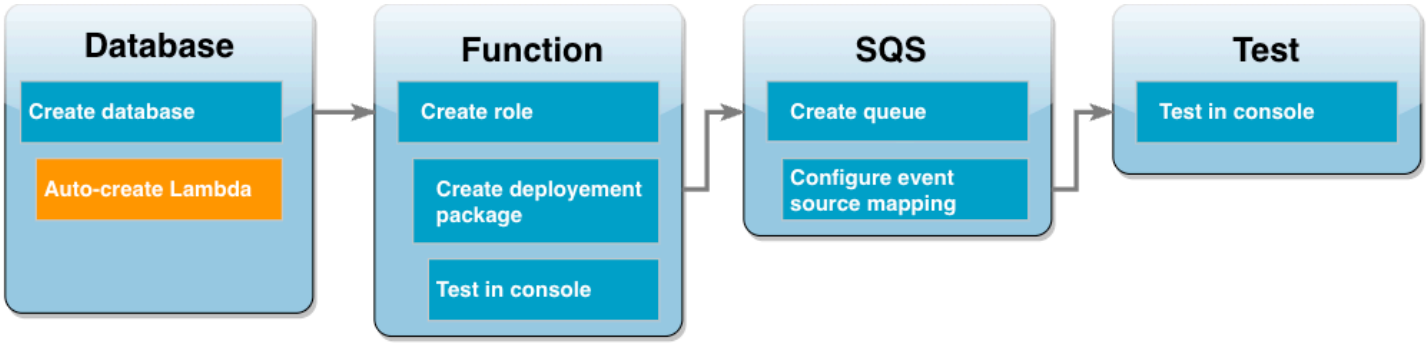
Eine DB-Instance von Amazon RDS ist eine isolierte Datenbankumgebung, die in der AWS Cloud ausgeführt wird. Eine Instance kann mehrere von Benutzern erstellte Datenbanken enthalten. Sofern Sie nichts anderes angeben, erstellt Amazon RDS neue Datenbank-Instances in der Standard-VPC, die in Ihrer AWS-Konto enthalten ist. Weitere Informationen zu Amazon VPC finden Sie im Benutzerhandbuch von Amazon Virtual Private Cloud.

In diesem Tutorial erstellen Sie eine neue Instance in Ihrer AWS-Konto Standard-VPC und erstellen eine Datenbank mit dem Namen dieser `ExampleDB` Instance. Sie können Ihre DB-Instance und Datenbank entweder mit dem AWS Management Console oder dem AWS CLI erstellen.

So erstellen Sie eine Datenbank-Instance

1. Öffnen Sie die Amazon-RDS-Konsole und wählen Sie Datenbank erstellen aus.
2. Lassen Sie die Standard-Erstellungsoption aktiviert und wählen Sie dann unter Engine-Optionen die Option MySQL aus.
3. Wählen Sie unter Vorlagen die Option Kostenloses Kontingent oder Sandbox aus. Das kostenlose Kontingent wird für Konten mit dem kostenlosen Kontingent angezeigt. Für Konten mit kostenpflichtigem Tarif wird eine Sandbox angezeigt.
4. Geben Sie unter Settings (Einstellungen) für die DB instance identifier (DB-Instance-Kennung) **`MySQLForLambda`** ein.
5. Gehen Sie wie folgt vor, um Ihren Benutzernamen und Ihr Passwort zu erstellen:
   a. Belassen Sie in den Einstellungen für Anmeldeinformationen den Hauptbenutzernamen bei `admin`.
   b. Geben Sie als Master-Passwort ein Passwort ein und bestätigen Sie es, um auf Ihre Datenbank zuzugreifen.
6. Geben Sie den Datenbanknamen an, indem Sie wie folgt vorgehen:
   - Lassen Sie alle verbleibenden Standardoptionen ausgewählt und scrollen Sie nach unten zum Abschnitt Zusätzliche Konfiguration.
   - Erweitern Sie diesen Bereich und geben Sie **`ExampleDB`** als Anfänglicher Datenbankname ein.
7. Lassen Sie alle verbleibenden Standardoptionen ausgewählt und wählen Sie Create database (Datenbank erstellen).

Erstellen einer Lambda-Funktion und eines Proxys



Sie können die RDS-Konsole verwenden, um eine Lambda-Funktion und einen Proxy in derselben VPC wie die Datenbank zu erstellen.

> **Note**
>
> Sie können diese zugehörigen Ressourcen nur erstellen, wenn Ihre Datenbank die Erstellung abgeschlossen hat und sich im Status Verfügbar befindet.

So erstellen Sie eine zugehörige Funktion und einen Proxy

1. Überprüfen Sie auf der Seite Datenbanken, ob sich Ihre Datenbank im Status Verfügbar befindet. Ist dies der Fall, fahren Sie mit dem nächsten Schritt fort. Andernfalls warten Sie, bis Ihre Datenbank verfügbar ist.
2. Wählen Sie Ihre Datenbank und die Option Lambda-Verbindung einrichten unter Aktionen aus.
3. Wählen Sie auf der Seite Lambda-Verbindung einrichten die Option Neue Funktion erstellen aus.

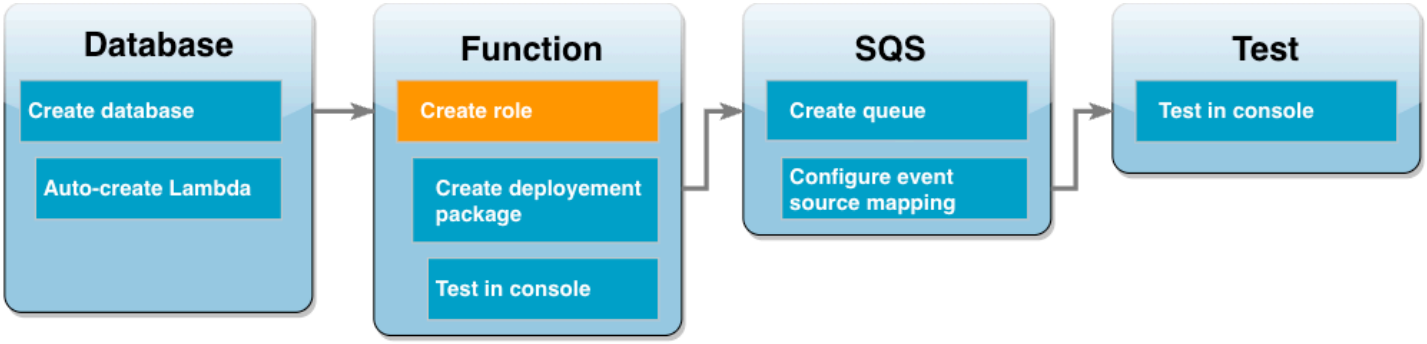
   Legen Sie den Wert im Feld Neuer Lambda-Funktionsname auf **`LambdaFunctionWithRDS`** fest.
4. Wählen Sie im Abschnitt RDS-Proxy die Option Über RDS-Proxy verbinden aus. Wählen Sie dann Neuen Proxy erstellen aus.

   - Wählen Sie für Datenbank-Anmeldeinformationen die Option Datenbank-Benutzername und Passwort aus.
   - Geben Sie für Benutzername den Namen `admin` an.
   - Geben Sie unter Passwort das Passwort ein, das Sie für Ihre DB-Instance erstellt haben.

5. Wählen Sie Einrichten aus, um die Erstellung des Proxys und der Lambda-Funktion abzuschließen.

Der Assistent schließt die Einrichtung ab und stellt einen Link zur Lambda-Konsole bereit, über den Sie Ihre neue Funktion überprüfen können. Notieren Sie sich den Proxy-Endpunkt, bevor Sie zur Lambda-Konsole wechseln.

Erstellen einer Funktionsausführungsrolle



Bevor Sie Ihre Lambda-Funktion erstellen, erstellen Sie eine Ausführungsrolle, um Ihrer Funktion die erforderlichen Berechtigungen zu geben. Für dieses Tutorial benötigt Lambda die Berechtigung, die Netzwerkverbindung zu der VPC, die Ihre Datenbank-Instance enthält, zu verwalten und Nachrichten aus einer Amazon-SQS-Warteschlange abzufragen.

Um Ihrer Lambda-Funktion die erforderlichen Berechtigungen zu geben, verwendet dieses Tutorial von IAM verwaltete Richtlinien. Dies sind Richtlinien, die Berechtigungen für viele häufige Anwendungsfälle erteilen und in Ihrem AWS-Konto verfügbar sind. Weitere Informationen über die Verwendung verwalteter Richtlinien finden Sie unter Bewährte Methoden für Richtlinien.

So erstellen Sie eine Lambda-Ausführungsrolle

1. Öffnen Sie die Seite Roles (Rollen) Ihrer IAM-Konsole und wählen Sie Create role (Rolle erstellen).
2. Wählen Sie als Vertrauenswürdiger Entitätstyp die Option AWS Service und als Anwendungsfall die Option Lambda aus.
3. Wählen Sie Weiter aus.
4. Fügen Sie die von IAM verwalteten Richtlinien hinzu, indem Sie wie folgt vorgehen:

    a. Verwenden Sie das Richtliniensuchfeld zur Suche nach **`AWSLambdaSQSQueueExecutionRole`**.

b. Aktivieren Sie in der Ergebnisliste das Kontrollkästchen neben der Rolle und wählen Sie Clear filters (Filter löschen).

c. Verwenden Sie das Richtliniensuchfeld zur Suche nach **`AWSLambdaVPCAccessExecutionRole`**.

d. Aktivieren Sie in der Ergebnisliste das Kontrollkästchen neben der Rolle und wählen Sie Next (Weiter).

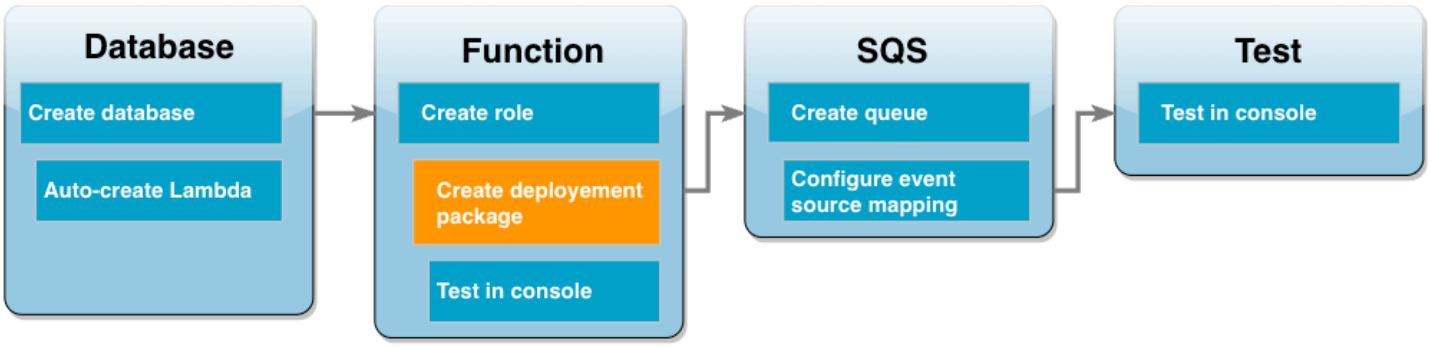
5. Geben Sie für Role name (Rollenname) den Namen **`lambda-vpc-sqs-role`** ein und wählen Sie dann Create role (Rolle erstellen).

Später im Tutorial benötigen Sie den Amazon-Ressourcennamen (ARN) der Ausführungsrolle, die Sie gerade erstellt haben.

So suchen Sie nach einem Ausführungsrollen-ARN

1. Öffnen Sie die Seite Rollen Ihrer IAM-Konsole und wählen Sie Ihre Rolle (`lambda-vpc-sqs-role`) aus.
2. Kopieren Sie den Rollen-ARN, der im Abschnitt Zusammenfassung angezeigt wird.

Erstellen eines Lambda-Bereitstellungspakets



Der folgende Python-Beispielcode verwendet das PyMySQL-Paket, um eine Verbindung zu Ihrer Datenbank zu öffnen. Wenn Sie Ihre Funktion zum ersten Mal aufrufen, wird auch eine neue Tabelle namens `Customer` erstellt. Die Tabelle verwendet das folgende Schema, wobei `CustID` der Primärschlüssel ist:

```
Customer(CustID, Name)
```

Die Funktion verwendet auch PyMy SQL, um dieser Tabelle Datensätze hinzuzufügen. Die Funktion fügt Datensätze mit Kunden IDs und Namen hinzu, die in Nachrichten angegeben sind, die Sie zu Ihrer Amazon SQS SQS-Warteschlange hinzufügen werden.

Der Code stellt die Verbindung mit Ihrer Datenbank außerhalb der Handlerfunktion her. Durch das Erstellen der Verbindung im Initialisierungscode kann die Verbindung durch nachfolgende Aufrufe Ihrer Funktion wiederverwendet werden, was die Leistung verbessert. In einer Produktionsanwendung können Sie die bereitgestellte Parallelität auch verwenden, um eine angeforderte Anzahl von Datenbankverbindungen zu initialisieren. Diese Verbindungen sind verfügbar, sobald Ihre Funktion aufgerufen wird.

```
import sys
import logging
import pymysql
import json
import os

# rds settings
user_name = os.environ['USER_NAME']
password = os.environ['PASSWORD']
rds_proxy_host = os.environ['RDS_PROXY_HOST']
db_name = os.environ['DB_NAME']

logger = logging.getLogger()
logger.setLevel(logging.INFO)

# create the database connection outside of the handler to allow connections to be
# re-used by subsequent function invocations.
try:
        conn = pymysql.connect(host=rds_proxy_host, user=user_name, passwd=password,
 db=db_name, connect_timeout=5)
except pymysql.MySQLError as e:
    logger.error("ERROR: Unexpected error: Could not connect to MySQL instance.")
    logger.error(e)
    sys.exit(1)

logger.info("SUCCESS: Connection to RDS for MySQL instance succeeded")

def lambda_handler(event, context):
    """
    This function creates a new RDS database table and writes records to it
    """
```

```
    message = event['Records'][0]['body']
    data = json.loads(message)
    CustID = data['CustID']
    Name = data['Name']

    item_count = 0
    sql_string = f"insert into Customer (CustID, Name) values(%s, %s)"

    with conn.cursor() as cur:
        cur.execute("create table if not exists Customer ( CustID  int NOT NULL, Name
varchar(255) NOT NULL, PRIMARY KEY (CustID))")
        cur.execute(sql_string, (CustID, Name))
        conn.commit()
        cur.execute("select * from Customer")
        logger.info("The following items have been added to the database:")
        for row in cur:
            item_count += 1
            logger.info(row)
    conn.commit()

    return "Added %d items to RDS for MySQL table" %(item_count)
```

> **Note**
>
> In diesem Beispiel werden Ihre Anmeldeinformationen für den Datenbankzugriff als Umgebungsvariablen gespeichert. In Produktionsanwendungen empfehlen wir, die Option AWS Secrets Manager als sicherere Option zu verwenden. Hinweis: Wenn sich Ihre Lambda-Funktion in einer VPC befindet, müssen Sie einen VPC-Endpunkt erstellen, um eine Verbindung zu Secrets Manager herzustellen. Weitere Informationen finden Sie unter Verwenden einer Lambda-Funktion für den Zugriff auf Amazon RDS in einer Amazon VPC.

Um die PyMy SQL-Abhängigkeit in Ihren Funktionscode aufzunehmen, erstellen Sie ein ZIP-Bereitstellungspaket. Die folgenden Befehle funktionieren für Linux, macOS oder Unix:

So erstellen Sie ein ZIP-Bereitstellungspaket

1. Speichern Sie den Beispielcode als Datei mit dem Namen `lambda_function.py`.

2. Erstellen Sie in demselben Verzeichnis, in dem Sie Ihre `lambda_function.py` Datei erstellt haben, ein neues Verzeichnis mit dem Namen `package` und installieren Sie die PyMy SQL-Bibliothek.

```
mkdir package
pip install --target package pymysql
```

3. Erstellen Sie eine ZIP-Datei, die Ihren Anwendungscode und die PyMy SQL-Bibliothek enthält. Führen Sie unter Linux oder MacOS die folgenden CLI-Befehle aus. Verwenden Sie unter Windows Ihr bevorzugtes ZIP-Tool, um die `lambda_function.zip`-Datei zu erstellen. Ihre Quellcodedatei `lambda_function.py` und die Ordner, die Ihre Abhängigkeiten enthalten, müssen im Stammverzeichnis der ZIP-Datei installiert sein.

```
cd package
zip -r ../lambda_function.zip .
cd ..
zip lambda_function.zip lambda_function.py
```

   Sie können Ihr Bereitstellungspaket auch in einer virtuellen Python-Umgebung erstellen. Siehe Bereitstellen von Python-Lambda-Funktionen mit ZIP-Dateiarchiven.

Aktualisieren der Lambda-Funktion

Mit dem soeben erstellten ZIP-Paket aktualisieren Sie nun Ihre Lambda-Funktion über die Lambda-Konsole. Damit Ihre Funktion auf Ihre Datenbank zugreifen kann, müssen Sie außerdem Umgebungsvariablen mit Ihren Zugangsdaten konfigurieren.

So aktualisieren Sie die Lambda-Funktion

1. Öffnen Sie die Seite Funktionen der Lambda-Konsole und wählen Sie Ihre Funktion `LambdaFunctionWithRDS` aus.
2. Wählen Sie auf der Registerkarte Runtime-Einstellungen die Option Bearbeiten aus, um die Laufzeit der Funktion auf Python 3.10 zu ändern.
3. Ändern Sie den Handler in `lambda_function.lambda_handler`.
4. Wählen Sie im Bereich Code die Option Hochladen von und dann ZIP-Datei aus.
5. Wählen Sie die `lambda_function.zip`-Datei aus, die Sie in der vorherigen Phase erstellt haben, und wählen Sie Save (Speichern).

Konfigurieren Sie nun die Funktion mit der zuvor erstellten Ausführungsrolle. Dadurch erhält die Funktion die Berechtigungen, die sie benötigt, um auf Ihre Datenbank-Instance zuzugreifen und eine Amazon-SQS-Warteschlange abzufragen.

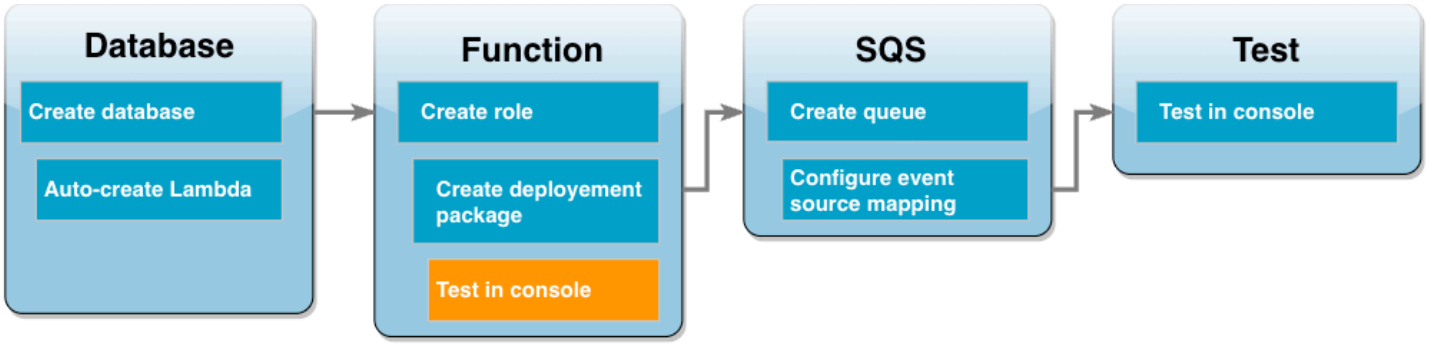
So konfigurieren Sie die Ausführungsrolle der Funktion

1. Wählen Sie auf der Seite Funktionen der Lambda-Konsole die Registerkarte Konfiguration und dann Berechtigungen aus.
2. Wählen Sie unter Ausführungsrolle die Option Bearbeiten aus.
3. Wählen Sie unter Vorhandene Rolle Ihre Ausführungsrolle (`lambda-vpc-sqs-role`) aus.
4. Wählen Sie Speichern.

So konfigurieren Sie dei Umgebungsvariablen Ihrer Funktion

1. Wählen Sie auf der Seite Funktionen der Lambda-Konsole die Registerkarte Konfiguration und dann Berechtigungsvariablen aus.
2. Wählen Sie Bearbeiten aus.
3. Gehen Sie wie folgt vor, um Ihre Anmeldeinformationen für den Datenbankzugriff hinzuzufügen:
    a. Wählen Sie Umgebungsvariable hinzufügen und geben Sie dann **`USER_NAME`** für Schlüssel und **`admin`** für Wert ein.
    b. Wählen Sie Umgebungsvariable hinzufügen und geben Sie dann **`DB_NAME`** für Schlüssel und **`ExampleDB`** für Wert ein.
    c. Wählen Sie Umgebungsvariable hinzufügen und geben Sie dann **`PASSWORD`** für Schlüssel und für Wert das Passwort ein, das Sie bei der Erstellung Ihrer Datenbank gewählt haben.
    d. Wählen Sie Umgebungsvariable hinzufügen aus und geben Sie dann **`RDS_PROXY_HOST`** für Schlüssel und für Wert den RDS-Proxy-Endpunkt ein, den Sie zuvor notiert haben.
    e. Wählen Sie Speichern.

Testen Ihrer Lambda-Funktion in der Konsole.



Sie können jetzt die Lambda-Konsole verwenden, um Ihre Funktion zu testen. Sie erstellen ein Testereignis, das die Daten nachahmt, die Ihre Funktion erhält, wenn Sie sie in der letzten Phase des Tutorials mit Amazon SQS aufrufen. Ihr Testereignis enthält ein JSON-Objekt, das eine Kunden-ID und einen Kundennamen angibt, die der von Ihrer Funktion erstellten `Customer`-Tabelle hinzugefügt werden sollen.

Lambda-Funktion testen

1. Öffnen Sie die Seite Functions (Funktionen) der Lambda-Konsole und wählen Sie eine Funktion aus.
2. Wählen Sie den Abschnitt Testen aus.
3. Wählen Sie Neues Ereignis erstellen aus und geben Sie **`myTestEvent`** als Ereignisnamen ein.
4. Kopieren Sie den folgenden Code in Event JSON (Ereignis-JSON) und wählen Sie Save (Speichern).

```
{
  "Records": [
    {
      "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
      "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
      "body": "{\n     \"CustID\": 1021,\n     \"Name\": \"Martha Rivera\"\n}",
      "attributes": {
        "ApproximateReceiveCount": "1",
        "SentTimestamp": "1545082649183",
        "SenderId": "AIDAIENQZJOLO23YVJ4VO",
        "ApproximateFirstReceiveTimestamp": "1545082649185"
      },
      "messageAttributes": {},
      "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
```

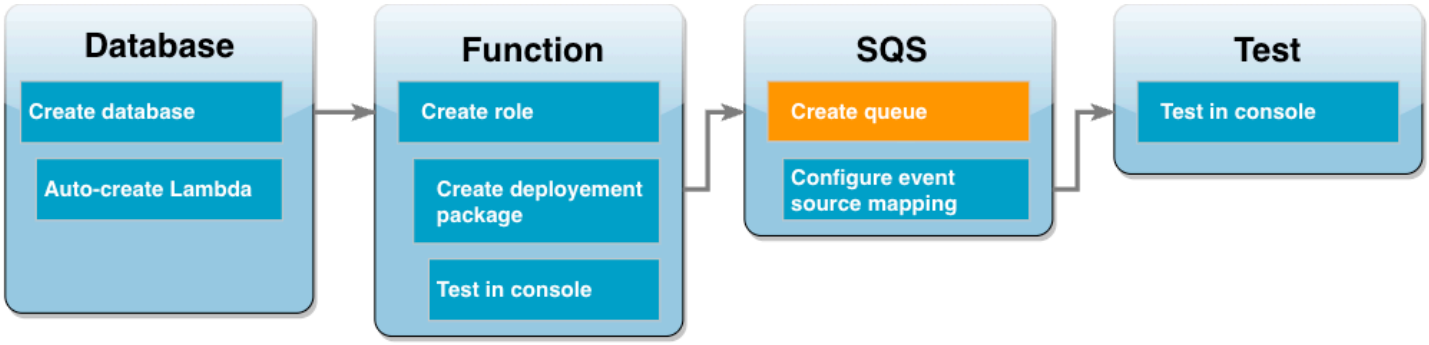
```
      "eventSource": "aws:sqs",
      "eventSourceARN": "arn:aws:sqs:us-west-2:123456789012:my-queue",
      "awsRegion": "us-west-2"
    }
  ]
}
```

5. Wählen Sie Test aus.

Auf der Registerkarte Ausführungsergebnisse sollten in den Funktionsprotokollen ähnliche Ergebnisse wie die folgenden angezeigt werden:

```
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f The following
 items have been added to the database:
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f (1021, 'Martha
 Rivera')
```

Erstellen einer Amazon-SQS-Warteschlange

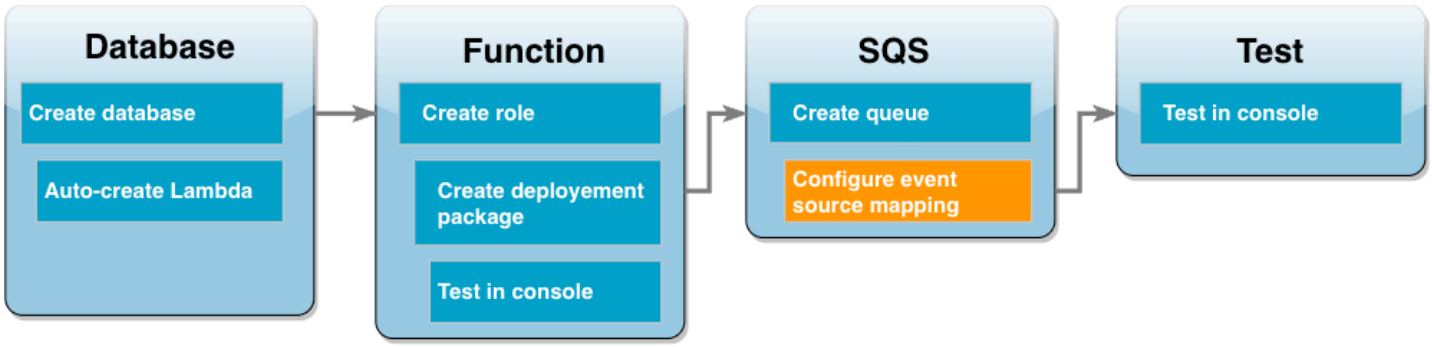


Sie haben die Integration Ihrer Lambda-Funktion und der Amazon-RDS-Datenbank-Instance erfolgreich getestet. Jetzt erstellen Sie die Amazon-SQS-Warteschlange, mit der Sie Ihre Lambda-Funktion in der letzten Phase des Tutorials aufrufen werden.

So erstellen Sie die Amazon-SQS-Warteschlange (Konsole)

1. Öffnen Sie die Seite Queues (Warteschlangen) der Amazon-SQS-Konsole und wählen Sie Create queue (Warteschlange erstellen) aus.
2. Belassen Sie den Type (Typ) auf Standard und geben Sie **`LambdaRDSQueue`** als Namen Ihrer Warteschlange ein.
3. Lassen Sie alle Standardoptionen ausgewählt und wählen Sie Create queue (Warteschlange erstellen).

Erstellen einer Zuordnung von Ereignisquellen, um Ihre Lambda-Funktion aufzurufen



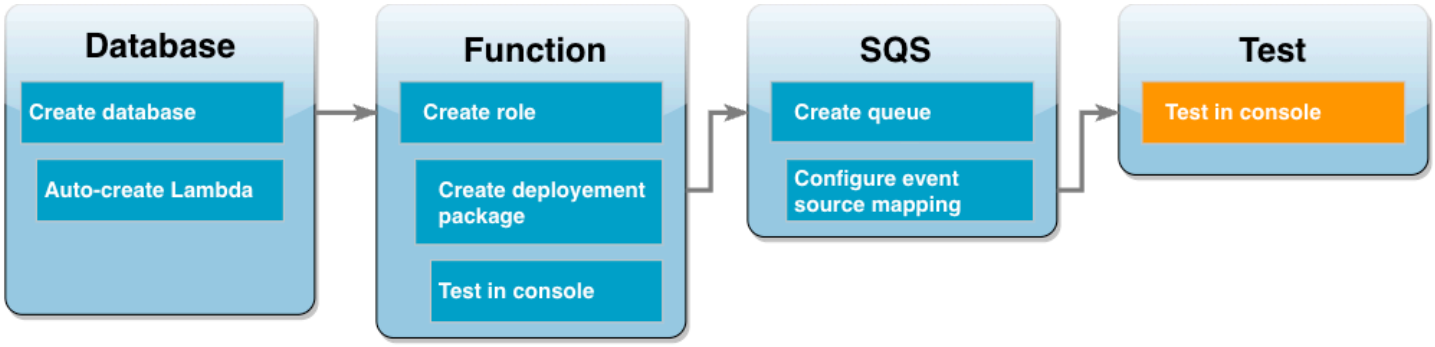
Ein Zuordnung von Ereignisquellen ist eine Ressource in Lambda, die Elemente aus einem Stream oder einer Warteschlange liest und eine Lambda-Funktion aufruft. Wenn Sie eine Zuordnung von Ereignisquellen konfigurieren, können Sie eine Batchgröße angeben, sodass Datensätze aus Ihrem Stream oder Ihrer Warteschlange zu einer einzigen Nutzlast zusammengefasst werden. In diesem Beispiel legen Sie die Batchgröße auf 1 fest, sodass Ihre Lambda-Funktion jedes Mal aufgerufen wird, wenn Sie eine Nachricht an Ihre Warteschlange senden. Sie können die Zuordnung der Ereignisquelle entweder mit der AWS CLI oder der Lambda-Konsole konfigurieren.

So erstellen Sie eine Zuordnung von Ereignisquellen (Konsole)

1. Öffnen Sie die Seite Functions (Funktionen) der Lambda-Konsole und wählen Sie Ihre Funktion (`LambdaFunctionWithRDS`) aus.
2. Wählen Sie im Abschnitt Funktionsübersicht die Option Auslöser hinzufügen aus.
3. Wählen Sie als Quelle Amazon SQS und dann den Namen Ihrer Warteschlange (`LambdaRDSQueue`) aus.
4. Geben Sie für Batch size (Stapelgröße) **1** ein.
5. Belassen Sie alle anderen Optionen auf den Standardwerten und wählen Sie Add (Hinzufügen).

Sie sind jetzt bereit, Ihre vollständige Einrichtung zu testen, indem Sie Ihrer Amazon-SQS-Warteschlange eine Nachricht hinzufügen.

Testen und Überwachen Ihrer Einrichtung



Um Ihre vollständige Einrichtung zu testen, fügen Sie mithilfe der Konsole Nachrichten zu Ihrer Amazon-SQS-Warteschlange hinzu. Anschließend überprüfen Sie mithilfe von CloudWatch Logs, dass Ihre Lambda-Funktion wie erwartet Datensätze in Ihre Datenbank schreibt.

So testen und überwachen Sie Ihre Einrichtung

1. Öffnen Sie die Seite Queues (Warteschlangen) der Amazon-SQS-Konsole und wählen Sie Ihre Warteschlange (`LambdaRDSQueue`) aus.
2. Wählen Sie Nachrichten senden und empfangen aus und fügen Sie den folgenden JSON in den Nachrichtentext im Abschnitt Nachricht senden ein.

   ```
   {
       "CustID": 1054,
       "Name": "Richard Roe"
   }
   ```

3. Klicken Sie auf Send Message (Nachricht senden).

   Wenn Sie Ihre Nachricht an die Warteschlange senden, ruft Lambda Ihre Funktion über Ihre Zuordnung von Ereignisquellen auf. Um zu überprüfen, ob Lambda Ihre Funktion wie erwartet aufgerufen hat, überprüfen Sie mithilfe von CloudWatch Logs, ob die Funktion den Kundennamen und die Kunden-ID in Ihre Datenbanktabelle geschrieben hat.
4. Öffnen Sie die Seite Protokollgruppen der CloudWatch Konsole und wählen Sie die Protokollgruppe für Ihre Funktion aus (`/aws/lambda/LambdaFunctionWithRDS`).
5. Wählen Sie im Abschnitt Protokollstreams den neuesten Protokollstream aus.

   Ihre Tabelle sollte zwei Kundendatensätze enthalten, einen aus jedem Aufruf Ihrer Funktion. Im Protokollstream werden Ihnen Nachrichten ähnlich der folgenden angezeigt:

```
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 The following
 items have been added to the database:
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1021, 'Martha
 Rivera')
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1054,
 'Richard Roe')
```

Bereinigen Ihrer Ressourcen

Sie können jetzt die Ressourcen, die Sie für dieses Tutorial erstellt haben, löschen, es sei denn, Sie möchten sie behalten. Indem Sie AWS Ressourcen löschen, die Sie nicht mehr verwenden, verhindern Sie, dass Ihr AWS Konto unnötig belastet wird.

So löschen Sie die Lambda-Funktion:

1. Öffnen Sie die Seite Funktionen der Lambda-Konsole.
2. Wählen Sie die Funktion aus, die Sie erstellt haben.
3. Wählen Sie Aktionen, Löschen aus.
4. Wählen Sie Löschen aus.

So löschen Sie die Ausführungsrolle

1. Öffnen Sie die Seite Roles in der IAM-Konsole.
2. Wählen Sie die von Ihnen erstellte Ausführungsrolle aus.
3. Wählen Sie Delete role (Rolle löschen) aus.
4. Wählen Sie Yes, delete (Ja, löschen) aus.

So löschen Sie die MySQL-DB-Instance

1. Öffnen Sie die Seite Datenbanken der Amazon-RDS-Konsole.
2. Wählen Sie die von Ihnen erstellte Datenbank aus.
3. Wählen Sie Aktionen, Löschen aus.
4. Deaktivieren Sie das Kontrollkästchen für Abschließenden Snapshot erstellen.
5. Geben Sie **`delete me`** in das Textfeld ein.

6. Wählen Sie Löschen aus.

So löschen Sie die Amazon-SQS-Warteschlange

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon SQS SQS-Konsole unter https://console.aws.amazon.com/sqs/.
2. Wählen Sie die Warteschlange aus, die Sie erstellt haben.
3. Wählen Sie Löschen aus.
4. Geben Sie **`delete`** in das Textfeld ein.
5. Wählen Sie Löschen aus.

Amazon RDS Tutorials und Beispiel-Code

Die AWS Dokumentation enthält mehrere Tutorials, die Sie durch gängige Amazon RDS Amazon führen. Viele dieser Tutorials zeigen Ihnen, wie Sie Amazon RDS Amazon mit anderen AWS Diensten verwenden können. Darüber hinaus können Sie in auf den Beispielcode zugreifen GitHub.

> **Note**
>
> Weitere Tutorials finden Sie im AWS -Datenbank-Blog. Informationen zu Schulungen finden Sie unter AWS Training and Certification.

Themen

- Tutorials in diesem Handbuch
- Tutorials in anderen AWS Leitfäden
- AWS Portal für Workshop- und Laborinhalte für Amazon RDS PostgreSQL
- AWS Portal für Workshop- und Laborinhalte für Amazon RDS MySQL
- Tutorials und Beispielcode in GitHub
- Verwenden Sie diesen Dienst mit einem AWS SDK

Tutorials in diesem Handbuch

Die folgenden Tutorials in diesem Handbuch veranschaulichen, wie Sie mit Amazon RDS gängige Aufgaben durchführen.

- Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur)

  Erfahren Sie, wie Sie eine DB-Instance in eine Virtual Private Cloud (VPC) aufnehmen, die auf dem Amazon-VPC-Service basiert. In diesem Fall teilt die VPC Daten mit einem Webserver, der auf einer EC2 Amazon-Instance in derselben VPC läuft.
- Tutorial: Erstellen einer VPC zur Verwendung mit einer DB–Instance (Dual-Stack-Modus)

  Erfahren Sie, wie Sie eine DB-Instance in eine Virtual Private Cloud (VPC) aufnehmen, die auf dem Amazon-VPC-Service basiert. In diesem Fall teilt die VPC Daten mit einer EC2 Amazon-Instance in derselben VPC. In diesem Tutorial erstellen Sie die VPC für dieses Szenario, die mit einer Datenbank arbeitet, die im Dual-Stack-Modus ausgeführt wird.

- Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance

  Weitere Informationen darüber, wie Sie einen Apache-Webserver mit PHP installieren und eine MySQL-Datenbank zu erstellen. Der Webserver läuft auf einer EC2 Amazon-Instance, die Amazon Linux verwendet, und die MySQL-Datenbank ist eine MySQL-DB-Instance, . Sowohl die EC2 Amazon-Instance als auch der werden in einer Amazon VPC ausgeführt.

- Tutorial: Eine RDS Amazon-DB-Instance aus einem DB-Snapshot wiederherstellen

  Weitere Informationen, wie Sie eine DB-Instance aus einem DB-Snapshot wiederherstellen.

- Tutorial: Verwenden einer Lambda-Funktion für den Zugriff auf eine Amazon-RDS-Datenbank

  Erfahren Sie, wie Sie über die RDS-Konsole eine Lambda Funktion erstellen, um über einen Proxy auf eine Datenbank zuzugreifen, eine Tabelle zu erstellen, einige Datensätze hinzuzufügen und die Datensätze aus der Tabelle abzurufen. Sie lernen auch, wie Sie die Lambda-Funktion aufrufen und die Abfrageergebnisse überprüfen.

- Tutorial: Geben Sie mithilfe von Tags an, welche DB-Instances gestoppt werden sollen

  Weitere Informationen, wie Sie Tags verwenden, um anzugeben, welche DB-Instances angehalten werden sollen.

- Tutorial: Statusänderungen der DB-Instance mithilfe von Amazon protokollieren EventBridge

  Erfahren Sie, wie Sie eine Änderung des DB-Instance-Status mithilfe von Amazon EventBridge und protokollieren AWS Lambda.

- Tutorial: Erstellen eines CloudWatch Amazon-Alarms für Multi-AZ-DB-Cluster-Replikatverzögerungen für Amazon RDS

  Erfahren Sie, wie Sie einen CloudWatch Alarm erstellen, der eine Amazon SNS SNS-Nachricht sendet, wenn die Replikatverzögerung für einen Multi-AZ-DB-Cluster einen Schwellenwert überschritten hat. Ein Alarm überwacht die `ReplicaLag`-Metrik über einen bestimmten, von Ihnen definierten Zeitraum. Die Aktion ist eine Benachrichtigung, die an ein Amazon SNS SNS-Thema oder eine Amazon EC2 Auto Scaling Scaling-Richtlinie gesendet wird.

Tutorials in anderen AWS Leitfäden

Die folgenden Tutorials in anderen AWS Handbüchern zeigen Ihnen, wie Sie allgemeine Aufgaben mit Amazon RDS ausführen:

- Tutorial: Ein Geheimnis für eine AWS Datenbank rotieren im AWS Secrets Manager Benutzerhandbuch

  Erfahren Sie, wie Sie ein Geheimnis für eine AWS Datenbank erstellen und das Geheimnis so konfigurieren, dass es nach einem Zeitplan rotiert. Sie lösen eine Rotation manuell aus und bestätigen dann, dass die neue Version des Secrets weiterhin Zugriff bietet.
- Tutorials und Beispiele im AWS Elastic Beanstalk -Entwicklerhandbuch

  Erfahren Sie, wie Sie Anwendungen bereitstellen, die Amazon RDS-Datenbanken verwenden AWS Elastic Beanstalk.
- Verwenden von Daten aus einer Amazon RDS-Datenbank zum Erstellen einer Amazon ML-Datenquelle im Amazon Machine Learning Developer Guide

  Erfahren Sie, wie Sie ein Amazon Machine Learning (Amazon ML)-Datenquellenobjekt aus Daten erstellen, die in einer MySQL-DB-Instance gespeichert sind.
- Manuelles Aktivieren des Zugriffs auf eine Amazon RDS-Instance in einer VPC im QuickSight Amazon-Benutzerhandbuch

  Erfahren Sie, wie Sie QuickSight den Zugriff auf eine Amazon RDS-DB-Instance in einer VPC aktivieren.

AWS Portal für Workshop- und Laborinhalte für Amazon RDS PostgreSQL

Die folgende Sammlung von Workshops und anderen praktischen Inhalten hilft Ihnen, die Funktionen und Möglichkeiten von Amazon RDS PostgreSQL zu verstehen:

- Erstellen einer DB-Instance

  Erfahren Sie, wie Sie eine DB-Instance erstellen.
- Leistungsüberwachung mit RDS-Tools

  Erfahren Sie, wie Sie SQL-Tools (Cloudwatch, Enhanced Monitoring, Slow Query Logs, Performance Insights, PostgreSQL Catalog Views) verwenden AWS , um Leistungsprobleme zu verstehen und Möglichkeiten zur Verbesserung der Leistung Ihrer Datenbank zu finden.

AWS Portal für Workshop- und Laborinhalte für Amazon RDS MySQL

Die folgende Sammlung von Workshops und anderen praktischen Inhalten hilft Ihnen, die Funktionen und Möglichkeiten von Amazon RDS MySQL zu verstehen:

- Erstellen einer DB-Instance

  Erfahren Sie, wie Sie eine DB-Instance erstellen.
- Verwenden von Performance-Insights

  Erfahren Sie, wie Sie Ihre DB-Instance mithilfe von Performance Insights überwachen und optimieren können.

Tutorials und Beispielcode in GitHub

Die folgenden Tutorials und der Beispielcode GitHub zeigen Ihnen, wie Sie allgemeine Aufgaben mit Amazon RDS ausführen:

- Erstellen der Elementverfolgung zum Amazon Relational Database Service

  Erfahren Sie, wie Sie eine Anwendung erstellen, die Arbeitselemente verfolgt und meldet. Diese Anwendung verwendet Amazon RDS, Amazon Simple Email Service, Elastic Beanstalk und SDK für Java 2.x.

Verwenden Sie diesen Dienst mit einem AWS SDK

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK für C++	AWS SDK für C++ Codebeispiele
AWS CLI	AWS CLI Code-Beispiele

SDK-Dokumentation	Codebeispiele
AWS SDK für Go	AWS SDK für Go Code-Beispiele
AWS SDK für Java	AWS SDK für Java Code-Beispiele
AWS SDK für JavaScript	AWS SDK für JavaScript Code-Beispiele
AWS SDK für Kotlin	AWS SDK für Kotlin Code-Beispiele
AWS SDK für .NET	AWS SDK für .NET Code-Beispiele
AWS SDK für PHP	AWS SDK für PHP Code-Beispiele
AWS -Tools für PowerShell	AWS -Tools für PowerShell Code-Beispiele
AWS SDK für Python (Boto3)	AWS SDK für Python (Boto3) Code-Beispiele
AWS SDK für Ruby	AWS SDK für Ruby Code-Beispiele
AWS SDK für Rust	AWS SDK für Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK für Swift	AWS SDK für Swift Code-Beispiele

Weitere Beispiele speziell für diesen Service finden Sie unter Codebeispiele für Amazon RDS mit AWS SDKs.

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

Bewährte Methoden für Amazon RDS

Informieren Sie sich über die bewährten Methoden für die Arbeit mit Amazon RDS. Dieser Abschnitt wird mit neuen bewährten Methoden aktualisiert, sobald diese bekannt sind.

Themen

- Grundlegende Anleitungen für den Amazon RDS-Betrieb
- RAM-Empfehlungen für DB-Instances
- Halten Sie die Versionen der Datenbank-Engine auf dem neuesten Stand
- AWS Datenbanktreiber
- Verwendung von „Enhanced Monitoring“ (Erweiterte Überwachung) zur Identifizierung von Betriebssystemproblemen
- Verwendung von Metriken zur Identifizierung von Problemen mit der Leistung
- Optimieren von Abfragen
- Best Practices für die Arbeit mit MySQL
- Best Practices für die Arbeit mit MariaDB
- Bewährte Methoden für die Arbeit mit Oracle
- Bewährte Methoden für die Arbeit mit PostgreSQL
- Bewährte Methoden für die Arbeit mit SQL Server
- Arbeiten mit DB-Parametergruppen
- Bewährte Methoden zur Automatisierung der DB-Instance-Erstellung
- Video zu den neuen Funktionen von Amazon RDS

> **Note**
>
> Allgemeine Empfehlungen für Amazon RDS finden Sie unter Empfehlungen von Amazon RDS.

Grundlegende Anleitungen für den Amazon RDS-Betrieb

Im Folgenden finden Sie einige grundlegende Anleitungen für die Ausführung, die bei der Arbeit mit Amazon RDS befolgt werden sollten. Beachten Sie, dass das Amazon RDS Service Level Agreement voraussetzt, dass Sie die folgenden Anleitungen befolgen:

- Verwenden Sie Metriken, um Speicher, CPU, Replikatverzögerung und Speichernutzung zu überwachen. Sie können Amazon so einrichten CloudWatch , dass Sie benachrichtigt werden, wenn sich die Nutzungsmuster ändern oder wenn Ihre Bereitstellung Kapazitätsgrenzen erreicht. Auf diese Weise können Sie die Systemleistung und Verfügbarkeit aufrechterhalten.
- Skalieren Sie Ihre DB-Instance, wenn Sie die Grenzen der Speicherkapazität beinahe erreicht haben. Sie sollten etwas Puffer in Speicher und Arbeitsspeicher haben, um unvorhergesehene Nachfragesteigerungen seitens Ihrer Anwendungen bewältigen zu können.
- Aktivieren Sie automatische Sicherungen und richten Sie das Sicherungsfenster so ein, dass Sicherungen während Zeiten mit nur wenigen Schreibvorgangs-IOPS ausgeführt werden. Dann ist eine Sicherung am wenigsten störend für Ihre Datenbanknutzung.
- Wenn Ihre Datenbank-Arbeitslast mehr erfordert, I/O als Sie bereitgestellt haben, ist die Wiederherstellung nach einem Failover oder Datenbankausfall langsam. Um die I/O Kapazität einer DB-Instance zu erhöhen, führen Sie einen oder alle der folgenden Schritte aus:
  - Migrieren Sie zu einer anderen DB-Instance-Klasse mit hoher I/O Kapazität.
  - Konvertieren Sie von einem magnetischen Speicher zu einem Speicher für allgemeine Zwecke oder zu einem Speicher mit bereitgestellten IOPS, abhängig davon, wie groß die benötigte Steigerung ist. Weitere Informationen zu den verfügbaren Speichertypen finden Sie unter Amazon RDS-Speichertypen.

    Wenn Sie zu einem Speicher mit bereitgestellten IOPS konvertieren, müssen Sie sicherstellen, dass Sie eine DB-Instance-Klasse verwenden, die für bereitgestellte IOPS optimiert ist. Weitere Informationen zu bereitgestellten IOPS finden Sie unter Bereitgestellter IOPS SSD-Speicher.
  - Wenn Sie bereits einen Speicher mit bereitgestellten IOPS verwenden, sollten Sie zusätzliche Durchsatzkapazitäten bereitstellen.
- Wenn Ihre Client-Anwendung die DNS-Daten (Domain Name Service) Ihrer DB-Instances zwischenspeichert, legen Sie einen time-to-live (TTL) -Wert von weniger als 30 Sekunden fest. Die zugrunde liegende IP-Adresse einer DB-Instance kann sich nach einem Failover ändern. Ein längeres Zwischenspeichern der DNS-Daten kann somit zu Verbindungsfehlern führen. Ihre Anwendung versucht möglicherweise, eine Verbindung zu einer IP-Adresse herzustellen, die nicht mehr in Betrieb ist.

- Testen Sie den Failover für Ihre DB-Instance, um zu verstehen, wie lange der Vorgang für Ihren besonderen Anwendungsfall dauert. Testen Sie außerdem die Failover-Funktion, um sicherzustellen, dass die Anwendung, mit der auf Ihre DB-Instance zugegriffen wird, nach einem Failover automatisch eine Verbindung mit der neuen DB-Instance herstellen kann.

RAM-Empfehlungen für DB-Instances

Eine bewährte Methode im Zusammenhang mit der Verbesserung der Leistung von Amazon RDS besteht in der Zuteilung von ausreichend RAM, damit sich Ihr Arbeitssatz beinahe vollständig im Arbeitsspeicher befindet. Der Arbeitssatz umfasst die Daten und Indizes, die häufig auf Ihrer Instance verwendet werden. Je häufiger Sie die DB-Instance verwenden, desto größer wird der Arbeitssatz.

Um festzustellen, ob sich Ihr Arbeitssatz fast vollständig im Arbeitsspeicher befindet, überprüfen Sie die ReadIOps-Metrik (mit Amazon CloudWatch), während die DB-Instance ausgelastet ist. Der Wert von ReadIOPS sollte klein und stabil sein. In einigen Fällen führt die Skalierung der DB-Instance-Klasse auf eine Klasse mit mehr RAM zu einer deutlichen Abnahme des ReadIOPS-Werts. In diesen Fällen war Ihr Arbeitssatz nicht fast vollständig im Speicher. Skalieren Sie weiter, bis der ReadIOPS-Wert nach einer Skalierungsoperation nicht mehr deutlich abnimmt oder zu einem sehr kleinen Wert reduziert wird. Informationen zur Überwachung der Metriken einer DB-Instance finden Sie unter Metriken in der RDS Amazon-Konsole anzeigen.

Halten Sie die Versionen der Datenbank-Engine auf dem neuesten Stand

Aktualisieren Sie Ihre Datenbank-Engine-Version regelmäßig, um Sicherheit, Leistung und Compliance zu gewährleisten. Amazon RDS veröffentlicht neue Neben- und Hauptversionen, die Sicherheitspatches, Leistungsverbesserungen und neue Funktionen enthalten. Der Betrieb einer veralteten Datenbank-Engine kann Ihre Workloads bekannten Sicherheitslücken, Kompatibilitätsproblemen und eingeschränktem Support durch AWS Datenbankanbieter aussetzen.

Um Unterbrechungen zu minimieren, sollten Sie bei der Planung von Upgrades Folgendes berücksichtigen:

- Testen Sie in einer Staging-Umgebung — Überprüfen Sie die neue Version anhand Ihrer Arbeitslast, bevor Sie die Produktionsdatenbanken aktualisieren.
- Verwenden Sie von Amazon RDS verwaltete Upgrades — Aktivieren Sie automatische Upgrades für kleinere Versionen, um das Patchen zu vereinfachen.

- Planen Sie größere Versions-Upgrades — Lesen Sie die Versionshinweise, testen Sie die Anwendungskompatibilität und planen Sie ein kontrolliertes Upgrade-Fenster ein.

Regelmäßige Upgrades tragen dazu bei, dass Ihre Datenbank sicher und optimiert bleibt und den AWS Best Practices entspricht.

AWS Datenbanktreiber

Wir empfehlen die AWS Treibersuite für die Anwendungskonnektivität. Die Treiber wurden so konzipiert, dass sie schnellere Switchover- und Failover-Zeiten sowie Authentifizierung mit AWS Secrets Manager, AWS Identity and Access Management (IAM) und Federated Identity unterstützen. Die AWS Treiber sind auf die Überwachung des DB-Instance-Status und die Kenntnis der Instance-Topologie angewiesen, um den neuen Writer zu ermitteln. Dieser Ansatz reduziert die Switchover- und Failover-Zeiten auf einstellige Sekunden, im Vergleich zu mehreren zehn Sekunden bei Open-Source-Treibern.

Bei der Einführung neuer Servicefunktionen besteht das Ziel der AWS Treibersuite darin, eine integrierte Unterstützung für diese Servicefunktionen zu bieten.

Weitere Informationen finden Sie unter Mit den AWS Treibern eine Verbindung zu DB-Instances herstellen.

Verwendung von „Enhanced Monitoring“ (Erweiterte Überwachung) zur Identifizierung von Betriebssystemproblemen

Wenn „Enhanced Monitoring“ (Erweiterte Überwachung) aktiviert ist, stellt Amazon RDS Metriken in Echtzeit für das Betriebssystem (OS) bereit, auf dem Ihre DB-Instance ausgeführt wird. Sie können die Metriken für Ihre DB-Instance über die Konsole anzeigen. Sie können die JSON-Ausgabe von Enhanced Monitoring von Amazon CloudWatch Logs auch in einem Überwachungssystem Ihrer Wahl verwenden. Weitere Informationen zu „Enhanced Monitoring“ (Erweiterte Überwachung) finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Verwendung von Metriken zur Identifizierung von Problemen mit der Leistung

Sie können die Metriken überwachen, die für Ihre Amazon RDS-DB-Instance verfügbar sind, um Leistungsprobleme aufgrund unzureichender Ressourcen und anderer häufiger Engpässe zu identifizieren.

Anzeigen von Leistungsmetriken

Sie sollten die Leistungsmetriken regelmäßig überwachen, um die Durchschnitts-, Höchst- und Mindestwerte für eine Vielzahl von Zeitbereichen anzuzeigen. Wenn Sie dies tun, können Sie feststellen, wenn die Leistung nachlässt. Sie können CloudWatch Amazon-Alarme auch für bestimmte Metrik-Schwellenwerte einrichten, sodass Sie benachrichtigt werden, wenn diese erreicht werden.

Um Probleme mit der Leistung zu beheben, müssen Sie die Basisleistung des Systems kennen. Wenn Sie eine DB-Instance einrichten und mit einer typischen Workload ausführen, erfassen Sie die Durchschnitts-, Maximal- und Minimalwerte aller Leistungsmetriken. Führen Sie diesen Vorgang in verschiedenen Intervallen aus (z. B. eine Stunde, 24 Stunden, eine Woche, zwei Wochen). Auf diese Weise erhalten Sie eine Vorstellung davon, was normal ist. Dies hilft, um Vergleichswerte für Betriebsstunden während und außerhalb von Spitzenbelastungen zu erhalten. Sie können diese Informationen anschließend verwenden, um festzustellen, wann die Leistung unter Standardwerte absinkt.

Wenn Sie Multi-AZ-DB-Cluster verwenden, überwachen Sie die Zeitdifferenz zwischen der letzten Transaktion auf der Writer-DB-Instance und der zuletzt angewendeten Transaktion auf einer Reader-DB-Instance. Dieser Unterschied heißt replica lag (Replikatverzögerung). Weitere Informationen finden Sie unter Replikatverzögerung und Multi-AZ-DB-Cluster.

Sie können die kombinierten Performance Insights und CloudWatch Metriken im Performance Insights Insights-Dashboard einsehen und Ihre DB-Instance überwachen. Performance Insights muss für Ihre DB-Instance aktiviert sein, damit diese Ansicht verwendet werden kann. Weitere Informationen zu dieser Überwachungsansicht finden Sie unter Kombinierte Metriken mit dem Performance Insights Insights-Dashboard anzeigen.

Sie können einen Leistungsanalysebericht für einen bestimmten Zeitraum erstellen und sich die ermittelten Erkenntnisse und Empfehlungen zur Lösung der Probleme ansehen. Weitere Informationen finden Sie unter Erstellen eines Leistungsanalyseberichts in Performance Insights.

So können Sie sich die Leistungsmessungen anzeigen

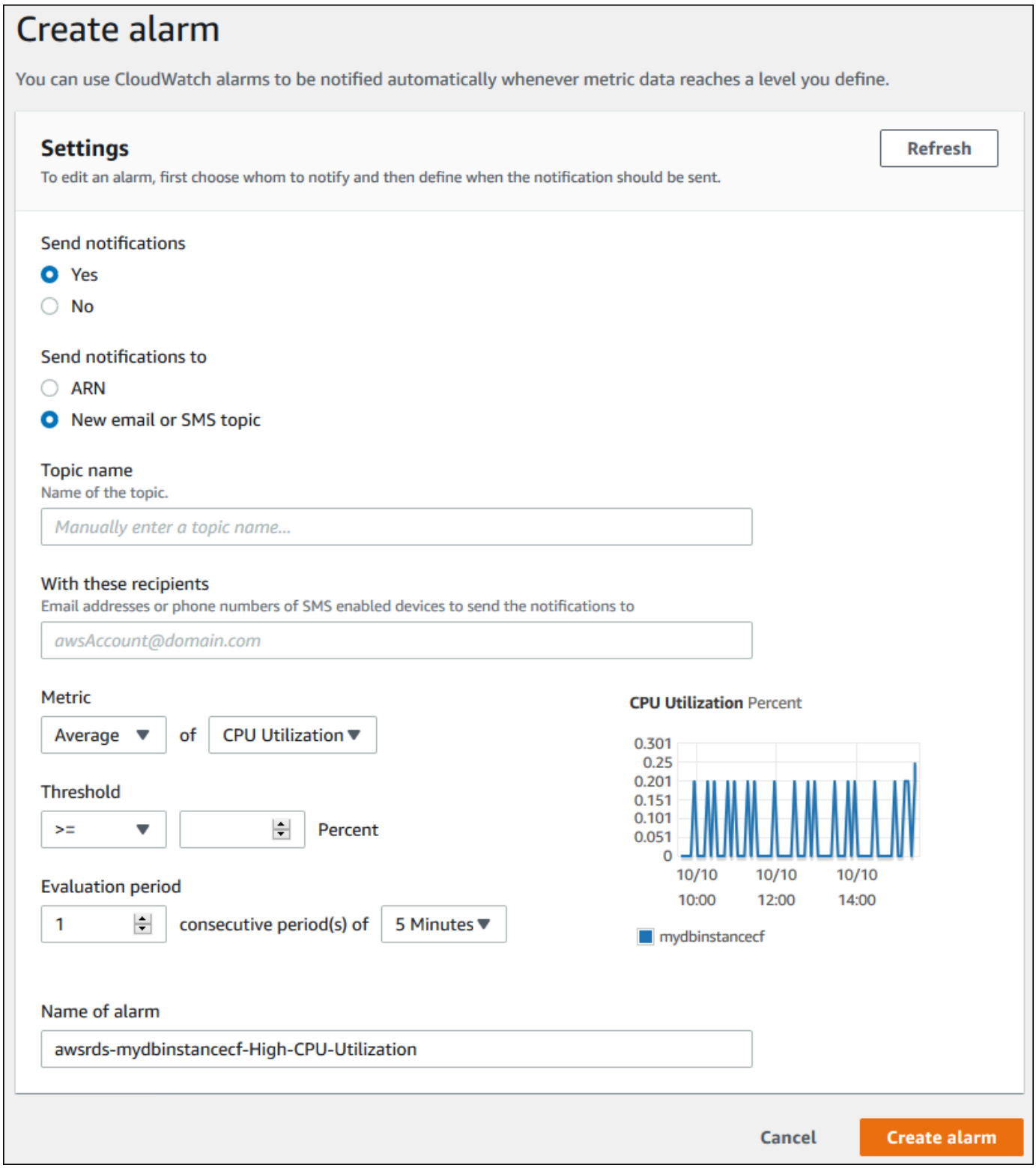
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Databases (Datenbanken) und anschließend eine DB-Instance.
3. Wählen Sie Monitoring.

   Das Dashboard stellt die Leistungsmetriken bereit. Standardmäßig werden die Metriken der letzten drei Stunden angezeigt.
4. Verwenden Sie die nummerierten Schaltflächen oben rechts, um durch die zusätzlichen Metriken zu blättern, oder passen Sie die Einstellungen an, um weitere Metriken anzuzeigen.
5. Wählen Sie eine Leistungsmetrik zur Anpassung des Zeitbereichs, um Daten für andere Tage als den aktuellen Tag anzuzeigen. Sie können die Werte für Statistik, Zeitraum und Intervall ändern, um die angezeigten Informationen anzupassen. Angenommen, Sie möchten beispielsweise die Spitzenwerte für eine Metrik für jeden Tag der letzten zwei Wochen anzeigen. Legen Sie in diesem Fall Statistic (Statistik) auf Maximum, Time Range (Zeitbereich) auf Last 2 Weeks (Letzte 2 Wochen) und Period (Zeitraum) auf Day (Tag) fest.

Sie können die Leistungsmetriken auch über die CLI oder API anzeigen. Weitere Informationen finden Sie unter Metriken in der RDS Amazon-Konsole anzeigen.

Um einen CloudWatch Alarm einzustellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Databases (Datenbanken) und anschließend eine DB-Instance.
3. Wählen Sie Logs & Events (Protokolle und Ereignisse).
4. Wählen Sie im Bereich CloudWatch Alarme die Option Alarm erstellen aus.



5. Wählen Sie für Send notifications (Benachrichtigungen senden) Yes (Ja) und für Send notifications to (Benachrichtigungen senden an= New email or SMS topic (Neue E-Mail oder SMS Thema).

6. Geben Sie unter Topic name (Themaname) einen Namen für die Benachrichtigung und unter With these recipients (Mit diesen Empfängern) eine kommagetrennte Liste von E-Mail-Adressen und Telefonnummern ein.
7. Wählen Sie für Metric (Metrik) die einzustellende Alarmstatistik und Metrik.
8. Geben Sie für Threshold (Schwellenwert) an, ob die Metrik größer, kleiner oder gleich dem Schwellenwert sein muss, und geben Sie den Schwellenwert an.
9. Wählen Sie unter Period (Zeitraum) den Auswertungszeitraum für den Alarm aus. Wählen Sie für consecutive period(s) of (aufeinanderfolgende Periode(n) von) den Zeitraum aus, in dem der Schwellenwert erreicht worden sein muss, um den Alarm auszulösen.
10. Geben Sie unter Alarmname einen aussagekräftigen Namen für Ihren Alarm ein.
11. Wählen Sie Create Alarm aus.

Der Alarm wird im Bereich CloudWatch Alarme angezeigt.

Auswerten von Leistungsmetriken

Eine DB-Instance besitzt eine Reihe unterschiedlicher Kategorien von Metriken. Die Entscheidung, welche Werte akzeptabel sind, ist von der Metrik abhängig.

CPU

- CPU-Nutzung – Prozentsatz der verwendeten Verarbeitungskapazität des Computers.

Arbeitsspeicher

- Freier Speicher — Wie viel RAM in Byte auf der DB-Instance verfügbar ist. Die rote Linie in der Metrik der Registerkarte Monitoring (Überwachung) kennzeichnet 75 % für CPU-, Arbeitsspeicher- und Speichermetriken. Wenn der Speicherverbrauch der Instance diese Linie häufig überschreitet, bedeutet dies, dass Sie die Workload prüfen sollten oder die Instance aktualisieren müssen.
- Swap-Nutzung — Wie viel Swap-Speicherplatz von der DB-Instance verwendet wird, in Byte.

Festplattenkapazität

- Freier Speicherplatz – Größe des Datenträgerbereichs, der zurzeit in der DB-Instance nicht verwendet wird (in Megabytes).

Eingabe-/Ausgabe-Operationen

- Lese-IOPS, Schreib-IOPS – die durchschnittliche Anzahl der Lese- oder Schreib-Datenträgeroperationen pro Sekunde.
- Leselatenz, Schreiblatenz – die durchschnittliche Zeit, die eine Lese- oder Schreiboperation benötigt (in Millisekunden).
- Lesedurchsatz, Schreibdurchsatz – die durchschnittliche Anzahl der Megabytes, die pro Sekunde aus dem Datenträger gelesen oder zum Datenträger geschrieben werden.
- Warteschlangentiefe — Die Anzahl der I/O Operationen, die darauf warten, auf die Festplatte geschrieben oder von ihr gelesen zu werden.

Netzwerkdatenverkehr

- Netzwerkeingangsdurchsatz, Netzwerkübertragungsdurchsatz – die Rate des Netzwerkdatenverkehrs zur und von der DB-Instance (in Bytes pro Sekunde).

Datenbankverbindungen

- Datenbankverbindungen – die Anzahl der Client-Sitzungen, die mit der DB-Instance verbunden sind.

Detailliertere einzelne Beschreibungen der verfügbaren Leistungsmetriken finden Sie unter Überwachung von Amazon RDS mit Amazon CloudWatch.

Allgemein ausgedrückt, sind die zulässigen Werte für Leistungsmetriken davon abhängig, wie die Basisleistung aussieht und welche Aufgaben von Ihrer Anwendung ausgeführt werden. Prüfen Sie, ob dauerhafte oder tendenzielle Abweichungen von Ihrer Ausgangsbasis vorliegen. Im Folgenden finden Sie ein paar Hinweise zu bestimmten Metriken:

- Hohe CPU- oder RAM-Nutzung – Hohe Werte für die CPU- oder RAM-Nutzung können angemessen sein. Dies kann zum Beispiel der Fall sein, wenn sie der Zielsetzung Ihrer Anwendung entsprechen (z. B. in Bezug auf Durchsatz oder Gleichzeitigkeit) und erwartet werden.
- Nutzung des Datenträgerplatzes – Überprüfen Sie die Nutzung des Datenträgerplatzes, wenn konsistent 85 Prozent oder mehr des gesamten Datenträgerplatzes belegt werden. Prüfen Sie, ob Daten in der Instance gelöscht oder auf einem anderen System archiviert werden können, um Speicherplatz freizugeben.

- Netzwerkdatenverkehr – Wenden Sie sich an Ihren Systemadministrator, um zu erfahren, welcher Durchsatz für Ihr Domänennetzwerk und Ihre Internetverbindung erwartet wird. Überprüfen Sie den Netzwerkdatenverkehr, wenn der Durchsatz dauerhaft unter dem erwarteten Wert liegt.
- Datenbankverbindungen – Ziehen Sie eine Einschränkung der Datenbankverbindungen in Betracht, wenn bei einer großen Anzahl von Benutzerverbindungen eine Abnahme der Instance-Leistung und der Reaktionszeit zu erkennen ist. Die optimale Anzahl der Benutzerverbindungen für Ihre DB-Instance ist von der Instance-Klasse und der Komplexität der Operationen abhängig, die ausgeführt werden. Wenn Sie die Anzahl der Datenbankverbindungen bestimmen möchten, ordnen Sie Ihre DB-Instance einer Parametergruppe zu. Legen Sie in dieser Gruppe den Parameter User Connections auf einen anderen Wert als 0 (unbegrenzt) fest. Sie können eine entweder eine vorhandene Parametergruppe verwenden oder eine neue erstellen. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
- IOPS-Metriken – Die erwarteten Werte für IOPS-Metriken sind von der Datenträgerspezifikation und der Serverkonfiguration abhängig. Verwenden Sie die Basiswerte als typische Werte. Prüfen Sie, ob dauerhafte Abweichungen von den Werten Ihrer Ausgangsbasis vorliegen. Sie erzielen eine optimale IOPS-Leistung, wenn Sie sicherstellen, dass die typischen zu verarbeitenden Datensätze komplett in den Arbeitsspeicher geladen werden können, sodass die Lese- und Schreibvorgänge auf ein Minimum beschränkt werden.

Bei Problemen mit Leistungsmetriken sollten Sie zunächst die am häufigsten verwendeten und kostspieligsten Abfragen anpassen, um die Leistung zu verbessern. Optimieren Sie sie, um festzustellen, ob dies den Druck auf die Systemressourcen verringert. Weitere Informationen finden Sie unter Optimieren von Abfragen.

Wenn Ihre Abfragen optimiert sind und ein Problem weiterhin besteht, sollten Sie ein Upgrade Ihrer von Amazon RDS in Betracht ziehen. Sie können es auf eine Version mit mehr Ressourcen (CPU, RAM, Festplattenspeicher, Netzwerkbandbreite, I/O Kapazität) aktualisieren, die mit dem Problem zusammenhängen.

Optimieren von Abfragen

Eine der besten Möglichkeiten zur Verbesserung der Leistung von DB-Instances besteht darin, die am häufigsten verwendeten und ressourcenintensivsten Abfragen zu optimieren. Hier optimieren Sie sie, damit sie kostengünstiger ausgeführt werden können. Verwenden Sie die folgenden Ressourcen, um Informationen zur Verbesserung von Abfragen zu erhalten:

- MySQL – Siehe SELECT-Anweisungen in der MySQL-Dokumentation optimieren. Weitere Ressourcen zur Abfrageoptimierung finden Sie unter MySQL-Performance-Tuning- und Optimierungsressourcen.
- Oracle – Siehe Database SQL Tuning Guide in der Oracle Datenbank-Dokumentation.
- SQL Server – Siehe Analysieren einer Abfrage in der Microsoft-Dokumentation. Sie können auch die ausführungs-, index- und I/O-bezogenen Datenverwaltungsansichten (DMVs) verwenden, die in der Microsoft-Dokumentation unter Dynamische Systemverwaltungsansichten beschrieben sind, um Probleme mit SQL Server-Abfragen zu beheben.

  Ein häufiger Aspekt beim Optimieren von Abfragen stellt die Erstellung effektiver Indizes dar. Weitere Informationen zu möglichen Indexverbesserungen für Ihre DB-Instance finden Sie in der Microsoft-Dokumentation unter Database Engine Tuning Ad Informationen zur Verwendung von Tuning Advisor für finden Sie unter . RDS for SQL Serv Analysieren Ihrer Datenbank-Arbeitslast auf einer Amazon RDS for SQL Server-DB-Instance mit Database Engine Tuning Advisor.
- PostgreSQL – Siehe EXPLAIN verwenden in der PostgreSQL-Dokumentation, um zu erfahren, wie man einen Abfrageplan analysiert. Sie können diese Informationen verwenden, um eine Abfrage oder zugrundeliegende Tabellen zu ändern, um die Abfrageleistung zu verbessern.

  Informationen darüber, wie Sie Joins in Ihrer Abfrage für die beste Leistung angeben, finden Sie unter Steuern des Planers mit expliziten JOIN-Klauseln.
- MariaDB – Siehe Abfrageoptimierungen in der MariaDB-Dokumentation.

Best Practices für die Arbeit mit MySQL

Sowohl die Tabellengröße als auch die Anzahl der Tabellen in einer MySQL-Datenbank können die Leistung beeinträchtigen.

Tabellengröße

In der Regel bestimmen Betriebssystemeinschränkungen für Dateigrößen die effektive maximale Tabellengröße für MySQL-Datenbanken. Daher werden die Grenzwerte normalerweise nicht durch interne MySQL-Einschränkungen bestimmt.

Vermeiden Sie es in MySQL-DB-Instances, dass Tabellen in Ihrer Datenbank zu groß werden. Obwohl die allgemeine Speichergrenze 64 TiB beträgt, beschränken die Begrenzungen für den bereitgestellten Speicher die maximale Größe einer MySQL-Tabellendatei auf 16 TiB. Partitionieren

Sie Ihre großen Tabellen so, dass die Dateigrößen deutlich unter der 16-TiB-Grenze liegen. Dieser Ansatz kann auch Leistung und Wiederherstellungszeit verbessern. Weitere Informationen finden Sie unter MySQL-Dateigrößenlimits in Amazon RDS.

Sehr große Tabellen (mehr als 100 GB) können sich negativ auf die Leistung sowohl bei Lese- als auch bei Schreibvorgängen auswirken (einschließlich DML-Anweisungen und insbesondere DDL-Anweisungen). Indizes für große Tabellen können die Select-Performance erheblich verbessern, sie können jedoch auch die Leistung von DML-Anweisungen beeinträchtigen. DDL-Anweisungen wie `ALTER TABLE` können für die großen Tabellen erheblich langsamer sein, da diese Vorgänge in einigen Fällen eine Tabelle vollständig neu aufbauen können. Diese DDL-Anweisungen könnten die Tabellen für die Dauer der Operation sperren.

Die Menge an Speicher, die MySQL für Lese- und Schreibvorgänge benötigt, hängt von den Tabellen ab, die an den Vorgängen beteiligt sind. Es ist eine bewährte Methode, mindestens genug RAM zu haben, um die Indizes aktiv genutzter Tabellen zu halten. Verwenden Sie die folgende Abfrage, um die zehn größten Tabellen und Indizes in einer Datenbank zu finden:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

Anzahl der Tabellen

Das zugrunde liegende Dateisystem hat möglicherweise eine Begrenzung für die Anzahl der Dateien, die Tabellen darstellen. MySQL hat jedoch keine Begrenzung für die Anzahl der Tabellen. Trotzdem kann die Gesamtzahl der Tabellen in der InnoDB-Speicher-Engine von MySQL unabhängig von der Größe dieser Tabellen zu Leistungseinbußen beitragen. Um die Auswirkungen auf das Betriebssystem zu begrenzen, können Sie die Tabellen auf mehrere Datenbanken in derselben MySQL-DB-Instance aufteilen. Dies könnte die Anzahl der Dateien in einem Verzeichnis begrenzen, löst jedoch nicht das Gesamtproblem.

Wenn es aufgrund einer großen Anzahl von Tabellen (mehr als 10 000) zu Leistungseinbußen kommt, wird dies dadurch verursacht, dass MySQL mit Speicherdateien arbeitet, sie öffnet und schließt. Um dieses Problem zu lösen, können Sie die Größe der Parameter `table_open_cache`

und `table_definition_cache` erhöhen. Eine Erhöhung der Parameterwerte kann jedoch die Menge des von MySQL verwendeten Speichers erheblich erhöhen und kann sogar den gesamten verfügbaren Speicher verwenden. Weitere Informationen finden Sie unter How MySQL Opens and Closes Tables (Wie MySQL Tabellen öffnet und schließt) in der MySQL-Dokumentation.

Darüber hinaus können sich zu viele Tabellen erheblich auf die Startzeit von MySQL auswirken. Sowohl ein sauberes Herunterfahren als auch ein Neustart sowie eine Absturzwiederherstellung können insbesondere in Versionen vor MySQL 8.0 beeinträchtigt werden.

Wir empfehlen, insgesamt weniger als 10 000 Tabellen in allen Datenbanken einer DB-Instance zu speichern. Informationen zu einem Anwendungsfall mit einer großen Anzahl von Tabellen in einer MySQL-Datenbank finden Sie unter One Million Tables in MySQL 8.0 (1 Million Tabellen in MySQL 8.0).

Speicher-Engine

Die point-in-time Wiederherstellungs- und Snapshot-Wiederherstellungsfunktionen von Amazon RDS for MySQL erfordern eine Speicher-Engine, die nach einem Absturz wiederhergestellt werden kann. Diese Funktionen werden nur für die InnoDB-Speicher-Engine unterstützt. MySQL unterstützt zwar verschiedene Speichermodule mit unterschiedlichen Kapazitäten. Von diesen sind jedoch nicht alle für die Wiederherstellung nach einem Absturz und Datenbeständigkeit optimiert. Beispielsweise unterstützt die MyISAM-Speicher-Engine keine zuverlässige Wiederherstellung nach einem Absturz und kann verhindern, dass eine point-in-time Wiederherstellung oder Snapshot-Wiederherstellung wie beabsichtigt funktioniert. Dies kann dazu führen, dass Daten verloren gehen oder beschädigt werden, wenn MySQL nach einem Absturz erneut gestartet wird.

InnoDB ist das empfohlene und unterstützte Speichermodul für MySQL-DB-Instances in Amazon RDS. InnoDB-Instances können auch zu Aurora migriert werden, während MyISAM-Instances nicht migriert werden können. MyISAM bietet jedoch eine bessere Leistung als InnoDB, wenn Sie intensive Volltextsuchfunktionen benötigen. Wenn Sie dennoch MyISAM mit Amazon RDS verwenden möchten, kann die Befolgung der unter Automatisierte Backups mit nicht unterstützten My SQL Storage Engines beschriebenen Schritte in bestimmten Szenarien helfen, eine Snapshot-Wiederherstellungsfunktion zu erhalten.

Wenn Sie bestehende MyISAM-Tabellen in InnoDB-Tabellen konvertieren möchten, können Sie den in Converting Tables from MyISAM to InnoDB beschriebenen Prozess in der MySQL-Dokumentation verwenden. MyISAM und InnoDB haben verschiedene Vor- und Nachteile. Daher sollten Sie die Auswirkungen vollständig auswerten, bevor Sie den Wechsel für Ihre Anwendungen ausführen.

Darüber hinaus wird die Federated Storage Engine aktuell von Amazon RDS for MySQL nicht unterstützt.

Best Practices für die Arbeit mit MariaDB

Sowohl Tabellengrößen als auch die Anzahl der Tabellen in einer MariaDB-Datenbank können sich auf die Performance auswirken.

Tabellengröße

In der Regel bestimmen Betriebssystemeinschränkungen für Dateigrößen die effektive maximale Tabellengröße für MariaDB-Datenbanken. Daher werden die Grenzwerte normalerweise nicht durch interne MariaDB-Einschränkungen festgelegt.

Vermeiden Sie es in MariaDB-Instances, dass Tabellen in Ihrer Datenbank zu groß werden. Obwohl die allgemeine Speichergrenze 64 TiB beträgt, beschränken die Begrenzungen für den bereitgestellten Speicher die maximale Größe einer MariaDB-Tabellendatei auf 16 TiB. Partitionieren Sie Ihre großen Tabellen so, dass die Dateigrößen deutlich unter der 16-TiB-Grenze liegen. Dieser Ansatz kann auch Leistung und Wiederherstellungszeit verbessern.

Sehr große Tabellen (mehr als 100 GB) können sich negativ auf die Leistung sowohl bei Lese- als auch bei Schreibvorgängen auswirken (einschließlich DML-Anweisungen und insbesondere DDL-Anweisungen). Indizes für große Tabellen können die Select-Performance erheblich verbessern, sie können jedoch auch die Leistung von DML-Anweisungen beeinträchtigen. DDL-Anweisungen wie `ALTER TABLE` können für die großen Tabellen erheblich langsamer sein, da diese Vorgänge in einigen Fällen eine Tabelle vollständig neu aufbauen können. Diese DDL-Anweisungen könnten die Tabellen für die Dauer der Operation sperren.

Die Menge an Speicher, die MariaDB für Lese- und Schreibvorgänge benötigt, hängt von den an den Operationen beteiligten Tabellen ab. Es ist eine bewährte Methode, mindestens genug RAM zu haben, um die Indizes aktiv genutzter Tabellen zu halten. Verwenden Sie die folgende Abfrage, um die zehn größten Tabellen und Indizes in einer Datenbank zu finden:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
```

```
order by 3 desc ) a
order by 3 desc
limit 10;
```

Anzahl der Tabellen

Das zugrunde liegende Dateisystem hat möglicherweise eine Begrenzung für die Anzahl der Dateien, die Tabellen darstellen. MariaDB hat jedoch keine Begrenzung für die Anzahl der Tabellen. Trotzdem kann die Gesamtzahl der Tabellen in der InnoDB-Speicher-Engine von MariaDB unabhängig von der Größe dieser Tabellen zu Leistungseinbußen beitragen. Um die Auswirkungen auf das Betriebssystem zu begrenzen, können Sie die Tabellen auf mehrere Datenbanken in derselben MariaDB-Instance aufteilen. Dies könnte die Anzahl der Dateien in einem Verzeichnis begrenzen, löst jedoch nicht das Gesamtproblem.

Wenn es aufgrund einer großen Anzahl von Tabellen (mehr als 10 000) zu Leistungseinbußen kommt, wird dies dadurch verursacht, dass MariaDB mit Speicherdateien arbeitet. Diese Arbeit umfasst das Öffnen und Schließen von Speicherdateien durch MariaDB. Um dieses Problem zu lösen, können Sie die Größe der Parameter `table_open_cache` und `table_definition_cache` erhöhen. Eine Erhöhung der Werte dieser Parameter könnte jedoch die Menge des von MariaDB verwendeten Speichers erheblich erhöhen. Es könnte sogar der gesamte verfügbare Speicher belegt werden. Weitere Informationen finden Sie unter Optimizing table_open_cache (table_open_cache optimieren) in der MariaDB-Dokumentation.

Darüber hinaus können zu viele Tabellen die Startzeit von MariaDB erheblich beeinflussen. Sowohl ein sauberes Herunterfahren und Neustart als auch eine Absturzwiederherstellung können beeinträchtigt werden. Wir empfehlen, insgesamt weniger als zehntausend Tabellen in allen Datenbanken einer DB-Instance zu haben.

Speicher-Engine

Die point-in-time Wiederherstellungs- und Snapshot-Wiederherstellungsfunktionen von Amazon RDS for MariaDB erfordern eine Speicher-Engine, die nach einem Absturz wiederhergestellt werden kann. MariaDB unterstützt zwar mehrere Speicher-Engines mit unterschiedlichen Fähigkeiten und Kapazitäten, jedoch sind nicht alle von ihnen für die Wiederherstellung nach Ausfall und für Datenbeständigkeit optimiert. Aria ist zwar ein absturzsicherer Ersatz für MyISAM, kann aber dennoch verhindern, dass eine point-in-time Wiederherstellung oder Snapshot-Wiederherstellung wie beabsichtigt funktioniert. Dies kann dazu führen, dass Daten verloren gehen oder beschädigt werden, wenn MariaDB nach einem Absturz erneut gestartet wird. InnoDB ist das empfohlene und

unterstützte Speichermodul für MariaDB-DB-Instances in Amazon RDS. Wenn Sie dennoch Aria mit Amazon RDS verwenden möchten, kann die Befolgung der unter Automatisierte Backups mit nicht unterstützten MariaDB-Speicher-Engines beschriebenen Schritte in bestimmten Szenarien helfen, eine Snapshot-Wiederherstellungsfunktion zu erhalten.

Wenn Sie bestehende MyISAM-Tabellen in InnoDB-Tabellen konvertieren möchten, können Sie den in Converting Tables from MyISAM to InnoDB beschriebenen Prozess in der MariaDB-Dokumentation verwenden. MyISAM und InnoDB haben verschiedene Vor- und Nachteile. Daher sollten Sie die Auswirkungen vollständig auswerten, bevor Sie den Wechsel für Ihre Anwendungen ausführen.

Bewährte Methoden für die Arbeit mit Oracle

Informationen zu bewährten Methoden für die Arbeit mit Amazon RDS for Oracle finden Sie unter Bewährte Methoden für die Ausführung von Oracle-Datenbanken in Amazon Web Services.

Ein AWS virtueller Workshop im Jahr 2020 beinhaltete eine Präsentation über den Betrieb von Oracle-Produktionsdatenbanken auf Amazon RDS. Das Video der Präsentation ist hier verfügbar.

Bewährte Methoden für die Arbeit mit PostgreSQL

Es gibt zwei wichtige Bereiche, in denen Sie die Leistung von RDS für PostgreSQL verbessern können. Einer davon ist das Laden von Daten in eine DB-Instance. Der andere Bereich betrifft die Verwendung der PostgreSQL-Selbstbereinigungsfunktion. In den folgenden Abschnitten werden einige der empfohlenen Verfahren für diese Bereiche beschrieben.

Informationen darüber, wie andere häufige PostgreSQL-DBA-Aufgaben Amazon RDS implementiert werden, finden Sie unter Häufige DBA-Aufgaben für Amazon RDS for PostgreSQL.

Laden von Daten in eine PostgreSQL-DB-Instance

Beim Laden von Daten in eine DB-Instance von Amazon RDS für PostgreSQL sollten Sie Ihre DB-Instance-Einstellungen und Ihre DB-Parametergruppenwerte ändern. Legen Sie diese fest, um den effizientesten Import von Daten in Ihre DB-Instance zu ermöglichen.

Ändern Sie die Einstellungen für Ihre DB-Instance wie folgt:

- Deaktivieren Sie die DB-Instance-Sicherungen (Sicherungsaufbewahrung auf 0 setzen).
- Deaktivieren Sie Multi-AZ.

Modifizieren Sie die DB-Parametergruppe so, dass sie die folgenden Einstellungen enthält. Testen Sie außerdem die Parametereinstellungen, um die effizientesten Einstellungen für Ihre DB-Instance zu ermitteln.

- Erhöhen Sie den Wert des Parameters `maintenance_work_mem`. Weitere Informationen zu PostgreSQL-Ressourcennutzungsparametern finden Sie in der PostgreSQL-Dokumentation.
- Erhöhen Sie den Wert der Parameter `max_wal_size` und `checkpoint_timeout`, um die Zahl der Schreibvorgänge zum Write-Ahead (WAL)-Protokoll zu reduzieren.
- Parameter `synchronous_commit` deaktivieren.
- Deaktivieren Sie den PostgreSQL-Selbstbereinigungsparameter.
- Stellen Sie sicher, dass sämtliche Tabellen, die Sie importieren, protokolliert sind. In nicht protokollierten Tabellen gespeicherte Daten können bei einem Failover verloren gehen. Weitere Informationen finden Sie unter CREATE TABLE UNLOGGED.

Verwenden Sie den Befehl `pg_dump -Fc` (komprimiert) oder den Befehl `pg_restore -j` (parallel) mit diesen Einstellungen.

Nachdem der Ladevorgang abgeschlossen ist, setzen Sie Ihre DB-Instance und DB-Parameter auf ihre normalen Einstellungen zurück.

Arbeiten mit der PostgreSQL-Selbstbereinigungsfunktion

Die Selbstbereinigungsfunktion für PostgreSQL-Datenbanken ist eine Funktion, deren Verwendung nachdrücklich empfohlen wird, um die Integrität Ihrer PostgreSQL-DB-Instance zu wahren.
Die Selbstbereinigung automatisiert die Ausführung der Befehle VACUUM und ANALYZE. Die Verwendung der Selbstbereinigung wird von PostgreSQL erfordert, nicht von Amazon RDS auferlegt, und ist von kritischer Bedeutung für eine gute Leistung. Die Funktion ist für alle neuen DB-Instances von Amazon RDS für PostgreSQL standardmäßig aktiviert. Die zugehörigen Konfigurationsparameter werden standardmäßig entsprechend festgelegt.

Ihr Datenbankadministrator muss diese Wartungsoperation kennen und verstehen. Die PostgreSQL-Dokumentation zu Autocauger finden Sie unter Der Autovacuum Daemon.

Die Selbstbereinigung ist keine „ressourcenlose“ Operation, sondern wird im Hintergrund ausgeführt und gibt Benutzeroperationen soweit möglich Vorrang. Bei Aktivierung prüft die Selbstbereinigung auf Tabellen mit einer großen Zahl von aktualisierten oder gelöschten Tupeln. Sie schützt darüber hinaus vor dem Verlust sehr alter Daten aufgrund von Transaktions-ID-Wraparounds. Weitere Informationen finden Sie unter Verhindern von Transaktions-ID-Wraparound-Fehlern.

Die Selbstbereinigung sollte nicht als Operation mit hohem Overhead betrachtet werden, die reduziert werden kann, um eine bessere Leistung zu erzielen. Im Gegenteil; die Leistung von Tabellen mit sehr häufigen Aktualisierungs- und Löschvorgängen wird mit der Zeit schnell abnehmen, wenn keine Selbstbereinigung ausgeführt wird.

> **Important**
>
> Die fehlende Ausführung von Selbstbereinigungen kann dazu führen, dass letzten Endes eine Ausfallzeit erforderlich ist, um eine VACUUM-Operation auszuführen, die sehr viel größere Auswirkungen hat. In einigen Fällen kann eine DB-Instance von RDS für PostgreSQL aufgrund einer zu konservativen Verwendung der Selbstbereinigungsfunktion nicht mehr verfügbar sein. In diesen Fällen wird die PostgreSQL-Datenbank heruntergefahren, um sich selbst zu schützen. Zu diesem Zeitpunkt muss Amazon RDS ein single-user-mode vollständiges Vakuum direkt auf der DB-Instance durchführen. Diese vollständige Bereinigung kann zu einem Ausfall von mehreren Stunden führen. Daher wird dringend empfohlen, die standardmäßig aktivierte Selbstbereinigung nicht zu deaktivieren.

Die Selbstbereinigungsparameter legen fest, wann und wie die harte Selbstbereinigung ausgeführt wird. Die Parameter`autovacuum_vacuum_threshold` und `autovacuum_vacuum_scale_factor` legen fest, wann die Selbstbereinigung ausgeführt wird. Die Parameter `autovacuum_max_workers`, `autovacuum_nap_time`, `autovacuum_cost_limit` und `autovacuum_cost_delay` legen fest, wie die harte Selbstbereinigung ausgeführt wird. Weitere Informationen zu Autovakuum, wann es ausgeführt wird und welche Parameter erforderlich sind, finden Sie unter Routine Vacuuming in der PostgreSQL-Dokumentation.

Die folgende Abfrage zeigt die Anzahl der „toten“ Tupel in einer Tabelle mit dem Namen table1:

```
SELECT relname, n_dead_tup, last_vacuum, last_autovacuum FROM
pg_catalog.pg_stat_all_tables
WHERE n_dead_tup > 0 and relname = 'table1';
```

Die Ergebnisse der Abfrage sehen ähnlich wie folgt aus:

```
relname | n_dead_tup | last_vacuum | last_autovacuum
---------+------------+-------------+-----------------
 tasks   |   81430522 |             |
(1 row)
```

Amazon RDS for PostgreSQL Video zu bewährten Praktiken

Die AWS re:Invent-Konferenz 2020 beinhaltete eine Präsentation über neue Funktionen und bewährte Methoden für die Arbeit mit PostgreSQL auf Amazon RDS. Das Video der Präsentation ist hier verfügbar.

Bewährte Methoden für die Arbeit mit SQL Server

Zu den bewährten Methoden für eine Multi-AZ-Bereitstellung mit einer SQL Server-DB-Instance gehören:

- Verwenden Sie Amazon RDS-DB-Ereignisse, um Failover zu überwachen. Beispielsweise können Sie per Textnachricht oder E-Mail benachrichtigt werden, wenn ein Failover für eine DB-Instance ausgeführt wird. Weitere Informationen über Amazon RDS-Ereignisse finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.
- Wenn Ihre Anwendung DNS-Werte zwischenspeichert, legen Sie den Time-to-Live (TTL)-Wert auf weniger als 30 Sekunden fest. Diese TTL-Einstellung stellt eine bewährte Methode für den Fall dar, dass ein Failover auftritt. Bei einem Failover kann sich die IP-Adresse ändern und der zwischengespeicherte Wert wird möglicherweise nicht mehr verwendet.
- Es wird empfohlen, die folgenden Modi nicht zu aktivieren, da sie die Transaktionsprotokollierung deaktivieren, die für Multi-AZ erforderlich ist:
  - Einfacher Wiederherstellungsmodus
  - Offlinemodus
  - Schreibgeschützter Modus
- Führen Sie Tests durch, um zu ermitteln, wie lange Ihre DB-Instance für einen Failover-Vorgang benötigt. Die Failover-Zeit kann unterschiedlich sein, abhängig von der Datenbank, der Instance-Klasse und des Speichertyps, die oder den Sie verwenden. Sie sollten auch die Fähigkeit Ihrer Anwendung testen, bei einem Failover weiter ausgeführt zu werden.
- Führen Sie die folgenden Schritte aus, um die Failover-Zeit zu verkürzen:
  - Stellen Sie sicher, dass Sie Ihrem Workload ausreichend bereitgestellte IOPS zugeteilt haben. Unzulänglich. I/O can lengthen failover times. Database recovery requires I/O
  - Verwenden Sie kleinere Transaktionen. Die Datenbankwiederherstellung ist von Transaktionen abhängig. Wenn Sie daher große Transaktionen in mehrere kleinere Transaktionen aufteilen können, sollte die Failover-Zeit verkürzt werden.

- Berücksichtigen Sie, dass es während eines Failovers zu erhöhten Latenzen kommt. Als Teil des Failover-Vorgangs repliziert Amazon RDS Ihre Daten automatisch zu einer neuen Standby-Instance. Diese Replikation bedeutet, dass neue Daten an zwei verschiedene DB-Instances übergeben werden. Daher kann es eine gewisse Latenz geben, bis die Standby-DB-Instance mit der neuen primäre DB-Instance aufgeschlossen hat.
- Stellen Sie Ihre Anwendungen in allen Availability Zones bereit. Wenn eine Availability Zone ausfällt, sind die Anwendungen in den anderen Availability Zones weiterhin verfügbar.

Denken Sie bei der Arbeit mit einer Multi-AZ-Bereitstellung von SQL Server daran, dass Amazon RDS Replicas für alle SQL Server-Datenbanken auf Ihrer Instance erstellt. Wenn Sie nicht möchten, dass bestimmte Datenbanken sekundäre Replicas aufweisen, richten Sie eine separate DB-Instance ein, die für diese Datenbanken keine Multi-AZ verwendet.

Video zu bewährten Methoden für Amazon RDS for SQL Server

Die AWS re:Invent-Konferenz 2019 beinhaltete eine Präsentation über neue Funktionen und bewährte Methoden für die Arbeit mit SQL Server auf Amazon RDS. Das Video der Präsentation ist hier verfügbar.

Arbeiten mit DB-Parametergruppen

Es wird empfohlen, DB-Parametergruppenänderungen stets zuerst in einer Test-DB-Instance durchzuführen, bevor Sie diese Parametergruppenänderungen auf Ihre Produktions-DB-Instances anwenden. Wenn die DB-Modulparameter in einer DB-Parametergruppe falsch festgelegt werden, kann dies unbeabsichtigte nachteilige Auswirkungen haben, einschließlich verminderter Leistung und Systeminstabilität. Gehen Sie stets vorsichtig vor, wenn Sie DB-Modulparameter ändern, und sichern Sie Ihre DB-Instance, bevor Sie eine DB-Parametergruppe ändern.

Informationen zum Sichern Ihrer DB-Instance finden Sie unter Daten sichern, wiederherstellen und exportieren.

Bewährte Methoden zur Automatisierung der DB-Instance-Erstellung

Es ist eine bewährte Methode für Amazon RDS, eine DB-Instance mit der bevorzugten Nebenversion des Datenbankmoduls zu erstellen. Sie können die AWS CLI Amazon RDS-API oder verwenden,

AWS CloudFormation um die Erstellung von DB-Instances zu automatisieren. Wenn Sie diese Methoden verwenden, können Sie nur die Hauptversion angeben und Amazon RDS erstellt die Instance mit der bevorzugten Nebenversion automatisch. ..Wenn zum Beispiel PostgreSQL 12.5 die bevorzugte Nebenversion ist und Sie Version 12 mit `create-db-instance` angeben, wird die DB-Instanz Version 12.5 sein.

Um die bevorzugte Nebenversion zu ermitteln, können Sie den Befehl `describe-db-engine-versions` mit der Option `--default-only` ausführen; siehe folgendes Beispiel.

```
aws rds describe-db-engine-versions --default-only --engine postgres

{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "EngineVersion": "12.5",
            "DBParameterGroupFamily": "postgres12",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 12.5-R1",
            ...some output truncated...
        }
    ]
}
```

Informationen zum programmgesteuerten Erstellen von DB-Instanzen finden Sie in den folgenden Ressourcen:

- Mit dem AWS CLI — create-db-instance
- Verwenden der Amazon RDS-API — Erstellen DBInstance
- Verwenden von AWS CloudFormation — AWS: :RDS:: DBInstance

Video zu den neuen Funktionen von Amazon RDS

Die AWS re:Invent-Konferenz 2023 beinhaltete eine Präsentation über neue Amazon RDS-Funktionen. Das Video der Präsentation ist hier verfügbar.

Programmatischer Zugriff auf Amazon RDS Amazon

Amazon RDS stellt Ihnen die folgenden Tools zur programmgesteuerten Verwaltung Ihrer Amazon RDS Amazon zur Verfügung.

AWS Command Line Interface (AWS CLI)

Sie können Ihre RDS erstellen und verwalten, indem Sie die AWS CLI Befehlszeilen-Shell verwenden. Das AWS CLI bietet direkten Zugriff auf APIs die AWS vier Dienste wie AmazonRDS. Syntax und Beispiele für die Befehle für Amazon RDS finden Sie unter rdsin der AWS CLI Befehlsreferenz.

AWS CloudFormation

Mit diesem AWS Infrastructure as Code (IaC) -Tool können Sie Vorlagen erstellen, die alle gewünschten Amazon RDS beschreiben und AWS CloudFormation diese Ressourcen für Sie bereitstellen und konfigurieren. Weitere Informationen finden Sie unter the section called “Ressourcen erstellen mit AWS CloudFormation”.

AWS Kits für die Softwareentwicklung () SDKs

AWS bietet SDKs für viele beliebte Technologien und Programmiersprachen. Sie erleichtern es Ihnen, AWS Dienste in Ihren Anwendungen in dieser Sprache oder Technologie aufzurufen. Weitere Informationen zu diesen finden Sie SDKs unter Tools für die Entwicklung und Verwaltung von Anwendungen auf AWS.

Amazon RDS API

Dies API ist die Schnittstelle auf Protokollebene für Amazon. RDS Wenn Sie dies verwendenAPI, müssen Sie jede HTTPS Anfrage korrekt formatieren und jeder Anfrage eine gültige digitale Signatur hinzufügen. Weitere Informationen finden Sie unter Amazon RDS-API-Referenz.

Console-to-Code

Mit diesem Tool können Sie Code für Aktionen generieren, die Sie in der RDS Amazon-Konsole ausführen, und diesen Code in anderen Tools verwenden, AWS CloudFormation z. Weitere Informationen finden Sie unter the section called “Console-to-Code”.

Wird verwendet Console-to-Code, um Code für Ihre Amazon RDS Amazon zu generieren

Die Konsole bietet eine Anleitung zum Erstellen von Ressourcen und zum Testen von Prototypen. Wenn Sie dieselben Ressourcen in großem Umfang erstellen möchten, benötigen Sie Automatisierungscode. Console-to-Codeist eine Funktion von Amazon Q Developer, die Ihnen den Einstieg in Ihren Automatisierungscode erleichtern kann. Console-to-Codezeichnet Ihre Konsolenaktionen auf, einschließlich der von Ihnen angegebenen Standardwerte und Parameterwerte. Anschließend schlägt es mithilfe generativer KI Code in Ihrer bevorzugten Sprache und Ihrem bevorzugten Format für die von Ihnen ausgewählten Aktionen vor. Da der Konsolen-Workflow sicherstellt, dass die von Ihnen angegebenen Parameterwerte zusammen gültig sind, Console-to-Code verfügt der Code, den Sie mithilfe von diesem Code generieren, über kompatible Parameterwerte. Sie können den Code als Ausgangspunkt verwenden und ihn dann so anpassen, dass er für Ihren speziellen Anwendungsfall produktionsbereit ist.

Mit können Sie Console-to-Code beispielsweise die Erstellung eines für eine RDSDB-Instance aufzeichnen und wählen, ob Code im AWS CloudFormation JSON Format generiert werden soll. Anschließend können Sie diesen Code kopieren und für die Verwendung in Ihrer AWS CloudFormation Vorlage anpassen.

Console-to-Code kann derzeit infrastructure-as-code (IaC) in den folgenden Sprachen und Formaten generieren:

- CDKJava
- CDKPython
- CDK TypeScript
- CloudFormation JSON
- CloudFormation YAML

Weitere Informationen und Anweisungen zur Verwendung Console-to-Code finden Sie unter Automatisieren von AWS Services mit Amazon Q Developer Console-to-Code im Amazon Q Developer User Guide.

Konfiguration einer Amazon RDS DB-Instance

In diesem Abschnitt wird gezeigt, wie Sie Ihre Amazon RDS DB-Instance einrichten. Bevor Sie eine DB-Instance erstellen, entscheiden Sie sich für die DB-Instance-Klasse, die die DB-Instance ausführt. Entscheiden Sie außerdem, wo die DB-Instance ausgeführt werden soll, indem Sie eine AWS Region auswählen. Als Nächstes erstellen Sie die DB-Instance.

Sie können eine DB-Instance mit einer Optionsgruppe und einer DB-Parametergruppe konfigurieren.

- Eine Optionsgruppe spezifiziert Funktionen, sogenannte Optionen, die für eine bestimmte Amazon RDS DB-Instance verfügbar sind.
- Eine DB-Parametergruppe dient als Container für Engine-Konfigurationswerte, die auf eine oder mehr DB-Instance angewendet werden.

Die verfügbaren Optionen und Parameter hängen von der DB-Engine und der DB-Engine-Version ab. Sie können eine Optionsgruppe und eine DB-Parametergruppe angeben, wenn Sie eine DB-Instance erstellen. Sie können eine DB-Instance auch ändern, um sie anzugeben.

Themen

- Erstellen einer Amazon RDS-DB-Instance
- Erstellen von Amazon RDS Amazon mit AWS CloudFormation
- Verbindung zu einer Amazon RDS DB-Instance herstellen
- Arbeiten mit Optionsgruppen
- Parametergruppen für Amazon RDS
- Erstellen eines ElastiCache Amazon-Caches mithilfe des Amazon RDS DB-Instance-Einstellungen
- Automatische Migration von EC2 Datenbanken zu Amazon RDS mit AWS Database Migration Service
- Tutorial: Erstellen eines mit einem benutzerdefinierten Parameter und einer neuen Optionsgruppe

Erstellen einer Amazon RDS-DB-Instance

Der grundlegende Baustein von Amazon RDS ist die DB-Instance, in der Sie Ihre Datenbanken erstellen. Die Engine-spezifischen Eigenschaften der DB-Instance wählen Sie beim Anlegen aus. Sie wählen auch die Speicherkapazität, die CPU, den Arbeitsspeicher usw. der AWS Instanz, auf der der Datenbankserver läuft.

Themen

- Voraussetzungen für DB-Instance
- Erstellen einer DB-Instance
- Einstellungen für DB-Instances

Voraussetzungen für DB-Instance

> **Important**
>
> Sie müssen die Aufgaben in Einrichtung Ihrer Amazon RDS-Umgebung abschließen, bevor Sie eine Amazon-RDS-DB-Instance erstellen können.

Die folgenden Voraussetzungen gelten für die Erstellung einer RDS-DB-Instance.

Themen

- Netzwerk für die DB-Instance konfigurieren
- Zusätzliche Voraussetzungen

Netzwerk für die DB-Instance konfigurieren

Sie können eine DB-Instance von Amazon RDS nur in einer Virtual Private Cloud (VPC) erstellen, die auf dem Amazon-VPC-Service basiert. Außerdem muss es sich in einer befinden AWS-Region , die mindestens zwei Availability Zones hat. Die DB-Subnetzgruppe, die Sie für die DB-Instance wählen, muss mindestens zwei Availability Zones abdecken. Diese Konfiguration stellt sicher, dass Sie eine Multi-AZ-Bereitstellung konfigurieren können, wenn Sie die DB-Instance erstellen oder in future problemlos zu einer solchen wechseln.

Um die Konnektivität zwischen Ihrer neuen DB-Instance und einer EC2 Amazon-Instance in derselben VPC einzurichten, tun Sie dies, wenn Sie die DB-Instance erstellen. Um von anderen Ressourcen als EC2 Instances in derselben VPC aus eine Verbindung zu Ihrer DB-Instance herzustellen, konfigurieren Sie die Netzwerkverbindungen manuell.

Themen

- Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2
- Manuelles Konfigurieren des Netzwerks

Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2

Wenn Sie eine RDS-DB-Instance erstellen, können Sie die verwenden, AWS Management Console um die Konnektivität zwischen einer EC2 Instance und der neuen DB-Instance einzurichten. In diesem Fall konfiguriert RDS Ihre VPC- und Netzwerkeinstellungen automatisch. Die DB-Instance wird in derselben VPC wie die EC2 Instance erstellt, sodass die EC2 Instance auf die DB-Instance zugreifen kann.

Die folgenden Anforderungen gelten für die Verbindung einer EC2 Instance mit der DB-Instance:

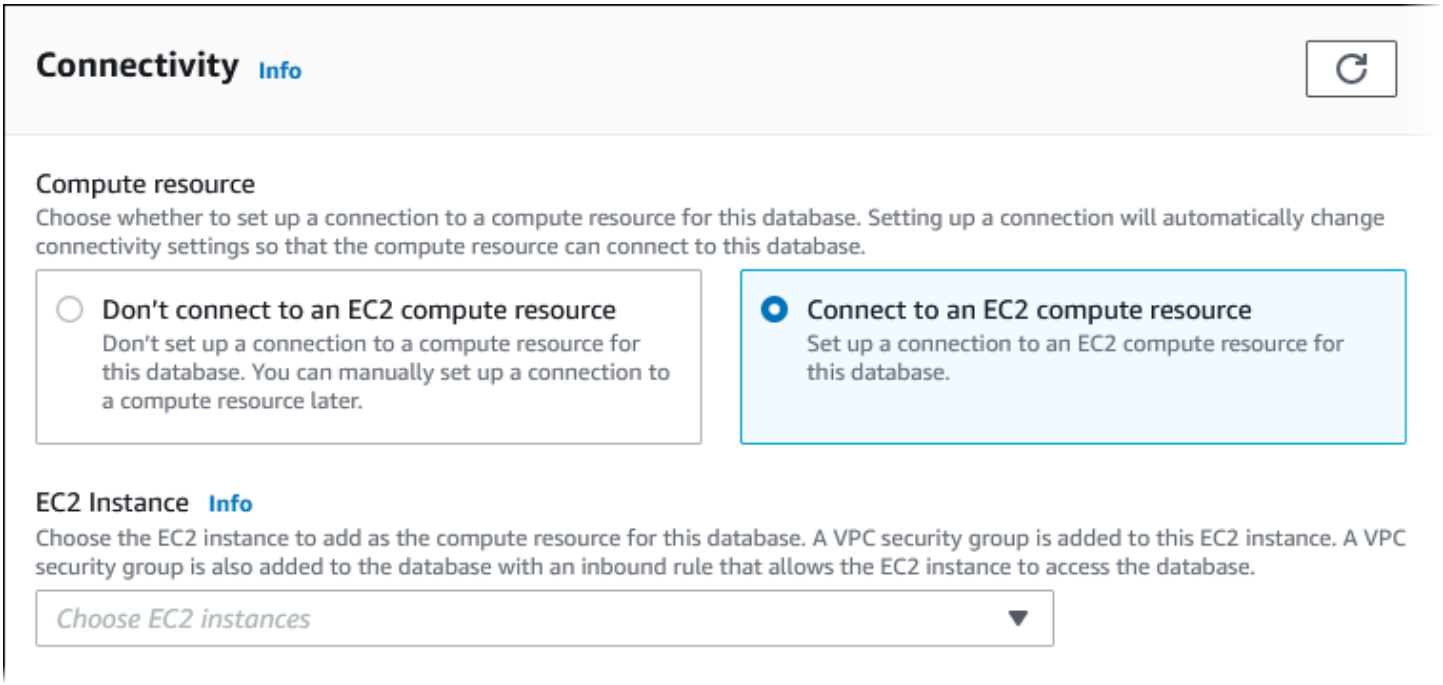
- Die EC2 Instance muss in der vorhanden sein, AWS-Region bevor Sie die DB-Instance erstellen.

  Wenn in der keine EC2 Instances vorhanden sind AWS-Region, bietet die Konsole einen Link zum Erstellen einer Instanz.
- Der Benutzer, der die DB-Instance erstellt, muss über Berechtigungen zum Ausführen der folgenden Vorgänge verfügen:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`

- `ec2:DescribeSubnets`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Mit dieser Option wird eine private DB-Instance erstellt. Die DB-Instance verwendet eine DB-Subnetzgruppe mit nur privaten Subnetzen, um den Zugriff auf Ressourcen innerhalb der VPC einzuschränken.

Um eine EC2 Instance Connect der DB-Instance zu verbinden, wählen Sie auf der Seite Datenbank erstellen im Abschnitt Konnektivität die Option Mit einer EC2 Rechenressource verbinden aus.



Wenn Sie Mit einer EC2 Rechenressource Connect wählen, legt RDS die folgenden Optionen automatisch fest. Sie können diese Einstellungen nur ändern, wenn Sie sich dafür entscheiden, keine Konnektivität mit einer EC2 Instance einzurichten, indem Sie „Keine Verbindung zu einer EC2 Rechenressource herstellen“ wählen.

Konsolenoption	Automatische Einstellung
Netzwerktyp	RDS setzt den Netzwerktyp auf IPv4. Derzeit wird der Dual-Stack-Modus nicht unterstützt, wenn Sie eine Verbindung zwischen einer EC2 Instance und der DB-Instance einrichten.

Konsolenoption	Automatische Einstellung
Virtual Private Cloud (VPC)	RDS setzt die VPC auf die mit der EC2 Instance verknüpfte.
DB subnet group (DB-Subnetzgruppe)	RDS benötigt eine DB-Subnetzgruppe mit einem privaten Subnetz in derselben Availability Zone wie die Instance. EC2 Wenn eine DB-Subnetzgruppe vorhanden ist, die diese Anforderung erfüllt, dann verwendet RDS die vorhandene DB-Subnetzgruppe. Standardmäßig ist diese Option auf Automatic setup (Automatische Einrichtung) eingestellt. Wenn Sie Automatic setup (Automatische Einrichtung) auswählen und es keine DB-Subnetzgruppe gibt, die diese Anforderung erfüllt, wird die folgende Aktion ausgeführt. RDS verwendet drei verfügbare private Subnetze in drei Availability Zones, wobei eine der Availability Zones mit der Instance identisch ist. EC2 Wenn kein privates Subnetz in einer Availability Zone verfügbar ist, erstellt RDS ein privates Subnetz in der Availability Zone. Anschließend erstellt RDS die DB-Subnetzgruppe. Wenn ein privates Subnetz verfügbar ist, verwendet RDS die zugehörige Routing-Tabelle und fügt alle Subnetze, die es erstellt, dieser Routing-Tabelle hinzu. Wenn kein privates Subnetz verfügbar ist, erstellt RDS eine Routing-Tabelle ohne Internet-Gateway-Zugriff und fügt die erstellten Subnetze der Routing-Tabelle hinzu. Mit RDS können Sie auch vorhandene DB-Subnetzgruppen verwenden. Wählen Sie Choose existing (Vorhandene wählen) aus, wenn Sie eine vorhandene DB-Subnetzgruppe Ihrer Wahl verwenden möchten.

Konsolenoption	Automatische Einstellung
Öffentlicher Zugriff	RDS entscheidet Nein, sodass die DB-Instance nicht öffentlich zugänglich ist. Aus Sicherheitsgründen ist es eine bewährte Methode, die Datenbank privat zu halten und sicherzustellen, dass sie nicht über das Internet zugänglich ist.
VPC-Sicherheitsgruppe (Firewall)	RDS erstellt eine neue Sicherheitsgruppe, die der DB-Instance zugeordnet ist. Die Sicherheitsgruppe heißt `rds-ec2-`*n*, wobei *n* eine Zahl ist. Diese Sicherheitsgruppe enthält eine Regel für eingehenden Datenverkehr mit der EC2 VPC-Sicherheitsgruppe (Firewall) als Quelle. Diese Sicherheitsgruppe, die der DB-Instance zugeordnet ist, ermöglicht der EC2 Instance den Zugriff auf die DB-Instance. RDS erstellt außerdem eine neue Sicherheitsgruppe, die der EC2 Instance zugeordnet ist. Die Sicherheitsgruppe heißt `ec2-rds-`*n*, wobei *n* eine Zahl ist. Diese Sicherheitsgruppe enthält eine ausgehende Regel mit der VPC-Sicherheitsgruppe der DB-Instance als Quelle. Diese Sicherheitsgruppe ermöglicht es der EC2 Instance, Traffic an die DB-Instance zu senden. Sie können eine weitere neue Sicherheitsgruppe hinzufügen, indem Sie Neu erstellen wählen und den Namen der neuen Sicherheitsgruppe eingeben. Sie können vorhandene Sicherheitsgruppen hinzufügen, indem Sie Bestehende auswählen und Sicherheitsgruppen auswählen, die hinzugefügt werden sollen.

Konsolenoption	Automatische Einstellung
Availability Zone	Wenn Sie unter Verfügbarkeit und Haltbarkeit (Single-AZ-Bereitstellung) die Option „Einzelne DB-Instance“ wählen, wählt RDS die Availability Zone der EC2 Instance aus. Wenn Sie unter Verfügbarkeit und Haltbarkeit (Multi-AZ-DB-Instance-Bereitstellung) die Option Multi-AZ-DB-Instance wählen, wählt RDS die Availability Zone der EC2 Instance für eine DB-Instance in der Bereitstellung. RDS wählt zufällig eine andere Availability Zone für die andere DB-Instance aus. Entweder die primäre DB-Instance oder das Standby-Replikat wird in derselben Availability Zone wie die Instance erstellt. EC2 Wenn Sie sich für eine Multi-AZ-DB-Instance entscheiden, besteht die Möglichkeit, dass Kosten für mehrere Availability Zones anfallen, wenn sich die DB-Instance und die EC2 Instance in unterschiedlichen Availability Zones befinden.

Weitere Informationen zu diesen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Wenn Sie diese Einstellungen ändern, nachdem die DB-Instance erstellt wurde, können sich die Änderungen auf die Verbindung zwischen der EC2 Instance und der DB-Instance auswirken.

Manuelles Konfigurieren des Netzwerks

Um von anderen Ressourcen als EC2 Instances in derselben VPC aus eine Verbindung zu Ihrer DB-Instance herzustellen, konfigurieren Sie die Netzwerkverbindungen manuell. Wenn Sie die verwenden, AWS Management Console um Ihre DB-Instance zu erstellen, können Sie Amazon RDS automatisch eine VPC für Sie erstellen lassen. Sie können auch eine bestehende VPC verwenden oder eine neue VPC für Ihre DB-Instance erstellen. Unabhängig vom jeweiligen Ansatz muss Ihre VPC mindestens ein Subnetz in jeder von mindestens zwei Availability Zones haben, damit Sie sie mit einer RDS-DB-Instance verwenden können.

Standardmäßig erstellt Amazon RDS die DB-Instance automatisch als Availability Zone für Sie. Um eine spezifische Availability Zone auszuwählen, müssen Sie die Einstellung für Verfügbarkeit und Haltbarkeit auf Single-Instance ändern. Dadurch wird eine Availability Zone-Einstellung verfügbar, mit der Sie zwischen den Availability Zones in Ihrer VPC wählen können. Wenn Sie sich jedoch für eine

Multi-AZ-Bereitstellung entscheiden, wählt RDS automatisch die Availability Zone der primären oder Writer-DB-Instance aus und die Availability Zone-Einstellung wird nicht angezeigt.

Es kann sein, dass Sie keine Standard-VPC besitzen oder keine VPC erstellt haben. In diesen Fällen kann Amazon RDS automatisch eine VPC für Sie erstellen, wenn Sie eine DB-Instance über die Konsole erstellen. Andernfalls gehen Sie wie folgt vor:

- Erstellen Sie eine VPC mit mindestens einem Subnetz in jeder der mindestens zwei Availability Zones in der Region, AWS-Region in der Sie Ihre DB-Instance bereitstellen möchten. Weitere Informationen erhalten Sie unter Arbeiten mit einer DB-Instance in einer VPC und Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur).
- Legen Sie eine VPC-Sicherheitsgruppe fest, die Verbindungen mit Ihrer DB-Instance autorisiert. Weitere Informationen erhalten Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe und Zugriffskontrolle mit Sicherheitsgruppen.
- Legen Sie eine RDS-DB-Subnetzgruppe fest, die mindestens zwei Subnetze in der VPC definiert, die von der DB-Instance verwendet werden können. Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.

Wenn Sie eine Verbindung mit einer Ressource herstellen möchten, die sich nicht in derselben VPC wie die DB-Instance Cluster befindet, sehen Sie sich die entsprechenden Szenarien unter Szenarien für den Zugriff auf einen in einem VPC an.

Zusätzliche Voraussetzungen

Bevor Sie eine DB-Instance erstellen, sollten Sie die folgenden zusätzlichen Voraussetzungen berücksichtigen:

- Wenn Sie eine Verbindung AWS mit AWS Identity and Access Management (IAM-) Anmeldeinformationen herstellen, muss Ihr AWS Konto über bestimmte IAM-Richtlinien verfügen. Diese gewähren die für die Durchführung von Amazon-RDS-Vorgängen erforderlichen Berechtigungen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.

  Um IAM für den Zugriff auf die RDS-Konsole zu verwenden, melden Sie sich AWS Management Console mit Ihren IAM-Benutzeranmeldedaten bei der an. Gehen Sie dann zur Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
- Wenn Sie die Konfigurationsparameter für Ihre DB-Instance anpassen möchten, müssen Sie eine DB-Parametergruppe mit den erforderlichen Parametereinstellungen festlegen.

Weitere Informationen zum Erstellen oder Ändern einer DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

> **Important**
>
> Wenn Sie das BYOL-Modell für Amazon RDS for Db2 verwenden, müssen Sie vor dem Erstellen einer DB-Instance zunächst eine benutzerdefinierte Parametergruppe erstellen, die Ihre IBM Site ID and IBM Customer ID. Weitere Informationen finden Sie unterBringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit.

- Bestimmen Sie die TCP/IP-Portnummer, die Sie für Ihre DB-Instance festlegen werden. Die Firewalls einiger Unternehmen blockieren Verbindungen zu den Standard-Ports für RDS-DB-Instances. Wenn die Firewall Ihres Unternehmens den Standardport blockiert, wählen Sie für Ihre DB-Instance einen anderen Port. Die Standardports für Amazon-RDS-DB-Engines sind:

RDS für Db2	RDS for MariaDB	RDS for MySQL	RDS für Oracle	RDS for PostgreSQL	RDS für SQL Server
50000	3306	3306	1521	5432	1433

Für RDS für SQL Server sind die folgenden Ports reserviert, und Sie können sie nicht verwenden, wenn Sie eine DB-Instance erstellen: `1234, 1434, 3260, 3343, 3389, 47001,` und`49152-49156`.

Erstellen einer DB-Instance

Sie können eine Amazon RDS-DB-Instance mithilfe der AWS Management Console AWS CLI, der oder der RDS-API erstellen.

> **Note**
>
> Für RDS for Db2 empfehlen wir, dass Sie die für Ihr Lizenzmodell erforderlichen Elemente einrichten, bevor Sie eine RDS for Db2-DB-Instance erstellen. Weitere Informationen finden Sie unter Lizenzierungsoptionen für Amazon RDS für Db2.

Konsole

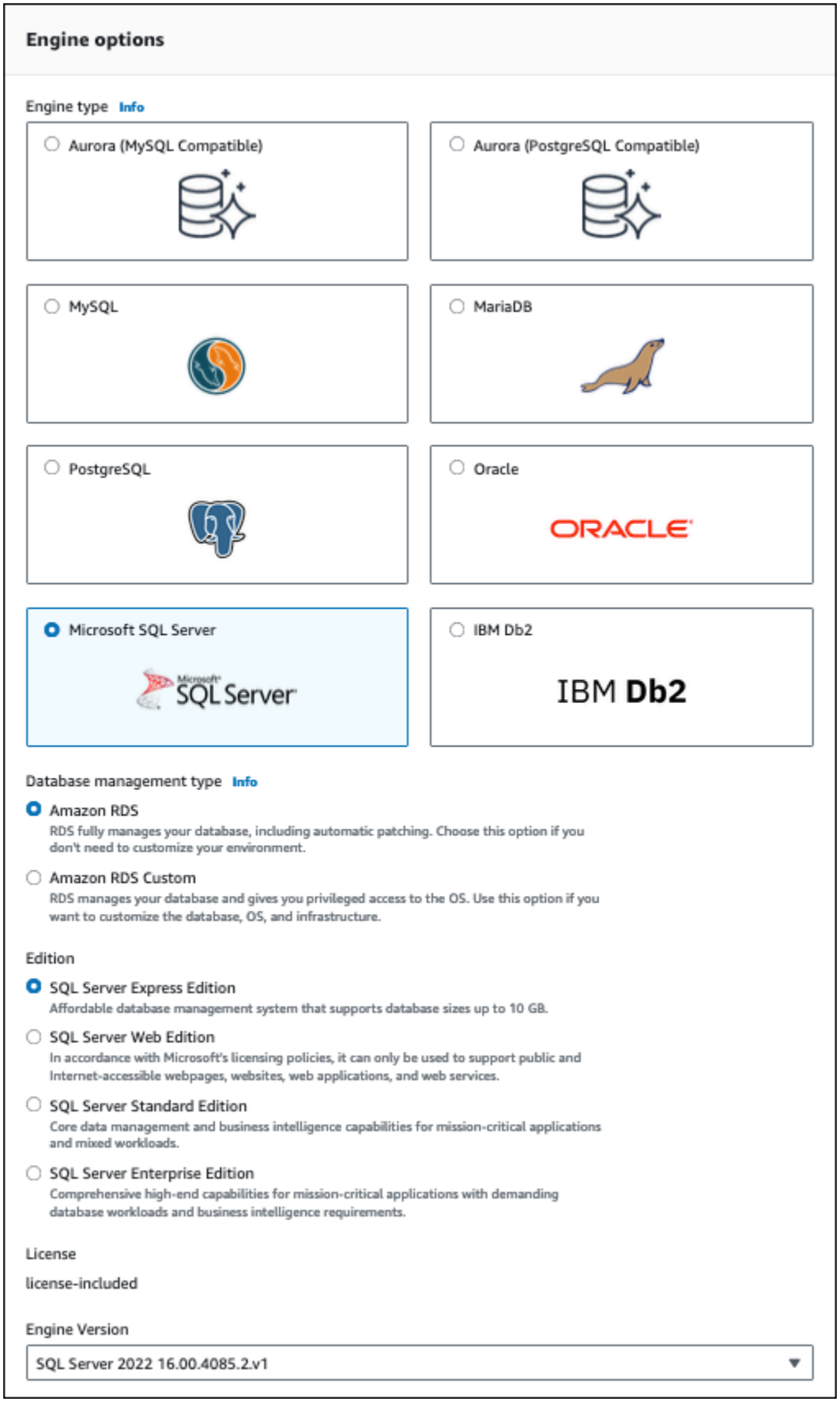
Sie können eine DB-Instance mit aktiviertem oder AWS Management Console deaktiviertem Easy Create erstellen. Wenn die Option Einfache Erstellung aktiviert ist, geben Sie nur den Engine-Typ, die Größe der Instance und die Kennung der Instance für die DB an. Mit der Option Einfache Erstellung werden für die anderen Konfigurationsoptionen die Standardeinstellungen verwendet. Ist die Option Einfache Erstellung nicht aktiviert, geben Sie bei der Erstellung einer Datenbank weitere Konfigurationsoptionen an, über die Verfügbarkeit, Sicherheit, Backups und Wartung der Datenbank konfiguriert werden.

> **Note**
>
> Im folgenden Verfahren ist Standard Create (Standarderstellung) aktiviert und Easy Create (Einfache Erstellung) ist nicht aktiviert. Dieses Verfahren verwendet Microsoft SQL Server als Beispiel.
> Beispiele, die Easy Create (Einfache Erstellung) verwenden, um Sie durch das Erstellen und Verbinden mit Beispiel-DB-Instances für jede Engine zu führen, finden Sie unter Erste Schritte mit Amazon RDS.

So erstellen Sie eine DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon-RDS-Konsole die AWS -Region aus, in der Sie die DB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Datenbank erstellen und danach Standarderstellung aus.
5. Wählen Sie als Engine-Typ IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle oder PostgreSQL aus.

   Microsoft SQL Server wird hier angezeigt.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
SQL Server
IBM Db2
IBM Db2
Database management type Info
Amazon RDS
RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.
Amazon RDS Custom
RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.
Edition
SQL Server Express Edition
Affordable database management system that supports database sizes up to 10 GB.
SQL Server Web Edition
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.
SQL Server Standard Edition
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
SQL Server Enterprise Edition
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.
License
license-included
Engine Version
SQL Server 2022 16.00.4085.2.v1

6. Wählen Sie als Datenbankverwaltungstyp Amazon RDS oder Amazon RDS Custom, wenn Sie Oracle oder SQL Server verwenden.

   Hier wird Amazon RDS angezeigt. Weitere Informationen zu RDS CUstom finden Sie unter Amazon RDS Custom.
7. Wenn Sie Db2, Oracle oder SQL Server verwenden, wählen Sie unter Edition die DB-Engine-Edition aus, die Sie verwenden möchten.

   MySQL hat nur eine Option für die Edition, und MariaDB und PostgreSQL haben keine.
8. Wählen Sie unter Version die Engine-Version aus.
9. Wählen Sie unter Templates (Vorlagen) die Vorlage für Ihr Anwendungsszenario aus. Bei Auswahl von Production (Produktion) ist Folgendes in einem späteren Schritt vorausgewählt:

   - Failover-Option Multi-AZ
   - Speicheroption Provisioned IOPS SSD (io1) (Bereitgestellte IOPS-SSD (io1))
   - Option Enable deletion protection (Löschschutz aktivieren)

   Wir empfehlen diese Funktionen für jede Produktionsumgebung.

   > **Note**
   >
   > Die zur Auswahl stehenden Vorlagen richten sich nach der jeweiligen Edition.
10. Öffnen Sie im Abschnitt Settings (Einstellungen) die Option Credential Settings (Einstellungen zu Anmeldeinformationen). Führen Sie dann die folgenden Schritte aus:

    a. (Optional) Ändern Sie den Wert des Haupt-Benutzernamens.

    b. Wählen Sie eine der folgenden Optionen zur Verwaltung von Anmeldeinformationen:

       - Verwaltet in AWS Secrets Manager

         Wählen Sie unter Verschlüsselungsschlüssel auswählen entweder einen KMS-Schlüssel aus, den Secrets Manager erstellt, oder einen Schlüssel, den Sie selbst erstellt haben.

         > **Note**
         >
         > Wir empfehlen AWS Secrets Manager dies als sicherste Methode zur Verwaltung von Anmeldeinformationen. Es fallen zusätzliche Gebühren an. Weitere

Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.

- Selbstverwaltet

  Um ein Passwort anzugeben, deaktivieren Sie das Kontrollkästchen Passwort automatisch generieren, falls es aktiviert ist. Geben Sie dasselbe Passwort in die Felder Master-Passwort und Master-Passwort bestätigen ein.

11. (Optional) Richten Sie eine Verbindung zu einer Rechenressource für diese DB-Instance ein.

    Sie können die Konnektivität zwischen einer EC2 Amazon-Instance und der neuen DB-Instance während der DB-Instance-Erstellung konfigurieren. Weitere Informationen finden Sie unter Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2 .

12. Wenn Sie im Abschnitt Konnektivität unter VPC-Sicherheitsgruppe (Firewall) die Option Neu erstellen auswählen, wird eine VPC-Sicherheitsgruppe mit einer Regel für eingehenden Datenverkehr erstellt, die es der IP-Adresse Ihres lokalen Computers ermöglicht, auf die Datenbank zuzugreifen.

13. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die DB-Instance an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

14. Wählen Sie Create database (Datenbank erstellen) aus.

    Wenn Sie ein automatisch generiertes Passwort verwenden, wird auf der Seite Databases (Datenbanken) die Schaltfläche View credential details (Details zu Anmeldeinformationen anzeigen) angezeigt.

    Um den Masterbenutzernamen und das zugehörige Passwort für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen).

    Verwenden Sie den angezeigten Benutzernamen und das angezeigte Passwort, um eine Verbindung zu der DB-Instance als Hauptbenutzer herzustellen.

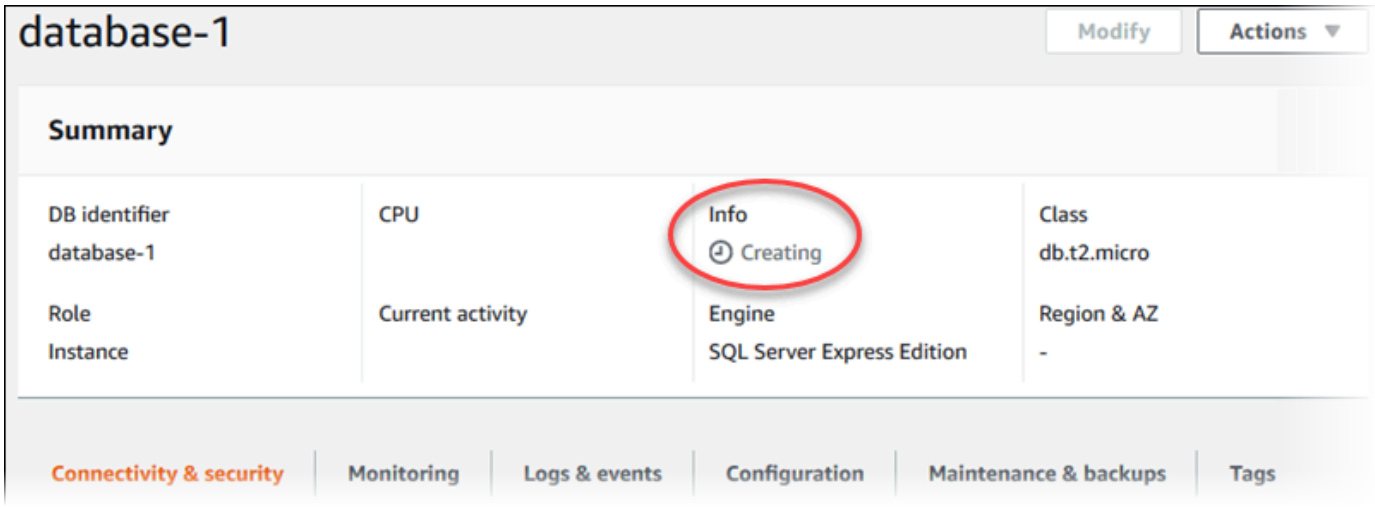
    Important

    Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern. Wenn Sie das Passwort für den Masterbenutzer ändern müssen, nachdem die DB-Instance verfügbar wurde, ändern

Sie die DB-Instance entsprechend. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

15. Wählen Sie in Databases (Datenbanken) den Namen der neuen DB-Instance aus.

    In der RDS-Konsole werden die Details der neuen DB-Instance angezeigt. Die DB-Instance wird mit dem Status Creating (Wird erstellt) angezeigt, bis die Erstellung abgeschlossen ist und sie verwendet werden kann. Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Je nach der Klasse und dem zugeteilten Speicher der DB-Instance kann es einige Minuten dauern, bis sie verfügbar ist.



AWS CLI

> **Note**
>
> Wenn Sie die Db2-Lizenz verwenden möchten AWS Marketplace, müssen Sie zuerst IBM abonnieren AWS Marketplace und sich über die AWS Management Console registrieren. Weitere Informationen finden Sie unter Angebote von Db2 Marketplace abonnieren und sich registrieren bei IBM.

Um eine DB-Instance mithilfe von zu erstellen AWS CLI, rufen Sie den create-db-instanceBefehl mit den folgenden Parametern auf:

- `--db-instance-identifier`
- `--db-instance-class`

- `--vpc-security-group-ids`
- `--db-subnet-group`
- `--engine`
- `--master-username`
- `--master-user-password` oder `--manage-master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

In diesem Beispiel wird Microsoft SQL Server verwendet.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --engine sqlserver-se \
    --db-instance-identifier mymsftsqlserver \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --vpc-security-group-ids mysecuritygroup \
    --db-subnet-group mydbsubnetgroup \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --engine sqlserver-se ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --vpc-security-group-ids mysecuritygroup ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
```

```
    --backup-retention-period 3
```

Die Ausgabe dieses Befehls sieht etwa wie folgt aus.

```
DBINSTANCE  mydbinstance  db.t3.large  sqlserver-se  250  sa  creating  3  ****  n
 10.50.2789
SECGROUP  default  active
PARAMGRP  default.sqlserver-se-14  in-sync
```

RDS-API

> **Note**
>
> Wenn Sie die Db2-Lizenz über verwenden möchten AWS Marketplace, müssen Sie zuerst IBM abonnieren AWS Marketplace und sich bei IBM registrieren, indem Sie den AWS Management Console verwenden. Weitere Informationen finden Sie unter Angebote von Db2 Marketplace abonnieren und sich registrieren bei IBM.

Um eine DB-Instance mithilfe der Amazon RDS-API zu erstellen, rufen Sie den DBInstance Vorgang Create auf.

Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Einstellungen für DB-Instances

In der folgenden Tabelle finden Sie Einzelheiten zu Einstellungen, die Sie beim Erstellen einer DB-Instance wählen können. Die Tabelle zeigt auch die DB-Engines, für die jede Einstellung unterstützt wird.

Sie können eine DB-Instance mit der Konsole, dem create-db-instanceCLI-Befehl oder dem Vorgang Create DBInstance RDS API erstellen.

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
Allocated storage	Die Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes. In	CLI-Option:	Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

	einigen Fällen kann die I/O Leistung verbessert werden, wenn Sie Ihrer DB-Instance eine höhere Speichermenge zuweisen als die Größe Ihrer Datenbank.
	
	Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.
		`--allocated-storage`
		
		API-Parameter:
		
		`AllocatedStorage`	

Architektur-Einstellungen	Wenn Sie Oracle-Multitenant-Architektur wählen, erstellt RDS für Oracle eine Container-Datenbank (CDB). Wenn Sie diese Option nicht wählen, erstellt RDS for Oracle eine Nicht-CDB. Eine Nicht-CDB verwendet die herkömmliche Oracle-Architektur. Eine CDB kann austauschbare Datenbanken (PDBs) enthalten, eine Nicht-CDB dagegen nicht. Oracle Database 21c verwendet nur die CDB-Architektur. Oracle Database 19c kann entweder die CDB- oder die Nicht-CDB-Architektur verwenden. Versionen, die niedriger als Oracle Database 19c sind, verwenden nur die Nicht-CDB-Architektur. Weitere Informationen finden Sie unter Überblick über RDS for Oracle CDBs.	CLI-Option: `--engine oracle-ee-cdb` (Oracle-Multitenant) `--engine oracle-se2-cdb` (Oracle-Multitenant) `--engine oracle-ee` (traditionell) `--engine oracle-se2` (traditionell) API-Parameter: `Engine`	Oracle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
Konfiguration der Architektur	Diese Einstellungen sind nur gültig, wenn Sie Oracle-Multitenant-Architektur unter Architektur-Einstellungen wählen. Wählen Sie eine der folgenden zusätzlichen Einstellungen: • Bei der Multi-Tenant-Konfiguration kann Ihre RDS for Oracle CDB-Instance je nach Datenbankedition und erforderlichen Optionslizenzen 1—30 Mandantendatenbanken enthalten. Im Kontext einer Oracle-Datenbank ist eine Tenant-Datenbank eine PDB. Anwendung PDBs und Proxy PDBs werden nicht unterstützt. Ihre DB-Instance wird mit einer ersten Tenant-Datenbank erstellt. Wählen Sie Werte für den Namen der Tenant-Datenbank, den Namen des Hauptbenutzers der Tenant-Datenbank, das Master-Passwort für die Tenant-Datenbank und den Zeichensatz der Tenant-Datenbank. Die Multi-Tenant-Konfiguration ist dauerhaft. Daher können Sie die Multi-Tenant-Konfiguration nicht wieder in die Single-Tenant-Konfiguration konvertieren. Das unterstützte Release-Update (RU) für die Multi-Tenant-Konfiguration ist mindestens 19.0.0.0.ru-2022-01.rur-2022.r1.	CLI-Option: `--multi-tenant` (Multi-Tenant-Konfiguration) `--no-multi-tenant` (Single-Tenant-Konfiguration) API-Parameter: `MultiTenant`	Oracle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

> **Note**
>
> Die Amazon RDS-Konfiguration wird als „Multi-Tenant“ und nicht als „Multitenant“ bezeichnet, da es sich um eine Funktion von Amazon RDS handelt, nicht nur um die Oracle-DB-Engine. In ähnlicher Weise bezieht sich der RDS-Begriff „Mandant“ auf jeden Mandanten in einer RDS-Konfiguration, nicht nur auf Oracle. PDBs In der RDS-Dokumentation bezieht sich der Begriff „Oracle Multitenant“ ohne Bindestrich ausschließlich auf die CDB-Architektur der Oracle-Datenbank, die sowohl mit lokalen als auch mit RDS-Bereitstellungen kompatibel ist.

- Bei der Single-Tenant-Konfiguration enthält Ihre RDS-für-Oracle-CDB 1 PDB. Dies ist die Standardkonfiguration beim Erstellen einer CDB. Sie können die ursprüngliche PDB nicht löschen oder weitere hinzufügen. PDBs Sie können die Single-Tenant-

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
	Konfiguration Ihrer CDB später in die Multi-Tenant-Konfiguration konvertieren, aber Sie können dann nicht wieder zur Single-Tenant-Konfiguration zurückkehren. Unabhängig davon, für welche Konfiguration Sie sich entscheiden, enthält Ihre CDB eine einzige anfängliche PDB. In der Mehrmandantenkonfiguration können Sie PDBs später mithilfe von RDS weitere erstellen. APIs Weitere Informationen finden Sie unter Überblick über RDS for Oracle CDBs.		

Automatische Nebenversions-Updates	Wählen Sie „auto Upgrade der Nebenversion aktivieren“, damit Ihre DB-Instance automatisch Upgrades für bevorzugte kleinere DB-Engine-Versionen erhält, sobald sie verfügbar sind. Dies ist das Standardverhalten. Amazon RDS führt im Wartungsfenster automatische Nebenversionenupgrades durch. Wenn Sie die Option auto Nebenversions-Upgrade aktivieren nicht auswählen, wird Ihre DB-Instance nicht automatisch aktualisiert, wenn neue Nebenversionen verfügbar werden.
	
	Weitere Informationen finden Sie unter Automatisches Upgraden der Engine-Unterversion.
		CLI-Option:
		
		`--auto-minor-version-upgrade`
		
		`--no-auto-minor-version-upgrade`
		
		API-Parameter:
		
		`AutoMinorVersionUpgrade`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Availability Zone	Die Availability Zone für Ihre DB-Instance. Verwenden Sie den Standardwert No Preference (Keine Präferenz), außer Sie möchten eine Availability Zone festlegen.
	
	Weitere Informationen finden Sie unter Regionen, Availability Zones und Local Zones.
		CLI-Option:
		
		`--availability-zone`
		
		API-Parameter:
		
		`AvailabilityZone`
			Alle

AWS KMS key	Nur verfügbar, wenn Verschlüsselung auf Verschlüsselung aktivieren festgelegt ist. Wählen Sie das Symbol AWS KMS key , die für die Verschlüsselung dieser DB-Instance verwendet werden soll. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.	CLI-Option:
		
		`--kms-key-id`
		
		API-Parameter:
		
		`KmsKeyId`
			Alle

AWS License Manager Konfiguration	Geben Sie einen Namen für eine AWS License Manager Lizenzkonfiguration ein. Der Name darf maximal 100 Zeichen lang sein und darf nur a-z, A-Z und 0-9 enthalten. Weitere Informationen finden Sie unter the section called “Integrieren mit AWS License Manager”.	CLI-Option: Weitere Informationen finden Sie unter AWS License Manager CLI. API-Parameter: Weitere Informationen finden Sie unter AWS License Manager API.	Db2

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
Sicherungs-Replikation	Wählen Sie Replikation in eine andere AWS -Region aktivieren, um Backups in einer zusätzlichen Region für die Notfallwiederherstellung zu erstellen. Wählen Sie dann die Zielregion für die zusätzlichen Backups aus.	Nicht verfügbar beim Erstellen einer DB-Instance. Informationen zur Aktivierung regionsübergreifender Backups mithilfe der AWS CLI oder RDS-API finden Sie unter Aktivierung regionsübergreifender automatisierter Backups für Amazon RDS	Db2 Oracle PostgreSQL SQL Server

Aufbewahrungszeitraum für Backups	Die Anzahl der Tage, für die automatische Backups der DB-Instance aufbewahrt werden sollen. Setzen Sie diesen Wert für jede nicht verzichtbare DB-Instance auf **1** oder höher.
	
	Weitere Informationen finden Sie unter Einführung in Backups.
		CLI-Option:
		
		`--backup-retention-period`
		
		API-Parameter:
		
		`BackupRetentionPeriod`
			Alle

Backup-Ziel	Wählen Sie AWS Cloud, ob automatische Backups und manuelle Snapshots in der übergeordneten AWS Region gespeichert werden sollen. Klicken Sie auf Outposts (lokal) um sie On-Premises auf Ihrem Outpost zu speichern.
	
	Diese Optionseinstellung gilt nur für RDS in Outposts. Weitere Informationen finden Sie unter Erstellen von DB-Instances für Amazon RDS in AWS Outposts.
		CLI-Option:
		
		`--backup-target`
		
		API-Parameter:
		
		`BackupTarget`
			MySQL, PostgreSQL, SQL Server

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Backup window	Der Zeitraum, in dem Amazon RDS automatisch ein Backup der DB-Instance erstellt. Wenn Sie keine bestimmte Zeit haben, zu der Sie Ihre Datenbank sichern möchten, verwenden Sie den Standardwert No Preference (Keine Präferenz).
	
	Weitere Informationen finden Sie unter Einführung in Backups.
		CLI-Option:
		
		`--preferred-backup-window`
		
		API-Parameter:
		
		`PreferredBackupWindow`
			Alle

Zertifizierungsstelle	Die Zertifizierungsstelle (CA) für das Serverzertifikat, das von der DB-Instance verwendet wird.
	
	Weitere Informationen finden Sie unter .
		CLI-Option:
		
		`--ca-certificate-identifier`
		
		RDS-API-Parameter:
		
		`CACertificateIdentifier`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Zeichensatz	Der Zeichensatz für die DB-Instance. Der Standardwert von AL32UTF8für den DB-Zeichensatz gilt für den universellen Unicode-5.0-UTF-8-Zeichensatz. Sie können den DB-Zeichensatz nicht ändern, nachdem Sie die DB-Instance erstellt haben.
	
	In einer Single-Tenant-Konfiguration wirkt sich ein nicht standardmäßiger DB-Zeichensatz nur auf die PDB aus, nicht auf die CDB. Weitere Informationen finden Sie unter Single-Tenant-Konfiguration der CDB-Architektur.
	
	Der DB-Zeichensatz unterscheidet sich vom nationalen Zeichensatz, der als NCHAR-Zeichensatz bezeichnet wird. Im Gegensatz zum DB-Zeichensatz spezifiziert der NCHAR-Zeichensatz die Kodierung für Spalten der NCHAR-Datentypen (NCHAR, und NCLOB) NVARCHAR2, ohne die Datenbank metadaten zu beeinflussen.
	
	Weitere Informationen finden Sie unter RDS for Oracle-Zeichensätze.
		CLI-Option:
		
		`--character-set-name`
		
		API-Parameter:
		
		`CharacterSetName`
			Oracle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Kollation	Eine Gruppierung auf Serverebene für Ihre DB-Instance.
	
	Weitere Informationen finden Sie unter Sortierung auf Serverebene bei Microsoft SQL Server.
		CLI-Option:
		
		`--character-set-name`
		
		API-Parameter:
		
		`CharacterSetName`
			SQL Server

Tags zu Snapshots kopieren	Diese Option kopiert alle DB-Instance-Tags in einen DB-Snapshot, wenn Sie einen Snapshot erstellen.
	
	Weitere Informationen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.
		CLI-Option:
		
		`--copy-tags-to-snapshot`
		
		`--no-copy-tags-to-snapshot`
		
		RDS-API-Parameter:
		
		`CopyTagsToSnapshot`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Datenbank-Authentifizierung	Die Datenbankauthentifizierungsoption, die Sie verwenden möchten.
	
	Wählen Sie Passwortauthentifizierung aus, um Datenbankbenutzer ausschließlich mit Datenbankpasswörtern zu authentifizieren.
	
	Wählen Sie Password and IAM DB authentication (Passwort- und IAM-DB-Authentifizierung) aus, um Datenbankbenutzer mit Datenbankpasswörtern und Benutzeranmeldeinformationen über Benutzer und Rollen zu authentifizieren. Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL. Diese Option wird nur für MySQL und PostgreSQL unterstützt.
	
	Wählen Sie Passwort- und Kerberos-Authentifizierung aus, um Datenbankbenutzer mit Datenbankkennwörtern und Kerberos-Authentifizierung über ein Created with zu authentifizieren. AWS Managed Microsoft AD AWS Directory Service Als Nächstes wählen Sie das Verzeichnis oder Create a new Directory (Ein neues Verzeichnis erstellen) aus.
	
	Weitere Informationen finden Sie unter einem der folgenden Themen:
	
	•
		IAM:
		
		CLI-Option:
		
		`--enable-iam-database-authentication`
		
		`--no-enable-iam-database-authentication`
		
		RDS-API-Parameter:
		
		`EnableIAMDatabaseAuthentication`
		
		Kerberos:
		
		CLI-Option:
		
		`--domain`
		
		`--domain-iam-role-name`
		
		RDS-API-Parameter:
		
		`Domain`
		
		`DomainIAMRoleName`
			Variiert je nach Authentifizierungsart

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
	KerberosAuthentifizierung für Amazon RDS for Db2 verwenden • KerberosAuthentifizierung für Amazon RDS for MySQL verwenden • Konfigurieren der Kerberos-Authentifizierung für Amazon RDS for Oracle • Verwenden der Kerberos-Authentifizierung mit Amazon RDS for PostgreSQL		
Datenbankverwaltungstyp	Klicken Sie auf Amazon RDS, wenn Sie Ihre Umgebung nicht anpassen müssen. Klicken Sie auf Amazon RDS Custom, wenn Sie die Datenbank, das Betriebssystem und die Infrastruktur anpassen möchten. Weitere Informationen finden Sie unter Amazon RDS Custom.	Für die CLI und API geben Sie den Typ des Datenbankmoduls an.	Oracle SQL Server

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Datenbank port	Der Port, über den Sie auf die DB-Instance zugreifen wollen. Der Standardport wird angezeigt.
	
	Note
	Die Firewalls einiger Unternehmen blockieren Verbindungen zu den standardmäßigen MariaDB-, MySQL- und PostgreSQL-Ports. Wenn Ihre Unternehmensfirewall den Standardport blockiert, geben Sie einen anderen Port für Ihre DB-Instance ein.
		CLI-Option:
		
		`--port`
		
		RDS-API-Parameter:
		
		`Port`
			Alle

DB-Engine-Version	Die Version der Datenbank-Engine, die Sie verwenden möchten.	CLI-Option:
		
		`--engine-version`
		
		RDS-API-Parameter:
		
		`EngineVersion`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

DB-Instance-Klasse	Die Konfiguration für Ihre DB-Instance. Eine DB-Instance-Klasse db.t3.small hat beispielsweise 2 GiB Arbeitsspeicher, 2 V, 1 virtuellen KernCPUs, eine variable ECU und eine moderate I/O-Kapazität.
	
	Wählen Sie möglichst eine DB-Instance-Klasse, die groß genug ist, um einen typischer Abfragesatz im Arbeitsspeicher halten zu können. Wenn Arbeitssätze im Arbeitsspeicher gehalten werden, kann das System das Schreiben auf die Festplatte vermeiden, was die Leistung verbessert. Weitere Informationen finden Sie unter .
	
	In RDS for Oracle können Sie Zusätzliche Speicherkonfigurationen einschließen. Diese Konfigurationen sind für ein hohes Verhältnis von Speicher zu vCPU optimiert. Beispiel, db.r5.6xlarge.tpc2.mem4x ist eine db.r5.8x DB-Instance, die 2 Threads pro Kern (tpc2) und 4x den Speicher einer standardmäßigen db.r5.6xlare DB-Instance hat. Weitere Informationen finden Sie unter RDS für Oracle-DB-Instance-Klassen.
		CLI-Option:
		
		`--db-instance-class`
		
		RDS-API-Parameter:
		
		`DBInstanceClass`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

DB-Instance-Kennung	Der Name der DB-Instance. Benennen Sie Ihre DB-Instances auf die gleiche Weise wie Ihre lokalen Server. Ihre DB-Instance-ID kann bis zu 63 alphanumerische Zeichen enthalten und muss für Ihr Konto in der von Ihnen ausgewählten Region eindeutig sein. AWS	CLI-Option:
		`--db-instance-identifier`
		RDS-API-Parameter:
		`DBInstanceIdentifier`
			Alle

DB-Parametergruppe	Eine Parametergruppe für die DB-Instance. Sie können die Standardparametergruppe wählen oder eine benutzerdefinierte Parametergruppe erstellen.
	Wenn Sie das BYOL-Modell für RDS for Db2 verwenden, müssen Sie vor dem Erstellen einer DB-Instance zunächst eine benutzerdefinierte Parametergruppe erstellen, die Ihr und enthält. IBM Site ID IBM Customer ID Weitere Informationen finden Sie unter Bringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit.
	Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
		CLI-Option:
		`--db-parameter-group-name`
		RDS-API-Parameter:
		`DBParameterGroupName`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

DB-Subnetzgruppe	Die DB-Subnetzgruppe, die Sie für den DB-Cluster verwenden möchten. Wählen Sie Choose existing (Vorhandene wählen) aus, um eine vorhandene DB-Subnetzgruppe zu verwenden. Wählen Sie dann die erforderliche Subnetzgruppe aus der Dropdown-Liste Existing DB subnet groups (Vorhandene DB-Subnetzgruppen) aus.
	
	Wählen Sie Automatic setup (Automatische Einrichtung) aus, damit RDS eine kompatible DB-Subnetzgruppe auswählen kann. Wenn keine vorhanden ist, erstellt RDS eine neue Subnetzgruppe für Ihren Cluster.
	
	Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.
		CLI-Option:
		
		`--db-subnet-group-name`
		
		RDS-API-Parameter:
		
		`DBSubnetGroupName`
			Alle

Dediziertes Protokoll-Volume	Verwenden Sie ein dediziertes Protokoll-Volume (DLV), um Datenbank-Transaktionsprotokolle auf einem Speicher-Volume zu speichern, das von dem Volume mit den Datenbanktabellen getrennt ist.
	
	Weitere Informationen finden Sie unter Verwendung eines dedizierten Protokoll-Volumes (DLV).
		CLI-Option:
		
		`--dedicated-log-volume`
		
		RDS-API-Parameter:
		
		`DedicatedLogVolume`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Löschschutz	Um zu verhindern, dass die DB-Instance gelöscht wird, können Sie die Option Enable deletion protection (Löschschutz aktivieren) aktivieren. Wenn Sie eine Produktions-DB-Instance mit dem erstellen AWS Management Console, ist der Löschschutz standardmäßig aktiviert.
	
	Weitere Informationen finden Sie unter Löschen einer DB-Instance.
		CLI-Option:
		
		`--deletion-protection`
		
		`--no-deletion-protection`
		
		RDS-API-Parameter:
		
		`DeletionProtection`
			Alle

Verschlüsselung	Mit Enable encryption (Verschlüsselung aktivieren) wird die Verschlüsselung ruhender Daten für diese DB-Instance aktiviert.
	
	Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
		CLI-Option:
		
		`--storage-encrypted`
		
		`--no-storage-encrypted`
		
		RDS-API-Parameter:
		
		`StorageEncrypted`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Verbesserte Überwachung	Wählen Sie Erweiterte Überwachung aktivieren, um Metriken in Echtzeit für das Betriebssystem zu erhalten, in dem Ihre DB-Instance ausgeführt wird.
	
	Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).
		CLI-Optionen:
		
		`--monitoring-interval`
		
		`--monitoring-role-arn`
		
		RDS-API-Parameter:
		
		`MonitoringInterval`
		
		`MonitoringRoleArn`
			Alle

Engine type (Engine-Typ)	Wählen Sie die Datenbank-Engine aus, die für diese DB-Instance verwendet werden soll.	CLI-Option:
		
		`--engine`
		
		RDS-API-Parameter:
		
		`Engine`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Anfänglicher Datenbankname	Der Name für die Datenbank in der DB-Instance. Wenn Sie keinen Namen angeben, erstellt Amazon RDS keine Datenbank in der DB-Instance (außer Oracle und PostgreSQL). Der Name darf kein von der Datenbank-Engine reserviertes Wort sein und hat je nach DB-Engine andere Einschränkungen.
	
	Db2:
	
	• Er muss 1–8 alphanumerische Zeichen enthalten.
	• Es muss mit a-z, A-Z, @, $ oder # beginnen, gefolgt von a-z, A-Z, 0-9, _, @, # oder $.
	• Leerzeichen dürfen nicht enthalten sein.
	• Weitere Informationen finden Sie unter Weitere Überlegungen.
	
	MariaDB und MySQL:
	
	• Er muss 1–64 alphanumerische Zeichen enthalten.
	
	Oracle:
	
	•
		CLI-Option:
		
		`--db-name`
		
		RDS-API-Parameter:
		
		`DBName`
			Alle außer SQL Server

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
	Er muss 1–8 alphanumerische Zeichen enthalten. • Er darf nicht sei `NULL`. Der Standardwert ist `ORCL`. • Er muss mit einem Buchstaben beginnen. PostgreSQL: • Er muss 1–63 alphanumerische Zeichen enthalten. • Er muss mit einem Buchstaben oder einem Unterstrich beginnen. Nachfolgende Zeichen können Groß-, Kleinbuchstaben oder Zahlen (0-9) sein. • Der anfängliche Datenbankname lautet `postgres`.		

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

License	Gültige Werte für das Lizenzmodell:
	• bring-your-own-licenseoder Marketplace-Lizenz für Db2.
	• general-public-licensefür MariaDB.
	• license-included für Microsoft SQL Server.
	• general-public-licensefür MySQL.
	• inklusive Lizenz oder bring-your-own-licensefür Oracle.
	• postgresql-license für PostgreSQL.
		CLI-Option:
		`--license-model`
		RDS-API-Parameter:
		`LicenseModel`
			Alle

Protokoll exporte	Die Typen von Datenbank-Protokoll ldateien, die in Amazon CloudWatch Logs veröffentlicht werden sollen.
	Weitere Informationen finden Sie unter Datenbankprotokolle in Amazon CloudWatch Logs veröffentlichen.
		CLI-Option:
		`--enable-cloudwatch-logs-exports`
		RDS-API-Parameter:
		`EnableCloudwatchLogsExports`
			Alle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Wartungsfenster	Das 30-Minuten-Fenster, in dem anstehende Änderungen an Ihrer DB-Instance durchgeführt werden. Wählen Sie No Preference (Keine Präferenz) aus, wenn der Zeitraum nicht wichtig ist.
	
	Weitere Informationen finden Sie unter Amazon RDS-Wartungsfenster.
		CLI-Option:
		
		`--preferred-maintenance-window`
		
		RDS-API-Parameter:
		
		`PreferredMaintenanceWindow`
			Alle

Hauptanmeldedaten verwalten in AWS Secrets Manager	Wählen Sie Master-Anmeldeinformationen verwalten in AWS Secrets Manager aus, um das Hauptbenutzerpasswort in Secrets Manager geheim zu verwalten.
	
	Wählen Sie optional einen KMS-Schlüssel zum Schutz des Secrets aus. Wählen Sie aus den KMS-Schlüsseln in Ihrem Konto oder geben Sie den Schlüssel eines anderen Kontos ein.
	
	Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.
		CLI-Option:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		RDS-API-Parameter:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Hauptpass wort	Das Passwort für das Masterben utzerkonto. Das Passwort hat die folgende Anzahl druckbarer ASCII-Zei chen (ausgenommen `/`, `"`, ein Leerzeich en und `@`), abhängig von der DB-Engine:
	• Db2:8—255
	• Oracle: 8–30
	• MariaDB und MySQL: 8–41
	• SQL Server und PostgreSQL: 8–128
		CLI-Option:
		`--master-user-pass word`
		RDS-API-Parameter:
		`MasterUserPassword`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Master-Benutzername	Der Name, den Sie als Hauptbenutzernamen für die Anmeldung bei der DB-Instance mit allen Datenbankberechtigungen verwenden. Beachten Sie die folgenden Benennungseinschränkungen:
	• Der Name kann 1–16 alphanumerische Zeichen und Unterstriche enthalten.
	• Das erste Zeichen muss ein Buchstabe sein.
	• Der Name darf kein von der Datenbank-Engine reserviertes Wort sein.
	
	Sie können den Namen des Hauptbenutzers nicht ändern, nachdem die DB-Instance erstellt wurde.
	
	Für Db2 empfehlen wir, denselben Master-Benutzernamen wie Ihren selbstverwalteten Db2-Instanznamen zu verwenden.
	
	Weitere Informationen zu Berechtigungen, die dem Masterbenutzer gewährt werden, finden Sie unter Berechtigungen von Hauptbenutzerkonten.
		CLI-Option:
		
		`--master-username`
		
		RDS-API-Parameter:
		
		`MasterUsername`
			Alle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Microsoft SQL Server Windows-Authentifizierung	Aktivieren Sie die Microsoft SQL Server Windows-Authentifizierung und durchsuchen Sie dann das Verzeichnis, um das Verzeichnis auszuwählen, in dem Sie autorisierten Domänenbenutzern die Authentifizierung mit dieser SQL Server-Instance mithilfe der Windows-Authentifizierung erlauben möchten.	CLI-Optionen:
		`--domain`
		`--domain-iam-role-name`
		RDS-API-Parameter:
		`Domain`
		`DomainIAMRoleName`
			SQL Server

Multi-AZ-Bereitstellung	Erstellen Sie eine Standby-Instance, um eine passive, sekundäre Kopie Ihrer DB-Instance in einer anderen Availability Zone für die Failover-Unterstützung zu erstellen. Wir empfehlen Multi-AZ, um die hohe Verfügbarkeit von Produktions-Workloads sicherzustellen.
	Für die Entwicklung und das Testen können Sie Do not create a standby instance (Keine Standby-Instance erstellen) auswählen.
	Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.
		CLI-Option:
		`--multi-az`
		`--no-multi-az`
		RDS-API-Parameter:
		`MultiAZ`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Nationaler Zeichensatz (NCHAR)	Der nationale Zeichensatz für Ihre DB-Instance, der allgemein als NCHAR-Zeichensatz bezeichnet wird. Sie können den nationalen Zeichensatz entweder auf AL16 UTF16 (Standard) oder UTF-8 setzen. Sie können den nationalen Zeichensatz nicht ändern, nachdem Sie die DB-Instance erstellt haben.
	
	Der nationale Zeichensatz unterscheidet sich vom DB-Zeichensatz. Im Gegensatz zum DB-Zeichensatz gibt der nationale Zeichensatz die Kodierung nur für Spalten der NCHAR-Datentypen (NCHAR, und NCLOB) an NVARCHAR2, ohne dass sich dies auf die Datenbankmetadaten auswirkt.
	
	Weitere Informationen finden Sie unter RDS for Oracle-Zeichensätze.
		CLI-Option:
		
		`--nchar-character-set-name`
		
		API-Parameter:
		
		`NcharCharacterSetName`
			Oracle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Network type (Netzwerktyp)	Die von der DB-Instance unterstützten IP-Adressierungsprotokolle.
	
	IPv4(Standard), um anzugeben, dass Ressourcen mit der DB-Instance nur über das Adressierungsprotokoll Version 4 (IPv4) des Internetprotokolls kommunizieren können.
	
	Dual-Stack-Modus, um anzugeben, dass Ressourcen mit der DB-Instance über IPv4 Internet Protocol Version 6 (IPv6) oder beides kommunizieren können. Verwenden Sie den Dual-Stack-Modus, wenn Sie über Ressourcen verfügen, die über das IPv6 Adressierungsprotokoll mit Ihrer DB-Instance kommunizieren müssen. Stellen Sie außerdem sicher, dass Sie allen Subnetzen in der von Ihnen angegebenen DB-Subnetzgruppe einen IPv6 CIDR-Block zuordnen.
	
	Weitere Informationen finden Sie unter Amazon-RDS-IP-Adressierung.
		CLI-Option:
		
		`--network-type`
		
		RDS-API-Parameter:
		
		`NetworkType`
			Alle

Option group	Eine Optionsgruppe für die DB-Instance. Sie können die Standardoptionsgruppe wählen oder eine benutzerdefinierte Optionsgruppe erstellen.
	
	Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.
		CLI-Option:
		
		`--option-group-name`
		
		RDS-API-Parameter:
		
		`OptionGroupName`
			Alle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Performance Insights	Aktivieren Sie Performance Insights, um die Auslastung Ihrer DB-Instance zu überwachen, damit Sie Ihre Datenbankleistung analysieren und Fehler beheben können.
	
	Wählen Sie eine Aufbewahrungsfrist, um festzulegen, in welchem Umfang die Performance Insights-Datenhistorie aufbewahrt werden soll. Die Aufbewahrungseinstellung ist Standard (7 Tage). Um Ihre Leistungsdaten länger aufzubewahren, geben Sie 1–24 Monate an. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.
	
	Wählen Sie einen KMS-Schlüssel aus, der zum Schutz des Schlüssels für die Verschlüsselung dieses Datenbankvolumes verwendet wird. Wählen Sie aus den KMS-Schlüsseln in Ihrem Konto oder geben Sie den Schlüssel eines anderen Kontos ein.
	
	Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.
		CLI-Optionen:
		
		`--enable-performance-insights`
		
		`--no-enable-performance-insights`
		
		`--performance-insights-retention-period`
		
		`--performance-insights-kms-key-id`
		
		RDS-API-Parameter:
		
		`EnablePerformanceInsights`
		
		`PerformanceInsightsRetentionPeriod`
		
		`PerformanceInsightsKMSKeyId`
			Alle außer Db2

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Bereitgestellte IOPS	Der bereitgestellte IOPS (I/O-Operationen pro Sekunde)-Wert für die DB-Instance. Diese Einstellung ist nur verfügbar, wenn Sie für Storage type (Speichertyp) eine der folgenden Optionen auswählen:
	• General purpose SSD (gp3) (Allzweck SSD (gp3))
	• Provisioned IOPS SSD (io1) (Bereitgestellte IOPS SSD (io1))
	• Bereitgestellte IOPS-SSD (io2)
	
	Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.
		CLI-Option:
		`--iops`
		RDS-API-Parameter:
		`Iops`
			Alle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Öffentlicher Zugriff	Yes (Ja), um der DB-Instance eine öffentliche IP-Adresse zuzuweisen (was bedeutet, dass sie von außerhalb der VPC zugänglich ist). Damit der öffentliche Zugriff für eine DB-Instance möglich ist, muss sie sich auch in einem öffentlichen Subnetz der VPC befinden.
	
	Nein, damit nur innerhalb der VPC auf die DB-Instance zugegriffen werden kann.
	
	Weitere Informationen finden Sie unter Ausblenden einer DB-Instance in einer VPC vor dem Internet.
	
	Um eine Verbindung zu einer DB-Instance von außerhalb ihrer VPC herzustellen, muss die DB-Instance öffentlich zugänglich sein. Außerdem muss der Zugriff unter Verwendung der Regeln für eingehenden Datenverkehr der Sicherheitsgruppe der DB-Instance gewährt werden. Darüber hinaus müssen andere Anforderungen erfüllt sein. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.
	
	Wenn Ihre DB-Instance nicht öffentlich zugänglich ist, verwenden Sie eine AWS Site-to-Site VPN-Verbindung oder eine Verbindung, um von einem AWS Direct Connect privaten Netzwerk aus
		CLI-Option:
		
		`--publicly-accessible`
		
		`--no-publicly-accessible`
		
		RDS-API-Parameter:
		
		`PubliclyAccessible`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
	darauf zuzugreifen. Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.		

Erweiterter RDS-Support	Wählen Sie Enable RDS Extended Support aus, damit unterstützte Engine-Hauptversionen auch nach Ablauf des RDS-Standard-Supports weiter ausgeführt werden können.
	
	Wenn Sie eine DB-Instance erstellen, verwendet Amazon RDS standardmäßig RDS Extended Support. Um zu verhindern, dass nach dem Ende des Standard-Supports für RDS eine neue DB-Instance erstellt wird, und um Gebühren für RDS Extended Support zu vermeiden, deaktivieren Sie diese Einstellung. Für Ihre vorhandenen DB-Instances fallen bis zum Startdatum der Preise für RDS Extended Support keine Gebühren an.
	
	Weitere Informationen finden Sie unter Erweiterter Support für Amazon RDS mit Amazon RDS.
		CLI-Option:
		
		`--engine-lifecycle-support`
		
		RDS-API-Parameter:
		
		`EngineLifecycleSupport`
			MySQL
			
			PostgreSQL

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines
RDS-Proxy	Wählen Sie Create an RDS Proxy (RDS-Proxy erstellen) aus, um einen Proxy für Ihre DB-Instance zu erstellen. Amazon RDS erstellt automatisch eine IAM-Rolle und ein Secrets-Manager-Secret für den Proxy. Weitere Informationen finden Sie unter Amazon RDS-Proxy .	Nicht verfügbar beim Erstellen einer DB-Instance.	MariaDB MySQL PostgreSQL

Automatische Speicherskalierung	Aktivieren Sie die automatische Skalierung des Speichers, um Amazon RDS zu ermöglichen, den Speicher bei Bedarf automatisch zu erhöhen. So wird vermieden, dass Ihrer DB-Instance der Speicherplatz ausgeht.
	
	Verwenden Sie Maximum storage threshold (Maximaler Speicherschwellenwert), um die Obergrenze für Amazon RDS festzulegen, bei der der Speicherplatz für Ihre DB-Instance automatisch vergrößert wird. Der Standardwert ist 1.000 GiB.
	
	Weitere Informationen finden Sie unter Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung.
		CLI-Option:
		
		`--max-allocated-storage`
		
		RDS-API-Parameter:
		
		`MaxAllocatedStorage`
			Alle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Storage throughput (Speicherdurchsatz)	Der Speicherdurchsatzwert für die DB-Instance. Diese Einstellung ist nur verfügbar, wenn Sie für Storage type (Speichertyp) die Option General purpose SSD (gp3) (Allzweck-SSD (gp3)) auswählen.
	
	Weitere Informationen finden Sie unter GP3-Speicher (empfohlen).
		CLI-Option:
		
		`--storage-throughput`
		
		RDS-API-Parameter:
		
		`StorageThroughput`
			Alle

Speichertyp	Der Speicherplatztyp für die DB-Instance.
	
	Wenn Sie sich für General Purpose SSD (gp3) (Allzweck-SSD (gp3)) entscheiden, können Sie unter Advanced Settings (Erweiterte Einstellungen) zusätzliche bereitgestellte IOPS) und zusätzlichen Speicherdurchsatz bereitstellen.
	
	Wenn Sie Provisioned IOPS SSD (io1) oder Provisioned IOPS SSD (io2) wählen, geben Sie den Wert Provisioned IOPS ein.
	
	Weitere Informationen finden Sie unter Amazon RDS-Speichertypen.
		CLI-Option:
		
		`--storage-type`
		
		RDS-API-Parameter:
		
		`StorageType`
			Alle

Subnetzgruppe	Eine DB-Subnetzgruppe, welche dieser DB-Instance zugeordnet werden soll.
	
	Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.
		CLI-Option:
		
		`--db-subnet-group-name`
		
		RDS-API-Parameter:
		
		`DBSubnetGroupName`
			Alle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Name der Tenant-Datenbank	Der Name Ihrer anfänglichen PDB in der Multi-Tenant-Konfiguration der Oracle-Architektur. Diese Einstellung ist nur verfügbar, wenn Sie für die Architekturkonfiguration die Multi-Tenant-Konfiguration wählen.
	
	Der Name der Tenant-Datenbank muss sich vom Namen Ihrer CDB unterscheiden, die als `RDSCDB` benannt ist. Sie können den Namen der CDB nicht ändern.
		CLI-Option:
		
		`--db-name`
		
		RDS-API-Parameter:
		
		`DBName`
			Oracle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Hauptbenutzername für die Tenant-Datenbank	Der Name, den Sie als Hauptbenutzernamen für die Anmeldung bei Ihrer Tenant-Datenbank (PDB) mit allen Datenbankberechtigungen verwenden wollen. Diese Einstellung ist nur verfügbar, wenn Sie für die Architekturkonfiguration die Multi-Tenant-Konfiguration wählen.
	
	Beachten Sie die folgenden Benennungseinschränkungen:
	
	• Der Name kann 1–16 alphanumerische Zeichen und Unterstriche enthalten.
	• Das erste Zeichen muss ein Buchstabe sein.
	• Der Name darf kein von der Datenbank-Engine reserviertes Wort sein.
	
	Sie haben nicht die folgenden Möglichkeiten:
	
	• Ändern Sie den Tenant-Hauptbenutzernamen, nachdem Sie die Tenant-Datenbank erstellt haben.
	• Melden Sie sich mit dem Tenant-Hauptbenutzernamen bei der CDB an.
		CLI-Option:
		
		`--master-username`
		
		RDS-API-Parameter:
		
		`MasterUsername`
			Oracle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Master-Passwort für die Tenant-Datenbank	Das Passwort für das Master-Benutzerkonto Ihrer Tenant-Datenbank (PDB). Diese Einstellung ist nur verfügbar, wenn Sie für die Architekturkonfiguration die Multi-Tenant-Konfiguration wählen.
	
	Das Passwort muss 8–30 druckbare ASCII-Zeichen enthalten (mit Ausnahme von `/`, `"`, Leerzeichen und `@`).
		CLI-Option:
		
		`--master-password`
		
		RDS-API-Parameter:
		
		`MasterPassword`
			Oracle

Zeichensatz für die Tenant-Datenbank	Der Zeichensatz der anfänglichen Tenant-Datenbank. Diese Einstellung ist nur verfügbar, wenn Sie für die Architekturkonfiguration die Multi-Tenant-Konfiguration wählen. Es werden nur CDB-Instances von RDS für Oracle unterstützt.
	
	Der Standardwert von AL32UTF8für den Zeichensatz der Tenant-Datenbank gilt für den universellen UTF-8-Zeichensatz Unicode 5.0. Sie können einen Zeichensatz für die Tenant-Datenbank wählen, der sich vom Zeichensatz der CDB unterscheidet.
	
	Weitere Informationen finden Sie unter RDS for Oracle-Zeichensätze.
		CLI-Option:
		
		`--character-set-name`
		
		RDS-API-Parameter:
		
		`CharacterSetName`
			Oracle

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Nationaler Zeichensatz für die Tenant-Datenbank	Der nationale Zeichensatz für Ihre Tenant–Datenbank, der allgemein als NCHAR-Zeichensatz bezeichnet wird. Diese Einstellung ist nur verfügbar, wenn Sie für die Architekturkonfiguration die Multi-Tenant-Konfiguration wählen. Es werden nur CDB-Instances von RDS für Oracle unterstützt.
	
	Sie können den nationalen Zeichensatz entweder auf AL16UTF16(Standard) oder UTF-8 setzen. Sie können den nationalen Zeichensatz nicht ändern, nachdem Sie die Tenant-Datenbank erstellt haben.
	
	Der nationale Zeichensatz der Tenant-Datenbank unterscheidet sich vom Zeichensatz der Tenant-Datenbank. Der nationale Zeichensatz spezifiziert die Kodierung nur für Spalten, die den NCHAR-Datentyp (NCHAR, NVARCHAR2 und NCLOB) verwenden, und wirkt sich nicht auf Datenbank-Metadaten aus.
	
	Weitere Informationen finden Sie unter RDS for Oracle-Zeichensätze.
		CLI-Option:
		
		`--nchar-character-set-name`
		
		API-Parameter:
		
		`NcharCharacterSetName`
			Oracle

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Unterstützte DB-Engines

Zeitzone	Die Zeitzone für Ihre DB-Instance. Wenn Sie keine Zeitzone auswählen , verwendet Ihre DB-Instance die Standardzeitzone. Sie können die Zeitzone nicht ändern, nachdem die DB-Instance erstellt wurde.
	
	Weitere Informationen erhalten Sie unter Lokale Zeitzone für Amazon RDS für Db2-DB-Instances und Lokale Zeitzone für Microsoft SQL Server-DB-Instances.
		CLI-Option:
		
		`--timezone`
		
		RDS-API-Parameter:
		
		`Timezone`
			Db2
			
			SQL Server
			
			RDS Custom für SQL Server

Virtual Private Cloud (VPC)	Eine VPC, die auf dem Amazon-VPC-Service basiert und mit dieser DB-Instance verknüpft werden soll. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.	Für die CLI und API geben Sie die VPC-Sicherheitsgruppe IDs an.	Alle

VPC-Sicherheitsgruppe (Firewall)	Die der DB-Instance zugeordneten Sicherheitsgruppe.
	
	Weitere Informationen finden Sie unter Überblick über VPC-Sicherheitsgruppen.
		CLI-Option:
		
		`--vpc-security-group-ids`
		
		RDS-API-Parameter:
		
		`VpcSecurityGroupIds`
			Alle

Erstellen von Amazon RDS Amazon mit AWS CloudFormation

Amazon RDS ist integriert AWS CloudFormation, ein Service, der Ihnen hilft, Ihre AWS Ressourcen zu modellieren und einzurichten, sodass Sie weniger Zeit mit der Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur verbringen müssen. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen (wie DB-Instances und DB-Parametergruppen) beschreibt und diese Ressourcen für Sie AWS CloudFormation bereitstellt und konfiguriert.

Wenn Sie Ihre Vorlage verwenden AWS CloudFormation, können Sie sie wiederverwenden, um Ihre RDS konsistent und wiederholt einzurichten. Beschreiben Sie Ihre Ressourcen einmal und stellen Sie dann dieselben Ressourcen immer wieder in mehreren AWS Konten und Regionen bereit.

RDS und AWS CloudFormation Vorlagen

AWS CloudFormation Vorlagen sind formatierte Textdateien in JSON oderYAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation Stacks bereitstellen möchten. Wenn Sie mit JSON oder nicht vertraut sind, können Sie AWS CloudFormation Designer verwendenYAML, um Ihnen bei den ersten Schritten mit AWS CloudFormation Vorlagen zu helfen. Weitere Informationen finden Sie unter Was ist AWS CloudFormation -Designer? im AWS CloudFormation -Benutzerhandbuch.

RDS unterstützt die Erstellung von Ressourcen in AWS CloudFormation. Weitere Informationen, einschließlich Beispielen JSON und YAML Vorlagen für diese Ressourcen, finden Sie in der Referenz zum RDS Ressourcentyp im AWS CloudFormation Benutzerhandbuch.

Erfahren Sie mehr über AWS CloudFormation

Weitere Informationen AWS CloudFormation dazu finden Sie in den folgenden Ressourcen:

- AWS CloudFormation
- AWS CloudFormation Benutzerhandbuch
- AWS CloudFormation APIReferenz
- AWS CloudFormation Benutzerhandbuch für die Befehlszeilenschnittstelle

Verbindung zu einer Amazon RDS DB-Instance herstellen

Bevor Sie eine Verbindung mit einer DB-Instance herstellen können, müssen Sie die DB-Instance erstellen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance. Nachdem Amazon RDS Ihre DB-Instance bereitgestellt hat, verwenden Sie eine beliebige Standard-Client-Anwendung oder ein beliebiges Standarddienstprogramm für Ihre DB-Engine, um eine Verbindung mit der DB-Instance herzustellen. Geben Sie in der Verbindungszeichenfolge die DNS Adresse vom DB-Instance-Endpunkt als Host-Parameter an. Sie geben auch die Portnummer vom DB-Instance-Endpunkt als Portparameter an.

Weitere Informationen zum Suchen von Verbindungsinformationen für eine RDS Amazon-DB-Instance oder zu Szenarien für den Zugriff auf eine DB-Instance in einer VPC finden Sie in den folgenden Themen.

- Suchen der Verbindungsinformationen für eine RDS Amazon-DB-Instance
- Szenarien für den Zugriff auf eine DB-Instance in einem VPC

Suchen der Verbindungsinformationen für eine RDS Amazon-DB-Instance

Die Verbindungsinformationen für eine DB-Instance umfassen ihren Endpunkt, ihren Port und einen gültigen Datenbankbenutzer, z. B. den Masterbenutzer. Nehmen wir beispielsweise für eine My SQL DB-Instance an, dass der Endpunktwert ist`mydb.123456789012.us-east-1.rds.amazonaws.com`. In diesem Fall ist `3306` der Port-Wert und der Datenbankbenutzer ist `admin`. Angesichts dieser Informationen geben Sie die folgenden Werte in einer Verbindungszeichenfolge an:

- Geben Sie für Host oder DNS Hostname oder -namen an`mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Als Port `3306`.
- Geben Sie für Benutzer a `admin`.

Der Endpunkt ist für jede DB-Instance eindeutig, und die Werte des Ports und des Benutzers können variieren. Die folgende Liste zeigt den gebräuchlichsten Port für jede DB-Engine:

- Db2 — 50000
- MariaDB – 3306

- Microsoft SQL Server — 1433
- Mein SQL — 3306
- Oracle – 1521
- Postkarte — 5432 SQL

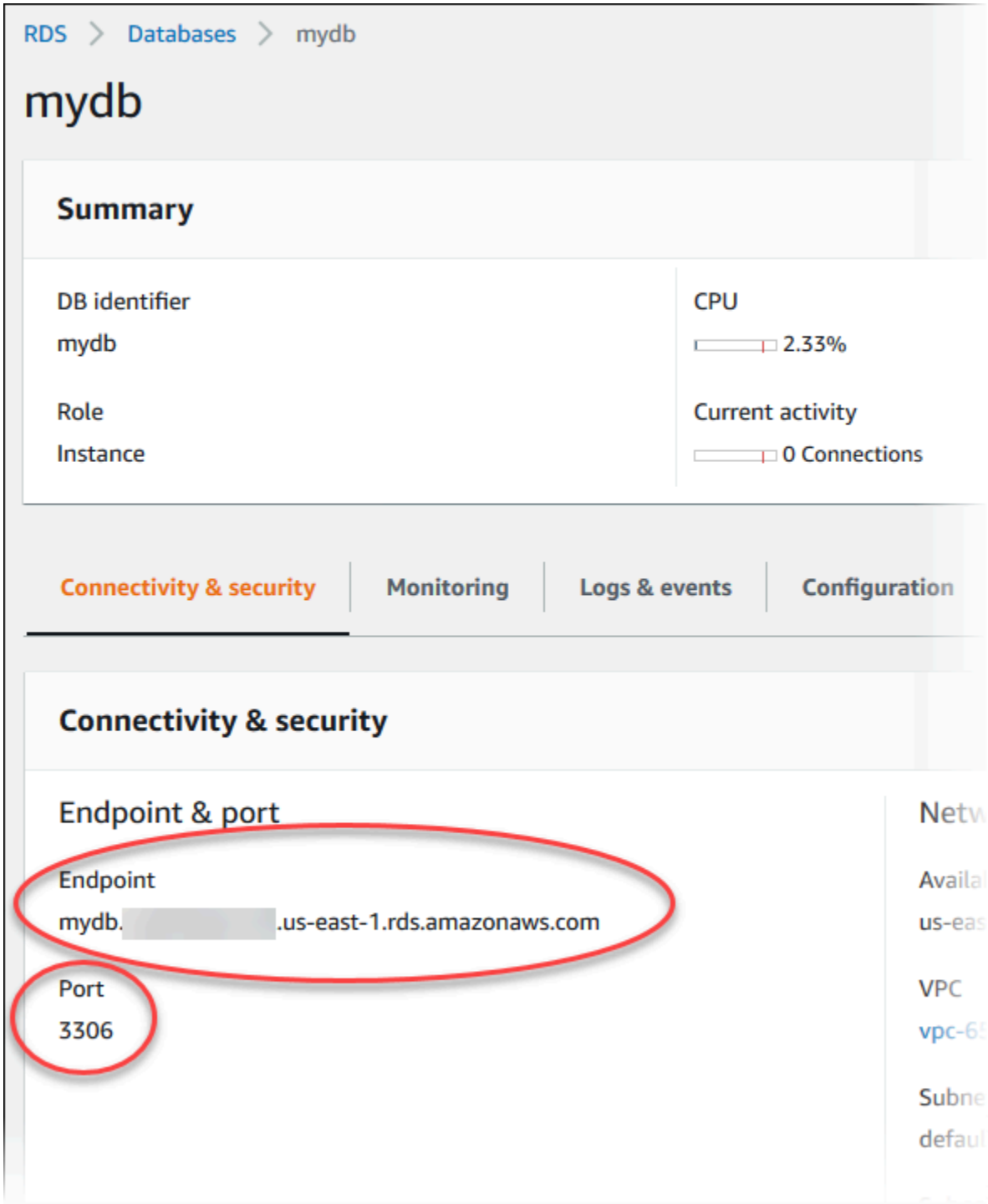
Um eine Verbindung mit einer DB-Instance herzustellen, verwenden Sie einen beliebigen Client für eine DB-Engine. Sie können beispielsweise das MySQL-Hilfsprogramm verwenden, um eine Verbindung zu einer MariaDB- oder My SQL DB-Instance herzustellen. Sie können Microsoft SQL Server Management Studio verwenden, um eine Verbindung zu einer SQL Server-DB-Instance herzustellen. Sie können Oracle SQL Developer verwenden, um eine Verbindung zu einer Oracle-DB-Instance herzustellen. In ähnlicher Weise können Sie das Befehlszeilenprogramm psql verwenden, um eine Verbindung zu einer SQL Postgre-DB-Instance herzustellen.

Verwenden Sie die AWS Management Console, um die Verbindungsinformationen für eine DB-Instance zu finden. Sie können auch den describe-db-instancesBefehl AWS Command Line Interface (AWS CLI) oder die RDS API escribeDBInstancesD-Operation verwenden.

Konsole

Um die Verbindungsinformationen für eine DB-Instance zu finden, finden Sie im AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie im Navigationsbereich auf Datenbanken, um eine Liste Ihrer DB-Instances anzuzeigen.
3. Wählen Sie den Namen der DB-Instance aus, um ihre Details anzuzeigen.
4. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



5. Wenn Sie den Masterbenutzernamen finden müssen, wählen Sie die Registerkarte Konfiguration und den Wert für den Masterbenutzernamen an.

AWS CLI

Rufen Sie den describe-db-instancesBefehl auf, um die Verbindungsinformationen für eine DB-Instance mithilfe von zu ermitteln. AWS CLI Fragen Sie beim Aufruf die DB-Instance-ID, den Endpunkt, den Port und den Masterbenutzernamen ab.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-instances \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-instances ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Die Ausgabe sollte in etwa wie folgt aussehen.

```
[
    [
        "mydb",
        "mydb.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "myoracledb",
        "myoracledb.123456789012.us-east-1.rds.amazonaws.com",
        1521,
        "dbadmin"
    ],
    [
        "mypostgresqldb",
        "mypostgresqldb.123456789012.us-east-1.rds.amazonaws.com",
        5432,
        "postgresadmin"
    ]
]
```

RDS API

Rufen Sie den escribeDBInstancesD-Vorgang auf, um die Verbindungsinformationen für eine DB-Instance mithilfe von Amazon RDS API zu ermitteln. Suchen Sie in der Ausgabe die Werte für die Endpunktadresse, den Endpunktport und den Masterbenutzernamen.

Szenarien für den Zugriff auf eine DB-Instance in einem VPC

Mit Amazon Virtual Private Cloud (AmazonVPC) können Sie AWS Ressourcen wie Amazon RDS DB-Instances in einer virtuellen privaten Cloud (VPC) starten. Wenn Sie Amazon verwendenVPC, haben Sie die Kontrolle über Ihre virtuelle Netzwerkumgebung. Sie können Ihren eigenen IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen und Zugriffskontrolllisten konfigurieren.

Eine VPC Sicherheitsgruppe kontrolliert den Zugriff auf DB-Instances innerhalb einerVPC. Jede VPC Sicherheitsgruppenregel ermöglicht einer bestimmten Quelle den Zugriff auf eine DB-Instance in einerVPC, die dieser VPC Sicherheitsgruppe zugeordnet ist. Die Quelle kann ein Adressbereich (z. B. 203.0.113.0/24) oder eine andere Sicherheitsgruppe sein. VPC Wenn Sie eine VPC Sicherheitsgruppe als Quelle angeben, lassen Sie eingehenden Datenverkehr von allen Instanzen (normalerweise Anwendungsservern) zu, die die Quellsicherheitsgruppe verwenden. VPC

Bevor Sie versuchen, eine Verbindung zu Ihrer DB-Instance herzustellen, konfigurieren Sie Ihre VPC für Ihren Anwendungsfall. Im Folgenden finden Sie gängige Szenarien für den Zugriff auf eine DB-Instance in einemVPC:

- Eine DB-Instance in einer, auf die von einer EC2 Amazon-Instance in derselben VPC zugegriffen wird VPC — Eine DB-Instance in einer VPC wird häufig verwendet, um Daten mit einem Anwendungsserver zu teilen, der in einer EC2 Instance in derselben ausgeführt wirdVPC. Die EC2 Instance kann einen Webserver mit einer Anwendung ausführen, die mit der DB-Instance interagiert.
- Eine DB-Instance in einer, auf die von einer EC2 Instance in einer anderen VPC zugegriffen wird VPC — In einigen Fällen befindet sich Ihre DB-Instance in einer anderen VPC EC2 Instance als der Instance, mit der Sie darauf zugreifen. Wenn ja, können Sie VPC Peering verwenden, um auf die DB-Instance zuzugreifen.
- Eine DB-Instance in einer Anwendung, auf die ein Client über das Internet VPC zugreift — Um über das Internet auf eine VPC DB-Instance in einer Client-Anwendung zuzugreifen, konfigurieren Sie eine VPC mit einem einzigen öffentlichen Subnetz. Außerdem konfigurieren Sie ein Internet-Gateway für die Kommunikation über das Internet.

Um von außerhalb eine Verbindung zu einer DB-Instance herzustellenVPC, muss die DB-Instance öffentlich zugänglich sein. Außerdem muss der Zugriff unter Verwendung der eingehenden Regeln der Sicherheitsgruppe der DB-Instance gewährt werden, und andere Anforderungen müssen erfüllt sein. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

- Eine DB-Instance in einem privaten Netzwerk, auf das über ein privates Netzwerk VPC zugegriffen wird — Wenn Ihre DB-Instance nicht öffentlich zugänglich ist, können Sie eine der folgenden Optionen verwenden, um von einem privaten Netzwerk aus darauf zuzugreifen:
  - Eine AWS Site-to-Site VPN Verbindung
  - Eine AWS Direct Connect Verbindung
  - Eine AWS Client VPN Verbindung

Weitere Informationen finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.

Mit den AWS Treibern eine Verbindung zu DB-Instances herstellen

Die AWS Treibersuite wurde so konzipiert, dass sie schnellere Switchover- und Failover-Zeiten sowie Authentifizierung mit AWS Secrets Manager, AWS Identity and Access Management (IAM) und Federated Identity unterstützt. Die AWS Treiber sind darauf angewiesen, den Status der DB-Instance zu überwachen und die Instance-Topologie zu kennen, um die neue primäre Instance zu ermitteln. Dieser Ansatz reduziert die Switchover- und Failover-Zeiten auf einstellige Sekunden, verglichen mit mehreren zehn Sekunden bei Open-Source-Treibern.

In der folgenden Tabelle sind die Funktionen aufgeführt, die für die einzelnen Treiber unterstützt werden. Im Zuge der Einführung neuer Servicefunktionen besteht das Ziel der AWS Treibersuite darin, eine integrierte Unterstützung für diese Servicefunktionen zu bieten.

Funktion	AWS JDBCTreiber	AWS Python-Treiber	AWS ODBCTreiber für meinen SQL
Failover-Unterstützung	Ja	Ja	Ja
Verbesserte Failover-Überwachung	Ja	Ja	Ja

Funktion	AWS JDBCTreiber	AWS Python-Treiber	AWS ODBCTreiber für meinen SQL
Aufteilung von Lese-/Schreibvorgängen	Ja	Ja	Nein
Verbindung mit Treibermetadaten	Ja	–	N/A
Telemetrie	Ja	Ja	Nein
Secrets Manager	Ja	Ja	Ja
IAMAuthentifizierung	Ja	Ja	Ja
Föderierte Identität (AD FS)	Ja	Ja	Nein
Föderierte Identität (Okta)	Ja	Nein	Nein
Multi-AZ-DB-Cluster	Ja	Ja	Nein

Weitere Informationen zu den AWS Treibern finden Sie im entsprechenden Sprachtreiber für Ihre RDSfür MariaDB, RDSfür My SQL oder RDS für Postgre SQL DB-Instance.

> **Note**
>
> Die einzigen Funktionen, die RDS für MariaDB unterstützt werden, sind Authentifizierung mit AWS Secrets Manager, AWS Identity and Access Management (IAM) und Federated Identity.

Verbindung zu einer DB-Instance herstellen, auf der eine bestimmte DB-Engine ausgeführt wird

Um zu erfahren, wie Sie eine Verbindung zu einer DB-Instance herstellen, auf der eine bestimmte DB-Engine ausgeführt wird, folgen Sie den Anweisungen für Ihre DB-Engine:

- RDSfür Db2

- RDSfür MariaDB
- RDSfür Server SQL
- RDSfür meinen SQL
- RDSfür Oracle
- RDSfür Postgre SQL

Verbindungen mit RDS Proxy verwalten

Sie können Amazon RDS Proxy auch verwenden, um Verbindungen zu RDS MariaDB-, Microsoft SQL Server-SQL, RDS My- und RDS Postgre-DB-Instances zu verwalten. RDS SQL RDSProxy ermöglicht es Anwendungen, Datenbankverbindungen zu bündeln und gemeinsam zu nutzen, um die Skalierbarkeit zu verbessern. Weitere Informationen finden Sie unter Amazon RDS-Proxy .

Datenbankauthentifizierungsoptionen

Amazon RDS unterstützt die folgenden Methoden zur Authentifizierung von Datenbankbenutzern:

- Passwortauthentifizierung – Ihre DB-Instance führt die gesamte Verwaltung von Benutzerkonten durch. Sie erstellen Benutzer und geben Passwörter mit SQL Anweisungen an. Die SQL Anweisungen, die Sie verwenden können, hängen von Ihrer DB-Engine ab.
- AWS Identity and Access Management (IAM) Datenbankauthentifizierung — Sie müssen kein Passwort verwenden, wenn Sie eine Verbindung zu einer DB-Instance herstellen. Stattdessen verwenden Sie ein Authentifizierungstoken.
- Kerberos-Authentifizierung – Sie verwenden die externe Authentifizierung von Datenbankbenutzern, die Kerberos und Microsoft Active Directory verwenden. Kerberos ist ein Netzwerk-Authentifizierungsprotokoll, das Tickets und symmetrische Schlüsselkryptographie verwendet, um die Notwendigkeit der Übertragung von Passwörtern über das Netzwerk zu vermeiden. Kerberos wurde in Active Directory integriert und wurde entwickelt, um Benutzer gegenüber Netzwerkressourcen wie Datenbanken zu authentifizieren.

IAMDatenbankauthentifizierung und Kerberos-Authentifizierung sind nur für bestimmte DB-Engines und Versionen verfügbar.

Weitere Informationen finden Sie unter Datenbankauthentifizierung mit Amazon RDS.

Verschlüsselte Verbindungen

Sie können Secure Socket Layer (SSL) oder Transport Layer Security (TLS) in Ihrer Anwendung verwenden, um eine Verbindung zu einer DB-Instance zu verschlüsseln. Jede DB-Engine hat ihren eigenen Prozess zur Implementierung vonSSL/TLS. Weitere Informationen finden Sie unter .

Arbeiten mit Optionsgruppen

Einige DB-Engines bieten zusätzliche Funktionen, welche die Verwaltung von Daten und Datenbanken erleichtern und zusätzliche Sicherheit für Ihre Datenbank bieten. Amazon RDS verwendet Optionsgruppen, um diese Funktionen zu aktivieren und zu konfigurieren. Eine Optionsgruppe kann Funktionen, sogenannte Optionen, angeben, die für eine bestimmte RDS Amazon-DB-Instance verfügbar sind. Optionen können Einstellungen enthalten, die angeben, wie die Option funktioniert. Wenn Sie eine DB-Instanz einer Optionsgruppe zuordnen, werden die angegebenen Optionen und Optionseinstellungen für diese DB-Instanz aktiviert.

Amazon RDS unterstützt Optionen für die folgenden Datenbank-Engines:

Datenbank-Engine	Relevante Dokumentation
`Db2`	Optionen RDS für Db2-DB-Instances
`MariaDB`	Optionen für MariaDB-Datenbank-Engine
`Microsoft SQL Server`	Optionen für die Microsoft SQL Server-Datenbank-Engine
`MySQL`	Optionen für Meine SQL DB-Instances
`Oracle`	Hinzufügen von Optionen zu Oracle DB-Instances
`PostgreSQL`	Postgre verwendet SQL keine Optionen und Optionsgruppen. Postgre SQL verwendet Erweiterungen und Module, um zusätzliche Funktionen bereitzustellen. Weitere Informationen finden Sie unter Unterstützte SQL Postgre-Erweiterungsversionen.

Übersicht über die Optionsgruppen

Amazon RDS stellt für jede neue DB-Instance eine leere Standardoptionsgruppe bereit. Sie können diese Standardoptionsgruppe nicht ändern oder löschen, aber jede neue von Ihnen erstellte Optionsgruppe leitet ihre Einstellungen von der Standardoptionsgruppe ab. Um eine Option auf eine DB-Instanz anzuwenden, müssen Sie Folgendes tun:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie der Optionsgruppe eine oder mehrere Optionen hinzu.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

   Um einer DB-Instance eine Optionsgruppe zuzuordnen, ändern Sie die DB-Instance. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sowohl DB-Instanzen als auch DB-Snapshots können einer Optionsgruppe zugeordnet werden. In einigen Fällen können Sie Daten aus einem DB-Snapshot wiederherstellen oder eine point-in-time Wiederherstellung für eine DB-Instance durchführen. In diesen Fällen ist die mit dem DB-Snapshot oder der DB-Instance verknüpfte Optionsgruppe standardmäßig mit der wiederhergestellten DB-Instance verknüpft. Sie können einer wiederhergestellten DB-Instance eine andere Optionsgruppe zuordnen. Die neue Optionsgruppe muss jedoch alle persistenten oder permanenten Optionen enthalten, die in der ursprünglichen Optionsgruppe enthalten waren. Persistente und permanente Optionen werden nachfolgend beschrieben.

Einige Optionen erfordern zusätzlichen Arbeitsspeicher, um auf einer DB-Instance ausgeführt werden zu können. Somit müssen Sie möglicherweise eine größere Instance für deren Verwendung ausführen. Dies hängt von der aktuellen Nutzung Ihrer DB-Instance ab. Beispielsweise verwendet Oracle Enterprise Manager Database Control etwa 300 MB vonRAM. Wenn Sie diese Option für eine kleine DB-Instance aktivieren, können Leistungsprobleme oder out-of-memory Fehler auftreten.

Persistente und permanente Optionen

Zwei Arten von Optionen, persistente und permanente, müssen besonders berücksichtigt werden, wenn Sie sie zu einer Optionsgruppe hinzufügen.

Persistente Optionen können nicht aus einer Optionsgruppe entfernt werden, solange DB-Instances mit der Optionsgruppe verknüpft sind. Ein Beispiel für eine persistente Option ist die TDE Option für transparente Datenverschlüsselung von Microsoft SQL Server (TDE). Sie müssen alle DB-Instances von der Optionsgruppe trennen, bevor eine persistente Option aus der Optionsgruppe entfernt werden kann. In einigen Fällen können Sie einen DB-Snapshot point-in-time wiederherstellen oder eine Wiederherstellung durchführen. Wenn in solchen Fällen die Optionsgruppe, die mit dem DB-Snapshot verknüpft ist, eine persistente Option enthält, können Sie nur die wiederhergestellte DB-Instance mit dieser Optionsgruppe verknüpfen.

Dauerhafte Optionen, wie die TDE Option für Oracle Advanced SecurityTDE, können niemals aus einer Optionsgruppe entfernt werden. Sie können die Optionsgruppe einer DB-Instance ändern, welche die permanente Option verwendet. Allerdings muss die mit der DB-Instance verknüpfte Optionsgruppe die gleiche permanente Option enthalten. In einigen Fällen können Sie einen DB-Snapshot point-in-time wiederherstellen oder eine Wiederherstellung durchführen. In diesen Fällen können Sie, wenn die mit diesem DB-Snapshot verknüpfte Optionsgruppe eine permanente Option enthält, nur die wiederhergestellte DB-Instance mit einer Optionsgruppe verknüpfen, die diese permanente Option enthält.

Bei Oracle DB-Instances können Sie gemeinsam genutzte DB-Snapshots mit den Optionen `Timezone` oder `OLS` (oder beiden) kopieren. Geben Sie dazu eine Zieloptionengruppe an, die diese Optionen enthält, wenn Sie den DB-Snapshot kopieren. Die OLS Option ist permanent und persistent nur für Oracle-DB-Instances, auf denen Oracle Version 12.2 oder höher ausgeführt wird. Weitere Informationen zu diesen Optionen finden Sie unter Oracle-Zeitzone und Oracle Label Security.

VPCÜberlegungen

Die der DB-Instance zugeordnete Optionsgruppe ist mit der der DB-Instance verknüpftVPC. Das bedeutet, dass Sie die einer DB-Instance zugewiesene Optionsgruppe nicht verwenden können, wenn Sie versuchen, die Instance auf einer anderen wiederherzustellenVPC. Wenn Sie eine DB-Instance auf einer anderen wiederherstellenVPC, können Sie einen der folgenden Schritte ausführen:

- Zuweisen einer Standard-Optionsgruppe zu einer DB-Instance.
- Weisen Sie eine Optionsgruppe zu, die damit verknüpft istVPC.
- Erstellen einer neuen Optionsgruppe und Zuweisen von dieser zur DB-Instance

Bei persistenten oder permanenten Optionen wie Oracle TDE müssen Sie eine neue Optionsgruppe erstellen. Diese Optionsgruppe muss die persistente oder permanente Option enthalten, wenn eine DB-Instance in einer anderen wiederhergestellt wirdVPC.

Optionseinstellungen steuern das Verhalten einer Option. Die Oracle Advanced Security Option `NATIVE_NETWORK_ENCRYPTION`hat beispielsweise eine Einstellung, mit der Sie den Verschlüsselungsalgorithmus für den Netzwerkverkehr zur und von der DB-Instanz angeben können. Einige Optionseinstellungen sind für die Verwendung mit Amazon optimiert RDS und können nicht geändert werden.

Einander ausschließende Optionen

Einige Optionen schließen sich gegenseitig aus. Sie können die eine oder die andere verwenden, aber nicht beide zugleich. Die folgenden Optionen schließen sich gegenseitig aus:

- Oracle Enterprise Manager Database Express und Oracle Management Agent für Enterprise Cloud Control.
- Oracle Native Network Encryption und Oracle Secure Sockets Layer.

Erstellen einer Optionsgruppe

Sie können eine neue Optionsgruppe erstellen, die ihre Einstellungen von der Standardoptionsgruppe ableitet. Fügen Sie der neuen Optionsgruppe dann eine oder mehrere Optionen hinzu. Wenn Sie bereits über eine vorhandene Optionsgruppe verfügen, können Sie diese Optionsgruppe auch mit allen Optionen in eine neue Optionsgruppe kopieren. Weitere Informationen finden Sie unter Kopieren einer Optionsgruppe.

Wenn Sie eine neue Optionsgruppe erstellen, hat diese keine Optionen. Wie Sie Optionen zur Optionsgruppe hinzufügen, finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe. Nachdem Sie die gewünschten Optionen hinzugefügt haben, können Sie die Optionsgruppe dann einer DB-Instance zuordnen. Auf diese Weise werden die Optionen auf der DB-Instance verfügbar. Informationen über das Zuordnen einer Optionsgruppe zu einer DB-Instance finden Sie in der Dokumentation für Ihre Engine unter Arbeiten mit Optionsgruppen.

Konsole

Eine Möglichkeit zur Erstellung einer Optionsgruppe besteht in der Verwendung der AWS Management Console.

So erstellen Sie eine neue Optionsgruppe mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Fenster Create option group (Optionsgruppe erstellen) Folgendes aus:

    a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihres AWS Kontos eindeutig ist. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.

b. Bei Description (Beschreibung) geben Sie eine kurze Beschreibung der Optionsgruppe ein. Die Beschreibung ist nur zur Information.

c. Bei Engine wählen Sie den gewünschten DB-Motor aus.

d. Bei Major engine version (Engine-Hauptversion) wählen Sie die Hauptversion der gewünschten DB-Engine aus.

5. Klicken Sie zum Fortfahren auf Create (Erstellen). Wählen Sie zum Abbrechen der Operation Cancel (Abbrechen) aus.

AWS CLI

Verwenden Sie den AWS CLI `create-option-group`Befehl mit den folgenden erforderlichen Parametern, um eine Optionsgruppe zu erstellen.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

Das folgende Beispiel erstellt eine Optionsgruppe namens `testoptiongroup`, die der Oracle Enterprise Edition DB-Engine zugeordnet ist. Die Beschreibung ist in Anführungszeichen gesetzt.

Für LinuxmacOS, oderUnix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name oracle-ee \
    --major-engine-version 19 \
    --option-group-description "Test option group for Oracle Database 19c EE"
```

Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name oracle-ee ^-
    --major-engine-version 19 ^
    --option-group-description "Test option group for Oracle Database 19c EE"
```

RDS API

Um eine Optionsgruppe zu erstellen, rufen Sie den RDS API `CreateOptionGroup`Amazon-Vorgang auf. Verwenden Sie die folgenden Parameter:

- `OptionGroupName`
- `EngineName`
- `MajorEngineVersion`
- `OptionGroupDescription`

Kopieren einer Optionsgruppe

Sie können die Optionsgruppe AWS CLI oder Amazon RDS API Copy a verwenden. Das Kopieren einer Optionsgruppe kann praktisch sein. Beispielsweise wenn Sie bereits eine Optionsgruppe haben und eine Vielzahl der darin enthaltenen benutzerdefinierten Parameter und Werte in eine neue Optionsgruppe übernehmen möchten. Sie können eine Optionsgruppe auch kopieren und dann die Kopie verändern, um andere Optionseinstellungen zu testen.

> **Note**
>
> Derzeit können Sie eine Optionsgruppe nicht in eine andere AWS Region kopieren.

AWS CLI

Verwenden Sie den AWS CLI copy-option-groupBefehl, um eine Optionsgruppe zu kopieren. Verwenden Sie den folgenden erforderlichen Parameter:

- `--source-option-group-identifier`
- `--target-option-group-identifier`

- `--target-option-group-description`

Example

Das folgende Beispiel erstellt eine Optionsgruppe namens `new-option-group`, eine lokale Kopie der Optionsgruppe `my-option-group`.

Für LinuxmacOS, oderUnix:

```
aws rds copy-option-group \
    --source-option-group-identifier my-option-group \
    --target-option-group-identifier new-option-group \
    --target-option-group-description "My new option group"
```

Windows:

```
aws rds copy-option-group ^
    --source-option-group-identifier my-option-group ^
    --target-option-group-identifier new-option-group ^
    --target-option-group-description "My new option group"
```

RDS API

Um eine Optionsgruppe zu kopieren, rufen Sie den RDS API CopyOptionGroupAmazon-Vorgang auf. Verwenden Sie die folgenden erforderlichen Parameter.

- `SourceOptionGroupIdentifier`
- `TargetOptionGroupIdentifier`
- `TargetOptionGroupDescription`

Hinzufügen einer Option zu einer Optionsgruppe

Sie können einer vorhandenen Optionsgruppe eine Option hinzufügen. Nachdem Sie die gewünschten Optionen hinzugefügt haben, können Sie die Optionsgruppe dann einer DB-Instanz zuordnen, damit die Optionen in der DB-Instanz verfügbar werden. Mehr über das Zuordnen einer Optionsgruppe zu einer DB-Instanz finden Sie im jeweiligen Handbuch unter Arbeiten mit Optionsgruppen.

Änderungen in Optionsgruppen müssen in zwei Fällen sofort angewendet werden:

- Wenn Sie eine Option hinzufügen, mit der ein Port-Wert hinzugefügt oder aktualisiert wird, z. B. die `OEM` Option.
- Wenn Sie eine Optionsgruppe hinzufügen oder entfernen, die eine Option mit einem Port-Wert enthält.

Wählen Sie in einem solchen Fall die Option Apply Immediately (Sofort anwenden) in der Konsole aus. Oder Sie können die `--apply-immediately` Option einbeziehen, wenn Sie den Amazon verwenden, AWS CLI oder den `ApplyImmediately` Parameter auf setzen, `true` wenn Sie Amazon verwenden RDSAPI. Optionen, die keine Port-Werte enthalten, können sofort oder während des nächsten Wartungsfensters für die DB-Instance übernommen werden.

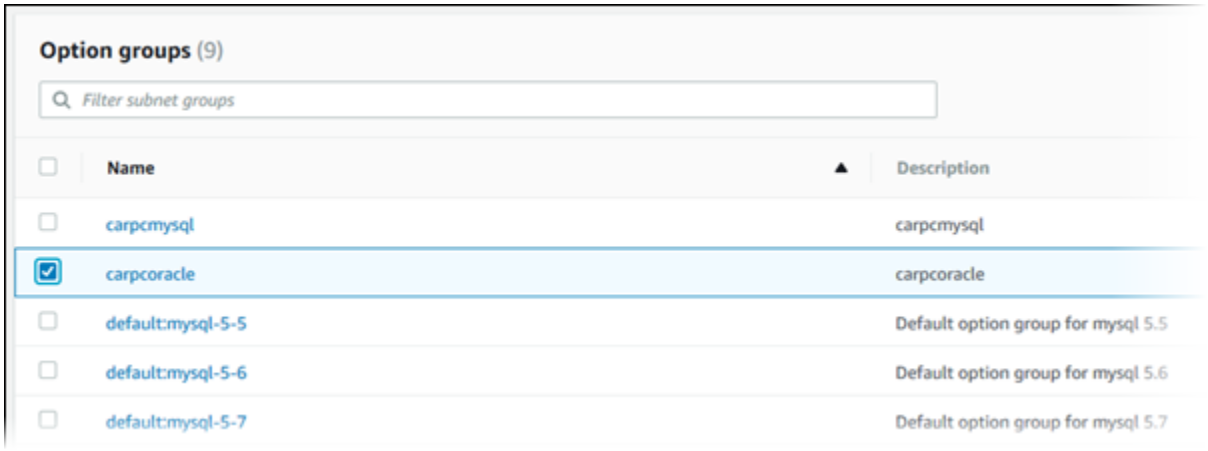
> **Note**
>
> Wenn Sie eine Sicherheitsgruppe als Wert für eine Option in einer Optionsgruppe angeben, verwalten Sie die Sicherheitsgruppe, indem Sie die Optionsgruppe ändern. Sie können diese Sicherheitsgruppe nicht ändern oder entfernen, indem Sie eine DB-Instance ändern. Außerdem erscheint die Sicherheitsgruppe nicht in den DB-Instance-Details in der AWS Management Console oder in der Ausgabe für den AWS CLI Befehl`describe-db-instances`.

Konsole

Sie können den verwenden AWS Management Console , um einer Optionsgruppe eine Option hinzuzufügen.

Um einer Optionsgruppe per Konsole eine Option hinzuzufügen

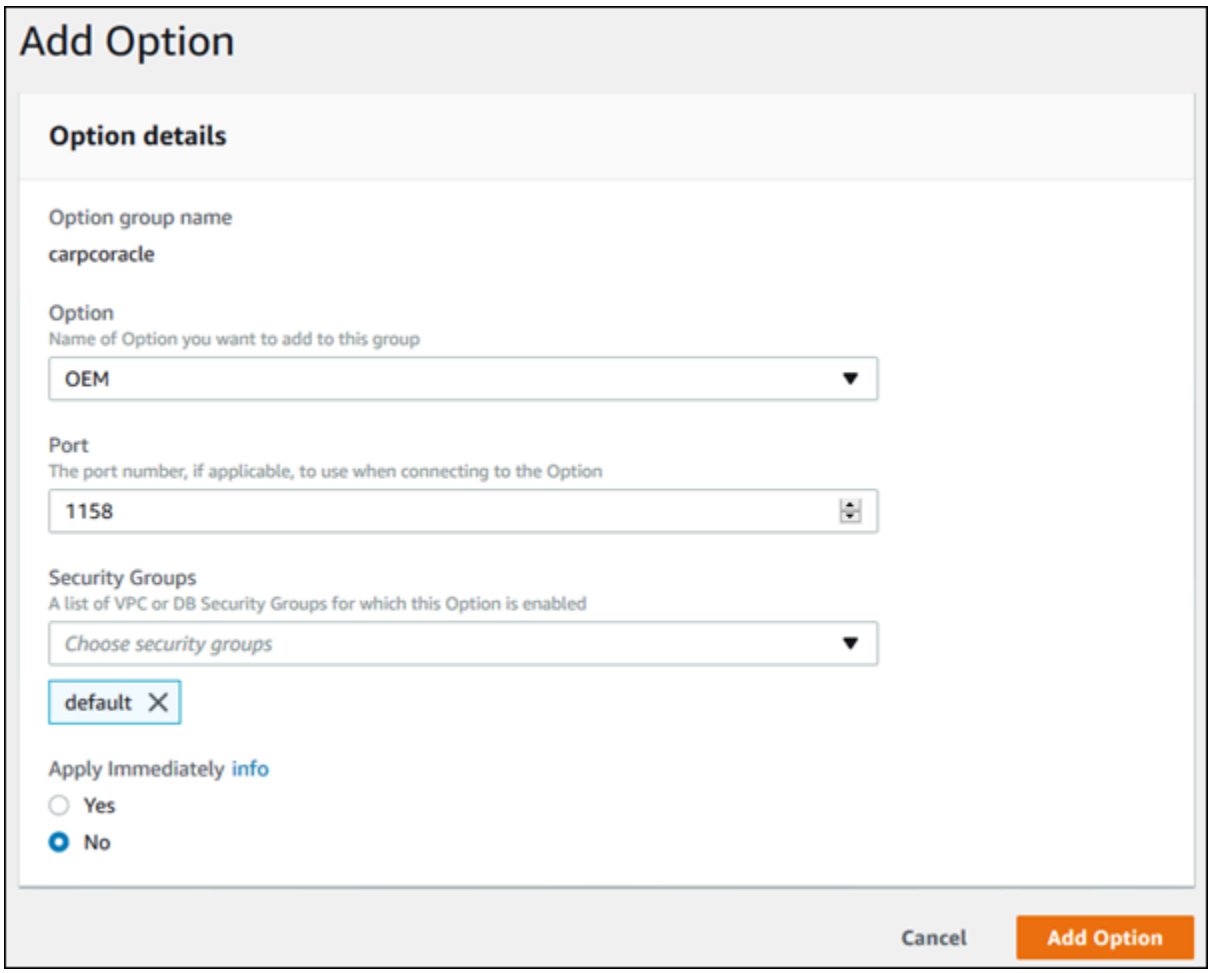
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus, die Sie ändern möchten, und wählen Sie dann Add option (Option hinzufügen).



4. Führen Sie im Fenster Add option (Option hinzufügen) die folgenden Schritte aus:

   a. Wählen Sie die Option aus, die Sie hinzufügen möchten. Abhängig von der ausgewählten Option müssen Sie möglicherweise zusätzliche Werte angeben. Wenn Sie beispielsweise die Option `OEM` auswählen, müssen Sie auch einen Port-Wert eingeben und eine Sicherheitsgruppe angeben.

   b. Um die Option in allen zugeordneten DB-Instanzen zu aktivieren, sobald Sie sie hinzufügen, wählen Sie für Apply Immediately (Direkt anwenden) Yes (Ja). Wenn Sie No (Nein) (Standard) wählen, wird die Option während des nächsten Wartungsfensters in jeder zugeordneten DB-Instanz aktiviert.



5. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Add option (Option hinzufügen) aus.

AWS CLI

Um einer Optionsgruppe eine Option hinzuzufügen, führen Sie den Befehl AWS CLI add-option-to-option-group mit der Option aus, die Sie hinzufügen möchten. Um die neue Option sofort auf allen zugeordneten DB-Instanzen zu aktivieren, verwenden Sie den `--apply-immediately`Parameter . Standardgemäß wird die Option für jede zugeordnete DB-Instanz während ihres nächsten Wartungsfensters aktiviert. Verwenden Sie den folgenden erforderlichen Parameter:

- `--option-group-name`

Example

Im folgenden Beispiel wird die `Timezone` Option zusammen mit der `America/Los_Angeles` Einstellung einer Optionsgruppe mit dem Namen hinzugefügt `testoptiongroup` und sofort aktiviert.

Für LinuxmacOS, oderUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

Die Befehlsausgabe wird ungefähr wie folgt aussehen:

```
...{
       "OptionName": "Timezone",
       "OptionDescription": "Change time zone",
       "Persistent": true,
       "Permanent": false,
       "OptionSettings": [
           {
               "Name": "TIME_ZONE",
               "Value": "America/Los_Angeles",
               "DefaultValue": "UTC",
               "Description": "Specifies the timezone the user wants to change the
 system time to",
               "ApplyType": "DYNAMIC",
               "DataType": "STRING",
               "AllowedValues": "Africa/Cairo,...",
               "IsModifiable": true,
               "IsCollection": false
           }
       ],
       "DBSecurityGroupMemberships": [],
```

```
        "VpcSecurityGroupMemberships": []
    }...
```

Example

Im folgenden Beispiel wird die OEM Oracle-Option zu einer Optionsgruppe hinzugefügt. Es spezifiziert auch einen benutzerdefinierten Port und ein Paar von EC2 VPC Amazon-Sicherheitsgruppen, die für diesen Port verwendet werden sollen.

Für LinuxmacOS, oderUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 \
    --apply-immediately
```

Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 ^
    --apply-immediately
```

Die Befehlsausgabe wird ungefähr wie folgt aussehen:

```
OPTIONGROUP  False  oracle-ee  19 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
 Test Option Group  testoptiongroup vpc-test
OPTIONS Oracle 12c EM Express   OEM     False   False   5500
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test1
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test2
```

Example

Im folgenden Beispiel wird die Oracle-Option `NATIVE_NETWORK_ENCRYPTION` zu einer Optionsgruppe hinzugefügt und die Optionseinstellungen angegeben. Wenn keine Optionseinstellungen angegeben werden, werden Standardwerte verwendet.

Für LinuxmacOS, oderUnix:

```
aws rds add-option-to-option-group \
```

```
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES"}],"OptionName":"NATIVE_NET
 \
    --apply-immediately
```

Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES"}],"OptionName"="NATIVE_N
 ^
    --apply-immediately
```

Die Befehlsausgabe wird ungefähr wie folgt aussehen:

```
...{
    "OptionName": "NATIVE_NETWORK_ENCRYPTION",
    "OptionDescription": "Native Network Encryption",
    "Persistent": false,
    "Permanent": false,
    "OptionSettings": [
        {
            "Name": "SQLNET.ENCRYPTION_TYPES_SERVER",
             "Value": "AES256,AES192,DES",
            "DefaultValue":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "Description": "Specifies list of encryption algorithms in order of
 intended use",
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "IsModifiable": true,
            "IsCollection": true
        },
        {
            "Name": "SQLNET.ENCRYPTION_SERVER",
            "Value": "REQUIRED",
            "DefaultValue": "REQUESTED",
            "Description": "Specifies the desired encryption behavior",
```

```
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues": "ACCEPTED,REJECTED,REQUESTED,REQUIRED",
            "IsModifiable": true,
            "IsCollection": false
        },...
```

RDS API

Um einer Optionsgruppe mithilfe von Amazon eine Option hinzuzufügen RDSAPI, rufen Sie den ModifyOptionGroupVorgang mit der Option auf, die Sie hinzufügen möchten. Um die neue Option sofort auf allen zugeordneten DB-Instanzen zu aktivieren, verwenden Sie den Parameter `ApplyImmediately` und setzen Sie ihn auf `true`. Standardgemäß wird die Option für jede zugeordnete DB-Instanz während ihres nächsten Wartungsfensters aktiviert. Verwenden Sie den folgenden erforderlichen Parameter:

- `OptionGroupName`

Auflisten der Optionen und Optionseinstellungen für eine Optionsgruppe

Sie können alle Optionen und Optionseinstellungen für eine Optionsgruppe auflisten.

Konsole

Sie können den verwenden AWS Management Console , um alle Optionen und Optionseinstellungen für eine Optionsgruppe aufzulisten.

die Optionen und Optionseinstellungen für eine Optionsgruppe auflisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie den Namen einer Optionengruppe aus, um deren Details anzuzeigen. Die Optionen und Optionseinstellungen für eine Optionengruppe werden aufgelistet.

AWS CLI

Verwenden Sie den AWS CLI `describe-option-groups`Befehl, um die Optionen und Optionseinstellungen für eine Optionsgruppe aufzulisten. Geben Sie den Namen der

Optionsgruppe an, deren Optionen und Einstellungen Sie anzeigen möchten. Wenn Sie keinen Optionsgruppennamen angeben, werden alle Optionsgruppen beschrieben.

Example

Das folgende Beispiel listet die Optionen und Optionseinstellungen für alle Optionsgruppen auf.

```
aws rds describe-option-groups
```

Example

Im folgenden Beispiel werden die Optionen und Optionseinstellungen für eine Optionsgruppe namens aufgeliste `testoptiongroup`.

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

RDS API

Verwenden Sie den RDS API `DescribeOptionGroups`Amazon-Vorgang, um die Optionen und Optionseinstellungen für eine Optionsgruppe aufzulisten. Geben Sie den Namen der Optionsgruppe an, deren Optionen und Einstellungen Sie anzeigen möchten. Wenn Sie keinen Optionsgruppennamen angeben, werden alle Optionsgruppen beschrieben.

Ändern einer Optionseinstellung

Nachdem Sie eine Option hinzugefügt haben, die über veränderbare Optionseinstellungen verfügt, können Sie die Einstellungen jederzeit ändern. Wenn Sie Optionen oder Optionseinstellungen in einer Optionsgruppe ändern, werden diese Änderungen auf alle DB-Instanzen angewendet, die dieser Optionsgruppe zugeordnet sind. Weitere Informationen zu den verfügbaren Einstellungen für die verschiedenen Optionen finden Sie in der Dokumentation für Ihre Engine unter Arbeiten mit Optionsgruppen.

Änderungen in Optionsgruppen müssen in zwei Fällen sofort angewendet werden:

- Wenn Sie eine Option hinzufügen, mit der ein Port-Wert hinzugefügt oder aktualisiert wird, z. B. die `OEM` Option.
- Wenn Sie eine Optionsgruppe hinzufügen oder entfernen, die eine Option mit einem Port-Wert enthält.

Wählen Sie in einem solchen Fall die Option Apply Immediately (Sofort anwenden) in der Konsole aus. Sie können die `--apply-immediately` Option auch einbeziehen, wenn Sie den verwenden, AWS CLI oder den `ApplyImmediately` Parameter auf setzen, `true` wenn Sie den verwenden RDSAPI. Optionen, die keine Port-Werte enthalten, können sofort oder während des nächsten Wartungsfensters für die DB-Instance übernommen werden.

> **Note**
>
> Wenn Sie eine Sicherheitsgruppe als Wert für eine Option in einer Optionsgruppe angeben, verwalten Sie die Sicherheitsgruppe, indem Sie die Optionsgruppe ändern. Sie können diese Sicherheitsgruppe nicht ändern oder entfernen, indem Sie eine DB-Instance ändern. Außerdem erscheint die Sicherheitsgruppe nicht in den DB-Instance-Details in der AWS Management Console oder in der Ausgabe des AWS CLI Befehls`describe-db-instances`.

Konsole

Sie können die verwenden AWS Management Console , um eine Optionseinstellung zu ändern.

eine Optionseinstellung mithilfe der Konsole ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus, deren Option Sie ändern möchten, und klicken Sie auf Modify option (Option ändern).
4. Wählen Sie im Fenster Modify option (Option ändern) unter Installed Options (Installierte Optionen) die Option aus, deren Einstellung Sie ändern möchten. Nehmen Sie die Änderungen vor, die Sie wollen.
5. Um die Option zu aktivieren, sobald Sie sie hinzufügen, wählen Sie für Apply Immediately (Direkt anwenden) Yes (Ja). Wenn Sie No (Nein) (Standard) wählen, wird die Option während des nächsten Wartungsfensters in jeder zugeordneten DB-Instanz aktiviert.
6. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Modify Option (Option ändern) aus.

AWS CLI

Um eine Optionseinstellung zu ändern, verwenden Sie den AWS CLI `add-option-to-option-group`Befehl mit der Optionsgruppe und Option, die Sie ändern möchten. Standardgemäß wird die Option für jede zugeordnete DB-Instanz während ihres nächsten Wartungsfensters aktiviert. Um die Änderung sofort auf alle zugeordneten DB-Instanzen anzuwenden, schließen Sie den `--apply-immediately`-Parameter. Um eine Optionseinstellung zu ändern, verwenden Sie das `--settings`-Argument.

Example

Das folgende Beispiel ändert den Port, den Oracle Enterprise Manager Database Control (OEM) in einer Optionsgruppe mit dem Namen verwendet, `testoptiongroup` und wendet die Änderung sofort an.

Für LinuxmacOS, oderUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \
    --apply-immediately
```

Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^
    --apply-immediately
```

Die Befehlsausgabe wird ungefähr wie folgt aussehen:

```
OPTIONGROUP   False  oracle-ee  19  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
   Test Option Group    testoptiongroup
OPTIONS Oracle 12c EM Express   OEM     False   False   5432
DBSECURITYGROUPMEMBERSHIPS   default  authorized
```

Example

Im folgenden Beispiel werden die Oracle-Option `NATIVE_NETWORK_ENCRYPTION` und die Optionseinstellungen geändert.

Für LinuxmacOS, oderUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES,RC4_256"}],"OptionName":"NA
 \
    --apply-immediately
```

Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES
\,RC4_256"}],"OptionName"="NATIVE_NETWORK_ENCRYPTION" ^
    --apply-immediately
```

Die Befehlsausgabe wird ungefähr wie folgt aussehen:

```
OPTIONGROUP   False  oracle-ee  19  arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
   Test Option Group    testoptiongroup
OPTIONS Oracle Advanced Security - Native Network Encryption
 NATIVE_NETWORK_ENCRYPTION       False   False
OPTIONSETTINGS
 RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40 STATIC
 STRING
   RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
 Specifies list of encryption algorithms in order of intended use
   True     True    SQLNET.ENCRYPTION_TYPES_SERVER  AES256,AES192,DES,RC4_256
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED    STATIC  STRING  REQUESTED
 Specifies the desired encryption behavior  False   True  SQLNET.ENCRYPTION_SERVER
 REQUIRED
OPTIONSETTINGS  SHA1,MD5   STATIC  STRING  SHA1,MD5    Specifies list of
 checksumming algorithms in order of intended use      True    True
 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER  SHA1,MD5
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED  STATIC  STRING
 REQUESTED     Specifies the desired data integrity behavior   False   True
 SQLNET.CRYPTO_CHECKSUM_SERVER  REQUESTED
```

RDS API

Um eine Optionseinstellung zu ändern, verwenden Sie den RDS API `ModifyOptionGroup`Amazon-Befehl mit der Optionsgruppe und Option, die Sie ändern möchten. Standardgemäß wird die Option für jede zugeordnete DB-Instanz während ihres nächsten Wartungsfensters aktiviert. Um die Änderung sofort auf alle zugeordneten DB-Instanzen anzuwenden, inkludieren Sie den `ApplyImmediately`-Parameter und setzen ihn zu `true`.

Entfernen einer Option aus einer Optionsgruppe

Einige Optionen können aus einer Optionsgruppe entfernt werden, andere nicht. Eine dauerhafte Option kann nicht aus einer Optionsgruppe entfernt werden, bis alle DB-Instanzen, die dieser Optionsgruppe zugeordnet sind, getrennt werden. Eine permanente Option kann nicht aus einer Optionsgruppe entfernt werden. Weitere Informationen zu den Optionen, die entfernt werden können, finden Sie in der Dokumentation zu Ihrer Information bei Arbeiten mit Optionsgruppen.

Wenn Sie alle Optionen aus einer Optionsgruppe entfernen, RDS löscht Amazon die Optionsgruppe nicht. DB-Instances, die der leeren Optionsgruppe zugeordnet sind, werden weiterhin zugeordnet. Sie haben einfach keine aktiven Optionen. Zum Entfernen aller Optionen aus einer DB-Instance können Sie alternativ die DB-Instance der Standardoptionsgruppe (leer) zuordnen.

Konsole

Sie können den verwenden AWS Management Console , um eine Option aus einer Optionsgruppe zu entfernen.

entfernen einer Option aus einer Optionsgruppe mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus, deren Option Sie entfernen möchten, und klicken Sie auf Delete option (Option löschen).
4. Führen Sie im Fenster Delete option (Option löschen) die folgenden Schritte aus:

   - Aktivieren Sie das Kontrollkästchen für die Option, die Sie löschen möchten.
   - Damit die Option sofort gelöscht wird, wählen Sie für Apply immediately (Sofort anwenden) Yes (Ja) aus. Wenn Sie No (Nein) (Standard) wählen, ist die Option für jede zugeordnete DB-Instanz während ihres nächsten Wartungsfensters aktiviert.



5. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Yes, Delete (Ja, löschen) aus.

AWS CLI

Um eine Option aus einer Optionsgruppe zu entfernen, verwenden Sie den AWS CLI `remove-option-from-option-group`Befehl mit der Option, die Sie löschen möchten. Standardgemäß wird die Option während des nächsten Wartungsfensters aus jeder zugeordneten DB-Instanz entfernt. Um die Änderung sofort zu übernehmen, fügen Sie den `--apply-immediately`-Parameter hinzu.

Example

Das folgende Beispiel entfernt die Option Oracle Enterprise Manager Database Control (OEM) aus einer Optionsgruppe mit dem Namen `testoptiongroup` und wendet die Änderung sofort an.

Für LinuxmacOS, oderUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name testoptiongroup \
    --options OEM \
    --apply-immediately
```

Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name testoptiongroup ^
    --options OEM ^
    --apply-immediately
```

Die Befehlsausgabe wird ungefähr wie folgt aussehen:

```
OPTIONGROUP    testoptiongroup oracle-ee   19    Test option group
```

RDS API

Verwenden Sie die RDS API `ModifyOptionGroup`Amazon-Aktion, um eine Option aus einer Optionsgruppe zu entfernen. Standardgemäß wird die Option während des nächsten Wartungsfensters aus jeder zugeordneten DB-Instanz entfernt. Um die Änderung sofort zu übernehmen, fügen Sie den `ApplyImmediately`-Parameter ein und setzen Sie es auf `true`.

Verwenden Sie die folgenden Parameter:

- `OptionGroupName`
- `OptionsToRemove.OptionName`

Löschen einer Optionsgruppe

Sie können eine Optionsgruppe nur löschen, wenn sie die folgenden Kriterien erfüllt:

- Es ist mit keiner RDS Amazon-Ressource verknüpft. Eine Optionsgruppe kann mit einer DB-Instance, einem manuellen DB-Snapshot oder einem automatisierten DB-Snapshot verknüpft werden.
- Es handelt sich nicht um eine Standardoptionsgruppe.

Um die Optionsgruppen zu identifizieren, die von Ihren DB-Instances und DB-Snapshots verwendet werden, können Sie die folgenden CLI Befehle verwenden:

```
aws rds describe-db-instances \
    --query 'DBInstances[*].
[DBInstanceIdentifier,OptionGroupMemberships[].OptionGroupName]'
```

```
aws rds describe-db-snapshots | jq -r '.DBSnapshots[] | "\(.DBInstanceIdentifier),
\(.OptionGroupName)"' | sort | uniq
```

Wenn Sie versuchen, eine Optionsgruppe zu löschen, die einer RDS Ressource zugeordnet ist, wird ein Fehler wie der folgende zurückgegeben.

```
An error occurred (InvalidOptionGroupStateFault) when calling the DeleteOptionGroup
 operation: The option group 'optionGroupName' cannot be deleted because it is in use.
```

So finden Sie die RDS Amazon-Ressourcen, die einer Optionsgruppe zugeordnet sind

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie den Namen einer Optionsgruppe aus, um deren Details anzuzeigen.
4. Im Abschnitt Associated Instances and Snapshots finden Sie die zugehörigen RDS Amazon-Ressourcen.

Ist eine DB-Instance mit der Optionsgruppe verknüpft, ändern Sie die DB-Instance, um eine andere Optionsgruppe zu verwenden. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Ist ein manueller DB-Snapshot mit der Optionsgruppe verknüpft, ändern Sie den DB-Snapshot so, dass eine andere Optionsgruppe verwendet wird. Sie können dies mit dem AWS CLI `modify-db-snapshot`Befehl tun.

> **Note**
>
> Sie können die Optionsgruppe eines automatisierten DB-Snapshots nicht ändern.

Konsole

Eine Möglichkeit zum Löschen einer Optionsgruppe besteht in der Verwendung der AWS Management Console.

So löschen Sie eine Optionsgruppe mit der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus.
4. Wählen Sie Delete group (Gruppe löschen) aus.
5. Wählen Sie auf der Bestätigungsseite Delete (Löschen) aus, um das Löschen der Optionsgruppe abzuschließen, oder wählen Sie Cancel (Abbrechen) aus, um den Löschvorgang abzubrechen.

AWS CLI

Um eine Optionsgruppe zu löschen, verwenden Sie den AWS CLI `delete-option-group`Befehl mit dem folgenden erforderlichen Parameter.

- `--option-group-name`

Example

Im folgenden Beispiel wird eine Optionsgruppe namens gelösch `testoptiongroup`.

Für LinuxmacOS, oderUnix:

```
aws rds delete-option-group \
    --option-group-name testoptiongroup
```

Windows:

```
aws rds delete-option-group ^
    --option-group-name testoptiongroup
```

RDS API

Um eine Optionsgruppe zu löschen, rufen Sie den RDS API `DeleteOptionGroup`Amazon-Vorgang auf. Schließen Sie den folgenden Parameter ein:

- `OptionGroupName`

Parametergruppen für Amazon RDS

Database parameters (Datenbankparameter) – geben Sie an, wie die Datenbank konfiguriert wird. Datenbankparameter können z. B. die Menge der Ressourcen angeben, die einer Datenbank zugewiesen werden sollen, wie etwa den Speicher.

Sie verwalten Ihre Datenbankkonfiguration, indem Sie Ihre DB-Instances und Multi-AZ-DB-Cluster mit Parametergruppen zuordnen. Amazon RDS definiert Parametergruppen mit Standardeinstellungen. Sie können auch eigene Parametergruppen mit angepassten Einstellungen definieren.

> **Note**
>
> Einige DB-Engines bieten zusätzliche Funktionen, um die Sie Ihre Datenbank optional in einer Optionsgruppe ergänzen können. Informationen über Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Themen

- Übersicht über Parametergruppen
- DB-Parametergruppen für Amazon RDS Amazon
- Arbeiten mit DB-Cluster-Parametergruppen für Multi-AZ-DB-Cluster
- Vergleichen von DB-Parametergruppen
- Festlegen von DB-Parametern

Übersicht über Parametergruppen

Eine DB-Parametergruppe dient als Container für Engine-Konfigurationswerte, die auf eine oder mehrere DB-Instances angewendet werden.

DB-Cluster-Parametergruppen gelten nur für Multi-AZ-DB-Cluster. In einem Multi-AZ-DB-Cluster werden die Einstellungen in der Parametergruppe des DB-Clusters auf alle DB-Instances im Cluster angewendet. Die Standard-DB-Parametergruppe für die DB-Engine und die DB-Engine-Version wird für jede DB-Instance im DB-Cluster verwendet.

Themen

- Standard- und benutzerdefinierte Parametergruppen
- Statische und dynamische DB-Instance-Parameter

- Statische und dynamische DB-Cluster-Parameter
- Zeichensatzparameter
- Unterstützte Parameter und Parameterwerte

Standard- und benutzerdefinierte Parametergruppen

Wenn Sie eine DB-Instance ohne Angabe einer DB-Parametergruppe erstellen, verwendet die DB-Instance eine Standard-DB-Parametergruppe. Beim Erstellen eines Multi-AZ-DB-Clusters ohne Angabe einer DB-Cluster-Parametergruppe verwendet der DB-Cluster ebenso eine Standard-DB-Cluster-Parametergruppe. Jede Standard-Parametergruppe enthält Standardeinstellungen für die Datenbank-Engine und das Amazon RDS-System, die auf der Engine, der Datenverarbeitungsklasse und dem zugeordneten Speicher der Instance basieren.

Sie können die Parametereinstellungen für eine Standard-Parametergruppe nicht ändern. Stattdessen können Sie Folgendes tun:

1. Neue Parametergruppe erstellen.
2. Ändern Sie die Einstellungen Ihrer gewünschten Parameter. In einer Parametergruppe können nicht alle DB-Engine-Parameter geändert werden.
3. Ändern Sie Ihre DB-Instance oder Ihren DB-Cluster, um die neue Parametergruppe zuzuordnen.

   Wenn Sie einer DB-Instance eine neue DB-Parametergruppe zuordnen, erfolgt die Zuordnung sofort. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance. Informationen zum Ändern eines Multi-AZ-DB-Clusters finden Sie unter Ändern eines Multi-AZ-DB-Clusters für Amazon RDS.

   > **Note**
   >
   > Wenn Sie Ihre DB-Instance so geändert haben, dass sie eine benutzerdefinierte Parametergruppe verwendet, und Sie die DB-Instance starten, startet RDS die DB-Instance im Rahmen des Startvorgangs automatisch neu. Bei Multi-AZ-Instances von RDS für SQL Server mit AlwaysOn aktivierter Spiegelungsoption wird ein Failover erwartet, wenn die Instanz nach dem Startvorgang neu gestartet wird.

RDS wendet die geänderten statischen und dynamischen Parameter in einer neu verknüpften Parametergruppe erst an, nachdem die DB-Instance neu gestartet wurde. Wenn Sie jedoch

dynamische Parameter in der DB-Parametergruppe ändern, nachdem Sie sie der DB-Instance zugeordnet haben, werden diese Änderungen sofort ohne Neustart angewendet. Weitere Informationen zum Ändern der DB-Parametergruppe finden Sie unter Ändern einer Amazon RDS DB-Instance.

Wenn Sie Parameter innerhalb einer DB-Parametergruppe aktualisieren, gelten die Änderungen für alle DB-Instances, die dieser Parametergruppe zugeordnet sind. Wenn Sie Parameter innerhalb einer Parametergruppe eines Multi-AZ-DB-Clusters aktualisieren, gelten die Änderungen ebenso für alle Aurora-DB-Cluster, die dieser DB-Cluster-Parametergruppe zugeordnet sind.

Wenn Sie eine Parametergruppe nicht von Grund auf neu erstellen möchten, können Sie eine vorhandene Parametergruppe mit dem AWS CLI copy-db-parameter-groupBefehl oder dem Befehl copy-db-cluster-parameter-group kopieren. Das Kopieren einer Parametergruppe kann sich in einigen Fällen als nützlich erweisen. Wenn Sie beispielsweise die am häufigsten verwendeten benutzerdefinierten Parameter und Werte einer vorhandenen DB-Parametergruppe in eine neue DB-Parametergruppe aufnehmen möchten.

Statische und dynamische DB-Instance-Parameter

Die DB-Instance-Parameter sind entweder statisch oder dynamisch. Sie weisen folgende Unterschiede auf:

- Wenn Sie einen statischen Parameter ändern und eine DB-Parametergruppe speichern, wird die Änderung des Parameters nach einem manuellen Neustart der zugeordneten DB-Instances angewendet. Bei statischen Parametern verwendet die Konsole immer `pending-reboot` als `ApplyMethod`.
- Wenn Sie einen dynamischen Parameter ändern, wird die Parameteränderung standardmäßig sofort wirksam, ohne dass ein Neustart erforderlich ist. Wenn Sie die AWS Management Console zum Ändern von DB-Instance-Parameterwerten verwenden, werden `immediate` für die dynamischen Parameter immer die `ApplyMethod` für die verwendet. Verwenden Sie die AWS CLI oder RDS-API, um die Änderung des Parameters zu verschieben, bis Sie eine zugeordnete DB-Instance neu gestartet haben. Legen Sie die `ApplyMethod` für die Parameteränderung auf `pending-reboot` fest.

> **Note**
>
> Bei der Verwendung `pending-reboot` mit dynamischen Parametern in der AWS CLI oder der RDS-API auf RDS für SQL Server-DB-Instances wird ein Fehler generiert. Verwenden Sie `apply-immediately` auf RDS for SQL Server.

Weitere Hinweise zur Verwendung von AWS CLI zum Ändern eines Parameterwerts finden Sie unter modify-db-parameter-group. Weitere Informationen zur Verwendung der RDS-API zum Ändern eines Parameterwerts finden Sie unter DBParameterGruppe ändern.

Wenn auf der DB-Instance noch nicht die neuesten Änderungen der zugeordneten DB-Parametergruppe übernommen wurden, gibt die Konsole für die DB-Parametergruppe den Status pending-reboot an. Dieser Status führt während des nächsten Wartungsfensters nicht zu einem automatischen Neustart. Damit die neuesten Parameteränderungen für diese DB-Instance übernommen werden, starten Sie die DB-Instance manuell neu.

Statische und dynamische DB-Cluster-Parameter

Die DB-Cluster-Parameter sind entweder statisch oder dynamisch. Sie weisen folgende Unterschiede auf:

- Wenn Sie einen statischen Parameter ändern und die DB-Cluster-Parametergruppe speichern, wird die Änderung des Parameters nach einem manuellen Neustart der zugeordneten DB-Cluster wirksam. Bei statischen Parametern verwendet die Konsole immer `pending-reboot` als `ApplyMethod`.
- Wenn Sie einen dynamischen Parameter ändern, wird die Parameteränderung standardmäßig sofort wirksam, ohne dass ein Neustart erforderlich ist. Wenn Sie den AWS Management Console zum Ändern von DB-Cluster-Parameterwerten verwenden, verwendet `immediate` er immer `ApplyMethod` für dynamische Parameter. Verwenden Sie die AWS CLI oder RDS-API, um die Änderung des Parameters zu verschieben, bis ein zugeordneter DB-Cluster neu gestartet wurde. Legen Sie die `ApplyMethod` für die Parameteränderung auf `pending-reboot` fest.

Weitere Informationen zur Verwendung von AWS CLI zum Ändern eines Parameterwerts finden Sie unter -group. modify-db-cluster-parameter Weitere Informationen zur Verwendung der RDS-API zum Ändern eines Parameterwerts finden Sie unter Ändern DBCluster ParameterGroup.

Zeichensatzparameter

Bevor Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen, legen Sie alle Parameter für den Zeichensatz oder die Datenbanksortierung in Ihrer Parametergruppe fest. Führen Sie diesen Schritt auch aus, bevor Sie darin eine Datenbank erstellen. Dadurch stellen Sie sicher, dass die Standard-Datenbank und neue Datenbanken den Zeichensatz und die Sortierungswerte verwenden, die Sie angeben. Wenn Sie einen Zeichensatz oder eine Sammlung von Parametern ändern, werden die Parameteränderungen nicht in Ihren bestehenden Datenbanken angewandt.

Bei einigen DB-Engines können Sie den Zeichensatz oder die Sortierreihenfolge für eine bestehende Datenbank ändern, indem Sie z. B. den Befehl `ALTER DATABASE` verwenden:

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

Weitere Informationen zum Ändern des Zeichensatzes oder der Sortierreihenfolge für eine Datenbank finden Sie in der Dokumentation zu Ihrer DB-Engine.

Unterstützte Parameter und Parameterwerte

Wenn Sie die unterstützten Parameter für Ihre DB-Engine ermitteln möchten, zeigen Sie die Parameter in der DB-Parametergruppe und in der DB-Cluster-Parametergruppe an, die vom DB-Cluster oder von der DB-Instance verwendet werden. Weitere Informationen erhalten Sie unter Parameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen und Parameterwerte für eine DB-Cluster-Parametergruppe anzeigen.

In vielen Fällen können Sie Ganzzahl- und Boolesche Parameter mithilfe von Ausdrücken, Formeln und Funktionen angeben. Funktionen können einen mathematischen "log"-Ausdruck enthalten. Nicht alle Parameter unterstützen jedoch Ausdrücke, Formeln und Funktionen für Parameterwerte. Weitere Informationen finden Sie unter Festlegen von DB-Parametern.

Werden die Parameter in einer Parametergruppe unpassend eingestellt, kann dies unbeabsichtigte unerwünschte Auswirkungen haben, einschließlich verminderter Leistung und Systeminstabilität. Gehen Sie immer mit Bedacht vor, wenn Sie Datenbankparameter ändern, und sichern Sie Ihre Daten, bevor Sie eine Parametergruppe ändern. Führen Sie Änderungen an einer Parametergruppe immer zuerst auf einer Test-DB-Instance oder einem DB-Cluster aus, bevor Sie diese Änderungen für eine Produktions-DB-Instance oder einen -DB-Cluster übernehmen.

DB-Parametergruppen für Amazon RDS Amazon

DB-Instances verwenden DB-Parametergruppen. Die folgenden Abschnitte beschreiben das Konfigurieren und Verwalten von DB-Instance-Parametergruppen.

Themen

- Erstellen einer DB-Parametergruppe in Amazon RDS
- Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon
- Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS
-
- Kopieren einer DB-Parametergruppe in Amazon RDS
- DB-Parametergruppen in Amazon RDS Amazon auflisten
- Parameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen
- Löschen einer DB-Parametergruppe in Amazon RDS

Erstellen einer DB-Parametergruppe in Amazon RDS

Sie können eine neue DB-Parametergruppe mit dem AWS Management Console AWS CLI, dem oder dem erstellen RDSAPI.

Die folgenden Einschränkungen gelten für den Namen der DB-Parametergruppe:

- Der Name muss zwischen 1 und 255 Buchstaben, Zahlen oder Bindestriche enthalten.

  Standardnamen für Parametergruppen können einen Punkt enthalten, z. B. `default.mysql8.0`. Namen von benutzerdefinierten Parametergruppen dürfen jedoch keinen Punkt enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Der Name darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Konsole

So erstellen Sie eine DB-Parametergruppe:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie Create parameter group (Parametergruppe erstellen).
4. Geben Sie unter Parametergruppenname den Namen Ihrer neuen DB-Parametergruppe ein.
5. Geben Sie unter Beschreibung eine Beschreibung für Ihre neue DB-Parametergruppe ein.
6. Wählen Sie als Engine-Typ Ihre DB-Engine aus.
7. Wählen Sie für Parametergruppenfamilie eine DB-Parametergruppenfamilie aus.
8. Wählen Sie für Typ, falls zutreffend, DB-Parametergruppe aus.
9. Wählen Sie Create (Erstellen) aus.

AWS CLI

Verwenden Sie den AWS CLI `create-db-parameter-group`Befehl, um eine DB-Parametergruppe zu erstellen. Im folgenden Beispiel wird eine DB-Parametergruppe namens mydbparametergroup für My SQL Version 8.0 mit der Beschreibung "My new parameter group“ erstellt.

Nutzen Sie die folgenden erforderlichen Parameter:

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Um alle verfügbaren Parametergruppenfamilien aufzulisten, führen Sie den folgenden Befehl aus:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> Die Ausgabe enthält Duplikate.

Example

FürLinux, oder: macOS Unix

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL8.0 \
    --description "My new parameter group"
```

Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --db-parameter-group-family MySQL8.0 ^
    --description "My new parameter group"
```

Die Ausgabe dieses Befehls sieht etwa so aus:

```
DBPARAMETERGROUP  mydbparametergroup  mysql8.0  My new parameter group
```

RDS API

Verwenden Sie die RDS API `CreateDBParameterGroup`Operation, um eine DB-Parametergruppe zu erstellen.

Nutzen Sie die folgenden erforderlichen Parameter:

- `DBParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon

Sie können Ihre eigenen DB-Parametergruppen mit benutzerdefinierten Einstellungen erstellen. Sie können eine DB-Parametergruppe mit dem AWS Management Console, dem oder dem einer DB-Instance zuordnen RDSAPI. AWS CLI Das können Sie tun, wenn Sie eine DB-Instance erstellen oder ändern.

Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS. Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Note**
>
> Wenn Sie eine neue DB-Parametergruppe einer DB-Instance zuordnen, werden die geänderten statischen und dynamischen Parameter erst nach Neustart der DB-Instance angewendet. Wenn Sie jedoch dynamische Parameter in der DB-Parametergruppe ändern, nachdem Sie sie der DB-Instance zugeordnet haben, werden diese Änderungen sofort ohne Neustart angewendet.

Konsole

So verknüpfen Sie eine DB-Parametergruppe mit einer DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Ändern Sie die Einstellung für DB-Parametergruppen .
5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.
6. (Optional) Klicken Sie auf Apply immediately (Sofort anwenden), um die Änderungen direkt zu übernehmen. Die Auswahl dieser Option kann in einigen Fällen einen Ausfall verursachen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.
7. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern.

   Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Um eine DB-Parametergruppe mit einer DB-Instance zu verknüpfen, verwenden Sie den AWS CLI `modify-db-instance`Befehl mit den folgenden Optionen:

- `--db-instance-identifier`
- `--db-parameter-group-name`

Im folgenden Beispiel wird die mydbpg-DB-Parametergruppe mit der database-1-DB-Instance verknüpft. Die Änderungen werden mit sofort übernomme --apply-immediately. Verwenden Sie --no-apply-immediately, um Änderungen im nächsten Wartungszeitraum anzuwenden. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier database-1 \
    --db-parameter-group-name mydbpg \
    --apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier database-1 ^
    --db-parameter-group-name mydbpg ^
    --apply-immediately
```

RDS API

Um eine DB-Parametergruppe mit einer DB-Instance zu verknüpfen, verwenden Sie den RDS API ModifyDBInstanceVorgang mit den folgenden Parametern:

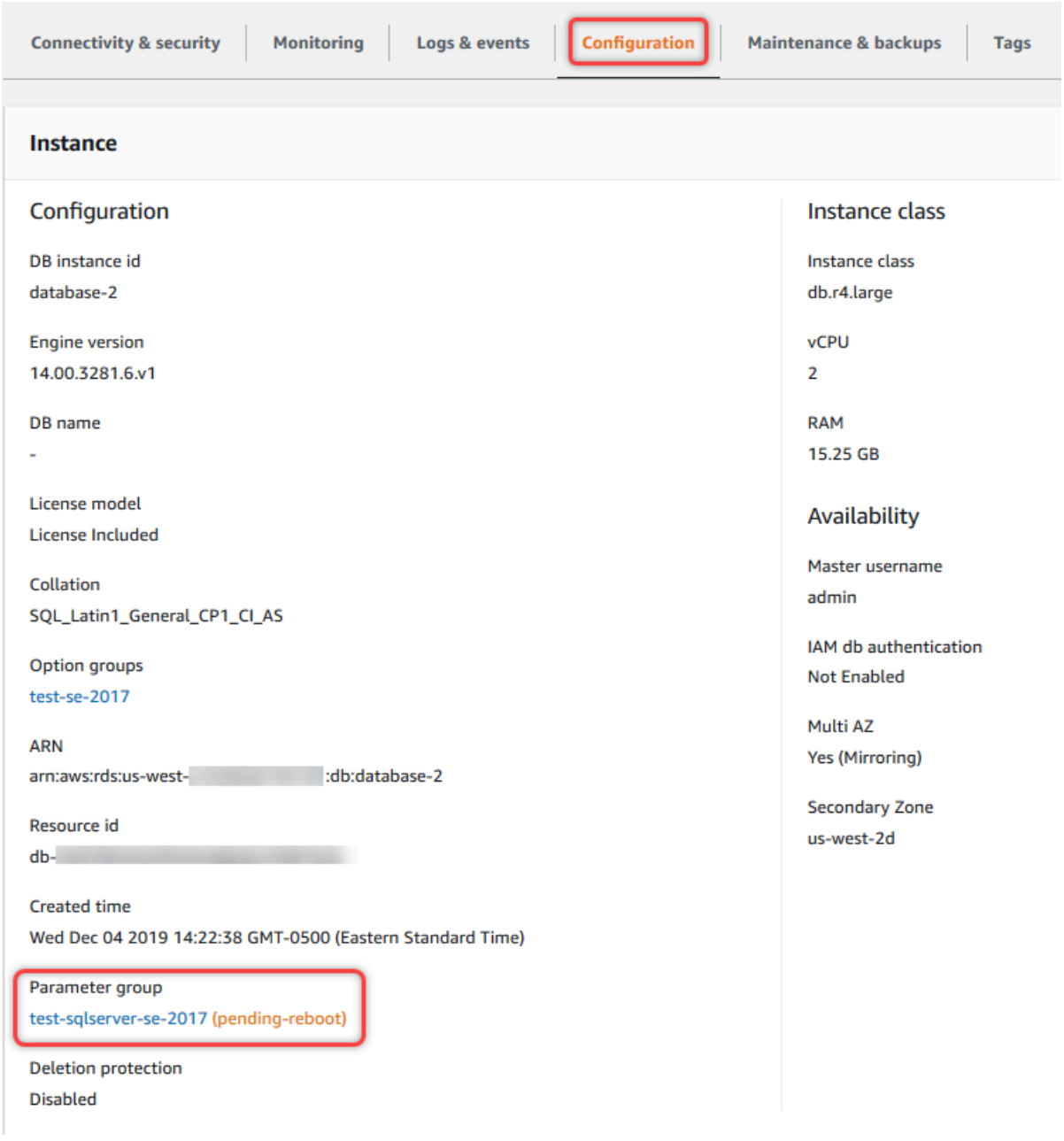
- DBInstanceName
- DBParameterGroupName

Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS

Sie können die Parameterwerte in einer benutzerdefinierten DB-Parametergruppe ändern. Die Parameterwerte in einer Standard-DB-Parametergruppe können nicht geändert werden. Änderungen bei Parametern in einer benutzerdefinierten DB-Parametergruppe werden auf alle DB-Instances angewandt, die dieser DB-Parametergruppe zugeteilt sind.

Änderungen an einigen Parametern werden sofort ohne Neustart auf die DB-Instance angewendet. Änderungen an anderen Parametern werden erst angewendet, nachdem die DB-Instance neu gestartet wurde. In der RDS-Konsole wird der Status einer DB-Parametergruppe, die einer DB-Instance zugeordnet ist, auf der Registerkarte Konfiguration angezeigt. Nehmen wir beispielsweise

an, dass auf der DB-Instance die neuesten Änderungen an der zugeordneten DB-Parametergruppe noch nicht übernommen wurden. In diesem Fall gibt die RDS-Konsole für die DB-Parametergruppe den Status pending-reboot an. Damit die neuesten Parameteränderungen für diese DB-Instance übernommen werden, starten Sie die DB-Instance manuell neu. Wir empfehlen, die DB-Instance immer neu zu starten, nachdem Sie die Parameter geändert haben. Wenn Sie die DB-Instance nicht neu starten, kann ein Failover-Vorgang länger als normal dauern.



Konsole

Um die Parameter in einer DB-Parametergruppe zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste den Namen der Parametergruppe aus, die Sie ändern möchten.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Bearbeiten.
5. Ändern Sie wie gewünscht die Werte der Parameter. Sie können durch die Parameter scrollen, in dem Sie die Pfeiltasten oben rechts im Dialogfeld verwenden.

   Die Werte in einer Standardparametergruppe können Sie nicht ändern.
6. Wählen Sie Save Changes.

AWS CLI

Um eine DB-Parametergruppe zu ändern, verwenden Sie den AWS CLI `modify-db-parameter-group`Befehl mit den folgenden erforderlichen Optionen:

- `--db-parameter-group-name`
- `--parameters`

Im folgenden Beispiel werden die Werte `max_connections` und `max_allowed_packet` in der DB-Parametergruppe mydbparametergroup geändert.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" \

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" ^

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBPARAMETERGROUP  mydbparametergroup
```

RDS-API

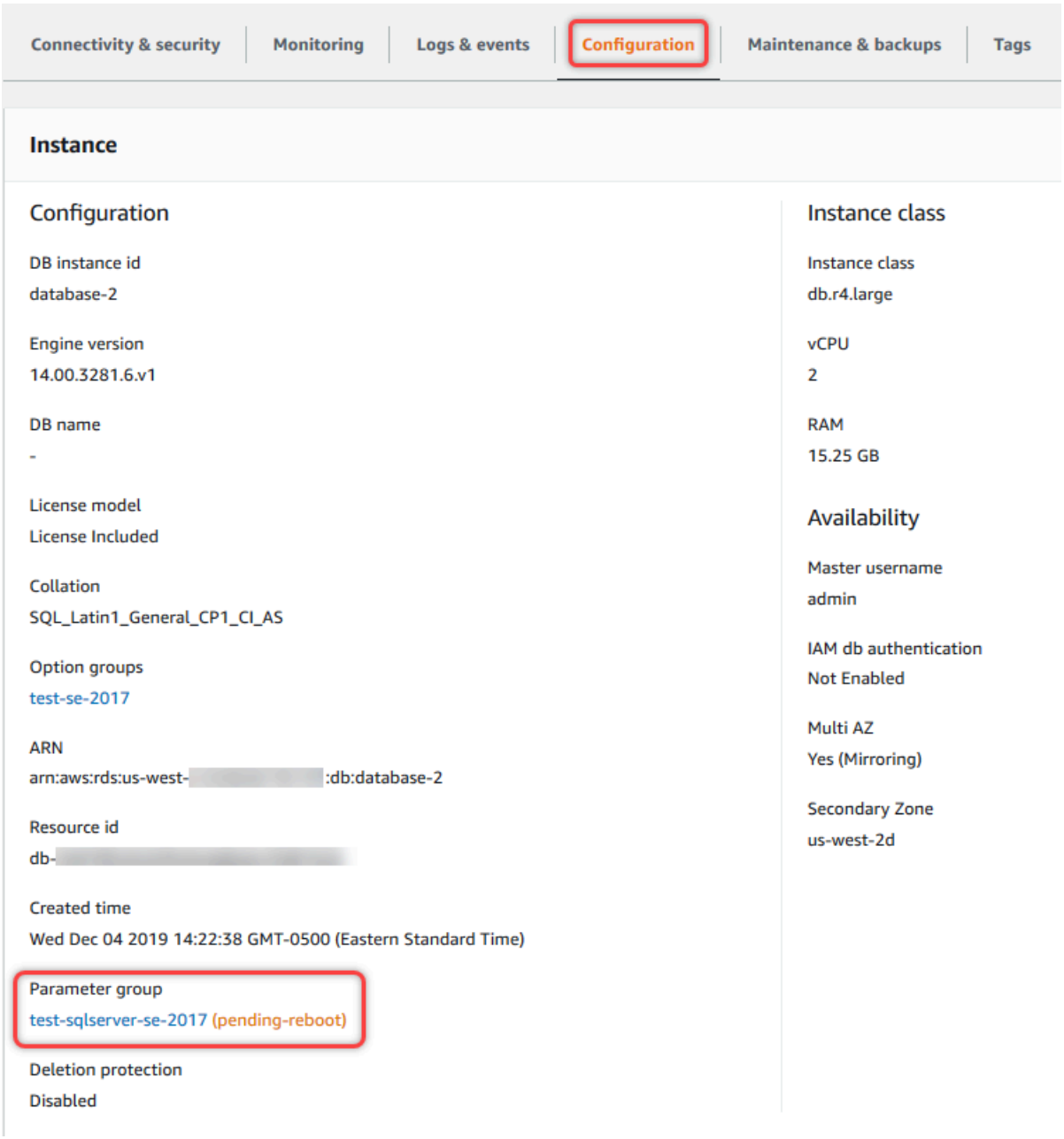
Zum Ändern einer DB-Parametergruppe verwenden Sie die RDS-API-Operation ModifyDBParameterGroup mit den folgenden erforderlichen Parametern:

- DBParameterGroupName
- Parameters

Sie können Parameterwerte in einer vom Kunden erstellten DB-Parametergruppe auf ihre Standardwerte zurücksetzen. Änderungen bei Parametern in einer benutzerdefinierten DB-Parametergruppe werden auf alle DB-Instances angewandt, die dieser DB-Parametergruppe zugeteilt sind.

Wenn Sie die Konsole verwenden, können Sie bestimmte Parameter auf ihre Standardwerte zurücksetzen. Sie können jedoch nicht einfach alle Parameter in der DB-Parametergruppe auf einmal zurücksetzen. Wenn Sie das AWS CLI oder verwenden RDSAPI, können Sie bestimmte Parameter auf ihre Standardwerte zurücksetzen. Sie können auch alle Parameter in der DB-Parametergruppe auf einmal zurücksetzen.

Änderungen an einigen Parametern werden sofort ohne Neustart auf die DB-Instance angewendet. Änderungen an anderen Parametern werden erst angewendet, nachdem die DB-Instance neu gestartet wurde. Die RDS Konsole zeigt den Status der DB-Parametergruppe, die einer DB-Instance zugeordnet ist, auf der Registerkarte Konfiguration an. Nehmen wir beispielsweise an, dass auf der DB-Instance die neuesten Änderungen an der zugeordneten DB-Parametergruppe noch nicht übernommen wurden. Wenn ja, zeigt die RDS Konsole die DB-Parametergruppe mit dem Status Pending-reboot an. Damit die neuesten Parameteränderungen für diese DB-Instance übernommen werden, starten Sie die DB-Instance manuell neu.



> **Note**
>
> In einer Standard-DB-Parametergruppe werden Parameter immer auf ihre Standardwerte festgelegt.

Konsole

So setzen Sie Parameter in einer DB-Parametergruppe auf ihre Standardwerte zurück

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste die Parametergruppe aus.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Bearbeiten.
5. Wählen Sie die Parameter aus, die Sie auf ihre Standardwerte zurücksetzen möchten. Sie können durch die Parameter scrollen, in dem Sie die Pfeiltasten oben rechts im Dialogfeld verwenden.

   Sie können die Werte in einer Standardparametergruppe nicht zurücksetzen.
6. Wählen Sie Reset (Zurücksetzen) und bestätigen Sie dann mit Reset parameters (Parameter zurücksetzen).

AWS CLI

Um einige oder alle Parameter in einer DB-Parametergruppe zurückzusetzen, verwenden Sie den AWS CLI `reset-db-parameter-group`Befehl mit der folgenden erforderlichen Option:`--db-parameter-group-name`.

Um alle Parameter in der DB-Parametergruppe zurückzusetzen, geben Sie die Option `--reset-all-parameters` an. Um bestimmte Parameter zurückzusetzen, geben Sie die Option `--parameters` an.

Im folgenden Beispiel werden alle Parameter in der DB-Parametergruppe namens mydbparametergroup auf ihre Standardwerte zurückgesetzt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

Im folgenden Beispiel werden die Optionen `max_connections` und `max_allowed_packet` in der DB-Parametergruppe namens mydbparametergroup auf ihre Standardwerte zurückgesetzt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" ^
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBParameterGroupName  mydbparametergroup
```

RDS API

Um Parameter in einer DB-Parametergruppe auf ihre Standardwerte zurückzusetzen, verwenden Sie den RDS API `ResetDBParameterGroup`Befehl mit dem folgenden erforderlichen Parameter:`DBParameterGroupName`.

Um alle Parameter in der DB-Parametergruppe zurückzusetzen, setzen Sie den Parameter `ResetAllParameters` auf `true`. Um bestimmte Parameter zurückzusetzen, geben Sie den Parameter `Parameters` an.

Kopieren einer DB-Parametergruppe in Amazon RDS

Sie können benutzerdefinierte DB-Parametergruppen, die Sie erstellt haben, kopieren. Das Kopieren einer Parametergruppe kann eine bequeme Lösung sein. Ein Beispiel ist, wenn Sie eine DB-

Parametergruppe erstellt haben und die am häufigsten verwendeten Parameter und Werte in einer neuen DB-Parametergruppe aufnehmen möchten. Sie können eine DB-Parametergruppe kopieren, indem Sie den verwenden AWS Management Console. Sie können auch den AWS CLI copy-db-parameter-groupBefehl oder die Operation RDS API C opyDBParameter Group verwenden.

Nachdem Sie eine DB-Parametergruppe kopiert haben, warten Sie mindestens fünf Minuten, bevor Sie die erste DB-Instance erstellen, die diese DB-Parametergruppe als Standardparametergruppe verwendet. Auf diese Weise kann Amazon RDS die Kopieraktion vollständig abschließen, bevor die Parametergruppe verwendet wird. Dies ist insbesondere für Parameter wichtig, die beim Erstellen der Standarddatenbank für eine DB-Instance wichtig sind. Ein Beispiel ist der Zeichensatz für die mit dem Parameter `character_set_database` definierte Standarddatenbank. Verwenden Sie die Option Parameter Groups der RDSAmazon-Konsole oder den describe-db-parametersBefehl, um zu überprüfen, ob Ihre DB-Parametergruppe erstellt wurde.

> **Note**
>
> Standardparametergruppen können nicht kopiert werden. Sie können jedoch eine neue Parametergruppe erstellen, die auf einer Standardparametergruppe basiert.
> Sie können eine DB-Parametergruppe nicht in eine andere AWS-Konto oder kopieren AWS-Region.

Konsole

So kopieren Sie eine DB-Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste die zu kopierende benutzerdefinierte Parametergruppe.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Kopieren.
5. Geben Sie unter New DB parameter group identifier (Neue DB-Parametergruppenkennung) einen Namen für die neue Parametergruppe ein.
6. Geben Sie unter Beschreibung eine Beschreibung für die neue Parametergruppe ein.
7. Wählen Sie die Option Copy aus.

AWS CLI

Um eine DB-Parametergruppe zu kopieren, verwenden Sie den AWS CLI `copy-db-parameter-group`Befehl mit den folgenden erforderlichen Optionen:

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

Im folgenden Beispiel wird eine neue DB-Parametergruppe mit dem Namen `mygroup2 that is a copy of the DB parameter group mygroup1` erstellt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds copy-db-parameter-group \
    --source-db-parameter-group-identifier mygroup1 \
    --target-db-parameter-group-identifier mygroup2 \
    --target-db-parameter-group-description "DB parameter group 2"
```

Windows:

```
aws rds copy-db-parameter-group ^
    --source-db-parameter-group-identifier mygroup1 ^
    --target-db-parameter-group-identifier mygroup2 ^
    --target-db-parameter-group-description "DB parameter group 2"
```

RDS API

Um eine DB-Parametergruppe zu kopieren, verwenden Sie den RDS API `CopyDBParameterGroup`Vorgang mit den folgenden erforderlichen Parametern:

- `SourceDBParameterGroupIdentifier`
- `TargetDBParameterGroupIdentifier`
- `TargetDBParameterGroupDescription`

DB-Parametergruppen in Amazon RDS Amazon auflisten

Sie können die DB-Parametergruppen auflisten, die Sie für Ihr AWS Konto erstellt haben.

> **Note**
>
> Standard-Parametergruppen werden automatisch aus einer Vorlage für Standard-Parameter erstellt, wenn Sie eine DB-Instance für eine bestimmte DB-Engine und -Version erstellen. Diese Standardparametergruppen enthalten bevorzugte Parametereinstellungen und können nicht geändert werden. Wenn Sie eine benutzerdefinierte Parametergruppe erstellen, können Sie Parametereinstellungen ändern.

Konsole

Um alle DB-Parametergruppen für ein AWS Konto aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   Die verfügbaren DB-Parametergruppen erscheinen in einer Liste.

AWS CLI

Verwenden Sie den AWS CLI `describe-db-parameter-groups`Befehl, um alle DB-Parametergruppen für ein AWS Konto aufzulisten.

Example

Im folgenden Beispiel werden alle verfügbaren DB-Parametergruppen für ein AWS -Konto aufgelistet.

```
aws rds describe-db-parameter-groups
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBPARAMETERGROUP  default.mysql8.0     mysql8.0  Default parameter group for MySQL8.0
DBPARAMETERGROUP  mydbparametergroup   mysql8.0  My new parameter group
```

Im folgenden Beispiel wird die Parametergruppe mydbparamgroup1 beschrieben.

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-parameter-groups \
    --db-parameter-group-name mydbparamgroup1
```

Windows:

```
aws rds describe-db-parameter-groups ^
    --db-parameter-group-name mydbparamgroup1
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBPARAMETERGROUP  mydbparametergroup1  mysql8.0  My new parameter group
```

RDS API

Verwenden Sie den RDS API DescribeDBParameterGroupsVorgang, um alle DB-Parametergruppen für ein AWS Konto aufzulisten.

Parameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen

Sie können eine Liste aller Parameter in einer DB-Parametergruppe und ihren Werten erhalten.

Konsole

So können Sie die Parameterwerte für eine DB-Parametergruppe ansehen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   Die verfügbaren DB-Parametergruppen erscheinen in einer Liste.
3. Wählen Sie den Namen der Parametergruppe, um deren Parameterliste anzuzeigen.

AWS CLI

Um die Parameterwerte für eine DB-Parametergruppe anzuzeigen, verwenden Sie den AWS CLI describe-db-parametersBefehl mit dem folgenden erforderlichen Parameter.

- `--db-parameter-group-name`

Example

Im folgenden Beispiel werden die Parameter und Parameterwerte für eine DB-Parametergruppe mit dem Namen mydbparametergroup. aufgelistet.

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBPARAMETER  Parameter Name            Parameter Value  Source           Data Type
 Apply Type  Is Modifiable
DBPARAMETER  allow-suspicious-udfs                      engine-default   boolean
 static      false
DBPARAMETER  auto_increment_increment                   engine-default   integer
 dynamic     true
DBPARAMETER  auto_increment_offset                      engine-default   integer
 dynamic     true
DBPARAMETER  binlog_cache_size         32768            system           integer
 dynamic     true
DBPARAMETER  socket                    /tmp/mysql.sock  system           string
 static      false
```

RDS API

Um die Parameterwerte für eine DB-Parametergruppe anzuzeigen, verwenden Sie den RDS API `DescribeDBParameters`Befehl mit dem folgenden erforderlichen Parameter.

- `DBParameterGroupName`

Löschen einer DB-Parametergruppe in Amazon RDS

Sie können eine DB-Parametergruppe mit dem AWS Management Console, AWS CLI, oder löschen RDSAPI. Eine Parametergruppe kann nur gelöscht werden, wenn sie keiner DB-Instance zugeordnet ist.

Konsole

Um eine DB-Parametergruppe zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   Die verfügbaren DB-Parametergruppen erscheinen in einer Liste.
3. Wählen Sie den Namen der Parametergruppen, die gelöscht werden sollen.
4. Wählen Sie Aktionen und dann Löschen.
5. Überprüfen Sie die Namen der Parametergruppen und wählen Sie dann Löschen.

AWS CLI

Um eine DB-Parametergruppe zu löschen, verwenden Sie den AWS CLI `delete-db-parameter-group`Befehl mit dem folgenden erforderlichen Parameter.

- `--db-parameter-group-name`

Example

Im folgenden Beispiel wird eine DB-Parametergruppe mit dem Namen mydbparametergroup gelöscht.

```
aws rds delete-db-parameter-group --db-parameter-group-name mydbparametergroup
```

RDS API

Um eine DB-Parametergruppe zu löschen, verwenden Sie den RDS API `DeleteDBParameterGroup`Befehl mit dem folgenden erforderlichen Parameter.

- `DBParameterGroupName`

Arbeiten mit DB-Cluster-Parametergruppen für Multi-AZ-DB-Cluster

Multi-AZ-DB-Cluster verwenden DB-Cluster-Parametergruppen. Die folgenden Abschnitte beschreiben das Konfigurieren und Verwalten von DB-Cluster-Parametergruppen.

Themen

- Eine DB-Cluster-Parametergruppe erstellen
- Ändern von Parametern in einer DB-Cluster-Parametergruppe

- Zurücksetzen von Parametern in einer DB-Cluster-Parametergruppe
- Kopieren einer DB-Cluster-Parametergruppe
- Auflisten von DB-Cluster-Parametergruppen
- Parameterwerte für eine DB-Cluster-Parametergruppe anzeigen
- Löschen einer DB-Cluster-Parametergruppe

Eine DB-Cluster-Parametergruppe erstellen

Sie können eine neue DB-Cluster-Parametergruppe mithilfe der AWS Management Console AWS CLI, der oder der RDS-API erstellen.

Nachdem Sie eine DB-Cluster-Parametergruppe erstellt haben, sollten Sie mindestens fünf Minuten warten, bevor Sie einen DB-Cluster erstellen, der diese DB-Cluster-Parametergruppe verwendet. Auf diese Weise kann Amazon RDS die Parametergruppe vollständig erstellen, bevor sie vom neuen DB-Cluster verwendet wird. Sie können die Seite Parametergruppen in der Amazon RDS-Konsole oder den describe-db-cluster-parametersBefehl verwenden, um zu überprüfen, ob Ihre DB-Cluster-Parametergruppe erstellt wurde.

Die folgenden Einschränkungen gelten für den Namen der DB-Cluster-Parametergruppe:

- Der Name muss zwischen 1 und 255 Buchstaben, Zahlen oder Bindestriche enthalten.

  Standardnamen für Parametergruppen können einen Punkt enthalten, z. B. `default.mysql5.7`. Namen von benutzerdefinierten Parametergruppen dürfen jedoch keinen Punkt enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Der Name darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Konsole

So erstellen Sie eine DB-Cluster-Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie Create parameter group (Parametergruppe erstellen).

4. Geben Sie unter Parametergruppenname den Namen der neuen DB-Cluster-Parametergruppe ein.
5. Geben Sie unter Beschreibung eine Beschreibung für die neue DB-Cluster-Parametergruppe ein.
6. Wählen Sie als Engine-Typ Ihre Datenbank-Engine aus.
7. Wählen Sie für Parametergruppenfamilie eine DB-Parametergruppenfamilie aus.
8. Wählen Sie Erstellen aus.

AWS CLI

Verwenden Sie den AWS CLI `create-db-cluster-parameter-group`Befehl, um eine DB-Cluster-Parametergruppe zu erstellen.

Im folgenden Beispiel wird eine DB-Cluster-Parametergruppe mit dem Namen mydbclusterparametergroup für RDS for MySQL Version 8.0 und der Beschreibung „My new cluster parameter group“ erstellt.

Nutzen Sie die folgenden erforderlichen Parameter:

- `--db-cluster-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Um alle verfügbaren Parametergruppenfamilien aufzulisten, führen Sie den folgenden Befehl aus:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> Die Ausgabe enthält Duplikate.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
```

```
    --db-parameter-group-family mysql8.0 \
    --description "My new cluster parameter group"
```

Windows:

```
aws rds create-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --db-parameter-group-family mysql8.0 ^
    --description "My new cluster parameter group"
```

Die Ausgabe dieses Befehls sieht etwa so aus:

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "mysql8.0",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup2"
    }
}
```

RDS-API

Um eine DB-Cluster-Parametergruppe zu erstellen, verwenden Sie die RDS-API-Aktion `CreateDBClusterParameterGroup`.

Nutzen Sie die folgenden erforderlichen Parameter:

- `DBClusterParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Ändern von Parametern in einer DB-Cluster-Parametergruppe

Sie können Parameterwerte in einer vom Kunden erstellten DB-Clusterparametergruppe ändern. Sie können die Parameterwerte in einer Standard-DB-Clusterparametergruppe nicht ändern. Änderungen an Parametern in einer benutzerdefinierten DB-Cluster-Parametergruppe gelten für alle DB-Cluster, die dieser DB-Cluster-Parametergruppe zugeordnet sind.

Konsole

So ändern Sie eine DB-Cluster-Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste die zu ändernde Parametergruppe.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Bearbeiten.
5. Ändern Sie die Werte der Parameter, die Sie ändern möchten. Sie können durch die Parameter scrollen, in dem Sie die Pfeiltasten oben rechts im Dialogfeld verwenden.

   Die Werte in einer Standardparametergruppe können Sie nicht ändern.
6. Wählen Sie Änderungen speichern aus.
7. Starten Sie die Cluster neu, um die Änderungen darauf anzuwenden.

   Wenn Sie den Cluster nicht neu starten, kann ein Failover-Vorgang länger als normal dauern.

AWS CLI

Verwenden Sie den AWS CLI `modify-db-cluster-parameter-group`Befehl mit den folgenden erforderlichen Parametern, um eine DB-Cluster-Parametergruppe zu ändern:

- `--db-cluster-parameter-group-name`
- `--parameters`

Im folgenden Beispiel werden die Werte `server_audit_logging` und `server_audit_logs_upload` in der DB-Cluster-Parametergruppe mydbclusterparametergroup geändert.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" \
```

```
 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Windows:

```
aws rds modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBCLUSTERPARAMETERGROUP  mydbclusterparametergroup
```

RDS-API

Um eine DB-Cluster-Parametergruppe zu ändern, verwenden Sie den RDS-API-Befehl `ModifyDBClusterParameterGroup` mit den folgenden erforderlichen Parametern:

- `DBClusterParameterGroupName`
- `Parameters`

Zurücksetzen von Parametern in einer DB-Cluster-Parametergruppe

Sie können Parameter in einer vom Kunden erstellten DB-Clusterparametergruppe auf ihre Standardwerte zurücksetzen. Änderungen an Parametern in einer benutzerdefinierten DB-Cluster-Parametergruppe gelten für alle DB-Cluster, die dieser DB-Cluster-Parametergruppe zugeordnet sind.

> **Note**
>
> In einer Standardparametergruppe des DB-Clusters werden die Parameter immer auf ihre Standardwerte eingestellt.

Konsole

So setzen Sie Parameter in einer DB-Cluster-Parametergruppe auf ihre Standardwerte zurück

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste die Parametergruppe aus.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Bearbeiten.
5. Wählen Sie die Parameter aus, die Sie auf ihre Standardwerte zurücksetzen möchten. Sie können durch die Parameter scrollen, in dem Sie die Pfeiltasten oben rechts im Dialogfeld verwenden.

   Sie können die Werte in einer Standardparametergruppe nicht zurücksetzen.
6. Wählen Sie Reset (Zurücksetzen) und bestätigen Sie dann mit Reset parameters (Parameter zurücksetzen).
7. Starten Sie die .

AWS CLI

Um Parameter in einer DB-Cluster-Parametergruppe auf ihre Standardwerte zurückzusetzen, verwenden Sie den AWS CLI `reset-db-cluster-parameter-group`Befehl mit der folgenden erforderlichen Option:`--db-cluster-parameter-group-name`.

Um alle Parameter in der Parametergruppe des DB-Clusters zurückzusetzen, wählen Sie die Option `--reset-all-parameters`. Um bestimmte Parameter zurückzusetzen, geben Sie die Option `--parameters` an.

Im folgenden Beispiel werden alle Parameter in der DB-Parametergruppe namens mydbparametergroup auf ihre Standardwerte zurückgesetzt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

Im folgenden Beispiel werden die Werte `server_audit_logging` und `server_audit_logs_upload` in der DB-Cluster-Parametergruppe mydbclusterparametergroup auf ihre Standardwerte zurückgesetzt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters "ParameterName=server_audit_logging,ApplyMethod=immediate" \
                 "ParameterName=server_audit_logs_upload,ApplyMethod=immediate"
```

Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
DBClusterParameterGroupName  mydbclusterparametergroup
```

RDS-API

Um Parameter in einer DB-Cluster-Parametergruppe auf ihre Standardwerte zurückzusetzen, verwenden Sie den `ResetDBClusterParameterGroup`-RDS-API-Befehl mit dem folgenden erforderlichen Parameter: `DBClusterParameterGroupName`.

Um alle Parameter in der Parametergruppe des DB-Clusters zurückzusetzen, legen Sie den Parameter `ResetAllParameters` auf `true` fest. Um bestimmte Parameter zurückzusetzen, geben Sie den Parameter `Parameters` an.

Kopieren einer DB-Cluster-Parametergruppe

Sie können die von Ihnen erstellten benutzerdefinierten DB-Cluster-Parametergruppen kopieren. Das Kopieren einer Parametergruppe ist eine praktische Lösung, wenn Sie bereits eine DB-Cluster-Parametergruppe erstellt haben und die am häufigsten verwendeten Parameter und Werte aus dieser Gruppe in eine neuen DB-Cluster-Parametergruppe übernehmen möchten. Sie können eine DB-Cluster-Parametergruppe kopieren, indem Sie den Befehl AWS CLI copy-db-cluster-parameter-group oder den DBClusterParameterGroupRDS-API-Kopiervorgang verwenden.

Nachdem Sie eine DB-Cluster-Parametergruppe kopiert haben, warten Sie mindestens fünf Minuten, bevor Sie einen DB-Cluster erstellen, der diese DB-Cluster-Parametergruppe verwendet. Auf diese Weise kann Amazon RDS die Parametergruppe vollständig kopieren, bevor sie vom neuen DB-Cluster verwendet wird. Sie können die Seite Parametergruppen in der Amazon RDS-Konsole oder den describe-db-cluster-parametersBefehl verwenden, um zu überprüfen, ob Ihre DB-Cluster-Parametergruppe erstellt wurde.

> **Note**
>
> Standardparametergruppen können nicht kopiert werden. Sie können jedoch eine neue Parametergruppe erstellen, die auf einer Standardparametergruppe basiert.
> Sie können eine DB-Cluster-Parametergruppe nicht in eine andere AWS-Konto oder kopieren AWS-Region.

Konsole

So kopieren Sie eine DB-Cluster-Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste die zu kopierende benutzerdefinierte Parametergruppe.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Kopieren.
5. Geben Sie unter New DB parameter group identifier (Neue DB-Parametergruppenkennung) einen Namen für die neue Parametergruppe ein.
6. Geben Sie unter Beschreibung eine Beschreibung für die neue Parametergruppe ein.
7. Wählen Sie die Option Copy aus.

AWS CLI

Um eine DB-Cluster-Parametergruppe zu kopieren, verwenden Sie den AWS CLI `copy-db-cluster-parameter-group`Befehl mit den folgenden erforderlichen Parametern:

- `--source-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-description`

Im folgenden Beispiel wird eine neue DB-Cluster-Parametergruppe mit dem Namen `mygroup2` als Kopie der DB-Cluster-Parametergruppe `mygroup1` erstellt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier mygroup1 \
    --target-db-cluster-parameter-group-identifier mygroup2 \
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

Windows:

```
aws rds copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier mygroup1 ^
    --target-db-cluster-parameter-group-identifier mygroup2 ^
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

RDS-API

Um eine DB-Cluster-Parametergruppe zu kopieren, verwenden Sie die RDS-API-Operation `CopyDBClusterParameterGroup` mit den folgenden erforderlichen Parametern:

- `SourceDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupDescription`

Auflisten von DB-Cluster-Parametergruppen

Sie können die DB-Cluster-Parametergruppen auflisten, die Sie für Ihr AWS Konto erstellt haben.

> **Note**
>
> Standardparametergruppen werden automatisch aus einer Standardparametervorlage generiert, wenn Sie ein DB-Cluster für eine bestimmte DB-Engine und -Version erstellen. Diese Standardparametergruppen enthalten bevorzugte Parametereinstellungen und können nicht geändert werden. Wenn Sie eine benutzerdefinierte Parametergruppe erstellen, können Sie Parametereinstellungen ändern.

Konsole

Um alle DB-Cluster-Parametergruppen für ein AWS Konto aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   Die DB-Cluster-Parametergruppen erscheinen in der Liste mit DB cluster parameter group (DB-Cluster-Parametergruppe) als Type (Typ).

AWS CLI

Verwenden Sie den AWS CLI `describe-db-cluster-parameter-groups`Befehl, um alle DB-Cluster-Parametergruppen für ein AWS Konto aufzulisten.

Example

Im folgenden Beispiel werden alle verfügbaren DB-Cluster-Parametergruppen für ein AWS -Konto aufgelistet.

```
aws rds describe-db-cluster-parameter-groups
```

Im folgenden Beispiel wird die Parametergruppe mydbclusterparametergroup beschrieben.

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Windows:

```
aws rds describe-db-cluster-parameter-groups ^
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Die Ausgabe des Befehls ähnelt der Folgenden:

```
{
    "DBClusterParameterGroups": [
        {
            "DBClusterParameterGroupName": "mydbclusterparametergroup2",
            "DBParameterGroupFamily": "mysql8.0",
            "Description": "My new cluster parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
        }
    ]
}
```

RDS-API

Verwenden Sie die DescribeDBClusterParameterGroupsRDS-API-Aktion, um alle DB-Cluster-Parametergruppen für ein AWS Konto aufzulisten.

Parameterwerte für eine DB-Cluster-Parametergruppe anzeigen

Sie können alle Parameter in einer DB-Cluster-Parametergruppe mit ihren Werten in einer Liste anzeigen.

Konsole

So zeigen Sie die Parameterwerte für eine DB-Cluster-Parametergruppe an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

Die DB-Cluster-Parametergruppen erscheinen in der Liste mit DB cluster parameter group (DB-Cluster-Parametergruppe) als Type (Typ).

3. Wählen Sie den Namen der DB-Cluster-Parametergruppe, um deren Parameterliste anzuzeigen.

AWS CLI

Um die Parameterwerte für eine DB-Cluster-Parametergruppe anzuzeigen, verwenden Sie den AWS CLI `describe-db-cluster-parameters`Befehl mit dem folgenden erforderlichen Parameter.

- `--db-cluster-parameter-group-name`

Example

Das folgende Beispiel listet die Parameter und Parameterwerte für eine DB-Cluster-Parametergruppe namens mydbclusterparametergroup im JSON-Format auf.

Die Ausgabe des Befehls ähnelt der Folgenden:

```
aws rds describe-db-cluster-parameters --db-cluster-parameter-group-
name mydbclusterparametergroup
```

```
{
    "Parameters": [
        {
            "ParameterName": "activate_all_roles_on_login",
            "ParameterValue": "0",
            "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
            "Source": "engine-default",
            "ApplyType": "dynamic",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
        {
```

```
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only an
 xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
...
```

RDS-API

Um die Parameterwerte für eine DB-Cluster-Parametergruppe anzuzeigen, verwenden Sie den RDS-API-Befehl `DescribeDBClusterParameters` mit dem folgenden erforderlichen Parameter.

- `DBClusterParameterGroupName`

In einigen Fällen werden die zulässigen Werte für einen Parameter nicht angezeigt. Es handelt sich dabei immer um Parameter, bei denen die Quelle die Standardeinstellung der Datenbank-Engine ist.

Um die Werte dieser Parameter anzuzeigen, können Sie die folgenden SQL-Anweisungen ausführen:

- MySQL:

```
-- Show the value of a particular parameter
mysql$ SHOW VARIABLES LIKE '%parameter_name%';

-- Show the values of all parameters
mysql$ SHOW VARIABLES;
```

- PostgreSQL:

```
-- Show the value of a particular parameter
postgresql=> SHOW parameter_name;

-- Show the values of all parameters
postgresql=> SHOW ALL;
```

Löschen einer DB-Cluster-Parametergruppe

Sie können eine DB-Cluster-Parametergruppe mithilfe der AWS Management Console AWS CLI, oder RDS-API löschen. Eine Parametergruppe für eine DB-Cluster-Parametergruppe kann nur gelöscht werden, wenn sie keinem DB-Cluster zugeordnet ist.

Konsole

Um Parametergruppen zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   Die Parametergruppen werden in einer Liste angezeigt.
3. Wählen Sie den Namen der DB-Cluster-Parametergruppen, die gelöscht werden sollen.
4. Wählen Sie Aktionen und dann Löschen.
5. Überprüfen Sie die Namen der Parametergruppen und wählen Sie dann Löschen.

AWS CLI

Um eine DB-Cluster-Parametergruppe zu löschen, verwenden Sie den AWS CLI `delete-db-cluster-parameter-group`Befehl mit dem folgenden erforderlichen Parameter.

- `--db-parameter-group-name`

Example

Im folgenden Beispiel wird eine DB-Cluster-Parametergruppe mit dem Namen mydbparametergroup gelöscht.

```
aws rds delete-db-cluster-parameter-group --db-parameter-group-name mydbparametergroup
```

RDS-API

Um eine DB-Cluster-Parametergruppe zu löschen, verwenden Sie den `DeleteDBClusterParameterGroup`RDS-API-Befehl mit dem folgenden erforderlichen Parameter.

- `DBParameterGroupName`

Vergleichen von DB-Parametergruppen

Sie können den verwenden AWS Management Console , um die Unterschiede zwischen zwei DB-Parametergruppen anzuzeigen.

Die angegebenen Parametergruppen müssen beide DB-Parametergruppen oder DB-Cluster-Parametergruppen sein. Dies gilt, auch wenn die DB-Engine und die Version identisch sind. Sie können beispielsweise eine DB-Parametergruppe `aurora-mysql8.0` (Aurora My SQL Version 3) und eine `aurora-mysql8.0` DB-Cluster-Parametergruppe nicht vergleichen.

Sie können die Parametergruppen Aurora My SQL und RDS for My SQL DB vergleichen, auch für verschiedene Versionen, aber Sie können Aurora Postgre SQL und RDS for Postgre SQL DB-Parametergruppen nicht vergleichen.

Um zwei DB-Parametergruppen zu vergleichen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie in der Liste die beiden zu vergleichenden Parametergruppen.

   > **Note**
   >
   > Um eine Standardparametergruppe mit einer benutzerdefinierten Parametergruppe zu vergleichen, wählen Sie zuerst die Standardparametergruppe auf der Registerkarte Standard und dann die benutzerdefinierte Parametergruppe auf der Registerkarte Benutzerdefiniert aus.

4. Wählen Sie unter Aktionen die Option Vergleichen aus.

Festlegen von DB-Parametern

Zu den DB-Parametertypen gehören die folgenden:

- Ganzzahl
- Boolesch
- String
- Long

- Double
- Zeitstempel
- Objekt anderer definierter Datentypen
- Array von Werten vom Typ Ganzzahlwert, Boolesch, String, Long, Double, Zeitstempel oder Objekt

Sie können auch Ganzzahl- und Boolesche Parameter mit Ausdrücken, Formeln und Funktionen angeben.

Für die Oracle-Engine können Sie die `DBInstanceClassHugePagesDefault`-Formelvariable zur Angabe eines booleschen DB-Parameters verwenden. Siehe DB-Parameter-Formel-Variablen.

Für die SQL Postgre-Engine können Sie einen Ausdruck verwenden, um einen booleschen DB-Parameter anzugeben. Siehe Boolesche DB-Parameterausdrücke.

Inhalt

- DB-Parameter-Formeln
  - DB-Parameter-Formel-Variablen
  - DB-Parameter-Formel-Operatoren
- DB-Parameter-Funktionen
- Boolesche DB-Parameterausdrücke
- Protokollausdrücke von DB-Parametern
- Beispiele für DB-Parameterwerte

DB-Parameter-Formeln

Eine DB-Parameterformel ist ein Ausdruck, der in einen Ganzzahlwert oder einen booleschen Wert aufgelöst wird. Sie schließen den Ausdruck in Klammern ein: {}. Sie können eine Formel entweder für einen DB-Parameterwert oder als Argument für eine DB-Parameterfunktion verwenden.

Syntax

```
{FormulaVariable}
{FormulaVariable*Integer}
{FormulaVariable*Integer/Integer}
{FormulaVariable/Integer}
```

DB-Parameter-Formel-Variablen

Jede Formelvariable gibt einen Ganzzahlwert oder einen booleschen Wert zurück. Bei den Namen aller Variablen muss die Groß- und Kleinschreibung beachtet werden.

AllocatedStorage

Gibt einen Ganzzahlwert zurück, der die Größe des Datenvolumens in Byte darstellt.

DBInstanceClassHugePagesDefault

Gibt einen booleschen Wert zurück. Sie wird derzeit nur für Oracle-Engines unterstützt.

Weitere Informationen finden Sie unter Aktivieren von HugePages für eine Instance von RDS für Oracle.

DBInstanceClassMemory

Gibt eine Ganzzahl für die Anzahl der Speicherbytes zurück, die für den Datenbankprozess verfügbar sind. Diese Zahl wird intern berechnet, indem mit der Gesamtspeichermenge für die DB-Instance-Klasse begonnen wird. Davon subtrahiert die Berechnung den Speicher, der für das Betriebssystem und die RDS Prozesse reserviert ist, die die Instanz verwalten. Daher ist die Zahl immer etwas niedriger als die Speicherzahlen, die in den Instance-Klassentabellen unter genannt werden. Der genaue Wert hängt von einer Kombination von Faktoren ab. Dazu gehören die Instance-Klasse, die DB-Engine und ob sie für eine RDS Instance oder eine Instance gilt, die Teil eines Aurora-Clusters ist.

DBInstanceVCPU

Gibt eine Ganzzahl zurück, die die Anzahl der virtuellen Zentraleinheiten (vCPUs) darstellt, die von Amazon RDS zur Verwaltung der Instance verwendet werden.

EndPointPort

Gibt eine Ganzzahl zurück, die den beim Herstellen einer Verbindung mit der DB-Instance verwendeten Port darstellt.

TrueIfReplica

Gibt 1 zurück, wenn es sich bei der DB-Instance um ein Lesereplikat handelt, und `0`, wenn dies nicht der Fall ist. Dies ist der Standardwert für den `read_only` Parameter in MySQL.

DB-Parameter-Formel-Operatoren

DB-Parameter-Formeln unterstützen zwei Operatoren: Division und Multiplikation.

Divisions-Operator: /

Dividiert den Dividend durch den Divisor, gibt einen Quotienten als Ganzzahl zurück. Dezimalzahlen in einem Quotienten werden gekürzt und nicht gerundet.

Syntax

```
dividend / divisor
```

Die Dividend- und Divisor-Argumente müssen Ausdrücke mit Ganzzahlen sein.

Multiplikations-Operator: *

Multipliziert die Ausdrücke und gibt das Produkt der Ausdrücke zurück. Dezimalstellen in Ausdrücken werden gekürzt (und nicht gerundet).

Syntax

```
expression * expression
```

Beide Ausdrücke müssen Ganzzahlen sein.

DB-Parameter-Funktionen

Sie geben die Argumente von DB-Parameter-Funktionen entweder als Ganzzahlen oder Formeln an. Jede Funktion muss mindestens ein Argument haben. Geben Sie mehrere Argumente als kommagetrennte Liste an. Die Liste darf keine leeren Elemente aufweisen, z. B. argument1,,argument3. Bei Funktionsnamen wird zwischen Groß- und Kleinschreibung unterschieden.

IF

Gibt ein Argument zurück.

Sie wird derzeit nur für Oracle-Engines unterstützt und das einzige unterstützte erste Argument ist `{DBInstanceClassHugePagesDefault}`. Weitere Informationen finden Sie unter Aktivieren von HugePages für eine Instance von RDS für Oracle.

Syntax

```
IF(argument1, argument2, argument3)
```

Gibt das zweite Argument zurück, wenn das erste Argument als wahr ausgewertet wird. Gibt andernfalls das dritte Argument zurück.

GREATEST

Gibt den größten Wert aus einer Liste von Ganzzahlen oder Parameter-Formeln zurück.

Syntax

```
GREATEST(argument1, argument2,...argumentn)
```

Gibt eine Ganzzahl zurück.

LEAST

Gibt den kleinsten Wert aus einer Liste von Ganzzahlen oder Parameter-Formeln zurück.

Syntax

```
LEAST(argument1, argument2,...argumentn)
```

Gibt eine Ganzzahl zurück.

SUM

Addiert die Werte der festgelegten Ganzzahlen oder Parameter-Formeln.

Syntax

```
SUM(argument1, argument2,...argumentn)
```

Gibt eine Ganzzahl zurück.

Boolesche DB-Parameterausdrücke

Ein boolescher DB-Parameterausdruck wird in einen booleschen Wert von 1 oder 0 aufgelöst. Der Ausdruch ist in Anführungszeichen gesetzt.

> **Note**
>
> Boolesche DB-Parameterausdrücke werden nur für die SQL Postgre-Engine unterstützt.

Syntax

```
"expression operator expression"
```

Beide Ausdrücke müssen in Ganzzahlen aufgelöst werden. Ein Ausdruck kann folgender sein:

- Konstante mit Ganzzahlwert
- DB-Parameter-Formel
- DB-Parameter-Funktion
- DB-Parametervariable

Boolesche DB-Parameterausdrücke unterstützen die folgenden Operatoren für Ungleichheit:

Der Operator größere als: **>**

Syntax

```
"expression > expression"
```

Der Operator kleiner als: **<**

Syntax

```
"expression < expression"
```

Die Operatoren größer als oder gleich: **>=**, **=>**

Syntax

```
"expression >= expression"
"expression => expression"
```

Die Operatoren kleiner als oder gleich: **<=**, **=<**

Syntax

```
"expression <= expression"
"expression =< expression"
```

Example Verwenden eines booleschen DB-Parameterausdrucks

Im folgenden Beispiel für einen booleschen DB-Parameterausdruck wird das Ergebnis einer Parameterformel mit einer Ganzzahl verglichen. Auf diese Weise wird der boolesche DB-Parameter `wal_compression` für eine Postgre-DB-Instance geändert. SQL Der Parameterausdruck vergleicht die Zahl von vCPUs mit dem Wert 2. Wenn die Anzahl von größer als 2 vCPUs ist, wird der `wal_compression` DB-Parameter auf true gesetzt.

```
aws rds modify-db-parameter-group --db-parameter-group-name group-name \
--parameters "ParameterName=wal_compression,ParameterValue=\"{DBInstanceVCPU} > 2\" "
```

Protokollausdrücke von DB-Parametern

Sie können einen DB-Parameterwert, der eine Ganzzahl ist, auf einen Protokollausdruck festlegen. Sie schließen den Ausdruck in Klammern ein: {}. Zum Beispiel:

```
{log(DBInstanceClassMemory/8187281418)*1000}
```

Die `log`-Funktion repräsentiert die Protokollbasis 2. In diesem Beispiel wird auch die `DBInstanceClassMemory`-Formelvariable verwendet. Siehe DB-Parameter-Formel-Variablen.

> **Note**
>
> Derzeit können Sie für den SQL `innodb_log_file_size` Parameter My keinen anderen Wert als eine Ganzzahl angeben.

Beispiele für DB-Parameterwerte

Diese Beispiele zeigen die Verwendung von Formeln, Funktionen und Ausdrücken für die Werte von DB-Parametern.

> **Warning**
>
> Wenn die Parameter in einer DB-Parametergruppe unpassend eingestellt werden, kann dies unbeabsichtigte unerwünschte Auswirkungen haben. Dies kann eine gestörte Leistung und Systeminstabilität umfassen. Gehen Sie mit Bedacht vor, wenn Sie Datenbank-Parameter ändern und sichern Sie Ihre Daten bevor Sie Ihre DB-Parametergruppe ändern. Testen Sie Parametergruppenänderungen an einer Test-DB-Instance, die mit, erstellt wurde point-in-

time-restores, bevor Sie diese Änderungen an den Parametergruppen auf Ihre Produktions-DB-Instances anwenden.

Example mithilfe der DB-Parameterfunktion GREATEST

Sie können die Funktion `GREATEST` in einem Oracle-Prozessparameter angeben. Verwenden Sie es, um die Anzahl der Benutzerprozesse auf größer als 80 oder `DBInstanceClassMemory` geteilt durch 9.868.951 einzustellen.

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

Example mit der DB-Parameter-Funktion LEAST

Sie können die `LEAST` Funktion in einem SQL `max_binlog_cache_size` Parameterwert vom Typ My angeben. Verwenden Sie diese Option, um die maximale Cachegröße, die eine Transaktion in einer SQL My-Instanz verwenden kann, auf 1 MB oder `DBInstanceClass` /256 festzulegen, je nachdem, welcher Wert kleiner ist.

```
LEAST({DBInstanceClassMemory/256},10485760)
```

Erstellen eines ElastiCache Amazon-Caches mithilfe des Amazon RDS DB-Instance-Einstellungen

ElastiCache ist ein vollständig verwalteter In-Memory-Caching-Dienst, der Lese- und Schreiblatenzen im Mikrosekundenbereich bietet und flexible Anwendungsfälle in Echtzeit unterstützt. ElastiCache kann Ihnen helfen, die Anwendungs- und Datenbankleistung zu beschleunigen. Sie können ihn ElastiCache als primären Datenspeicher für Anwendungsfälle verwenden, die keine Datenbeständigkeit erfordern, z. B. Bestenlisten für Spiele, Streaming und Datenanalysen. ElastiCache trägt dazu bei, die Komplexität zu verringern, die mit der Bereitstellung und Verwaltung einer verteilten Computerumgebung verbunden ist. Weitere Informationen finden Sie unter Häufige ElastiCache Anwendungsfälle und Wie ElastiCache kann ich für Memcached helfen? und Allgemeine ElastiCache Anwendungsfälle und Wie ElastiCache kann ich für Redis helfen? OSS Sie können die RDS Amazon-Konsole zum Erstellen des ElastiCache Caches verwenden.

Sie können Amazon ElastiCache in zwei Formaten betreiben. Sie können mit einem Serverless-Cache beginnen oder einen eigenen Cache-Cluster entwerfen. Wenn Sie sich dafür entscheiden, Ihren eigenen Cache-Cluster zu entwerfen, ElastiCache funktioniert er sowohl mit der Redis OSS - als auch mit der Memcached-Engine. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, finden Sie weitere Informationen unter Memcached und Redis im VergleichOSS. Weitere Informationen zu Amazon ElastiCache finden Sie im ElastiCache Amazon-Benutzerhandbuch.

Themen

- Überblick über die ElastiCache Cache-Erstellung mit den
- Einen ElastiCache Cache mit Einstellungen aus einer erstellen

Überblick über die ElastiCache Cache-Erstellung mit den

Sie können einen ElastiCache Cache von Amazon RDS mit denselben Konfigurationseinstellungen wie eine neu erstellte oder bestehende erstellen.

Einige Anwendungsfälle, um einen ElastiCache Cache mit Ihrer zu verknüpfen:

- Sie können Kosten sparen und Ihre Leistung verbessern, indem Sie es ElastiCache zusammen RDS verwenden und RDS nicht alleine laufen.

Sie können beispielsweise bis zu 55% an Kosten sparen und eine bis zu 80-mal schnellere Leseleistung erzielen, wenn Sie ElastiCache mit RDS for My SQL im Vergleich zu RDS My SQL allein arbeiten.

- Sie können den ElastiCache Cache als primären Datenspeicher für Anwendungen verwenden, für die keine Datenbeständigkeit erforderlich ist. Ihre Anwendungen, die Redis OSS oder Memcached verwenden, können fast ElastiCache ohne Änderungen verwendet werden.

Wenn Sie einen ElastiCache Cache aus erstellenRDS, erbt der ElastiCache Cache die folgenden Einstellungen von der zugehörigen :

- ElastiCache Konnektivitätseinstellungen
- ElastiCache Sicherheitseinstellungen

Sie können die Cache-Konfigurationseinstellungen auch Ihren Anforderungen entsprechend festlegen.

Einrichtung ElastiCache in Ihren Anwendungen

Ihre Anwendungen müssen so eingerichtet sein, dass sie den ElastiCache Cache nutzen. Sie können die Cache-Leistung auch optimieren und verbessern, indem Sie Ihre Anwendungen so einrichten, dass sie je nach Ihren Anforderungen Caching-Strategien verwenden.

- Informationen zum Zugriff auf Ihren ElastiCache Cache und zu den ersten Schritten finden Sie unter Erste Schritte mit ElastiCache (RedisOSS) und Erste Schritte mit ElastiCache (Memcached).
- Weitere Informationen zu Caching-Strategien finden Sie unter Caching-Strategien und bewährte Methoden für Memcached und Caching-Strategien und bewährte Methoden für Redis. OSS
- Weitere Informationen zur Hochverfügbarkeit in ElastiCache (Redis-OSS) Clustern finden Sie unter Hochverfügbarkeit mithilfe von Replikationsgruppen.
- Möglicherweise fallen Kosten im Zusammenhang mit Backup-Speicher, Datenübertragung innerhalb oder zwischen Regionen oder der Nutzung von an. AWS Outposts Preisdetails finden Sie unter ElastiCache Amazon-Preise.

Einen ElastiCache Cache mit Einstellungen aus einer erstellen

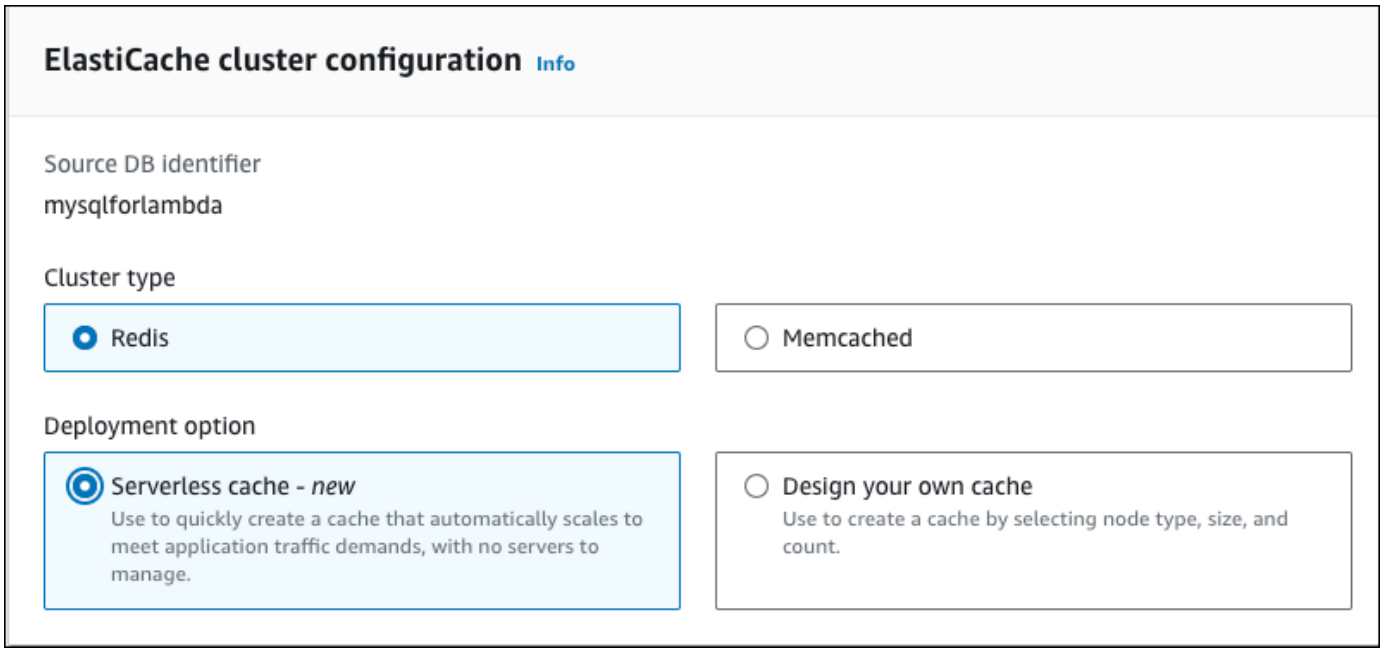
Sie können einen ElastiCache Cache für Ihre mit Einstellungen für die Übernahme von der erstellen.

Erstellen Sie einen ElastiCache Cache mit Einstellungen aus einer

1. Wenn Sie eine DB-Instance erstellen möchten, folgen Sie den Anweisungen in Erstellen einer Amazon RDS-DB-Instance.
2. Nach dem Erstellen einer zeigt die Konsole das Fenster Vorgeschlagene Add-Ons an. Wählen Sie RDSMit Ihren DB-Einstellungen einen ElastiCache Cluster erstellen aus.

   Wählen Sie für eine bestehende Datenbank auf der Seite Datenbanken die erforderliche aus. Wählen Sie im Dropdownmenü Aktionen die Option ElastiCache Cluster erstellen aus, um einen ElastiCache Cache zu erstellenRDS, der dieselben Einstellungen wie Ihre bestehende hat.

   Im Abschnitt ElastiCache Konfiguration zeigt die Quell-DB-ID an, von welcher der ElastiCache Cache die Einstellungen erbt.
3. Wählen Sie aus, ob Sie einen Redis OSS - oder Memcached-Cluster erstellen möchten. Weitere Informationen finden Sie unter Memcached und Redis im VergleichOSS.



4. Wählen Sie anschließend aus, ob Sie einen serverlosen Cache erstellen oder Ihren eigenen Cache entwerfen möchten. Weitere Informationen finden Sie unter Auswahl zwischen Bereitstellungsoptionen.

   Wenn Sie Serverless Cache wählen:

   a. Geben Sie in den Cache-Einstellungen Werte für Name und Beschreibung ein.

b. Behalten Sie unter Standardeinstellungen anzeigen die Standardeinstellungen bei, um die Verbindung zwischen Ihrem Cache und der herzustellen.

c. Sie können die Standardeinstellungen auch bearbeiten, indem Sie Standardeinstellungen anpassen wählen. Wählen Sie die ElastiCache Konnektivitätseinstellungen, ElastiCache Sicherheitseinstellungen und Maximale Nutzungsbeschränkungen aus.

5. Wenn Sie Ihren eigenen Cache entwerfen wählen:

a. Wenn Sie sich für OSSRedis-Cluster entschieden haben, wählen Sie aus, ob Sie den Clustermodus Aktiviert oder Deaktiviert beibehalten möchten. Weitere Informationen finden Sie unter Replikation: Redis OSS (Clustermodus deaktiviert) im Vergleich zu Redis OSS (Clustermodus aktiviert).

b. Geben Sie Werte für Name, Beschreibung und Engine-Version ein.

Für Engine-Version wird als Standardwert die neueste Engine-Version empfohlen. Sie können auch eine Engine-Version für den ElastiCache Cache wählen, die Ihren Anforderungen am besten entspricht.

c. Wählen Sie den Knotentyp in der Option Knotentyp aus. Weitere Informationen finden Sie unter Verwalten von Knoten.

Wenn Sie einen OSS Redis-Cluster mit aktiviertem Clustermodus erstellen möchten, geben Sie die Anzahl der Shards (Partitionen/Knotengruppen) in die Option Anzahl der Shards ein.

Geben Sie im Feld Anzahl der Replikate die Zahl der Replikate jedes Shards ein.

> **Note**
>
> Der gewählte Knotentyp, die Anzahl der Shards und die Anzahl der Replikate wirken sich alle auf Ihre Cache-Leistung und die Ressourcenkosten aus. Stellen Sie sicher, dass diese Einstellungen Ihren Datenbankanforderungen entsprechen. Preisinformationen finden Sie unter ElastiCache Amazon-Preise.

d. Wählen Sie die ElastiCache Konnektivitäts - und ElastiCache Sicherheitseinstellungen aus. Sie können die Standardeinstellungen beibehalten oder diese Einstellungen an Ihre Anforderungen anpassen.

6. Überprüfen Sie die Standardeinstellungen und die geerbten Einstellungen Ihres ElastiCache Caches. Einige Einstellungen können nach der Erstellung nicht geändert werden.

> **Note**
>
> RDSkönnte das Backup-Fenster Ihres ElastiCache Caches so anpassen, dass das Mindestfenster von 60 Minuten eingehalten wird. Das Backup-Fenster Ihrer Quelldatenbank bleibt gleich.

7. Wenn Sie bereit sind, wählen Sie ElastiCache Cache erstellen.

In der Konsole wird ein Bestätigungsbanner für die ElastiCache Cache-Erstellung angezeigt. Folgen Sie dem Link im Banner zur ElastiCache Konsole, um die Cache-Details einzusehen. Die ElastiCache Konsole zeigt den neu erstellten ElastiCache Cache an.

Automatische Migration von EC2 Datenbanken zu Amazon RDS mit AWS Database Migration Service

Sie können die verwenden, um eine EC2 Datenbank zu zu migrieren RDS. verwendet AWS Database Migration Service (AWS DMS), um Ihre EC2 Quelldatenbank zu migrieren. AWS DMS ermöglicht es Ihnen, relationale Datenbanken in Ihre AWS Cloud zu migrieren. Weitere Informationen zu finden Sie AWS Database Migration Service unter Was ist AWS Database Migration Service? im AWS Database Migration Service Benutzerhandbuch.

Um mit der Migration zu beginnen, müssen Sie eine entsprechende erstellen, in die die Daten migriert werden sollen. Nachdem Sie Ihre Zieldatenbank erstellt haben, können Sie Ihre EC2 Datenbank in diese importieren.

Übersicht

Mit der können Sie EC2 Datenbanken in entsprechende migrieren. Sie müssen eine erstellen, um die Migration von der Konsole aus zu ermöglichen.

Sie können EC2 Datenbanken für die folgenden Datenbank-Engines migrieren:

- Mein SQL
- MariaDB
- Postgret SQL

Der Migrationsprozess umfasst die folgenden Schritte:

- Erstellen Sie eine äquivalente Datenbank in . Damit die Datenbanken gleichwertig sind, müssen sie dieselbe Datenbank-Engine und kompatible Engine-Versionen haben. Sie müssen sich auch in derselben Version befindenVPC. Anweisungen zum Erstellen Ihrer Datenbank finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Wählen Sie den Replikationstyp für Ihre Datenbank aus:
  - Volllastmigration — kopiert die komplette Quelldatenbank in die Zieldatenbank und erstellt bei Bedarf neue Tabellen in der Zieldatenbank.

    > **Note**
    >
    > Diese Option verursacht einen Ausfall in Ihrer .

- Volllast und Change Data Capture (CDC) -Migration — Ähnlich wie bei der Volllastmigration RDSkopiert mit dieser Option die komplette Quelldatenbank in die Zieldatenbank. Nach der Volllastmigration RDSwendet jedoch alle erfassten Änderungen in der Quelle auf die Zieldatenbank an. Change Data Capture erfasst Änderungen an den Datenbankprotokollen mithilfe der systemeigenen Datenbank-EngineAPI.

  > **Note**
  >
  > Diese Option verursacht einen Ausfall in Ihrer .

- Datenerfassung ändern (CDC) — Verwenden Sie diese Option, um Ihre Zieldatenbank während der Migration verfügbar zu halten. migriert laufende Änderungen in Ihrer Quelldatenbank in die Zieldatenbank.

- schafft die notwendigen Netzwerkressourcen, um die Migration zu erleichtern. Sobald die erforderlichen Ressourcen RDSerstellt hat, werden Sie über die erstellten Ressourcen informiert und Sie können die Datenübertragung einleiten.

  Die Zeit, die für den Abschluss der Migration benötigt wird, hängt von der Art der Replikation und der Größe der Quelldatenbank ab.

Voraussetzungen

My SQL und MariaDB

Bevor Sie beginnen, mit einer My SQL - oder MariaDB-Datenbank als Quelldatenbank zu arbeiten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen. Diese Voraussetzungen gelten für AWS-verwaltete Quellen.

Sie müssen über ein Konto verfügen AWS DMS , das die Rolle des Replikationsadministrators innehat. Die Rolle benötigt die folgenden Berechtigungen:

- REPLICATIONCLIENT— Dieses Recht ist nur für CDC Aufgaben erforderlich. Mit anderen Worten, für full-load-only Aufgaben ist dieses Privileg nicht erforderlich.
- REPLICATIONSLAVE— Dieses Recht ist nur für CDC Aufgaben erforderlich. Mit anderen Worten, für full-load-only Aufgaben ist dieses Privileg nicht erforderlich.

Der AWS DMS Benutzer muss außerdem über SELECT Berechtigungen für die Quelltabellen verfügen, die für die Replikation vorgesehen sind.

Gewähren Sie die folgenden Rechte, wenn Sie My SQL -specific Premigration Assessments verwenden.

```
grant select on mysql.user to <dms_user>;
grant select on mysql.db to <dms_user>;
grant select on mysql.tables_priv to <dms_user>;
grant select on mysql.role_edges to <dms_user>  #only for MySQL version 8.0.11 and
 higher
```

Postgret SQL

Gehen Sie vor der Migration von Daten aus einer AWS-verwalteten SQL Postgre-Quelldatenbank wie folgt vor:

- Wir empfehlen, dass Sie ein AWS Benutzerkonto mit den erforderlichen Mindestberechtigungen für die SQL Postgre-DB-Instance als Benutzerkonto für den SQL Postgre-Quellendpunkt verwenden. AWS DMS Von der Verwendung des Hauptkontos wird abgeraten. Das Konto muss über die Rolle `rds_superuser` und die Rolle `rds_replication` verfügen. Die Rolle `rds_replication` erteilt Berechtigungen zur Verwaltung von logischen Slots und zum Streamen von Daten mithilfe von logischen Slots.

> **Note**
>
> Einige AWS DMS Transaktionen sind für einige Zeit inaktiv, bevor die DMS Engine sie erneut verwendet. Wenn Sie den Parameter `idle_in_transaction_session_timeout` in SQL Postgre-Versionen 9.6 und höher verwenden, können Sie dazu führen, dass bei inaktiven Transaktionen ein Timeout auftritt und diese fehlschlagen.

Einschränkungen

Die folgenden Einschränkungen gelten für den Automigrationsprozess:

- Ihr Zieldatenbankstatus muss verfügbar sein, um mit der Quelldatenbankmigration zu beginnen.
- Bei der Migration von einer My SQL Source Database muss Ihr über die Rolle Replication Admin verfügen. Sie müssen auch über die entsprechenden Rechte für diese Rolle verfügen.

- Ihre EC2 Instanz und Ihre Zieldatenbank müssen sich in derselben befindenVPC.
- Sie können Ihre EC2 Datenbank nicht zu den folgenden Zieldatenbanken migrieren, wenn Sie die Aktion Daten aus EC2 Datenbank migrieren verwenden:
  - Datenbank, die Mitglied eines Clusters ist
  - Oracle-, SQL Server- und Db2-Datenbanken
  - Datenbanken mit einer SQL Version unter 5.7
  - Datenbanken mit einer SQL Postgre-Version unter 10.4
  - Datenbanken mit einer MariaDB-Version unter 10.2

Erstellung von IAM-Ressourcen für homogene Migrationen

RDS verwendet AWS DMS , um Ihre Daten zu migrieren. Damit Sie auf Ihre Datenbanken zugreifen und Daten migrieren können, AWS DMS wird eine serverlose Umgebung für homogene Datenmigrationen geschaffen. In dieser Umgebung ist Zugriff auf VPC-Peering, Routing-Tabellen, Sicherheitsgruppen und andere AWS Ressourcen AWS DMS erforderlich. AWS DMS Speichert außerdem Protokolle, Metriken und den Fortschritt für jede Datenmigration in Amazon CloudWatch. Um ein Datenmigrationsprojekt zu erstellen, AWS DMS ist Zugriff auf diese Dienste erforderlich.

AWS DMS Erfordert außerdem Zugriff auf die Geheimnisse, die eine Reihe von Benutzeranmeldeinformationen darstellen, um die Datenbankverbindung für die Quell- und Zielverbindung zu authentifizieren.

> **Note**
>
> Mithilfe der Aktion Daten aus EC2 Instance migrieren können Sie die RDS-Konsole verwenden, um diese IAM-Ressourcen zu generieren. Überspringen Sie diesen Schritt, wenn Sie die von der Konsole generierten IAM-Ressourcen verwenden.

Für diesen Prozess benötigen Sie die folgenden IAM-Ressourcen:

Themen

- Erstellung einer IAM-Richtlinie für homogene Datenmigrationen
- Eine IAM-Rolle für homogene Datenmigrationen erstellen
- Eine geheime Zugriffsrichtlinie und -rolle erstellen
- Eine IAM Rolle für die Verwaltung AWS DMS von Amazon erstellen VPC

Erstellung einer IAM-Richtlinie für homogene Datenmigrationen

In diesem Schritt erstellen Sie eine IAM-Richtlinie, die den Zugriff auf Amazon EC2 und CloudWatch Ressourcen ermöglicht AWS DMS . Anschließend erstellen Sie eine IAM-Rolle und fügen dieser diese Richtlinie an.

Um eine IAM-Richtlinie für die Datenmigration zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie auf der Seite Richtlinie erstellen die Registerkarte JSON aus.
5. Fügen Sie den folgenden JSON-Code in den Editor ein.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ec2:DescribePrefixLists",
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "servicequotas:GetServiceQuota"
            ],
            "Resource": "arn:aws:servicequotas:*:*:vpc/L-0EA8095F"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DescribeLogStreams"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:dms-data-migration-*:log-
stream:dms-data-migration-*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": "arn:aws:ec2:*:*:route-table/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:security-group-rule/*",
                "arn:aws:ec2:*:*:route-table/*",
                "arn:aws:ec2:*:*:vpc-peering-connection/*",
                "arn:aws:ec2:*:*:vpc/*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group-rule/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AcceptVpcPeeringConnection",
                "ec2:ModifyVpcPeeringConnectionOptions"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-peering-connection/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:AcceptVpcPeeringConnection",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        }
    ]
}
```

6. Wählen Sie Weiter: Tags und Weiter: Prüfen aus.
7. Geben Sie unter Name* **`HomogeneousDataMigrationsPolicy`** ein und wählen Sie dann Richtlinie erstellen aus.

Eine IAM-Rolle für homogene Datenmigrationen erstellen

In diesem Schritt erstellen Sie eine IAM-Rolle, die Zugriff auf AWS Secrets Manager Amazon EC2 und CloudWatch bietet.

Um eine IAM-Rolle für Datenmigrationen zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie auf der Seite Vertrauenswürdige Entität auswählen unter Typ der vertrauenswürdigen Entität die Option AWS -Service aus. Unter Anwendungsfälle für andere AWS -Services wählen Sie DMS.
5. Markieren Sie das Kontrollkästchen DMS und wählen Sie Weiter.
6. Wählen Sie auf der Seite „Berechtigungen hinzufügen“ HomogeneousDataMigrationsPolicydie Option aus, die Sie zuvor erstellt haben. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Rollenname festlegen und überprüfen **`HomogeneousDataMigrationsRole`** für Rollenname ein und wählen Sie Rolle erstellen aus.
8. Geben Sie auf der Seite Rollen **`HomogeneousDataMigrationsRole`** für Rollenname ein. Wählen Sie HomogeneousDataMigrationsRole.
9. Wählen Sie auf der HomogeneousDataMigrationsRoleSeite die Registerkarte Vertrauensbeziehungen aus. Wählen Sie Vertrauensrichtlinie bearbeiten aus.
10. Fügen Sie auf der Seite Vertrauensrichtlinie bearbeiten den folgenden JSON-Code in den Editor ein und ersetzen Sie den vorhandenen Text.

    JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "dms-data-migrations.amazonaws.com",
                    "dms.your_region.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
```

```
    ]
}
```

Ersetzen Sie es im vorherigen Beispiel *your_region* durch den Namen Ihres AWS-Region.

Die oben genannte ressourcenbasierte Richtlinie gewährt AWS DMS Service Principals Berechtigungen zur Ausführung von Aufgaben gemäß der vom Kunden verwalteten Richtlinie. HomogeneousDataMigrationsPolicy

11. Wählen Sie Richtlinie aktualisieren.

Eine geheime Zugriffsrichtlinie und -rolle erstellen

Gehen Sie wie folgt vor, um Ihre geheime Zugriffsrichtlinie und Rolle zu erstellen, die es DMS ermöglichen, auf die Benutzeranmeldeinformationen für Ihre Quell- und Zieldatenbanken zuzugreifen.

Um die geheime Zugriffsrichtlinie und Rolle zu erstellen, die Amazon RDS den AWS Secrets Manager Zugriff auf Ihr entsprechendes Geheimnis ermöglicht

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Identity and Access Management (IAM-) Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie Richtlinien und anschließend Richtlinie erstellen aus.
3. Wählen Sie JSON aus und geben Sie die folgende Richtlinie ein, um den Zugriff auf Ihr Secret und dessen Entschlüsselung zu ermöglichen.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:SecretName-ABCDEF"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Hier *secret_arn* ist der ARN Ihres Geheimnisses, den Sie je nach `SecretsManagerSecretId` Bedarf von beiden abrufen können, und *kms_key_arn* der ARN des AWS KMS Schlüssels, mit dem Sie Ihr Geheimnis verschlüsseln, wie im folgenden Beispiel.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "arn:aws:secretsmanager:us-
east-2:123456789012:secret:MySQLTestSecret-qeHamH"
        },
        {
             "Effect": "Allow",
             "Action": [
                        "kms:Decrypt",
                        "kms:DescribeKey"
                      ],
             "Resource": "arn:aws:kms:us-
east-2:123456789012:key/761138dc-0542-4e58-947f-4a3a8458d0fd"
        }
     ]
}
```

> **Note**
>
> Wenn Sie den standardmäßigen Verschlüsselungsschlüssel verwenden AWS Secrets Manager, der von erstellt wurde, müssen Sie die AWS KMS Berechtigungen für *kms_key_arn* nicht angeben.
> Wenn Sie möchten, dass Ihre Richtlinie Zugriff auf beide Geheimnisse gewährt, geben Sie einfach ein zusätzliches JSON-Ressourcenobjekt für das andere an*secret_arn*.

4. Überprüfen und erstellen Sie die Richtlinie mit einem Anzeigenamen und einer optionalen Beschreibung.
5. Wählen Sie Rollen und anschließend Rolle erstellen aus.
6. Wählen Sie als Typ der vertrauenswürdigen Entität AWS -Service aus.
7. Wählen Sie in der Liste der Services DMS als vertrauenswürdigen Service aus und wählen Sie dann Weiter: Berechtigungen aus.
8. Suchen Sie nach der Richtlinie, die Sie in Schritt 4 erstellt haben, und fügen Sie sie an. Fügen Sie dann alle Tags hinzu und überprüfen Sie Ihre Rolle. Bearbeiten Sie jetzt die Vertrauensbeziehungen für die Rolle, um Ihren Amazon RDS Regional Service Principal als vertrauenswürdige Entität zu verwenden. Dieser Prinzipal weist das folgende Format auf:

```
dms.region-name.amazonaws.com
```

Hier ist *region-name* der Name Ihrer Region, z. B. `us-east-1`. Somit folgt ein regionaler Amazon RDS-Servicechef für diese Region.

```
dms.us-east-1.amazonaws.com
dms-data-migrations.amazonaws.com
```

Eine IAM Rolle für die Verwaltung AWS DMS von Amazon erstellen VPC

Sie müssen eine IAM Rolle für erstellen AWS DMS , um die VPC Einstellungen für Ihre Ressourcen zu verwalten. Diese Rolle muss für eine erfolgreiche Migration verfügbar sein.

`dms-vpc-role`Für die Datenbankmigration wird das erstellt

<result>

Dadurch wird die Rolle für die erstelltDMS, um die VPC Einstellungen für die Migration zu verwalten.
</result>

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie die AWS-ServiceOption für die Option Vertrauenswürdige Entität auswählen.

   Wählen Sie für Anwendungsfall die Option DMS.
4. Wählen Sie für den Schritt Berechtigungen hinzufügen die Option `AmazonDMSVPCManagementRole` Weiter aus.
5. Legen Sie auf der Seite Name, Überprüfung und Erstellung den Rollennamen auf fest `dms-vpc-role` und wählen Sie Rolle erstellen aus.

Datenmigration für die EC2 Datenbank einrichten

Um mit der Migration von Daten aus Ihrer EC2 Quelldatenbank zu beginnen, müssen Sie eine entsprechende erstellen. Anweisungen zum Erstellen Ihrer Datenbank finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Nachdem Sie Ihre Zieldatenbank erstellt haben, gehen Sie wie folgt vor, um die Datenmigration einzurichten:

Richten Sie ein Datenmigrationsprojekt ein

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der RDS Konsole aus.
2. Wählen Sie das Drop-down-Menü Aktionen und dann die Option Daten aus EC2 Datenbank migrieren aus. Informationen zu den unterstützten Zieldatenbanken finden Sie unterEinschränkungen.
3. Gehen Sie im Abschnitt EC2 Quelldatenbank auswählen wie folgt vor:

   1. Überprüfen Sie den Engine-Typ und stellen Sie sicher, dass er mit Ihrer Quelldatenbank übereinstimmt.

      Überprüfen Sie außerdem, ob die Engine-Versionen kompatibel sind.
   2. Wählen Sie beispielsweise die EC2EC2Instanz aus, in der sich Ihre Quelldatenbank befindet.

3. Geben Sie für Port den Port ein, auf dem Ihre Quelldatenbank Datenverkehr zulässt.
4. Wählen Sie für Secret die Option Create and use a new secret aus, falls Sie noch keinen Secret haben. Geben Sie den Benutzernamen und das Passwort für Ihre Quelldatenbank ein. Wählen Sie auch den KMS Schlüssel, mit dem Sie Ihr Geheimnis verschlüsseln möchten.

   Wenn Sie über ein vorhandenes Geheimnis verfügen, wählen Sie Bestehendes Geheimnis verwenden und wählen Sie dann in der Drop-down-Liste die Option Geheimnis aus.
5. Wenn Sie über eine bestehende IAMRolle verfügen, wählen Sie unter IAM Rolle für geheim die Option Bestehende IAM Rolle verwenden aus und wählen Sie aus dem Drop-down-Menü eine IAM Rolle aus, die auf die geheime ID aus dem vorherigen Schritt zugreifen kann.

   Wenn Sie noch keine IAM Rolle haben, wählen Sie Neue IAM Rolle erstellen und verwenden aus. Geben Sie den neuen Namen für Ihre Rolle als IAMRollennamen ein. Die mit dieser Rolle verknüpften Berechtigungen finden Sie unter dem folgenden Link.

4. Gehen Sie im Abschnitt RDS Zieldatenbank anzeigen wie folgt vor:

   1. Bestätigen Sie die Einstellungen Ihrer Zieldatenbank oben im Abschnitt.
   2. Wählen Sie unter Geheim die Option Create and use a new secret aus, falls Sie noch kein Secret haben, das Ihre Zieldatenbank-Anmeldeinformationen enthält.

      Wenn Sie über ein vorhandenes Geheimnis verfügen, wählen Sie das Geheimnis aus der Dropdownliste aus.
   3. Wählen Sie IAMunter Rolle für den geheimen Schlüssel eine IAM Rolle aus, die auf den geheimen Schlüssel aus dem vorherigen Schritt zugreifen kann. Sie können auch eine neue IAM Rolle erstellen, wenn Sie noch keine IAM Rolle haben.

      Wenn das Drop-down-Menü die IAM Rollen nicht auffüllt, geben Sie die IAMRolle ARN im Format an. `arn:aws:iam:`*`account_id`*`:role/`*`roleName`*

5. Gehen Sie im Abschnitt Datenmigration konfigurieren wie folgt vor:

   1. Wählen Sie die Art der Datenmigration aus, indem Sie zwischen Volllast, Volllast und Datenänderungserfassung (CDC) oder Datenänderungserfassung (CDC) wählen. Weitere Informationen zu diesen Optionen finden Sie unter Übersicht.

      Sie können den Migrationstyp nicht ändern, nachdem die Migration gestartet wurde.
   2. Wenn Sie bereits über eine IAMRolle für die Datenmigration verfügen, wählen Sie Bestehende IAM Rolle verwenden aus und wählen Sie eine IAM Rolle aus der Dropdownliste aus, DMS die

die Berechtigungen zum Erstellen der für die Migration erforderlichen Ressourcen gewährt. IAM Wenn Sie noch keine IAM Rolle haben, wählen Sie Neue IAM Rolle erstellen und verwenden aus.

6. Vergewissern Sie sich, dass auf der Registerkarte „Migrationseinstellungen anzeigen“ die erforderlichen Einstellungen angezeigt werden, damit Ihre Datenmigration erfolgreich eingerichtet werden kann.
7. Wählen Sie Migrieren aus, um die Einrichtung der Migration abzuschließen.

Nachdem Sie diese Schritte abgeschlossen haben, können Sie sehen, welche Ressourcen für die Datenmigration eingerichtet werden, indem Sie im Fortschrittsbanner in der Konsole auf Details anzeigen klicken. Sobald die erforderlichen Ressourcen eingerichtet sind, wird die Migration automatisch gestartet. Wenn Sie erstellen

Um mehrere Datenbanken in die Zieldatenbank zu migrieren, starten Sie diesen Vorgang erneut mit Details zur neuen EC2 Datenbank.

Verwaltung von Datenmigrationen

Nachdem Sie die Aktion Daten aus EC2 Datenbank migrieren in der RDS Konsole verwendet haben, RDSstartet die Migration automatisch.

Wenn Sie die AWS DMS Konsole zum Erstellen der Migrationsressourcen verwendet haben, können Sie den Migrationsprozess starten.

Starten der Datenmigration

Gehen Sie wie folgt vor, um die Datenmigration zu starten:

Starten einer Datenmigration

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der RDS Konsole aus.
2. Wählen Sie auf der Seite mit den Datenbankdetails den Tab Datenmigrationen aus.
3. Auf der Registerkarte Datenmigrationen werden im Bereich Zugeordnete Datenmigrationen die verfügbaren Datenmigrationen aufgeführt.

   Migrationen, die mit der eingerichtet wurden, werden automatisch gestartet, sobald die erforderlichen Ressourcen eingerichtet sind.

Migrationen, die mit der DMS Konsole eingerichtet wurden, sind auf Bereit gesetzt.

Um mit diesen Migrationen zu beginnen, wählen Sie das Drop-down-Menü Aktionen und dann Start aus.

4. Damit wird die Datenmigration für Ihre EC2 Datenbank gestartet.

Stoppt die Datenmigration

Bei Datenmigrationen, deren Replikationstyp Volllast ist, führt das Beenden der Migration dazu, dass der Prozess gestoppt wird und nicht wieder aufgenommen werden kann. Nach dem Beenden müssen Sie die Migration neu starten.

Bei Migrationen, bei denen der Replikationstyp auf Datenerfassung ändern (CDC) oder Volllast und eingestellt istCDC, können Sie den kontinuierlichen Replikationsprozess beenden und den Vorgang später fortsetzen.

Eine Datenmigration beenden

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der RDS Konsole aus.
2. Wählen Sie auf der Seite mit den Datenbankdetails den Tab Datenmigrationen aus.
3. Auf der Registerkarte Datenmigrationen werden im Bereich Zugeordnete Datenmigrationen die laufenden Datenmigrationen aufgeführt.

   Um eine Migration zu beenden, wählen Sie eine Datenmigration aus und wählen Sie in der Dropdownliste Aktionen die Option Stopp aus.
4. Dadurch wird die Datenmigration für Ihre EC2 Datenbank gestoppt.

Wiederaufnahme der Datenmigration

Bei Datenmigrationen, deren Replikationstyp Volllast und Change Data Capture (CDC) oder Change Data Capture (CDC) -Migration ist, können Sie den CDC Vorgang vom letzten Endpunkt aus fortsetzen.

Wiederaufnahme einer Datenmigration

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der RDS Konsole aus.
2. Wählen Sie auf der Seite mit den Datenbankdetails den Tab Datenmigrationen aus.

3. Auf der Registerkarte Datenmigrationen werden im Bereich Zugeordnete Datenmigrationen die gestoppten Datenmigrationen aufgeführt.

   Um eine Migration fortzusetzen, wählen Sie eine Datenmigration aus und wählen Sie in der Dropdownliste Aktionen die Option Verarbeitung fortsetzen aus.
4. Setzen Sie damit die Datenmigration für Ihre EC2 Datenbank fort.

Löschen der Datenmigration

Gehen Sie wie folgt vor, um eine zugehörige Datenmigration zu löschen

Löschen einer Datenmigration

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der RDS Konsole aus.
2. Wählen Sie auf der Seite mit den Datenbankdetails den Tab Datenmigrationen aus.
3. Um eine Migration zu löschen, wählen Sie eine Datenmigration aus und wählen Sie im Drop-down-Menü Aktionen die Option Löschen aus.
4. Dadurch wird die Datenmigration gelöscht.

Das Löschen einer laufenden Datenmigration hat keine Auswirkungen auf Daten, die bereits in die Zieldatenbank geladen wurden.

Die Datenmigration neu starten

Gehen Sie wie folgt vor, um eine zugehörige Datenmigration von einem CDC Startpunkt aus neu zu starten

Eine Datenmigration neu starten

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der RDS Konsole aus.
2. Wählen Sie auf der Seite mit den Datenbankdetails den Tab Datenmigrationen aus.
3. Um eine Migration neu zu starten, wählen Sie eine Datenmigration aus und wählen Sie im Drop-down-Menü Aktionen die Option Neu starten aus.
4. Dadurch wird die Datenmigration von einem CDC Startpunkt aus neu gestartet.

Der Neustart einer laufenden Datenmigration hat keine Auswirkungen auf Daten, die bereits in die Zieldatenbank geladen wurden.

Überwachung Ihrer Datenmigrationen

Nach dem Start der Datenmigrationen können Sie deren Status und Fortschritt überwachen. Die Durchführung von Datenmigrationen großer Datensätze dauert Stunden. Überwachen Sie den Fortschritt Ihrer Datenmigration regelmäßig, um die Zuverlässigkeit, Verfügbarkeit und hohe Leistung Ihrer Datenmigration aufrechtzuerhalten.

So prüfen Sie den Status und den Fortschritt Ihrer Datenmigration

1. Wählen Sie die Zieldatenbank auf der Seite Datenbanken in der Konsole ausRDS.
2. Wählen Sie auf der Seite mit den Datenbankdetails den Tab Datenmigrationen.
3. Im Abschnitt Verknüpfte Datenmigrationen werden Ihre Datenmigrationen aufgeführt. Überprüfen Sie die Spalte Status.
4. Bei laufenden Datenmigrationen wird in der Spalte Migrationsprozess der Prozentsatz der migrierten Daten angezeigt.
5. Verwenden Sie den Link in der Spalte „In“ CloudWatch, um den Prozess zu überwachen. CloudWatch

Migrationsstatus

Für jede Datenmigration, die Sie ausführen, zeigt die den Status an. Die folgende Liste enthält die Status:

- `Ready`: Die Datenmigration ist startbereit.
- `Starting`: erstellt RDSdie serverlose Umgebung für Ihre Datenmigration.
- `Load running`: führt die Volllastmigration durch.
- `Load complete, replication ongoing`: hat den vollen Ladevorgang abgeschlossen und repliziert nun die laufenden Änderungen. Dieser Status gilt nur für Volllast- und CDC Typmigrationen.
- `Replication ongoing`: repliziert laufende Änderungen. Dieser Status gilt nur für CDC Typmigrationen.
- `Stopping`: stoppt RDSdie Datenmigrationen. Dieser Status gilt, wenn Sie die Datenmigration im Menü Aktionen beenden möchten.
- `Stopped`: hat die Datenmigration gestoppt.

- `Failed`: Die Datenmigration ist fehlgeschlagen. Weitere Informationen finden Sie in den Protokolldateien.
- `Restarting`: Die Datenmigration hat eine laufende Datenreplikation von einem CDC Startpunkt aus neu gestartet.

Tutorial: Erstellen eines mit einem benutzerdefinierten Parameter und einer neuen Optionsgruppe

In diesem Tutorial erstellen Sie einen mit einem benutzerdefinierten Parameter und einer neuen Optionsgruppe. Weitere Informationen zu benutzerdefinierten Parameter - und Optionsgruppen finden Sie unter Parametergruppen für Amazon RDSundArbeiten mit Optionsgruppen.

> **Important**
>
> Für die Erstellung eines AWS Kontos fallen keine Gebühren an. Wenn Sie dieses Tutorial abschließen, können Ihnen jedoch Kosten für die von Ihnen verwendeten AWS Ressourcen entstehen. Sie können diese Ressourcen nach Abschluss des Tutorials löschen, wenn sie nicht mehr benötigt werden.

Um einen mit benutzerdefinierten Konfigurationen und Einstellungen zu erstellen, können Sie benutzerdefinierte Parameter und neue Optionsgruppen verwenden. Benutzerdefinierte Parameter und neue Optionsgruppen sind besonders hilfreich, wenn Sie mit mehreren Datenbanken arbeiten und die Einstellungen für diese einheitlich konfigurieren möchten.

Wenn Sie diese Schritte ausführen, lernen Sie Folgendes:

- So verwenden Sie Amazon RDS Amazon , um einen mit einem benutzerdefinierten Parameter und einer neuen Optionsgruppe zu erstellen.
- So verwenden Sie spezifische Parameter und Optionen für .

Führen Sie für dieses Tutorial die folgenden Aufgaben aus:

1. Erstellen Sie eine benutzerdefinierte mit dem und `disconnect_on_expired_password`.
2. Erstellen Sie eine neue Optionsgruppe mit der MySQL-OptionsfunktionMariaDB `Audit Plugin`. Anweisungen zum Erstellen einer Optionsgruppe finden Sie unterArbeiten mit Optionsgruppen.
3. Erstellen Sie einen mit der benutzerdefinierten und der neuen Optionsgruppe, die Sie erstellt haben.

Themen

- Voraussetzungen

- Eine Amazon RDS Amazon erstellen
- Ändern Sie den der Parameterwerte in Ihrer benutzerdefinierten Parametergruppe
- Eine neue Amazon RDS-Optionsgruppe erstellen
- Fügen Sie Ihrer neuen Optionsgruppe eine Option hinzu
- Erstellen Sie einen für eine MySQL-DB-Instance mit einem benutzerdefiniertenund einer neuen Optionsgruppe

Voraussetzungen

Für dieses Tutorial benötigen Sie einen AWS-Konto und einen Benutzer mit Administratorzugriff. Wenn Sie diese noch nicht eingerichtet haben, führen Sie die Schritte in den folgenden Abschnitten aus:

- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Eine Amazon RDS Amazon erstellen

Der Parameter steuert die Anzahl der Tage, bevor das Client-Passwort automatisch abläuft. `default_password_lifetime` Der `disconnect_on_expired_password` Parameter steuert, ob die MySQL-DB-Instance die Verbindung zum Client trennt, wenn das Passwort abläuft. Weitere Informationen zu anderen Parametern, die für verfügbar sind, finden Sie unter Parameter für My SQL.

So erstellen Sie eine Parametergruppe:

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie Parametergruppen aus.
2. Wählen Sie für Benutzerdefinierte Parametergruppen die Option Parametergruppe erstellen aus.
3. Legen Sie die Details der Parametergruppe fest.
    1. Geben Sie einen Namen für die Parametergruppe ein.
    2. Geben Sie eine Beschreibung der Parametergruppe ein.
    3. Wählen Sie als Engine-Typ die Option MySQL Community aus.
    4. Wählen Sie für Parametergruppenfamilie MySQL 8.0 aus.
4. Wählen Sie Create (Erstellen) aus.

Die neue wird auf der Seite Parametergruppen in der Amazon RDS-Konsole angezeigt. Die folgenden Schritte veranschaulichen, wie Sie Parameterwerte ändern, um Ihre Parametergruppe anzupassen.

Ändern Sie den der Parameterwerte in Ihrer benutzerdefinierten Parametergruppe

Gehen Sie wie folgt vor, um die in der Parametergruppe zu ändern, in der Sie sie erstellt habenEine Amazon RDS Amazon erstellen.

So ändern Sie Parameterwerte in Ihrer Parametergruppe

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie Parametergruppen aus.
2. Wählen Sie für Benutzerdefinierte Parametergruppen den Namen der , die Sie erstellt haben.
3. Wählen Sie Edit (Bearbeiten) aus.
4. Suchen Sie im Suchfeld Filterparameter nach dem benutzerdefinierten Parameter`default_password_lifetime`.
5. Aktivieren Sie das Kontrollkästchen neben dem Parameter und geben Sie einen Wert für die Anzahl der Tage ein, die für diesen Parameter für die Gültigkeitsdauer des Kennworts festgelegt werden sollen.
6. Wählen Sie Save Changes (Änderungen speichern).
7. Wiederholen Sie dieselben Schritte für den Parameter `disconnect_on_expired_password`. Wenn Sie diesen Parameter auswählen, werden Sie aufgefordert, einen Wert von 0 oder 1 aus dem Dropdownmenü auszuwählen. Wählen Sie 1, um die Verbindung zu trennen, wenn das Passwort abgelaufen ist.

Die benutzerdefinierte Parametergruppe kann jetzt mit der Amazon RDS Amazon for MySQL 8.0-DB-Instance verknüpft werden. Erstellen Sie als Nächstes eine neue Optionsgruppe für Ihre DB-Instance.

Eine neue Amazon RDS-Optionsgruppe erstellen

Erstellen Sie eine neue Optionsgruppe mit der Option MariaDB Audit Plugin. Dieses Plugin protokolliert Serveraktivitäten aus Sicherheits- und Compliance-Gründen. Weitere Informationen zu anderen benutzerdefinierten Optionen, die für MySQL-DB-Instances verfügbar sind, finden Sie unterOptionen für Meine SQL DB-Instances.

So erstellen Sie eine Optionsgruppe

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie Optionsgruppen.
2. Wählen Sie für Optionsgruppen die Option Gruppe erstellen aus.
3. Legen Sie die Details der Optionsgruppe fest.
   - Geben Sie einen Namen für die Optionsgruppe ein.
   - Geben Sie eine Beschreibung der Optionsgruppe ein.
   - Wählen Sie für Engine die Option mysql aus.
   - Wählen Sie für Major Engine-Version 8.0 aus.
4. Wählen Sie Create (Erstellen) aus.

Die neue Optionsgruppe wird auf der Seite Optionsgruppen in der Amazon RDS-Konsole angezeigt. Die folgenden Schritte zeigen, wie Sie der Optionsgruppe Optionen hinzufügen.

Fügen Sie Ihrer neuen Optionsgruppe eine Option hinzu

Gehen Sie wie folgt vor, um der neuen Optionsgruppe, in der Sie sie erstellt haben, eine Option hinzuzufügenEine neue Amazon RDS-Optionsgruppe erstellen.

So fügen Sie Ihrer Optionsgruppe eine Option hinzu

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie Optionsgruppen.
2. Wählen Sie für Optionsgruppen den Namen der Optionsgruppe aus, die Sie erstellt haben.
3. Wählen Sie unter Optionen die Option Hinzufügen aus.
4. Legen Sie die Details der Optionsgruppe fest.
   - Wählen Sie als Optionsname die Option MariaDB Audit Plugin, MARIADB_AUDIT_PLUGIN.
   - Lassen Sie für Optionseinstellungen alle Standardoptionen ausgewählt.
   - Wählen Sie für Sofort anwenden die Option Ja aus.
5. Wählen Sie die Option Erstellen.

Die neue Optionsgruppe sollte jetzt für alle zugehörigen DB-Instances verfügbar sein. Als Nächstes erstellen Sie eine MySQL-DB-Instance mit dem benutzerdefinierten Parameter und der neuen Optionsgruppe.

Erstellen Sie einen für eine MySQL-DB-Instance mit einem benutzerdefiniertenund einer neuen Optionsgruppe

Erstellen Sie abschließend einen mit dem benutzerdefinierten Parameter und der neuen Optionsgruppe, die Sie in den vorherigen Schritten erstellt haben. Die folgenden Schritte zeigen, wie Sie den mit Ihrem benutzerdefinierten Parameter und der neuen Optionsgruppe erstellen.

Um einen mit einem benutzerdefinierten Parameter und einer neuen Optionsgruppe zu erstellen

1. Öffnen Sie die Amazon RDS-Konsole und wählen Sie Datenbanken.
2. Wählen Sie Datenbank erstellen aus.
3. Wählen Sie für Wählen Sie eine Methode zur Datenbankerstellung die Option Standard Create aus.
4. Wählen Sie für Engine-Optionen MySQL .
5. Wählen Sie für Verfügbarkeit und Haltbarkeit die Option Einzelne DB-Instance aus. Dieser Schritt ist erforderlich, um einen benutzerdefinierten Parameter oder eine neue Optionsgruppe zu unterstützen.
6. Wählen Sie Zusätzliche Konfiguration aus.
   - Wählen Sie unter Ursprünglicher Datenbankname einen Namen für Ihren aus.
   - Wählen Sie in der Dropdownliste den Namen der benutzerdefinierten aus, die Sie zuvor erstellt haben.
   - Wählen Sie im Dropdownmenü „Optionsgruppe“ den Namen der neuen Optionsgruppe aus, die Sie zuvor erstellt haben.
7. Für dieses Tutorial können Sie die Standardeinstellungen für alle anderen DB-Einstellungen beibehalten oder sie nach Bedarf ändern.
8. Wählen Sie Datenbank erstellen aus.

RDS erstellt einen neuen für MySQL-DB-Instances mit einer benutzerdefinierten Parametergruppe und einer neuen Optionsgruppe. Weitere Informationen zu dieser Datenbank finden Sie auf der Datenbankseite der Amazon RDS-Konsole.

In diesem Tutorial haben Sie einen mit maßgeschneiderten Einstellungen unter Verwendung eines benutzerdefinierten Parameters und einer neuen Optionsgruppe konfiguriert.
Dieser neu erstellte verwaltet die Lebensdauer des Benutzerkennworts mithilfe des Parameters`default_password_lifetime`. Dieser trennt auch Benutzer, die mithilfe des

Parameters `disconnect_on_expired_password` eine Verbindung mit einem abgelaufenen Passwort herstellen. Sie verwenden die Option auch`MariaDB Audit Plugin`, um die Serveraktivität zu verfolgen. Um Ihre Datenbank zu optimieren, können Sie zusätzliche Einstellungen in Ihrer benutzerdefinierten Parametergruppe vornehmen und Optionen hinzufügen.

Nachdem Sie die Erstellung Ihres benutzerdefinierten abgeschlossen haben, sollten Sie Ihre Ressourcen löschen, um unerwünschte Kosten zu vermeiden. Um einen zu löschen, folgen Sie den Anweisungen unter. Löschen einer DB-Instance

Verwaltung einer Amazon RDS DB-Instance

Im Folgenden finden Sie Anweisungen zur Verwaltung und Wartung Ihrer Amazon RDS DB-Instance.

Themen

- Eine Amazon RDS-DB-Instance temporär stoppen
- Starten einer Amazon RDS DB-Instance, die zuvor gestoppt wurde
- Neustarten einer DB-Instance
- Automatisches Verbinden einer EC2 Instance und einer DB-Instance mit
- Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance
- Ändern einer Amazon RDS DB-Instance
- Warten einer DB-Instance
- Upgrade der Engine-Version für eine DB-Instance
- Umbenennen einer DB-Instance
- Arbeiten mit DB-Instance-Lesereplikaten
- Verschlagwortung von Amazon-Ressourcen RDS
- Amazon-Ressourcennamen (ARNs) in Amazon RDS
- Arbeiten mit Speicher für Amazon RDS DB-Instances
- Löschen einer DB-Instance
- Tutorial: Verwaltung einer My SQL DB-Instance-Umgebung von der Entwicklung bis zur Produktion

Eine Amazon RDS-DB-Instance temporär stoppen

Sie können Ihre DB-Instance für temporäre Tests oder für eine tägliche Entwicklungsaktivität zeitweise für maximal 7 aufeinanderfolgende Tage anhalten. Der häufigste Anwendungsfall ist die Kostenoptimierung.

Wie lange es dauert, bis Ihre DB-Instance gestoppt wird, hängt von Faktoren wie Instance-Klasse, Netzwerkstatus, DB-Engine-Typ und Datenbankstatus ab. Der Vorgang kann mehrere Minuten dauern. Der Dienst muss die folgenden Aktionen ausführen:

- Fahren Sie die Datenbank-Engine-Prozesse herunter.
- Fahren Sie die RDS-Plattformprozesse herunter.
- Trennen Sie die EBS-Speichervolumes, die Ihrer DB-Instance zugeordnet sind.
- Beenden Sie die zugrunde liegende EC2 Amazon-Instance.

> **Warning**
>
> Das Starten einer DB-Instance erfordert die Wiederherstellung der Instance und kann zwischen Minuten und Stunden dauern. Wenn die Verfügbarkeit von Instances ein Problem darstellt, sollten Sie daher vorsichtig sein, wenn Sie eine Produktionsinstanz vorübergehend anhalten. Weitere Informationen finden Sie unter Starten einer Amazon RDS DB-Instance, die zuvor gestoppt wurde.

Um Ihre DB-Instance im selben Vorgang zu beenden und zu starten, starten Sie die DB-Instance neu. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Themen

- Anwendungsfälle für das Stoppen Ihrer DB-Instance
- Unterstützte DB-Engines, Instance-Klassen und Regionen
- Stoppen einer DB-Instance in einer Multi-AZ-Bereitstellung
- So funktioniert das Stoppen einer DB-Instance
- Einschränkungen beim Stoppen Ihrer DB-Instance
- Überlegungen zu Options- und Parametergruppen
- Überlegungen zu öffentlichen IP-Adressen

- Vorübergehendes Stoppen einer DB-Instance: grundlegende Schritte

Anwendungsfälle für das Stoppen Ihrer DB-Instance

Das Stoppen und Starten einer DB-Instance ist schneller als das Erstellen eines DB-Snapshots, das Löschen Ihrer DB-Instance und das anschließende Wiederherstellen des Snapshots, wenn Sie auf die Instance zugreifen möchten. Zu den häufigsten Anwendungsfällen für das Stoppen einer Instance gehören die folgenden:

- Kostenoptimierung — Für Datenbanken, die nicht zur Produktion gehören, können Sie Ihre Amazon RDS-DB-Instance vorübergehend anhalten, um Geld zu sparen. Solange die Instance gestoppt ist, werden Ihnen keine DB-Instance-Stunden in Rechnung gestellt.

  > **Important**
  >
  > Während Ihre DB-Instance angehalten wird, werden nur Gebühren für bereitgestellten Speicher in Rechnung gestellt (einschließlich bereitgestellter IOPS). Es werden Ihnen auch Gebühren für den Backup-Speicher berechnet, einschließlich manuelle Snapshots und automatische Backups innerhalb des von Ihnen festgelegten Aufbewahrungsfensters. Für DB-Instance-Stunden werden Ihnen jedoch keine Gebühren in Rechnung gestellt. Weitere Informationen finden Sie unter Abrechnung FAQs.

- Tägliche Entwicklung — Wenn Sie eine DB-Instance zu Entwicklungszwecken verwalten, können Sie die Instance starten, wenn sie benötigt wird, und die Instance dann herunterfahren, wenn sie nicht benötigt wird.
- Testen — Möglicherweise benötigen Sie eine temporäre DB-Instance, um Sicherungs- und Wiederherstellungsverfahren, Migrationen, Anwendungsupgrades oder ähnliche Aktivitäten zu testen. In diesen Anwendungsfällen können Sie die DB-Instance beenden, wenn sie nicht benötigt wird.
- Schulung — Wenn Sie eine Schulung in RDS durchführen, müssen Sie möglicherweise DB-Instances während der Schulungssitzung starten und anschließend herunterfahren.

Unterstützte DB-Engines, Instance-Klassen und Regionen

Amazon-RDS-DB-Instances, die auf den folgenden DB-Engines ausgeführt werden, können gestoppt und gestartet werden:

- Db2
- MariaDB
- Microsoft SQL Server, einschließlich RDS Custom für SQL Server
- MySQL
- Oracle
- PostgreSQL

Das Anhalten und Starten einer DB-Instance wird für alle Arten von DB-Instance-Klassen und in allen AWS -Regionen unterstützt.

Stoppen einer DB-Instance in einer Multi-AZ-Bereitstellung

Sie können eine DB-Instance in einer Multi-AZ-Bereitstellung beenden und starten. Es gelten die folgenden Einschränkungen:

- Sie können eine Multi-AZ-Bereitstellung nur erstellen, wenn Ihre Datenbank-Engine dies unterstützt. Weitere Informationen zur Engine-Unterstützung finden Sie unterUnterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS.
- RDS für SQL Server unterstützt das Stoppen einer DB-Instance in einer Multi-AZ-Bereitstellung nicht. Weitere Informationen finden Sie unter Einschränkungen, Hinweise und Empfehlungen für Microsoft SQL Server Multi-AZ-Bereitstellung.
- Das Stoppen einer DB-Instance kann viel Zeit in Anspruch nehmen. Wenn Sie nach einem vorherigen Failover mindestens ein Backup haben, können Sie den Stoppvorgang beschleunigen, indem Sie einen Neustart mit Failover-Vorgang durchführen. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

So funktioniert das Stoppen einer DB-Instance

Das Stoppen erfolgt in den folgenden Phasen:

1. Die DB-Instance initiiert den normalen Vorgang zum Herunterfahren.

   Der Status der DB-Instance ändert sich in `stopping`.
2. Die Instance wird an maximal 7 aufeinanderfolgenden Tagen nicht mehr ausgeführt.

   Der Status der DB-Instance ändert sich in `stopped`.

Eigenschaften einer gestoppten DB-Instance

In einem gestoppten Zustand weist Ihre DB-Instance die folgenden Merkmale auf:

- Ihre gestoppte DB-Instance behält Folgendes bei:
  - Instance-ID
  - Domain Name Server (DNS)-Endpunkt
  - Parametergruppe
  - Sicherheitsgruppe
  - Option group
  - Amazon S3 S3-Transaktionsprotokolle (für eine point-in-time Wiederherstellung erforderlich)

  Wenn Sie eine DB-Instance neu starten, weist sie die gleiche Konfiguration wie zum Zeitpunkt des Stoppens auf.
- Speicher-Volumes bleiben an die DB-Instance angehängt und die Daten bleiben erhalten. RDS löscht alle im RAM der DB-Instance gespeicherten Daten.

  Während Ihre DB-Instance angehalten wird, werden nur Gebühren für bereitgestellten Speicher in Rechnung gestellt (einschließlich bereitgestellter IOPS). Es werden Ihnen auch Gebühren für den Backup-Speicher berechnet, einschließlich manuelle Snapshots und automatische Backups innerhalb des von Ihnen festgelegten Aufbewahrungsfensters.
- RDS entfernt ausstehende Aktionen, einschließlich geplanter Wartungsupdates, mit Ausnahme von ausstehenden Aktionen für die Optionsgruppe oder DB-Parametergruppe der DB-Instance.

> **Note**
>
> Gelegentlich wird eine RDS-for-PostgreSQL-DB-Instance nicht ordnungsgemäß heruntergefahren. In diesem Fall durchläuft die Instance bei einem späteren Neustart einen Wiederherstellungsprozess. Dies ist das erwartete Verhalten der Datenbank-Engine, das die Integrität der Datenbank schützen soll. Einige speicherbasierte Statistiken und Zähler behalten den Verlauf nicht bei und werden nach dem Neustart neu initialisiert, um die betriebliche Workload zu erfassen.

Automatischer Neustart einer gestoppten DB-Instance

Wenn Sie Ihre DB-Instance nicht manuell starten, nachdem sie an sieben aufeinanderfolgenden Tagen gestoppt war, startet RDS die DB-Instance automatisch. Auf diese Weise fällt Ihre Instance nicht hinter die erforderlichen Wartungsupdates zurück. Informationen dazu, wie Sie Ihre Instance nach einem Zeitplan stoppen und starten können, finden Sie unter Wie kann ich Step Functions verwenden, um eine Amazon-RDS-Instance für mehr als 7 Tage zu stoppen?.

Einschränkungen beim Stoppen Ihrer DB-Instance

Im Folgenden sind einige Einschränkungen des Stoppvorgangs aufgeführt:

- Sie können eine DB-Instance nicht anhalten, wenn sie ein Lesereplikat aufweist oder eine Lesereplikat ist.
- Sie können keine angehaltene DB-Instance ändern.
- Sie können keine Optionsgruppe löschen, die einer gestoppten DB-Instance zugeordnet ist.
- Sie können keine DB-Parametergruppe löschen, die einer angehaltenen DB-Instance zugeordnet ist.
- Beachten Sie bei einer Multi-AZ-Bereitstellung die folgenden Einschränkungen:
  - Sie können eine RDS for SQL Server-DB-Instance nicht stoppen.
  - Die primären und sekundären Availability Zones werden möglicherweise gewechselt, nachdem Sie die DB-Instance gestartet haben.

Es gelten zusätzliche Einschränkungen für RDS Custom für SQL Server. Weitere Informationen finden Sie unter Starten und Stoppen einer RDS Custom for SQL Server-DB-Instance.

Überlegungen zu Options- und Parametergruppen

Sie können keine persistenten Optionen (einschließlich permanenter Optionen) aus einer Optionsgruppe entfernen, wenn der betreffenden Optionsgruppe DB-Instances zugeordnet sind. Diese Funktionalität gilt auch für angehaltene DB-Instances mit dem Status `stopping`, `stopped` oder `starting`.

Sie können die Options- oder DB-Parametergruppe ändern, die einer angehaltenen DB-Instance zugeordnet ist. Die Änderung wird jedoch erst wirksam, wenn Sie die DB-Instance das nächste Mal starten. Wenn Sie die umgehende Anwendung der Änderungen wählen, erfolgt die Änderung

beim nächsten Starten der DB-Instance. Ansonsten erfolgt die Änderung während des nächsten Wartungsfensters, nachdem Sie die DB-Instance gestartet haben.

Überlegungen zu öffentlichen IP-Adressen

Wenn Sie eine DB-Instance anhalten, bewahrt sie ihren DNS-Endpunkt. Wenn Sie eine DB-Instance mit einer öffentlichen IP-Adresse anhalten, gibt Amazon RDS seine öffentliche IP-Adresse frei. Wenn die DB-Instance neu gestartet wird, hat sie eine andere öffentliche IP-Adresse.

> **Note**
>
> Sie sollten die Verbindung zu einer DB-Instance immer dem DNS-Endpunkt herstellen, nicht mit der IP-Adresse.

Vorübergehendes Stoppen einer DB-Instance: grundlegende Schritte

Sie können eine DB mithilfe der AWS Management Console AWS CLI, der oder der RDS-API stoppen.

Konsole

So halten Sie eine DB-Instance an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance aus, die Sie anhalten möchten.
3. Wählen Sie für Actions (Aktionen) die Option Stop temporarily (Temporär anhalten) aus.
4. Wählen Sie im Fenster Stop DB instance temporarily (DB-Instance vorübergehend anhalten) die Bestätigung aus, dass die DB-Instance nach 7 Tagen automatisch neu gestartet wird.
5. (Optional) Wählen Sie Save the DB instance in a snapshot (Die DB-Instance in einem Snapshot speichern) aus und geben Sie den Snapshot-Namen im Feld Snapshot name (Snapshot-Name) ein. Wählen Sie diese Option aus, wenn Sie einen Snapshot der DB-Instance erstellen möchten, bevor Sie sie anhalten.
6. Klicken Sie auf Stop temporarily (Temporär anhalten), um die DB-Instance anzuhalten, oder auf Cancel (Abbrechen), wenn Sie den Vorgang abbrechen möchten.

AWS CLI

Um eine DB-Instance mithilfe von zu beenden AWS CLI, rufen Sie den stop-db-instanceBefehl mit der folgenden Option auf:

- `--db-instance-identifier`: der Name der DB-Instance

Example

```
aws rds stop-db-instance --db-instance-identifier mydbinstance
```

RDS-API

Um eine DB-Instance mithilfe der Amazon RDS-API zu stoppen, rufen Sie den DBInstanceStop-Vorgang mit dem folgenden Parameter auf:

- `DBInstanceIdentifier`: der Name der DB-Instance

Starten einer Amazon RDS DB-Instance, die zuvor gestoppt wurde

In diesem Thema wird erklärt, wie eine Amazon RDS DB-Instance gestartet wird, die zuvor gestoppt wurde, und es werden die notwendigen Schritte und wichtigsten Überlegungen für die Wiederaufnahme der Nutzung der Datenbank beschrieben.

Sie können Ihre Amazon RDS DB-Instance vorübergehend beenden, um Geld zu sparen. Nachdem Sie die DB-Instance angehalten haben, können Sie sie neu starten und weiter verwenden. Weitere Informationen zum Stoppen einer DB-Instance finden Sie unterEine Amazon RDS-DB-Instance temporär stoppen.

Wenn Sie eine DB-Instance starten, die Sie zuvor gestoppt haben, speichert die DB-Instance Informationen wie die folgenden:

- Instance-ID
- Endpunkt des Domainnamenservers (DNS)
- DB-Parametergruppe
- VPCSicherheitsgruppe
- DB-Optionsgruppe

Amazon RDS berechnet Datenbank-Instances und angehängten Speicher in Sekundenschritten. Beim Start einer Instance fällt eine Mindestgebühr von 10 Minuten an.

Um die Instance zu starten, führt der RDS Amazon-Service Aktionen wie die folgenden aus:

- Bereitstellung der zugrunde liegenden Amazon-Instance EC2
- Die RDS Prozesse starten
- Die Datenbank-Engine-Prozesse werden gestartet
- Anhängen der EBS Speichervolumes
- Performance Insights aktivieren, falls es zuvor aktiviert war
- Wiederherstellung der DB-Instance (die Wiederherstellung erfolgt auch nach einem normalen Herunterfahren)

Die Zeit bis zum Start Ihrer DB-Instance hängt von Faktoren wie der Instance-Klasse, dem Netzwerkstatus, dem DB-Engine-Typ, der Datenbankgröße und dem Datenbankstatus ab, als die Instance heruntergefahren wurde. Der Startvorgang kann Minuten bis Stunden dauern. Wir

empfehlen Ihnen, bei der Erstellung Ihres Verfügbarkeitsplans die Variabilität der Startzeit zu berücksichtigen.

Konsole

So starten Sie eine DB-Instance

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance aus, die Sie starten möchten.
3. Wählen Sie für Actions (Aktionen) die Option Start.

AWS CLI

Um eine DB-Instance mit dem zu starten AWS CLI, rufen Sie den start-db-instanceBefehl mit der folgenden Option auf:

- `--db-instance-identifier` – der Name der DB-Instance

Example

```
aws rds start-db-instance --db-instance-identifier mydbinstance
```

RDS API

Um eine DB-Instance mithilfe von Amazon zu starten RDSAPI, rufen Sie die tartDBInstanceS-Operation mit dem folgenden Parameter auf:

- `DBInstanceIdentifier`: der Name der DB-Instance

Neustarten einer DB-Instance

Sie können den Datenbankdienst auf Ihrer RDS DB-Instance in einem einzigen Vorgang beenden und starten, der als Neustart bezeichnet wird. Ein Neustart kann erforderlich sein, um Konfigurationsänderungen vorzunehmen, kleinere Probleme zu beheben oder Netzwerkprobleme zu lösen, ohne dass Sie Ihre Datenbank vollständig neu starten oder migrieren müssen.

> **Note**
>
> Dieses Thema bezieht sich nur auf den Neustart einer Single-AZ- oder Multi-AZ-DB-Instance. Anweisungen zum Neustart eines Multi-AZ-DB-Clusters finden Sie unter. the section called “Neustarten von Multi-AZ-DB-Clustern”

Themen

- Anwendungsfälle für den Neustart einer DB-Instance einem DB-Cluster
- Wie funktioniert der Neustart einer DB-Instance DB-Cluster
- So funktioniert der Neustart einer DB-Instance in einer Multi-AZ-Bereitstellung
-
-
- Neustarten einer DB-Instance : grundlegende Schritte

Anwendungsfälle für den Neustart einer DB-Instance einem DB-Cluster

In der Regel starten Sie Ihre DB-Instance aus Wartungsgründen neu, damit Ihre Änderungen wirksam werden. Die folgenden Anwendungsfälle sind häufig:

- Zuordnen einer neuen DB-Parametergruppe — Wenn Sie einer DB-Instance eine neue DB-Parametergruppe zuordnen, werden die geänderten statischen und dynamischen Parameter erst nach dem Neustart der DB-Instance RDS angewendet. Wenn Sie jedoch dynamische Parameter in der DB-Parametergruppe ändern, nachdem Sie sie der DB-Instance zugeordnet haben, werden diese Änderungen sofort und ohne Neustart übernommen. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
- Anwenden einer Änderung auf einen statischen Parameter in einer vorhandenen DB-Parametergruppe — Wenn Sie einen statischen Parameter ändern und die DB-Parametergruppe

speichern, ändert sich der Status der mit dieser Parametergruppe verknüpften DB-Instances in der Konsole auf pending-reboot. Die Parameteränderung wird erst wirksam, nachdem die zugehörigen DB-Instances neu gestartet wurden. Wenn Sie einen dynamischen Parameter in einer vorhandenen Parametergruppe ändern, wird die Änderung standardmäßig sofort wirksam, ohne dass ein Neustart erforderlich ist.

> **Note**
>
> Der Status „Ausstehender Neustart“ führt nicht zu einem automatischen Neustart während des nächsten Wartungsfensters. Um die neuesten Parameteränderungen auf Ihre DB-Instance anzuwenden, starten Sie die DB-Instance manuell neu. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

- Fehlerbehebung — Möglicherweise treten Leistungs- oder andere Betriebsprobleme auf, die einen Neustart erforderlich machen. Beispielsweise reagiert Ihre DB-Instance möglicherweise nicht.

Wie funktioniert der Neustart einer DB-Instance DB-Cluster

Wenn Amazon Ihre DB-Instance RDS neu startet, führt es die folgenden sequentiellen Aufgaben aus:

1. Stoppt den Datenbankdienst auf Ihrer DB-Instance
2. Startet den Datenbankdienst auf Ihrer DB-Instance

Der Neustartvorgang führt zu einem kurzen Ausfall. Während dieses Ausfalls lautet der Status der DB-Instance „Neustart“. Ein Ausfall tritt sowohl bei einer Single-AZ-Bereitstellung als auch bei einer Multi-AZ-Bereitstellung von DB-Instances auf, selbst wenn Sie mit einem Failover neu starten.

So funktioniert der Neustart einer DB-Instance in einer Multi-AZ-Bereitstellung

Wenn sich die Amazon RDS DB-Instance in einer Multi-AZ-Bereitstellung befindet, können Sie einen Neustart mit einem Failover durchführen. Dieser Vorgang ist nützlich, um einen Ausfall einer DB-Instance zu simulieren oder den Betrieb nach einem Failover in der ursprünglichen Availability Zone wiederherzustellen.

Während des Neustarts mit Failover RDS geht Amazon wie folgt vor

- Unterbricht die Datenbank abrupt. Die DB-Instance und ihre Client-Sitzungen haben möglicherweise keine Zeit, um ordnungsgemäß herunterzufahren.

  > **Warning**
  >
  > Um die Möglichkeit eines Datenverlusts auszuschließen, empfehlen wir, Transaktionen auf Ihrer DB-Instance anzuhalten, bevor Sie mit einem Failover neu starten.

- Führt bei Bedarf eine Wiederherstellung der Datenbank nach einem Absturz durch.
- Wechselt automatisch zu einem Standby-Replikat in einer anderen AZ. Die AZ-Änderung spiegelt sich möglicherweise einige Minuten lang nicht in der und in Aufrufen an AWS CLI und RDS API wider. AWS Management Console
- Aktualisiert den DNS Datensatz für die DB-Instance so, dass er auf die Standby-DB-Instance verweist. Als Ergebnis müssen Sie alle bestehenden Verbindungen zu Ihrer DB-Instance bereinigen und neu erstellen. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.
- Erzeugt nach dem Neustart ein RDS Amazon-Event.

RDSBei Microsoft SQL Server wird beim Failover nur die primäre DB-Instance neu gestartet. Nach dem Failover wird die primäre DB-Instance zur neuen sekundären DB-Instance. Die Parameter werden für Multi-AZ-Instances möglicherweise nicht aktualisiert. Für einen Neustart ohne Failover starten sowohl die primäre als auch die sekundäre DB-Instance neu. Die Parameter werden nach dem Neustart aktualisiert. Wenn die DB-Instance nicht reagiert, empfehlen wir einen Neustart ohne Failover.

Bevor Sie Ihre Instance neu starten, sollten Sie Folgendes beachten:

- Bei einer DB-Instance mit Lesereplikaten können Sie die Quell-DB-Instance und ihre Lesereplikate unabhängig voneinander neu starten. Nach Abschluss eines Neustarts wird die Replikation automatisch fortgesetzt.
- Die Neustartzeit hängt vom Wiederherstellungsprozess nach einem Absturz, der Datenbankaktivität zum Zeitpunkt des Neustarts und dem Verhalten Ihrer spezifischen DB-Engine ab. Um die Neustartzeit zu verkürzen, empfehlen wir, die Datenbankaktivität während des Neustarts so weit wie möglich zu reduzieren. Diese Technik reduziert die Rollback-Aktivität für Transaktionen während der Übertragung.

Stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Ihre DB-Instance muss sich im Status `available` befinden. Ihre Datenbank kann aus verschiedenen Gründen nicht verfügbar sein, z. B. aufgrund eines laufenden Backups, einer zuvor angeforderten Änderung oder aufgrund eines Wartungsfensters.
- Wenn Sie einen Failover auf eine andere AZ erzwingen, muss Ihre DB-Instance für Multi-AZ konfiguriert sein.
- Wenn Sie einen Failover auf eine andere AZ erzwingen, empfehlen wir, zunächst die Transaktionen auf Ihrer DB-Instance zu stoppen, um einen möglichen Datenverlust zu verhindern.

Neustarten einer DB-Instance : grundlegende Schritte

Sie können Ihre DB-Instance mit AWS Management Console AWS CLI, oder neu starten. RDS API

Konsole

Neustarten einer DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance aus, die Sie neu starten möchten.
3. Wählen Sie unter Aktionen die Option Neustart aus.

   Die Seite DB-Instance neu starten wird angezeigt.
4. (Optional) Wählen Sie Reboot with failover? (Neustart mit Failover?) aus, um ein Failover von einer AZ zu einer anderen zu erzwingen.
5. Wählen Sie Neustart aus, um Ihrer DB-Instance neu zu starten.

   Alternativ können Sie Cancel (Abbrechen) aufrufen.

AWS CLI

Rufen Sie den `reboot-db-instance`Befehl auf AWS CLI, um eine DB-Instance mithilfe von neu zu starten.

Example Einfacher Neustart

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds reboot-db-instance \
    --db-instance-identifier mydbinstance
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds reboot-db-instance ^
    --db-instance-identifier mydbinstance
```

Example Neustart mit Failover

Verwenden Sie den Parameter, um in einem Multi-AZ-DB-Cluster einen Failover von einer AZ zur anderen zu erzwingen. `--force-failover`

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds reboot-db-instance \
    --db-instance-identifier mydbinstance \
    --force-failover
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

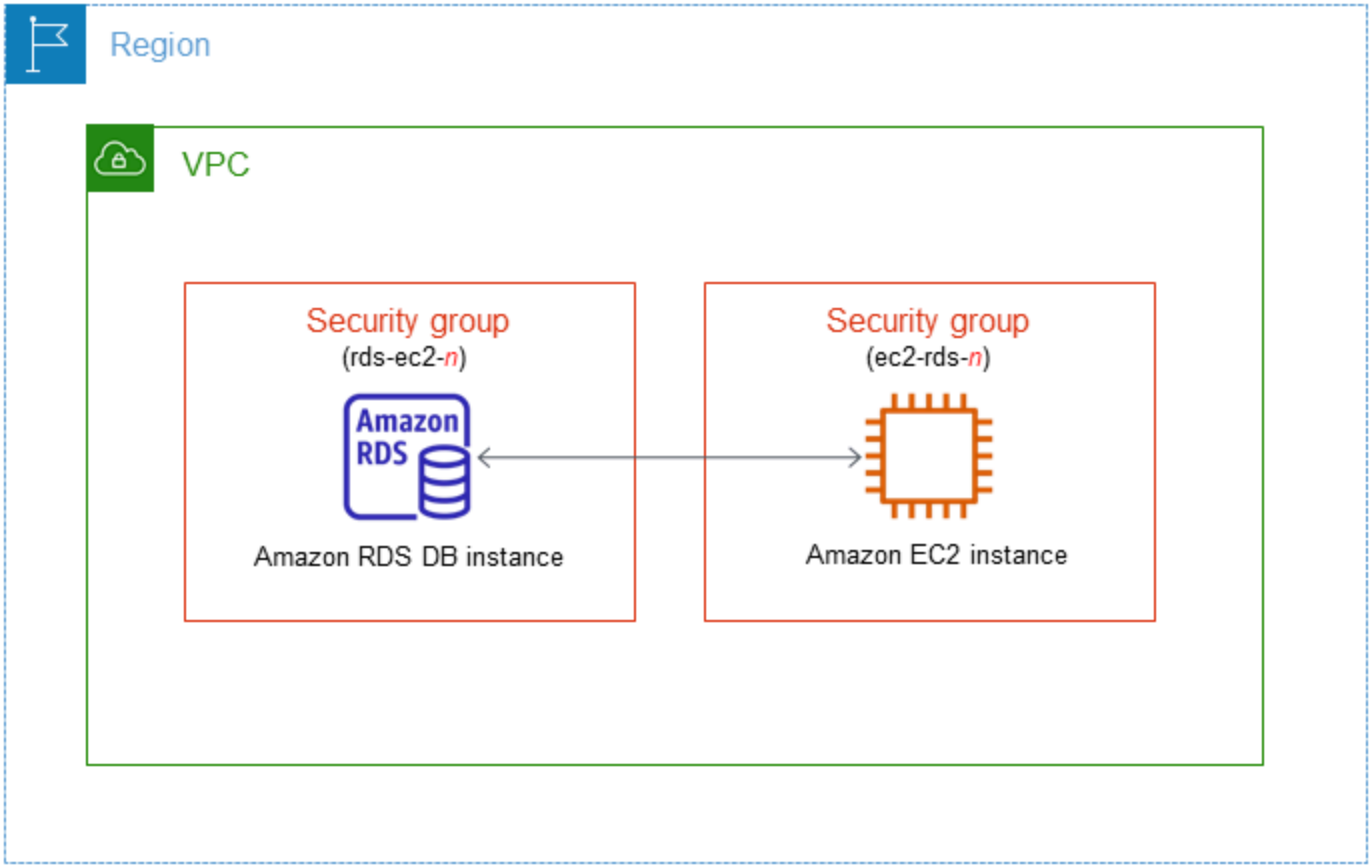
```
aws rds reboot-db-instance ^
    --db-instance-identifier mydbinstance ^
    --force-failover
```

RDS API

Um eine DB-Instance mithilfe von Amazon neu zu starten RDSAPI, rufen Sie den `RebootDBInstance`Vorgang auf.

Automatisches Verbinden einer EC2 Instance und einer DB-Instance mit

Sie können die RDS Amazon-Konsole verwenden, um die Einrichtung einer Verbindung zwischen einer Amazon Elastic Compute Cloud (AmazonEC2) -Instance und einer DB-Instance, einem , zu vereinfachen. Oft befindet sich Ihr in einem privaten Subnetz und Ihre EC2 Instance in einem öffentlichen Subnetz innerhalb eines. VPC Sie können einen SQL Client auf Ihrer EC2 Instance verwenden, um eine Verbindung zu Ihrem herzustellen. Die EC2 Instance kann auch Webserver oder Anwendungen ausführen, die auf Ihren privaten zugreifen. Anweisungen zum Einrichten einer Verbindung zwischen einer EC2 Instance und einem Multi-AZ-DB-Cluster finden Sie unterthe section called “Verbinden einer EC2-Instance mit einem Multi-AZ-DB-Cluster”.



Wenn Sie eine Verbindung zu einer EC2 Instance herstellen möchten, die sich nicht in derselben Instanz VPC wie der befindet, finden Sie die Szenarien unterSzenarien für den Zugriff auf einen in einem VPC.

Themen

- Überblick über die automatische Konnektivität mit einer EC2 Instance
- Automatisches Verbinden einer EC2 Instance und eines RDS
- Anzeigen verbundener Rechenressourcen
- Herstellen einer Verbindung mit einer DB-Instance, die eine bestimmte DB-Engine ausführt

Überblick über die automatische Konnektivität mit einer EC2 Instance

Wenn Sie eine Verbindung zwischen einer EC2 Instance und einem RDS einrichten, RDSautomatically konfiguriert Amazon die VPC Sicherheitsgruppe für Ihre EC2 Instance und für Ihren RDS.

Die folgenden Anforderungen gelten für die Verbindung einer EC2 Instance mit einem RDS:

- Die EC2 Instance muss in derselben Umgebung VPC wie der RDS existieren.

  Wenn in derselben EC2 Instanz keine Instanzen vorhanden sindVPC, bietet die Konsole einen Link zum Erstellen einer Instanz.
- Der Benutzer, der die Konnektivität einrichtet, muss über Berechtigungen verfügen, um die folgenden EC2 Amazon-Operationen auszuführen:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Wenn sich die DB-Instance und die EC2 Instance in unterschiedlichen Availability Zones befinden, können für Ihr Konto Kosten für mehrere Availability Zones anfallen.

Wenn Sie eine Verbindung zu einer EC2 Instance einrichten, RDS handelt Amazon gemäß der aktuellen Konfiguration der Sicherheitsgruppen, die dem RDS und der EC2 Instance zugeordnet sind, wie in der folgenden Tabelle beschrieben.

Aktuelle Konfiguration der RDS Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	RDSAktion
Dem RDS sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht `rds-ec2-`*n* (wobei eine Zahl *n* steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der EC2 Instance als Quelle.	Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht `ec2-rds-`*n* (wobei eine Zahl *n* steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC Sicherheitsgruppe des RDS als Quelle.	RDSergreift keine Maßnahmen. Eine Verbindung zwischen der EC2 Instance und dem RDS wurde bereits automatisch konfiguriert. Da bereits eine Verbindung zwischen der EC2 Instance und der RDS Datenbank besteht, werden die Sicherheitsgruppen nicht geändert.
Es gilt eine der folgenden Bedingungen: • Dem RDS ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entsprichtrds-ec2-*n*. • Dem RDS sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich trds-ec2-*n*. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2 Instance verwenden. Amazon RDS kann keine Sicherheitsgruppe verwenden, die nicht über	Es gilt eine der folgenden Bedingungen: • Der EC2 Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entsprichtec2-rds-*n*. • Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprichtec2-rds-*n*. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem RDS verwenden. Amazon RDS kann keine Sicherheitsgruppe verwenden, die keine	RDS action: create new security groups

Aktuelle Konfiguration der RDS Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	RDSAktion
eine Regel für eingehend en Datenverkehr mit der VPC Sicherheitsgruppe der EC2 Instance als Quelle verfügt. Amazon kann RDS auch keine Sicherhei tsgruppe verwenden, die geändert wurde. Beispiele für Änderungen sind das Hinzufügen einer Regel oder das Ändern des Ports einer vorhandenen Regel.	einzige ausgehende Regel mit der VPC Sicherhei tsgruppe des RDS als Quelle hat. Amazon kann RDS auch keine Sicherhei tsgruppe verwenden, die geändert wurde.	
Dem RDS sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich`rds-ec2-`*n*. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der EC2 Instance als Quelle.	Der EC2 Instanz sind eine oder mehrere Sicherhei tsgruppen zugeordnet, deren Name dem Muster `ec2-rds-`*n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem RDS verwenden. Amazon RDS kann keine Sicherhei tsgruppe verwenden, die keine einzige ausgehende Regel mit der VPC Sicherheitsgruppe des RDS als Quelle hat. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	RDS action: create new security groups

Aktuelle Konfiguration der RDS Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	RDSAktion
Dem RDS sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht`rds-ec2-`*n*. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der EC2 Instance als Quelle.	Eine gültige EC2 Sicherheitsgruppe für die Verbindung ist vorhanden, aber sie ist der EC2 Instanz nicht zugeordnet. Die Sicherheitsgruppe trägt einen Namen, der dem Muster `ec2-rds-`*n* entspricht. Sie wurde nicht geändert. Sie hat nur eine ausgehende Regel mit der VPC Sicherheitsgruppe des RDS als Quelle.	RDS action: associate EC2 security group

Aktuelle Konfiguration der RDS Sicherheitsgruppe	Aktuelle Konfiguration der EC2 Sicherheitsgruppe	RDSAktion
Es gilt eine der folgenden Bedingungen: • Dem RDS ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. • Dem RDS sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich `trds-ec2-`*n*. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2 Instance verwenden . Amazon RDS kann keine Sicherheitsgruppe verwenden, die nicht über eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der EC2 Instance als Quelle verfügt. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Der EC2 Instance sind eine oder mehrere Sicherhei tsgruppen zugeordnet, deren Name dem Muster entsprich`ec2-rds-`*n*. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC Sicherheitsgruppe des RDS als Quelle.	RDS action: create new security groups

RDSAktion: neue Sicherheitsgruppen erstellen

Amazon RDS ergreift die folgenden Maßnahmen:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `rds-ec2-`*n* entspricht. Diese Sicherheitsgruppe hat eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der EC2 Instance als Quelle. Diese Sicherheitsgruppe ist dem RDS zugeordnet und ermöglicht der EC2 Instance den Zugriff auf den RDS.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `ec2-rds-`*n* entspricht. Diese Sicherheitsgruppe hat eine ausgehende Regel mit der VPC Sicherheitsgruppe des RDS als Ziel. Diese Sicherheitsgruppe ist der Instance zugeordnet und ermöglicht es der EC2 EC2 Instance, Traffic an den RDS zu senden.

RDSAktion: EC2 Sicherheitsgruppe zuordnen

Amazon RDS ordnet der EC2 Instance die gültige, bestehende EC2 Sicherheitsgruppe zu. Diese Sicherheitsgruppe ermöglicht es der EC2 Instance, Datenverkehr an den RDS zu senden.

Automatisches Verbinden einer EC2 Instance und eines RDS

Bevor Sie eine Verbindung zwischen einer EC2 Instance und einem RDS einrichten, stellen Sie sicher, dass Sie die unter beschriebenen Anforderungen erfüllenÜberblick über die automatische Konnektivität mit einer EC2 Instance.

Wenn Sie nach der Konfiguration der Konnektivität Änderungen an Sicherheitsgruppen vornehmen, können sich die Änderungen auf die Verbindung zwischen der EC2 Instance und dem RDS auswirken.

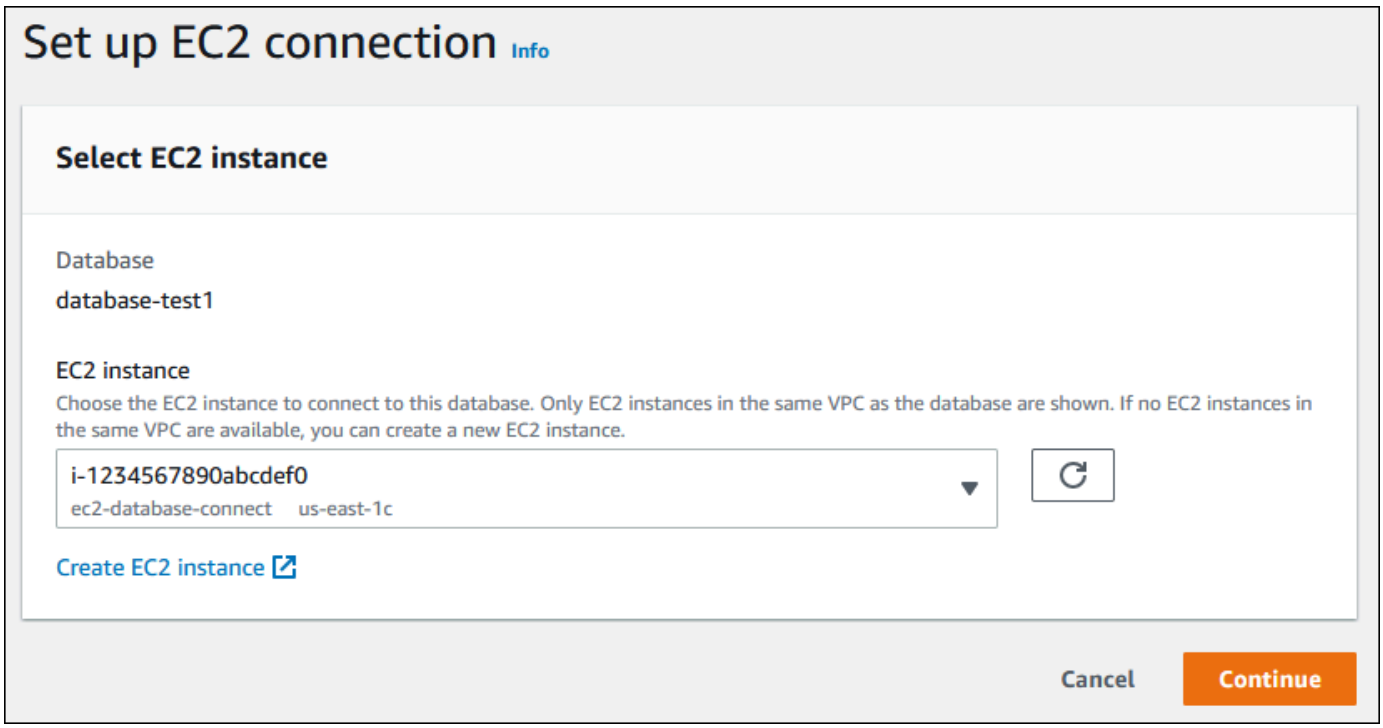
> **Note**
>
> Sie können eine Verbindung zwischen einer EC2 Instance und einem RDS nur automatisch einrichten, indem Sie den verwenden AWS Management Console. Mit dem AWS CLI oder können Sie keine automatische Verbindung einrichten RDSAPI.

Um eine EC2 Instance und einen RDS automatisch zu verbinden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den RDS aus.
3. Wählen Sie unter Aktionen die Option EC2Verbindung einrichten aus.

Die Seite EC2Verbindung einrichten wird angezeigt.

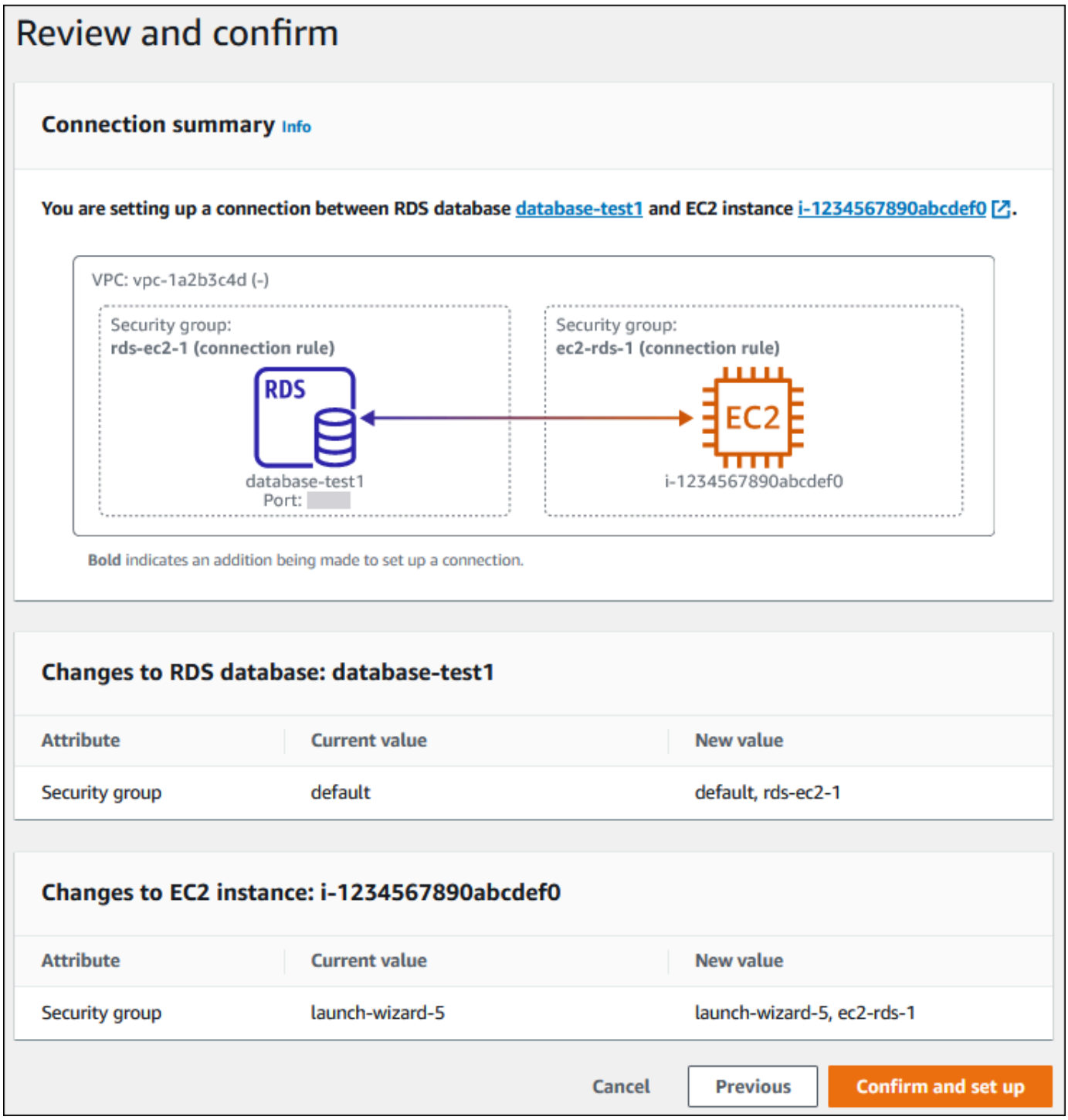
4. Wählen Sie auf der Seite EC2Verbindung einrichten die EC2 Instanz aus.



Wenn in derselben Instanz keine EC2 Instanzen vorhanden sindVPC, wählen Sie EC2Instanz erstellen, um eine zu erstellen. Stellen Sie in diesem Fall sicher, dass sich die neue EC2 Instance im selben VPC RDS befindet.

5. Klicken Sie auf Weiter.

Die Seite Review and confirm (Überprüfen und bestätigen) wird angezeigt.



6. Überprüfen Sie auf der Seite Überprüfen und bestätigen die Änderungen, die zur Einrichtung der Konnektivität mit der EC2 Instance vorgenommen RDS werden.

   Wenn die Änderungen korrekt sind, wählen Sie Bestätigen und einrichten.

   Sind die Änderungen nicht korrekt, wählen Sie Previous (Zurück) oder Cancel (Abbrechen) aus.

Anzeigen verbundener Rechenressourcen

Sie können den verwenden AWS Management Console , um die Computing-Ressourcen anzuzeigen, die mit einem für RDSDatenbanken verbunden sind. Zu den angezeigten Ressourcen gehören Rechenressourcenverbindungen, die automatisch eingerichtet wurden. Sie können die Konnektivität mit Rechenressourcen auf folgende Weise automatisch einrichten:

- Sie können die Rechenressource auswählen, wenn Sie die Datenbank erstellen.

  Weitere Informationen erhalten Sie unter Erstellen einer Amazon RDS-DB-Instance und Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen.
- Sie können die Konnektivität zwischen einer vorhandenen Datenbank und einer Rechenressource einrichten.

  Weitere Informationen finden Sie unter Automatisches Verbinden einer EC2 Instance und eines RDS.

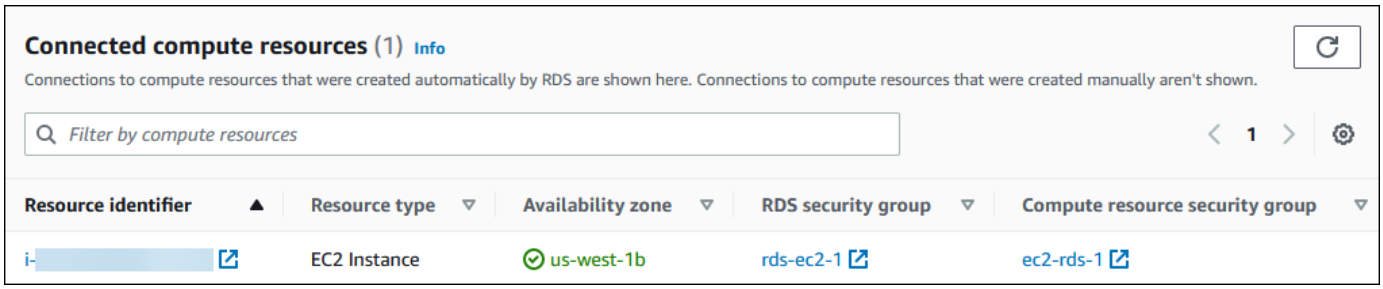
Die aufgelisteten Rechenressourcen enthalten keine Ressourcen, die manuell mit der Datenbank verbunden wurden. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf eine Datenbank ermöglichen, indem Sie der VPC Sicherheitsgruppe, die der Datenbank zugeordnet ist, eine Regel hinzufügen.

Für die Auflistung einer Rechenressource müssen die folgenden Bedingungen erfüllt sei:

- Der Name der Sicherheitsgruppe, die der Rechenressource zugeordnet ist, entspricht dem Muster `ec2-rds-`*n* (wobei *n* für eine Zahl steht).
- Die der Rechenressource zugeordnete Sicherheitsgruppe hat eine ausgehende Regel, bei der der Portbereich auf den Port festgelegt ist, den der RDS verwendet.
- Die der Rechenressource zugeordnete Sicherheitsgruppe hat eine ausgehende Regel, bei der die Quelle auf eine Sicherheitsgruppe gesetzt ist, die dem RDS zugeordnet ist.
- Der Name der Sicherheitsgruppe, die dem RDS zugeordnet ist, entspricht dem Muster `rds-ec2-`*n* (wobei eine Zahl *n* steht).
- Die dem RDS zugeordnete Sicherheitsgruppe hat eine Regel für eingehenden Datenverkehr, bei der der Portbereich auf den Port festgelegt ist, den der RDS verwendet.
- Die dem RDS zugeordnete Sicherheitsgruppe hat eine Regel für eingehenden Datenverkehr, wobei die Quelle auf eine Sicherheitsgruppe gesetzt ist, die der Rechenressource zugeordnet ist.

So zeigen Sie Rechenressourcen an, die mit einem in der RDSDatenbank verbunden sind

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den Namen des RDS aus.
3. Sehen Sie sich auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) die Rechenressourcen unter Verbundene Rechenressourcen an.



Herstellen einer Verbindung mit einer DB-Instance, die eine bestimmte DB-Engine ausführt

Informationen zum Herstellen einer Verbindung mit einer DB-Instance, die eine bestimmte DB-Engine ausführt, finden Sie in den Anweisungen für Ihre DB-Engine:

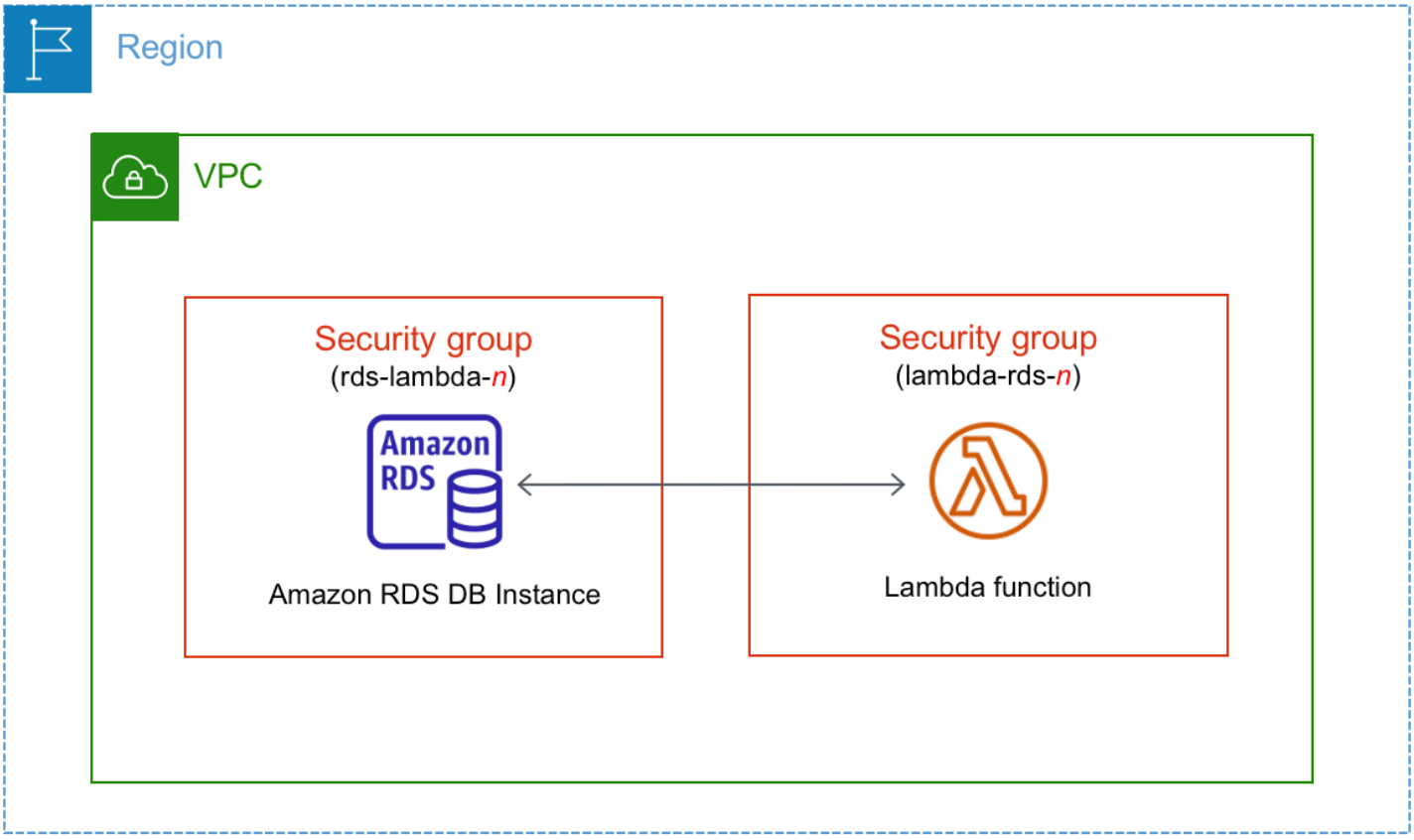
- Verbindung zu Ihrer MariaDB-DB-Instance herstellen
- Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen
- Verbindung zu Ihrer MySQL-DB-Instance herstellen
- Herstellen der Verbindung zu Ihrer Oracle-DB-Instance
- Herstellen einer Verbindung zu einer DB-Instance, in der die PostgreSQL-Datenbank-Engine ausgeführt wird

Automatisches Verbinden einer Lambda-Funktion mit einer DB-Instance

Sie können die RDS Amazon-Konsole verwenden, um die Einrichtung einer Verbindung zwischen einer Lambda-Funktion und einer DB-Instance, , zu vereinfachen. Oft befindet sich Ihr in einem privaten Subnetz innerhalb eines. VPC Die Lambda-Funktion kann von Anwendungen verwendet werden, um auf Ihre private DB-Instance zuzugreifen.

Anweisungen zum Einrichten einer Verbindung zwischen einer Lambda-Funktion und einem Multi-AZ-DB-Cluster finden Sie unter the section called “Verbinden einer Lambda-Funktion und eines Multi-AZ-DB-Clusters”.

Das folgende Bild zeigt eine direkte Verbindung zwischen Ihrer DB-Instance und Ihrer Lambda-Funktion.

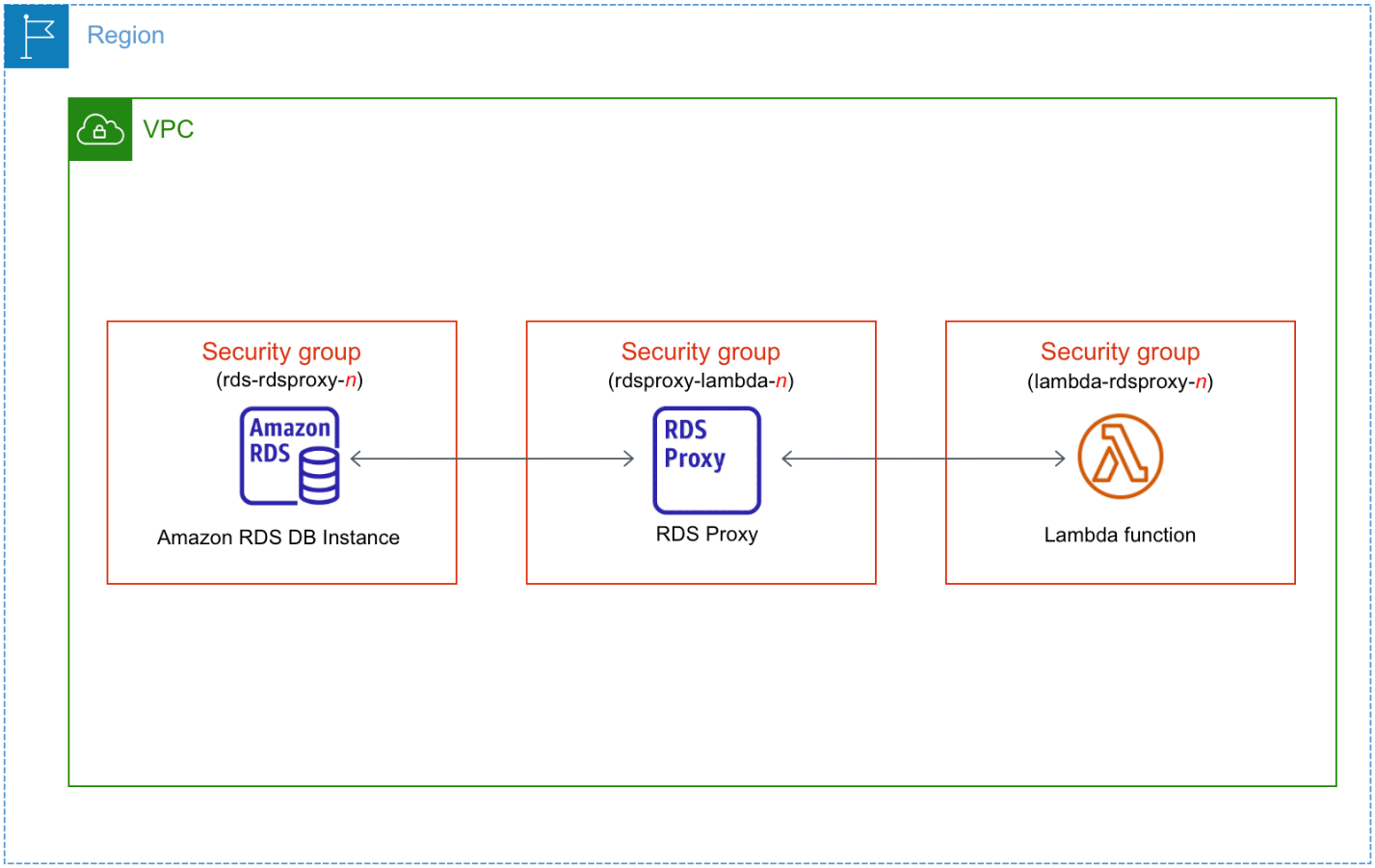


Sie können die Verbindung zwischen Ihrer Lambda-Funktion und Ihrem über RDS Proxy einrichten, um die Leistung und Resilienz Ihrer Datenbank zu verbessern. Oft stellen Lambda-Funktionen häufige, kurze Datenbankverbindungen her, die vom Verbindungspooling profitieren, das RDS Proxy bietet. Sie können jede AWS Identity and Access Management (IAM) -Authentifizierung nutzen,

die Sie bereits für Lambda-Funktionen haben, anstatt Datenbankanmeldedaten in Ihrem Lambda-Anwendungscode zu verwalten. Weitere Informationen finden Sie unter Amazon RDS-Proxy .

Wenn Sie die Konsole verwenden, um eine Verbindung mit einem vorhandenen Proxy herzustellen, RDS aktualisiert Amazon die Proxy-Sicherheitsgruppe, um Verbindungen von Ihrem und der Lambda-Funktion aus zuzulassen.

Sie können auf dieser Konsolenseite auch einen neuen Proxy erstellen. Wenn Sie in der Konsole einen Proxy erstellen, müssen Sie für den Zugriff auf den Ihre Datenbankanmeldedaten eingeben oder ein AWS Secrets Manager Geheimnis auswählen.



Tip

Um eine Lambda-Funktion schnell mit einer DB-Instance oder einem zu verbinden, können Sie auch den in der Konsole integrierten Assistenten verwenden. Gehen Sie wie folgt vor, um den Assistenten zu öffnen:

1. Öffnen Sie die Seite Funktionen der Lambda-Konsole.
2. Wählen Sie die Funktion aus, mit der Sie eine Datenbank verbinden möchten.

3. Wählen Sie auf der Registerkarte Konfiguration die Option RDSDatenbanken aus.
4. Wählen Sie Mit RDS Datenbank verbinden.

Nachdem Sie Ihre Funktion mit einer Datenbank verbunden haben, können Sie einen Proxy erstellen, indem Sie Proxy hinzufügen wählen.

Themen

- Überblick über die automatische Konnektivität mit einer Lambda-Funktion
- Automatisches Verbinden einer Lambda-Funktion und eines RDS
- Anzeigen verbundener Rechenressourcen

Überblick über die automatische Konnektivität mit einer Lambda-Funktion

Im Folgenden sind die Anforderungen für die Verbindung einer Lambda-Funktion mit einem aufgeführt:

- Die Lambda-Funktion muss im selben DB-Cluster VPC wie der vorhanden sein.
- Der Benutzer, der die Konnektivität einrichtet, muss über Berechtigungen verfügen, um die folgenden Amazon-RDS, Amazon-EC2, Lambda-, Secrets Manager- und IAM Operationen auszuführen:
  - Amazon RDS
    - `rds:CreateDBProxies`
    - `rds:DescribeDBInstances`
    - `rds:DescribeDBProxies`
    - `rds:ModifyDBInstance`
    - `rds:ModifyDBProxy`
    - `rds:RegisterProxyTargets`
  - Amazon EC2
    - `ec2:AuthorizeSecurityGroupEgress`
    - `ec2:AuthorizeSecurityGroupIngress`
    - `ec2:CreateSecurityGroup`
    - `ec2:DeleteSecurityGroup`

    - `ec2:DescribeSecurityGroups`
    - `ec2:RevokeSecurityGroupEgress`
    - `ec2:RevokeSecurityGroupIngress`
- Lambda
    - `lambda:CreateFunctions`
    - `lambda:ListFunctions`
    - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
    - `secretsmanager:CreateSecret`
    - `secretsmanager:DescribeSecret`
- IAM
    - `iam:AttachPolicy`
    - `iam:CreateRole`
    - `iam:CreatePolicy`
- AWS KMS
    - `kms:describeKey`

> **Note**
>
> Wenn sich die DB-Instance und die Lambda-Funktion in unterschiedlichen Availability Zones befinden, fallen auf Ihrem Konto ggf. Kosten über Availability Zones hinweg an.

Wenn Sie eine Verbindung zwischen einer Lambda-Funktion und einem RDS einrichten, RDS konfiguriert Amazon die VPC Sicherheitsgruppe für Ihre Funktion und für Ihren . Wenn Sie RDS Proxy verwenden, konfiguriert Amazon RDS auch die VPC Sicherheitsgruppe für den Proxy. Amazon RDS handelt gemäß der aktuellen Konfiguration der Sicherheitsgruppen, die dem zugeordnet sind, wie in der folgenden Tabelle beschrieben.

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Dem sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht. Falls ein Proxy bereits mit Ihrem verbunden ist, wird RDS geprüft, ob `TargetHealth` der eines zugehörigen Proxys vorhanden ist`AVAILABLE` . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Lambda-Proxys als Quelle.	Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht (wobei *n* für eine Zahl steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit entweder der VPC Sicherheitsgruppe des oder dem Proxy als Ziel.	Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht (wobei *n* für eine Zahl steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert.	Amazon RDS ergreift keine Maßnahmen. Es wurde bereits automatisch eine Verbindung zwischen der Lambda-Funktion, dem Proxy (optional) und der DB-Instance konfiguriert. Da bereits eine Verbindung zwischen der Funktion, dem Proxy und der Datenbank besteht, werden die Sicherheitsgruppen nicht geändert.
Es gilt eine der folgenden Bedingungen:	Es gilt eine der folgenden Bedingungen:	Es gilt eine der folgenden Bedingungen:	RDS action: create new security groups

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
• Der DB-Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. • Eine oder mehrere Sicherheitsgruppen sind der DB-Instance zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. Keine dieser Sicherheitsgruppen kann jedoch für die Verbindung mit der Lambda-Funktion verwendet werden. Amazon RDS kann keine Sicherheit	• Der Lambda-Funktion ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. • Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem verwenden. Amazon RDS kann keine Sicherheitsgruppe verwenden, die keine einzige ausgehende Regel	• Dem Proxy ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. • Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem oder der Lambda-Funktion verwenden. Amazon RDS kann keine Sicherheitsgruppe ohne Regeln für eingehenden und ausgehenden Datenverkehr mit der VPC Sicherheitsgruppe des und der Lambda-Fu	

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
tsgruppe verwenden, die keine einzige Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Proxys als Quelle hat. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde. Beispiele für Änderungen sind das Hinzufügen einer Regel oder das Ändern des Ports einer vorhandenen Regel.	mit der VPC Sicherheitsgruppe des oder Proxys als Ziel hat. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	nktion verwenden. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Eine oder mehrere Sicherheitsgruppen sind der DB-Instance zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Lambda-Proxys als Quelle.	Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem verwenden. Amazon RDS kann keine Sicherheitsgruppe verwenden, die keine einzige ausgehende Regel mit der VPC Sicherheitsgruppe des oder Proxys als Ziel hat. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem oder der Lambda-Funktion verwenden. Amazon RDS kann keine Sicherheitsgruppe ohne Regeln für eingehenden und ausgehenden Datenverkehr mit der VPC Sicherheitsgruppe des und der Lambda-Funktion verwenden. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	RDS action: create new security groups

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Eine oder mehrere Sicherheitsgruppen sind der DB-Instance zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Lambda-Proxys als Quelle.	Eine gültige Lambda-Sicherheitsgruppe für die Verbindung ist vorhanden, jedoch nicht mit der Lambda-Funktion verknüpft. Die Sicherheitsgruppe hat einen Namen, der dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Sie wurde nicht geändert. Sie hat nur eine ausgehende Regel mit der VPC Sicherheitsgruppe des als Ziel.	Eine gültige Proxy-Sicherheitsgruppe für die Verbindung ist vorhanden, jedoch nicht mit dem Proxy verknüpft. Die Sicherheitsgruppe trägt einen Namen, der dem Muster `rdsproxy-lambda-` *n* entspricht. Sie wurde nicht geändert. Es hat Regeln für eingehenden und ausgehenden Datenverkehr mit der VPC Sicherheitsgruppe des und der Lambda-Funktion.	RDS action: associate Lambda security group

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Es gilt eine der folgenden Bedingungen: • Der DB-Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. • Eine oder mehrere Sicherheitsgruppen sind der DB-Instance zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Lambda-	Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC Sicherheitsgruppe der DB-Instance oder des Proxys der als Ziel.	Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat Regeln für eingehenden und ausgehenden Datenverkehr mit der VPC Sicherheitsgruppe des und der Lambda-Funktion.	RDS action: create new security groups

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Funktion oder dem Lambda-Proxy verwenden. Amazon RDS kann keine Sicherheitsgruppe verwenden, die keine einzige Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Proxys als Quelle hat. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.			

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Es gilt eine der folgenden Bedingungen: • Der DB-Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. • Eine oder mehrere Sicherheitsgruppen sind der DB-Instance zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugehörigen Proxys `AVAILABLE` ist. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Lambda-	Es gilt eine der folgenden Bedingungen: • Der Lambda-Funktion ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. • Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem verwenden.	Es gilt eine der folgenden Bedingungen: • Dem Proxy ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. • Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem oder der Lambda-Funktion verwenden. Amazon RDS kann keine Sicherheitsgruppe ohne Regeln für eingehenden und ausgehenden	RDS action: create new security groups

Aktuelle Konfiguration RDS der Sicherheitsgruppe	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Funktion oder dem Lambda-Proxy verwenden. Amazon RDS kann keine Sicherheitsgruppe verwenden, die keine einzige Regel für eingehenden Datenverkehr mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Proxys als Quelle hat. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Amazon RDS kann keine Sicherheitsgruppe verwenden, die nicht über eine einzige ausgehende Regel mit der VPC Sicherheitsgruppe des oder Proxys als Quelle verfügt. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Datenverkehr mit der VPC Sicherheitsgruppe des und der Lambda-Funktion verwenden. Amazon kann RDS auch keine Sicherheitsgruppe verwenden, die geändert wurde.	

RDSAktion: neue Sicherheitsgruppen erstellen

Amazon RDS ergreift die folgenden Maßnahmen:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster entspricht `rds-lambda-`*n* oder `rds-rdsproxy-`*n* (falls Sie RDS Proxy verwenden möchten). Diese Sicherheitsgruppe hat eine eingehende Regel mit der VPC Sicherheitsgruppe der Lambda-Funktion oder des Lambda-Proxys als Quelle. Diese Sicherheitsgruppe ist der DB-Instance zugeordnet und ermöglicht der Funktion oder dem Proxy den Zugriff auf die DB-Instance.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `lambda-rds-`*n* oder `lambda-rdsproxy-`*n* entspricht. Diese Sicherheitsgruppe hat eine ausgehende Regel mit der VPC Sicherheitsgruppe des als Ziel. Diese Sicherheitsgruppe ist der Lambda-Funktion zugeordnet und

ermöglicht es der Funktion, Datenverkehr an die DB-Instance zu senden oder Datenverkehr über einen Proxy zu senden.

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `rdsproxy-lambda-`*n* entspricht. Diese Sicherheitsgruppe hat Regeln für eingehenden und ausgehenden Datenverkehr mit der VPC Sicherheitsgruppe des und der Lambda-Funktion.

RDSAktion: Lambda-Sicherheitsgruppe zuordnen

Amazon RDS verknüpft die gültige, bestehende Lambda-Sicherheitsgruppe mit der Lambda-Funktion. Diese Sicherheitsgruppe ermöglicht es der Funktion, Datenverkehr an die DB-Instance zu senden oder Datenverkehr über einen Proxy zu senden.

Automatisches Verbinden einer Lambda-Funktion und eines RDS

Sie können die RDS Amazon-Konsole verwenden, um automatisch eine Lambda-Funktion mit Ihrem zu verbinden. Dies vereinfacht das Einrichten einer Verbindung zwischen diesen Ressourcen.

Sie können RDS Proxy auch verwenden, um einen Proxy in Ihre Verbindung aufzunehmen. Lambda-Funktionen stellen häufig kurze Datenbankverbindungen her, die vom Verbindungspooling profitieren, das RDS Proxy bietet. Sie können auch jede IAM Authentifizierung verwenden, die Sie bereits für Ihre Lambda-Funktion eingerichtet haben, anstatt die Datenbankanmeldedaten in Ihrem Lambda-Anwendungscode zu verwalten.

Sie können eine vorhandene DB-Instance mit neuen oder bestehenden Lambda-Funktionen unter Verwendung der Seite Lambda-Verbindung einrichtenverbinden. Beim Einrichtungsvorgang werden automatisch die erforderlichen Sicherheitsgruppen für Sie eingerichtet.

Vor dem Einrichten einer Verbindung zwischen einer Lambda-Funktion und einer DB-Instance stellen Sie Folgendes sicher:

- Ihre Lambda-Funktion und Ihr befinden sich im selbenVPC.
- Sie verfügen über die richtigen Berechtigungen für Ihr Benutzerkonto. Weitere Informationen zu den Anforderungen finden Sie unter Überblick über die automatische Konnektivität mit einer Lambda-Funktion.

Wenn Sie Sicherheitsgruppen nach dem Konfigurieren der Verbindung ändern, können sich die Änderungen auf die Verbindung zwischen der Lambda-Funktion und der DB-Instance auswirken.

> **Note**
>
> Sie können automatisch eine Verbindung zwischen einem und einer Lambda-Funktion nur in der AWS Management Console einrichten. Zum Herstellen einer Verbindung mit einer Lambda-Funktion muss sich die DB-Instance im Status Verfügbar befinden.

So verbinden Sie eine Lambda-Funktion automatisch mit einer DB-Instance

<result>

Nachdem Sie die Einrichtung bestätigt haben, RDS beginnt Amazon mit der Verbindung Ihrer Lambda-Funktion, Ihres RDS Proxys (falls Sie einen Proxy verwendet haben) und Ihres . Die Konsole zeigt das Dialogfeld Verbindungsdetails an, in dem die Änderungen der Sicherheitsgruppe aufgeführt sind, die Verbindungen zwischen Ihren Ressourcen ermöglichen.
</result>

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die DB-Instance zum Verbinden mit einer Lambda-Funktion aus.
3. Wählen Sie für Aktionen die Option Lambda-Verbindung einrichten aus.
4. Führen Sie auf der Seite Lambda-Verbindung einrichten unter Lambda-Funktion auswählen einen der folgenden Schritte aus:
   - Wenn Sie eine bestehende Lambda-Funktion in derselben Datei VPC wie Ihr haben, wählen Sie Vorhandene Funktion auswählen und dann die Funktion aus.
   - Wenn Sie keine Lambda-Funktion in derselben habenVPC, wählen Sie Neue Funktion erstellen aus, und geben Sie dann einen Funktionsnamen ein. Die Standard-Laufzeit ist auf Nodejs.18 festgelegt. Sie können die Einstellungen für Ihre neue Lambda-Funktion in der Lambda-Konsole ändern, nachdem Sie die Verbindungseinrichtung abgeschlossen haben.
5. (Optional) Wählen Sie unter RDSProxy die Option Mit RDS Proxy Connect aus, und führen Sie dann einen der folgenden Schritte aus:
   - Wenn Sie einen vorhandenen Proxy haben, den Sie verwenden möchten, klicken Sie auf Vorhandenen Proxy auswählen und wählen Sie dann den Proxy aus.
   - Wenn Sie keinen Proxy haben und möchten, dass Amazon RDS automatisch einen für Sie erstellt, wählen Sie Neuen Proxy erstellen. Führen Sie dann für Datenbankanmeldeinformationen einen der folgenden Schritte aus:

a. Wählen Sie Datenbankbenutzername und Passwort aus und geben Sie dann den Benutzernamen und das Passwort für Ihre DB-Instance ein.

b. Wählen Sie Secrets-Manager-Secret aus. Wählen Sie dann für Select Secret ein AWS Secrets Manager Secret aus. Wenn Sie kein Secrets-Manager-Secret haben, wählen Sie Neues Secrets-Manager-Secret erstellen aus, um ein neues Secret zu erstellen. Nachdem Sie das Secret erstellt haben, wählen Sie das neue Secret unter Secret auswählen aus.

Nachdem Sie den neuen Proxy erstellt haben, wählen Sie Vorhandenen Proxy auswählen aus und wählen Sie dann den Proxy aus. Beachten Sie, dass es einige Zeit dauern kann, bis Ihr Proxy für die Verbindung verfügbar ist.

6. (Optional) Erweitern Sie die Verbindungsübersicht und überprüfen Sie die hervorgehobenen Updates für Ihre Ressourcen.
7. Wählen Sie Set up (Festlegen).

Anzeigen verbundener Rechenressourcen

Sie können den verwenden AWS Management Console , um die Lambda-Funktionen anzuzeigen, die mit Ihrem verbunden sind. Zu den angezeigten Ressourcen gehören Verbindungen zu Rechenressourcen, die Amazon automatisch RDS eingerichtet hat.

Die aufgelisteten Rechenressourcen umfassen keine Ressourcen, die manuell mit der DB-Instance verbunden sind. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf Ihren ermöglichen, indem Sie Ihrer mit der Datenbank verknüpften VPC Sicherheitsgruppe eine Regel hinzufügen.

Damit die Konsole eine Lambda-Funktion auflistet, müssen die folgenden Bedingungen gelten:

- Der Name der Sicherheitsgruppe, die der Rechenressource zugeordnet ist, entspricht dem Muster `lambda-rds-`*n* oder `lambda-rdsproxy-`*n* (wobei *n* für eine Zahl steht).
- Die Sicherheitsgruppe, die der Rechenressource zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei der Portbereich auf den Port der DB-Instance oder einen zugeordneten Proxy festgelegt ist. Das Ziel für die Regel für ausgehende Nachrichten muss auf eine der DB-Instance zugeordneten Sicherheitsgruppe oder auf einen zugeordneten Proxy festgelegt werden.

- Wenn die Konfiguration einen Proxy enthält, entspricht der Name der Sicherheitsgruppe, die an den mit Ihrer Datenbank verknüpften Proxy angefügt ist, dem Muster `rdsproxy-lambda-`*n* (wobei *n* für eine Zahl steht).
- Die Sicherheitsgruppe, die der Funktion zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei der Port auf den Port festgelegt ist, den die DB-Instance oder ein zugeordneter Proxy verwendet. Das Ziel muss auf eine der DB-Instance zugeordneten Sicherheitsgruppe oder auf einen zugeordneten Proxy festgelegt werden.

So zeigen Sie Rechenressourcen an, die automatisch mit einer DB-Instance verbunden sind

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die DB-Instance aus.
3. Sehen Sie sich auf der Registerkarte Konnektivität und Sicherheit die Rechenressourcen unter Verbundene Rechenressourcen an.

Ändern einer Amazon RDS DB-Instance

Sie können die Einstellungen einer DB-Instance ändern, um Aufgaben wie das Hinzufügen von Speicher oder das Ändern der DB-Instance-Klasse durchführen zu können. In diesem Thema erfahren Sie, wie Sie eine RDS Amazon-DB-Instance ändern und erfahren mehr über die Einstellungen für DB-Instances.

Wir empfehlen Ihnen alle Änderungen in einer Test-Instance zu prüfen, bevor Sie eine produktive Instance ändern. Auf diese Weise können Sie die Auswirkungen jeder Änderung vollständig verstehen. Das Testen ist besonders wichtig, wenn Sie Upgrades von Datenbankversionen durchführen.

Die meisten Änderungen an einer DB-Instance können Sie entweder sofort anwenden oder bis zum nächsten Wartungsfenster verschieben. Einige Änderungen, wie Änderungen der Parametergruppe, erfordern einen manuellen Neustart Ihrer DB-Instance, damit die Änderungen wirksam werden.

> **Important**
>
> Einige Änderungen führen zu Ausfallzeiten, da Amazon Ihre DB-Instance neu starten RDS muss, damit die Änderung wirksam wird. Überprüfen Sie die Auswirkungen auf Ihre Datenbank und Anwendungen, bevor Sie die Einstellungen für Ihre DB-Instance ändern.

Konsole

So ändern Sie eine DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Ändern Sie alle Einstellungen nach Bedarf. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.
5. Nachdem Sie die gewünschten Änderungen vorgenommen haben, wählen Sie Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.

6. (Optional) Klicken Sie auf Apply immediately (Sofort anwenden), um die Änderungen direkt zu übernehmen. Die Auswahl dieser Option kann in einigen Fällen Ausfallzeiten verursachen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.
7. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern.

   Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Um eine DB-Instance mithilfe von zu ändern AWS CLI, rufen Sie den modify-db-instanceBefehl auf. Geben Sie die DB-Instance-Kennung und die Werte für die Optionen an, die geändert werden sollen. Informationen zu den jeweiligen Optionen finden Sie unter Einstellungen für DB-Instances.

Example

Mit folgendem Code wird `mydbinstance` geändert, da der Aufbewahrungszeitraum für Backups auf 1 Woche (7 Tage) festgelegt wird. Der Code ermöglicht den Löschschutz durch Verwendung von `--deletion-protection`. Um den Löschschutz zu deaktivieren, verwenden Sie `--no-deletion-protection`. Die Änderungen werden während des nächsten Wartungsfensters (mit) übernomme `--no-apply-immediately`. Verwenden Sie `--apply-immediately`, damit Änderungen sofort angewendet werden. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --backup-retention-period 7 ^
    --deletion-protection ^
```

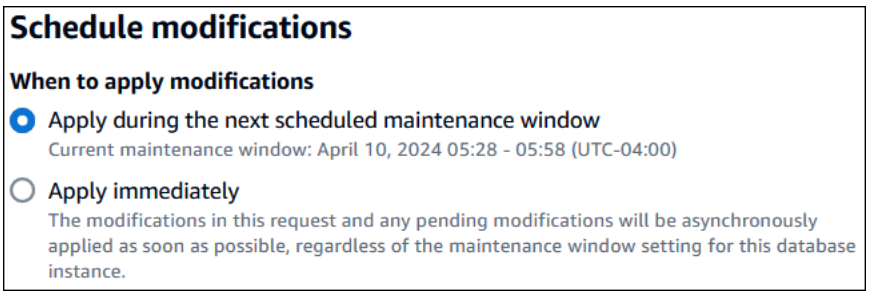
```
    --no-apply-immediately
```

RDS API

Um eine DB-Instance mithilfe von Amazon zu ändern RDSAPI, rufen Sie die odifyDBInstanceM-Operation auf. Geben Sie die DB-Instance-Kennung und die Parameter für die Einstellungen an, die geändert werden sollen. Weitere Informationen zu den einzelnen Parametern finden Sie unter Einstellungen für DB-Instances.

Einstellung „Zeitplanänderungen“ verwenden

Wenn Sie Ihre DB-Instance ändern, entscheiden Sie, wann die Änderungen vorgenommen werden sollen.



Um Änderungen sofort und nicht erst im nächsten Wartungsfenster anzuwenden, wählen Sie die Option Sofort anwenden in der AWS Management Console. Oder Sie verwenden den `--apply-immediately` Parameter beim Aufrufen des AWS CLI oder setzen den `ApplyImmediately` Parameter auf, `true` wenn Sie Amazon verwenden RDSAPI.

Wenn Sie sich nicht dafür entscheiden, Änderungen sofort zu übernehmen, werden RDS die Änderungen in die Warteschlange mit ausstehenden Änderungen verschoben. RDSWendet im nächsten Wartungsfenster alle ausstehenden Änderungen in der Warteschlange an. Wenn Sie sich entscheiden, die Änderungen sofort zu übernehmen, werden alle Ihre neuen Änderungen sowie alle ausstehenden Änderungen in der Warteschlange übernommen.

Um die Änderungen zu sehen, die für das nächste Wartungsfenster noch ausstehen, verwenden Sie den describe-db-instances AWS CLI Befehl und markieren Sie das `PendingModifiedValues` Feld.

> **Important**
>
> Wenn es eine der ausstehenden Änderungen erfordert, dass die DB-Instance vorübergehend nicht verfügbar ist (Ausfallzeiten), kann die Auswahl der Option „Sofort anwenden“ zu unerwarteten Ausfallzeiten führen.

Wenn Sie sich dafür entscheiden, eine Änderung sofort anzuwenden, werden alle anstehenden Änderungen ebenfalls sofort und nicht erst im nächsten Wartungsfenster übernommen.
Wenn Sie nicht möchten, dass eine ausstehende Änderung im nächsten Wartungsfenster übernommen wird, können Sie die DB-Instance so abändern, dass die Änderung rückgängig gemacht wird. Sie können dies tun, indem Sie die `--apply-immediately` Option AWS CLI und angeben.

Änderungen an Datenbankeinstellungen werden unmittelbar übernommen, auch wenn Sie sich entscheiden, Ihre Änderungen auf einen späteren Zeitpunkt zu verschieben. Informationen darüber, wie die verschiedenen Datenbankeinstellungen mit der Einstellung „Apply Immediately (Sofort Anwenden)“ interagieren, finden Sie unter Einstellungen für DB-Instances.

Einstellungen für DB-Instances

In der folgenden Tabelle finden Sie Details darüber, welche Einstellungen Sie ändern können und welche nicht. Sie können auch herausfinden, wann Änderungen angewendet werden können und ob die Änderungen Ausfallzeiten für Ihre DB-Instance verursachen. Indem Sie Amazon-RDS-Funktionen wie Multi-AZ verwenden, können Sie Ausfallzeiten minimieren, wenn Sie die DB-Instance später bearbeiten. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Sie können eine DB-Instance mithilfe der Konsole, des CLI-Befehls `modify-db-instance` oder der RDS-API-Operation `ModifyDBInstance` ändern.

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Allocated storage Der Speicherplatz (in Gibibyte), der Ihrer DB-Instance zugewiesen werden soll. Sie können nur den zugewiesenen Speicher	CLI-Option: `--allocated-storage`	Wenn Sie die Änderung sofort anwenden	Während dieser Änderung treten keine Ausfallze	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
erhöhen. Sie können den zugewiesenen Speicher nicht reduzieren. Sie können den Speicherplatz einiger älterer DB-Instances nicht ändern, oder von DB-Instances, die aus älteren DB-Snapshots wiederhergestellt wurden. Die Option Allocated storage (Zugewiesener Speicher) ist in der Konsole deaktiviert, wenn Ihre DB-Instance nicht geeignet ist. Mit dem describe-valid-db-instanceCLI-Befehl -modifications können Sie überprüfen, ob Sie mehr Speicherplatz zuweisen können. Dieser Befehl gibt die gültigen Speicheroptionen für Ihre DB-Instance zurück. Sie können den zugewiesenen Speicher nicht ändern, wenn der Status der DB-Instance storage-optimization lautet. Außerdem können Sie den zugewiesenen Speicher für eine DB-Instance nicht ändern, wenn er in den letzten sechs Stunden geändert wurde.	RDS-API-Parameter: `AllocatedStorage`	möchten, tritt sie sofort auf. Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.	iten auf. Die Leistung kann während des Änderungsvorgangs vermindert sein.	

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Der maximal zulässige Speicherplatz hängt von Ihrer DB-Engine und dem Speichertyp ab. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.				

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Konfiguration der Architektur Eine Konfiguration, die mehrere Tenant-Datenbanken in Ihrer DB-Instance zulässt. Derzeit unterstützen nur RDS für Oracle-Container-Datenbanken (CDBs) diese Einstellung. Wenn Ihre CDB die Single-Tenant-Konfiguration aufweist, können Sie sie so ändern, dass sie die Multi-Tenant-Konfiguration verwendet. In dieser Konfiguration können Sie RDS verwenden, APIs um 1—30 Mandantendatenbanken zu erstellen, abhängig von der Datenbankedition und den erforderlichen Optionslizenzen. Anwendung PDBs und Proxy PDBs werden nicht unterstützt. Die Multi-Tenant-Konfiguration ist dauerhaft, Sie können Ihre CDB also später nicht wieder in die Single-Tenant-Konfiguration konvertieren. **Note** Die Amazon RDS-Konfiguration wird als „Multi-	CLI-Option: `--multi-tenant` (Multi-Tenant-Konfiguration der CDB-Architektur) `--no-multi-tenant` (Single-Tenant-Konfiguration der CDB-Architektur) API-Parameter: `MultiTenant`	Die Änderung wird sofort übernommen.	Während dieser Änderung treten keine Ausfallzeiten auf.	Oracle

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Tenant“ und nicht als „Multitenant“ bezeichnet, da es sich um eine Funktion von Amazon RDS handelt, nicht nur um die Oracle-DB-Engine. In ähnlicher Weise bezieht sich der RDS-Begriff „Mandant“ auf jeden Mandanten in einer RDS-Konfiguration, nicht nur auf Oracle. PDBs In der RDS-Dokumentation bezieht sich der Begriff „Oracle Multitenant“ ohne Bindestrich ausschließlich auf die CDB-Architektur der Oracle-Datenbank, die sowohl mit lokalen als auch mit RDS-Bereitstellungen kompatibel ist. Weitere Informationen finden Sie unter Überblick über RDS for Oracle CDBs.				

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Architektur-Einstellungen Die Architektur der Oracle-Datenbank: CDB oder Nicht-CDB. Wenn Sie Oracle-Multitenant-Architektur auswählen, konvertiert RDS für Oracle Ihre Nicht-CDB in eine CDB, die die Single-Tenant-Konfiguration verwendet. Diese Einstellung wird nur unterstützt, wenn es sich bei Ihrer Datenbank um eine Nicht-CDB handelt, die Oracle Database 19c mit der RU von April 2021 oder höher ausführt. Nach der Konvertierung enthält Ihre CDB eine erste Pluggable Database (PDB). Die Änderung der Architektur ist dauerhaft, Sie können Ihre CDB also nicht wieder in eine Nicht-CDB konvertieren. **Note** Um eine CDB in der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration zu konvertieren, ändern	CLI-Option: `--engine oracle-ee-cdb` (Oracle-Multitenant) `--engine oracle-se2-cdb` (Oracle-Multitenant) API-Parameter: `Engine`	Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf. Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.	Während dieser Änderung treten Ausfallzeiten auf.	Oracle

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Sie Ihre CDB-Instance erneut und wählen Sie Konfiguration für mehrere Mandanten für Ihre Konfiguration der Architektur aus. Weitere Informationen finden Sie unter Single-Tenant-Konfiguration der CDB-Architektur.				

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Automatische Nebenversions-Updates
 	
Wählen Sie „auto Upgrade der Nebenversion aktivieren“, damit Ihre DB-Instance automatisch Upgrades für bevorzugte kleinere DB-Engine-Versionen erhält, sobald sie verfügbar sind. Dies ist das Standardverhalten. Amazon RDS führt im Wartungsfenster automatische Nebenversionenupgrades durch. Wenn Sie die Option auto Nebenversions-Upgrade aktivieren nicht auswählen, wird Ihre DB-Instance nicht automatisch aktualisiert, wenn neue Nebenversionen verfügbar werden.
 	
Weitere Informationen finden Sie unter Automatisches Upgraden der Engine-Unterversion.
	CLI-Option:
	
	`--auto-minor-version-upgrade | --no-auto-minor-version-upgrade`
	
	RDS-API-Parameter:
	
	`AutoMinorVersionUpgrade`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
Sicherungs-Replikation Wählen Sie Replizierung in eine andere AWS Region aktivieren, um Backups in einer weiteren Region für die Notfallwiederherstellung zu erstellen. Wählen Sie dann die Zielregion für die zusätzlichen Backups aus.	Nicht verfügbar beim Ändern einer DB-Instance. Informationen zur Aktivierung regionsübergreifender Backups mithilfe der AWS CLI oder RDS-API finden Sie unter Aktivierung regionsübergreifender automatisierter Backups für Amazon RDS.	Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Aufbewahrungszeitraum für Backups
 	
Die Anzahl der Tage, für die automatische Backups aufbewahrt werden. Setzen Sie den Wert des Aufbewahrungszeitraums für Backups auf 0, um automatische Backups zu deaktivieren.
 	
Weitere Informationen finden Sie unter Einführung in Backups.
 	
Note
Wenn Sie AWS Backup Ihre Backups verwalten, gilt diese Option nicht. Informationen dazu AWS Backup finden Sie im AWS Backup Developer Guide.
	CLI-Option:
	
	`--backup-retention-period`
	
	RDS-API-Parameter:
	
	`BackupRetentionPeriod`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten und Sie die Einstellung von einem Nicht-Null-Wert in einen anderen Nicht-Null-Wert ändern, wird die Änderung asynchron zum nächstgelegenen Zeitpunkt angewandt. Andernfalls wird die Änderung während des nächsten Wartungsf
			Im Falle, dass Sie den Wert von Null in einen Wert ungleich Null ändern (oder umgekehrt), verursachen Sie Ausfallzeiten.
			
			Dies gilt sowohl für Single-AZ- als auch Multi-AZ-DB-Instances.
				Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
		ensters übernommen.		

Backup window
 	
Der Zeitraum, in dem automatisierte Backups der Datenbanken erstellt werden. Das Sicherungsfenster wird mit einer Startzeit (in koordinierter Weltzeit (UTC)) und einer Dauer (in Stunden) angegeben.
 	
Weitere Informationen finden Sie unter Einführung in Backups.
 	
Note
Wenn Sie diese Option AWS Backup zur Verwaltung Ihrer Backups verwenden, wird diese Option nicht angezeigt. Informationen dazu AWS Backup finden Sie im AWS Backup Entwicklerhandbuch.
	CLI-Option:
	
	`--preferred-backup-window`
	
	RDS-API-Parameter:
	
	`PreferredBackupWindow`
		Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Zertifizierungsstelle
 	
Die Zertifizierungsstelle (CA) für das Serverzertifikat, das von der DB-Instance verwendet wird.
 	
Weitere Informationen finden Sie unter .
	CLI-Option:
	
	`--ca-certificate-identifier`
	
	RDS-API-Parameter:
	
	`CACertificateIdentifier`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Ein Ausfall tritt nur auf, wenn die DB-Engine keine Rotation ohne Neustart unterstützt. Sie können den describe-db-engine-versions AWS CLI Befehl verwenden, um festzustellen, ob die DB-Engine die Rotation ohne Neustart unterstützt.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Tags zu Snapshots kopieren
 	
Wenn Sie über DB-Instance-Tags verfügen, aktivieren Sie diese Option, um sie zu kopieren, wenn Sie einen DB-Snapshot erstellen.
 	
Weitere Informationen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.
	CLI-Option:
	
	`--copy-tags-to-snapshot` oder `--no-copy-tags-to-snapshot`
	
	RDS-API-Parameter:
	
	`CopyTagsToSnapshot`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Datenbankport
 	
Der Port, den Sie für den Zugriff auf die DB-Instance verwenden möchten.
 	
Der Wert für den Port muss nicht einem der Port-Werte entsprechen, die für Optionen in der Optionsgruppe angegeben wurden, die der DB-Instance zugeordnet ist.
 	
Weitere Informationen finden Sie unter Verbindung zu einer Amazon RDS DB-Instance herstellen.
	CLI-Option:
	
	`--db-port-number`
	
	RDS-API-Parameter:
	
	`DBPortNumber`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Die DB-Instance wird sofort neu gestartet.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

DB-Engine-Version
 	
Die Versionsnummer der DB-Engine, die Sie verwenden möchten. Bevor Sie die Produktions-DB-Instance aktualisieren, empfehlen wir, den Upgrade-Prozess auf einer Test-DB-Instance zu testen. Auf diese Weise können Sie die Dauer überprüfen und Ihre Anwendungen validieren.
 	
Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.
	CLI-Option:
	
	`--engine-version`
	
	RDS-API-Parameter:
	
	`EngineVersion`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

DB-Instance-Klasse
 	
Die zu verwendende DB-Instance-Klasse.
 	
Weitere Informationen finden Sie unter .
	CLI-Option:
	
	`--db-instance-class`
	
	RDS-API-Parameter:
	
	`DBInstanceClass`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

DB-Instance-Kennung
 	
Die neue DB-Instance-Kennung. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.
 	
Weitere Informationen zu den Auswirkungen einer Umbenennung der DB-Instance finden Sie unter Umbenennen einer DB-Instance.
	CLI-Option:
	
	`--new-db-instance-identifier`
	
	RDS-API-Parameter:
	
	`NewDBInstanceIdentifier`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten Ausfallzeiten auf, sofern Ihre DB-Engine-Version keinen dynamischen SSL-Upload unterstützt. Führen Sie den folgenden AWS CLI Befehl aus, um festzustellen, ob für Ihre Version ein Neustart erforderlich ist:
			
			`aws rds describe-db-engine-versions \ --default-only \ --engine your-db-engine \`
				Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
			`--query 'DBEngineVersions[*].SupportsCertificateRotationWithoutRestart'`	

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

DB-Parametergruppe
 	
Die DB-Parametergruppe, die Sie mit der DB-Instance verknüpfen möchten.
 	
Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
	CLI-Option:
	
	`--db-parameter-group-name`
	
	RDS-API-Parameter:
	
	`DBParameterGroupName`
		Die Zuordnung der neuen DB-Parametergruppe zur DB-Instance erfolgt sofort.	Ausfallzeiten treten nicht auf, wenn Sie Ihrer DB-Instance eine neue DB-Parametergruppe zuordnen.
			
			Die Zuordnung einer DB-Parametergruppe unterscheidet sich von der Anwendung von Parameteränderungen innerhalb einer Parametergruppe. RDS wendet geänderte statische und dynamische Parametereinstellungen in der neu verknüpften Gruppe erst an, nachdem
				Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
			Sie die DB-Instance manuell neu gestartet haben. Wenn Sie jedoch dynamische Parameter in der DB-Parametergruppe ändern, nachdem Sie sie der DB-Instance zugeordnet haben, werden diese Parametereinstellungen sofort angewendet, ohne dass ein Neustart erforderlich ist. Weitere Informationen erhalten Sie unter Parameter	

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
			gruppen für Amazon RDS und Neustarten einer DB-Instance.	

Dediziertes Protokoll-Volume
 	
Verwenden Sie ein dediziertes Protokoll-Volume (DLV), um Datenbank-Transaktionsprotokolle auf einem Speicher-Volume zu speichern, das von dem Volume mit den Datenbanktabellen getrennt ist.
 	
Weitere Informationen finden Sie unter Verwendung eines dedizierten Protokoll-Volumes (DLV).
	CLI-Option:
	
	`-dedicated-log-volume`
	
	RDS-API-Parameter:
	
	`DedicatedLogVolume`
		Die Änderung wird angewendet, wenn die DB-Instance neu gestartet wird.	Während des Neustarts der DB-Instance kommt es zu Ausfallzeiten.	MariaDB, MySQL, PostgreSQL

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Löschschutz
 	
Um zu verhindern, dass die DB-Instance gelöscht wird, können Sie die Option Enable deletion protection (Löschschutz aktivieren) aktivieren.
 	
Weitere Informationen finden Sie unter Löschen einer DB-Instance.
	CLI-Option:
	
	`--deletion-protection|--no-deletion-protection`
	
	RDS-API-Parameter:
	
	`DeletionProtection`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Verbesserte Überwachung
 	
Aktivieren Sie die erweiterte Überwachung, um das Sammeln von Metriken in Echtzeit für das Betriebssystem zu ermöglichen, auf dem Ihre DB-Instance ausgeführt wird.
 	
Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).
	CLI-Option:
	
	`--monitoring-interval` und `--monitoring-role-arn`
	
	RDS-API-Parameter:
	
	`MonitoringInterval` und `MonitoringRoleArn`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

IAM-DB-Authentifizierung
 	
Aktivieren Sie die IAM-DB-Authentifizierung, um Datenbankbenutzer über Benutzer und Rollen zu authentifizieren.
 	
Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.
	CLI-Option:
	
	`--enable-iam-database-authentication|--no-enable-iam-database-authentication`
	
	RDS-API-Parameter:
	
	`EnableIAMDatabaseAuthentication`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten keine Ausfallzeiten auf.	Nur MariaDB, MySQL und PostgreSQL

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Kerberos-Authentifizierung
 	
Wählen Sie das Active Directory, in das die DB-Instance verschoben werden soll. Das Verzeichnis muss vor diesem Vorgang bereits vorhanden sein. Wenn ein Verzeichnis bereits ausgewählt ist, können Sie None (Keine) angeben, um die DB-Instance aus dem aktuellen Verzeichnis zu entfernen.
 	
Weitere Informationen finden Sie unter Kerberos-Authentifizierung.
	CLI-Option:
	
	`--domain` und `--domain-iam-role-name`
	
	RDS-API-Parameter:
	
	`Domain` und `DomainIAMRoleName`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung kommt es zu einer kurzen Ausfallzeit.	Nur Microsoft SQL Server, MySQL, Oracle und PostgreSQL

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

License model
 	
Wählen Sie bring-your-own-license, ob Sie Ihre Lizenz für Db2 und Oracle verwenden möchten.
 	
Wählen Sie license-included (Lizenz enthalten) aus, um die allgemeine Lizenzvereinbarung für Microsoft SQL Server oder Oracle zu verwenden.
 	
Weitere Informationen finden Sie unter Lizenzierungsoptionen für Amazon RDS für Db2, Lizenzierung Microsoft SQL Server auf Amazon RDS und RDSfür Oracle-Lizenzoptionen.
	CLI-Option:
	
	`--license-model`
	
	RDS-API-Parameter:
	
	`LicenseModel`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten Ausfallzeiten auf.	Nur Microsoft SQL Server und Oracle

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Protokollexporte
 	
Die Typen von Datenbank-Protokolldateien, die in Amazon CloudWatch Logs veröffentlicht werden sollen.
 	
Weitere Informationen finden Sie unter Datenbankprotokolle in Amazon CloudWatch Logs veröffentlichen.
	CLI-Option:
	
	`--cloudwatch-logs-export-configuration`
	
	RDS-API-Parameter:
	
	`CloudwatchLogsExportConfiguration`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Wartungsfenster
 	
Der Zeitraum, in dem die Systemwartung durchgeführt wird. Die Systemwartung umfasst auch Upgrades (sofern verfügbar). Das Wartungsfenster wird mit einer Startzeit (in koordinierter Weltzeit (UTC)) und einer Dauer (in Stunden) angegeben.
 	
Wenn Sie das Fenster auf die aktuelle Zeit einstellen, müssen zwischen der aktuellen Zeit und dem Ende des Fensters mindestens 30 Minuten liegen. Dies trägt dazu bei, sicherzustellen, dass alle ausstehenden Änderungen übernommen werden.
 	
Weitere Informationen finden Sie unter Amazon RDS-Wartungsfenster.
	CLI-Option:
	
	`--preferred-maintenance-window`
	
	RDS-API-Parameter:
	
	`PreferredMaintenanceWindow`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Sofern eine oder mehrere Aktionen ausstehen, die Ausfallzeiten verursachen, und Sie das Wartungsfenster auf die aktuelle Zeit ändern, werden die ausstehenden Aktionen sofort angewendet und es kommt zu Ausfallzeiten.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Hauptanmeldedaten verwalten in AWS Secrets Manager
 	
Wählen Sie Master-Anmeldeinformationen verwalten in AWS Secrets Manager aus, um das Hauptbenutzerpasswort in Secrets Manager geheim zu verwalten.
 	
Wählen Sie optional einen KMS-Schlüssel zum Schutz des Secrets aus. Wählen Sie aus den KMS-Schlüsseln in Ihrem Konto oder geben Sie den Schlüssel eines anderen Kontos ein.
 	
Wenn RDS bereits das Hauptbenutzerpasswort für die DB-Instance verwaltet, können Sie dieses Passwort mit der Option Rotate secret immediately (Sofortige Secret-Drehung) rotieren.
 	
Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.
	CLI-Option:
	
	`--manage-master-user-password | --no-manage-master-user-password`
	
	`--master-user-secret-kms-key-id`
	
	`--rotate-master-user-password | --no-rotate-master-user-password`
	
	RDS-API-Parameter:
	
	`ManageMasterUserPassword`
		Wenn Sie die automatische Passwortverwaltung für Hauptbenutzer ein- oder ausschalten, erfolgt die Änderung sofort.
		Bei dieser Änderung wird die Einstellung zum sofortigen Anwenden ignoriert.
		
		Wenn Sie das Hauptbenutzerpasswort ändern, müssen Sie angeben, dass die Änderung sofort übernommen wird.
			Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

	`MasterUserSecretKmsKeyId`
	
	`RotateMasterUserPassword`			

Multi-AZ-Bereitstellung
 	
Yes (Ja), um Ihre DB-Instance in mehreren Availability Zones bereitzustellen. Andernfalls No (Nein).
 	
Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.
	CLI-Option:
	
	`--multi-az|--no-multi-az`
	
	RDS-API-Parameter:
	
	`MultiAZ`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten keine Ausfallzeiten auf. Es kann jedoch zur Beeinträchtigung der Leistung kommen. Weitere Informationen finden Sie unter Konvertierung einer DB-Instance in eine Multi-AZ-Bereitstellung für Amazon RDS.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Network type (Netzwerktyp)
 	
Die von der DB-Instance unterstützten IP-Adressierungsprotokolle.
 	
IPv4um anzugeben, dass Ressourcen nur über das Adressierungsprotokoll Version 4 (IPv4) mit der DB-Instance kommunizieren können.
 	
Dual-Stack-Modus, um anzugeben, dass Ressourcen mit der DB-Instance über IPv4 Internet Protocol Version 6 (IPv6) oder beides kommunizieren können. Verwenden Sie den Dual-Stack-Modus, wenn Sie über Ressourcen verfügen, die über das IPv6 Adressierungsprotokoll mit Ihrer DB-Instance kommunizieren müssen. Stellen Sie außerdem sicher, dass Sie allen Subnetzen in der von Ihnen angegebenen DB-Subnetzgruppe einen IPv6 CIDR-Block zuordnen.
 	
Weitere Informationen finden Sie unter Amazon-RDS-IP-Adressierung.
	CLI-Option:
	
	`--network-type`
	
	RDS-API-Parameter:
	
	`NetworkType`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung sind Ausfallzeiten möglich.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Neues Master-Passwort
 	
Das Passwort für den Hauptbenutzer. Das Passwort muss 8–41 alphanumerische Zeichen enthalten.
	CLI-Option:
	
	`--master-user-password`
	
	RDS-API-Parameter:
	
	`MasterUserPassword`
		Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Option group
 	
Die Optionsgruppe, die der DB-Instance zugeordnet werden soll.
 	
Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.
	CLI-Option:
	
	`--option-group-name`
	
	RDS-API-Parameter:
	
	`OptionGroupName`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten keine Ausfallzeiten auf. Eine Ausnahme ist das Hinzufügen des MariaDB-Audit-Plugins zu einer DB-Instance von RDS für MariaDB oder RDS für MySQL, das zu einem Ausfall führen kann.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Performance Insights
 	
Aktivieren Sie Performance Insights, um die Auslastung Ihrer DB-Instance zu überwachen, damit Sie Ihre Datenbankleistung analysieren und Fehler beheben können.
 	
Performance Insights ist für einige DB-Engine-Versionen und DB-Instance-Klassen nicht verfügbar. Der Abschnitt Performance-Insights wird nicht in der Konsole angezeigt, wenn er für Ihre DB-Instance nicht verfügbar ist.
 	
Weitere Informationen erhalten Sie unter Überwachung mit Performance Insights auf Amazon RDS und DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance Insights.
	CLI-Option:
	
	`--enable-performance-insights| --no-enable-performance-insights`
	
	RDS-API-Parameter:
	
	`EnablePerformanceInsights`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle außer Db2

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Performance Insights AWS KMS key
 	
Die AWS KMS Schlüssel-ID AWS KMS key für die Verschlüsselung von Performance Insights Insights-Daten. Die Schlüssel-ID ist der Amazon-Ressourcenname (ARN), die AWS KMS Schlüssel-ID oder der Schlüsselalias für den KMS-Schlüssel.
 	
Weitere Informationen finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten.
	CLI-Option:
	
	`--performance-insights-kms-key-id`
	
	RDS-API-Parameter:
	
	`PerformanceInsightsKMSKeyId`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle außer Db2

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Performance-Insights-Aufbewahrungszeitraum
 	
Der Zeitraum in Tagen, über den Performance Insights-Daten aufbewahrt werden sollen. Die Aufbewahrungseinstellung ist Standard (7 Tage). Um Ihre Leistungsdaten länger aufzubewahren, geben Sie 1–24 Monate an. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.
 	
Weitere Informationen finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten.
	CLI-Option:
	
	`--performance-insights-retention-period`
	
	RDS-API-Parameter:
	
	`PerformanceInsightsRetentionPeriod`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle außer Db2

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Prozessorfunktionen
 	
Die Zahl der CPU-Kerne und Threads pro Kern für die DB-Instance-Klasse des DB-Instance.
 	
Weitere Informationen finden Sie unter Konfigurieren des Prozessors für eine DB-Instance-Klasse in RDS für Oracle.
	CLI-Option:
	
	`--processor-features` und `--use-default-processor-features` | `--no-use-default-processor-features`
	
	RDS-API-Parameter:
	
	`ProcessorFeatures` und `UseDefaultProcessorFeatures`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten Ausfallzeiten auf.	Nur Oracle

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Bereitgestellte IOPS
 	
Der bereitgestellte IOPS (I/O-Operationen pro Sekunde)-Wert für die DB-Instance. Diese Einstellung ist nur verfügbar, wenn Sie für Storage type (Speichertyp) eine der folgenden Optionen auswählen:
 	
• General purpose SSD (gp3) (Allzweck SSD (gp3))
• Provisioned IOPS SSD (io1) (Bereitgestellte IOPS SSD (io1))
• Bereitgestellte IOPS-SSD (io2)
 	
Weitere Informationen erhalten Sie unter the section called “Bereitgestellter IOPS-Speicher” und the section called “GP3-Speicher (empfohlen)”.
	CLI-Option:
	
	`--iops`
	
	RDS-API-Parameter:
	
	`Iops`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Öffentlicher Zugriff
 	
Publicly accessible (Öffentlich zugänglich), um der DB-Instance eine öffentliche IP-Adresse zuzuweisen (was bedeutet, dass sie von außerhalb der VPC aus zugänglich ist). Damit der öffentliche Zugriff für eine DB-Instance möglich ist, muss sie sich auch in einem öffentlichen Subnetz der VPC befinden.
 	
Not publicly accessible (Nicht öffentlich zugänglich), um die DB-Instance nur innerhalb der VPC zugänglich zu machen.
 	
Weitere Informationen finden Sie unter Ausblenden einer DB-Instance in einer VPC vor dem Internet.
 	
Wenn Sie eine Verbindung mit einer DB-Instance von außerhalb ihrer VPC herstellen möchten, muss die DB-Instance öffentlich zugänglich sein. Außerdem muss der Zugriff unter Verwendung der Regeln für eingehenden Datenverkehr der Sicherheitsgruppe der DB-Instance
	CLI-Option:
	
	`--publicly-accessible | --no-publicly-accessible`
	
	RDS-API-Parameter:
	
	`PubliclyAccessible`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
gewährt werden. Darüber hinaus müssen andere Anforderungen erfüllt sein. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden. Wenn Ihre DB-Instance nicht öffentlich zugänglich ist, können Sie auch eine AWS Site-to-Site VPN-Verbindung oder eine Verbindung verwenden, um von einem AWS Direct Connect privaten Netzwerk aus darauf zuzugreifen. Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.				

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Sicherheitsgruppe
 	
Die VPC-Sicherheitsgruppe, die der DB-Instance zugeordnet werden soll.
 	
Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
	CLI-Option:
	
	`--vpc-security-group-ids`
	
	RDS-API-Parameter:
	
	`VpcSecurityGroupIds`
		Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Automatische Speicherskalierung
 	
Aktivieren Sie die automatische Skalierung des Speichers, um Amazon RDS zu ermöglichen, den Speicher bei Bedarf automatisch zu erhöhen. So wird vermieden, dass Ihrer DB-Instance der Speicherplatz ausgeht.
 	
Verwenden Sie Maximum storage threshold (Maximaler Speicherschwellenwert), um die Obergrenze für Amazon RDS festzulegen, bei der der Speicherplatz für Ihre DB-Instance automatisch vergrößert wird. Der Standardwert ist 1.000 GiB.
 	
Weitere Informationen finden Sie unter Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung.
	CLI-Option:
	
	`--max-allocated-storage`
	
	RDS-API-Parameter:
	
	`MaxAllocatedStorage`
		Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Storage throughput (Speicherdurchsatz)
 	
Der neue Speicherdurchsatzwert für die DB-Instance. Diese Einstellung ist nur verfügbar, wenn Sie für Storage type (Speichertyp) die Option General purpose SSD (gp3) (Allzweck-SSD (gp3)) auswählen.
 	
Weitere Informationen finden Sie unter the section called “GP3-Speicher (empfohlen)”.
	CLI-Option:
	
	`--storage-throughput`
	
	RDS-API-Parameter:
	
	`StorageThroughput`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten keine Ausfallzeiten auf.	Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

Speichertyp
 	
Der Speichertyp, der verwendet werden soll.
 	
Wenn Sie sich für General Purpose SSD (gp3) (Allzweck-SSD (gp3)) entscheiden, können Sie unter Advanced Settings (Erweiterte Einstellungen) zusätzliche Provisioned IOPS (Bereitgestellte IOPS) und Storage throughput (Speicherdurchsatz) bereitstellen.
 	
Wenn Sie Provisioned IOPS SSD (io1) oder Provisioned IOPS SSD (io2) wählen, geben Sie den Wert Provisioned IOPS ein.
 	
Nachdem Amazon RDS beginnt, Ihre DB-Instance zu modifizieren, um die Speichergröße oder den Speichertyp zu ändern, können Sie sechs Stunden lang keine weiteren Anfrage zur Änderung der Speichergröße, der Leistung oder des Speichertyps stellen.
 	
Weitere Informationen finden Sie unter Amazon RDS-Speichertypen.
	CLI-Option:
	
	`--storage-type`
	
	RDS-API-Parameter:
	
	`StorageType`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Die folgenden Änderungen führen zu kurzen Ausfallzeiten, bis der Vorgang gestartet wird. Anschließend können Sie die Datenbank normal verwenden, während die Änderungen durchgeführt werden.
			
			• Von General Purpose (SSD) (Allgemeine Zwecke (SSD)) oder Provisioned IOPS (SSD) (Bereitgestellte IOPS (SSD)) nach Magnetic
				Alle DB-Engines

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines
			(Magnetfestplatten). • Von Magnetic (Magnetfestplatten) nach General Purpose (SSD) (Allgemeine Zwecke (SSD)) oder Provisioned IOPS (SSD) (Bereitgestellte IOPS (SSD)).	

Konsoleneinstellung und Beschreibung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit	Unterstützte DB-Engines

DB-Subnetzgruppe
 	
Die DB-Subnetzgruppe für die DB-Instance. Sie können diese Einstellung verwenden, um Ihre DB-Instance in eine andere VPC zu verschieben.
 	
Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.
	CLI-Option:
	
	`--db-subnet-group-name`
	
	RDS-API-Parameter:
	
	`DBSubnetGroupName`
		Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
		
		Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
			Während dieser Änderung treten Ausfallzeiten auf.	Alle DB-Engines

Warten einer DB-Instance

Amazon RDS führt in regelmäßigen Abständen Wartungsarbeiten an Amazon RDS-Ressourcen durch. In den folgenden Themen werden diese Wartungsmaßnahmen und ihre Anwendung beschrieben.

Überblick über die Wartungsupdates für DB-Instances

Die Wartung beinhaltet in den meisten Fällen Aktualisierungen der folgenden Ressourcen in Ihrer DB-Instance:

- Zugrundeliegende Hardware
- Zugrundeliegendes Betriebssystem
- Datenbank-Engine-Version

Häufig werden Betriebssystemupdates wegen Sicherheitsproblemen herausgegeben. Wir empfehlen, dass Sie sie so schnell wie möglich durchführen. Weitere Informationen zu Betriebssystemupdates finden Sie unter the section called “Anwenden von Updates”.

Themen

- Offline-Ressourcen während Wartungsupdates
- Aufgeschobene Änderungen an der DB-Instance
- Eventuelle Konsistenz für die API DescribePendingMaintenanceActions

Offline-Ressourcen während Wartungsupdates

Einige Wartungselemente erfordern, dass Amazon RDS Ihre DB-Instance für kurze Zeit in den Offlinebetrieb versetzt. Zu den Wartungselementen, für die eine Ressource offline sein muss, gehört z. B. das Ausführen erforderlicher Patches für das Betriebssystem oder die Datenbank. Das erforderliche Patching wird automatisch und nur für Patches eingeplant, welche die Sicherheit und Instance-Zuverlässigkeit betreffen. Solche Patches treten selten auf, in der Regel einmal alle paar Monate. Es ist selten mehr als ein Bruchteil Ihres Wartungsfensters dafür erforderlich.

Aufgeschobene Änderungen an der DB-Instance

Aufgeschobene Änderungen der DB-Instance, die nicht sofort zur Anwendung kommen sollen, werden während des Wartungszeitraums umgesetzt. Sie können z. B. während des

Wartungszeitraums die DB-Instance-Klasse oder die Parametergruppe ändern. Solche Änderungen, die Sie mit der Einstellung für ausstehenden Neustart angeben, werden nicht in der Liste der ausstehenden Wartungsarbeiten angezeigt. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Um die Änderungen zu sehen, die für das nächste Wartungsfenster noch ausstehen, verwenden Sie den describe-db-instances AWS CLI Befehl und überprüfen Sie das `PendingModifiedValues` Feld.

Eventuelle Konsistenz für die API DescribePendingMaintenanceActions

Die Amazon `DescribePendingMaintenanceActions` RDS-API folgt einem eventuellen Konsistenzmodell. Das bedeutet, dass das Ergebnis des `DescribePendingMaintenanceActions` Befehls möglicherweise nicht sofort für alle nachfolgenden RDS-Befehle sichtbar ist. Beachten Sie dies, wenn Sie `DescribePendingMaintenanceActions` unmittelbar nach der Verwendung eines vorherigen API-Befehls verwenden.

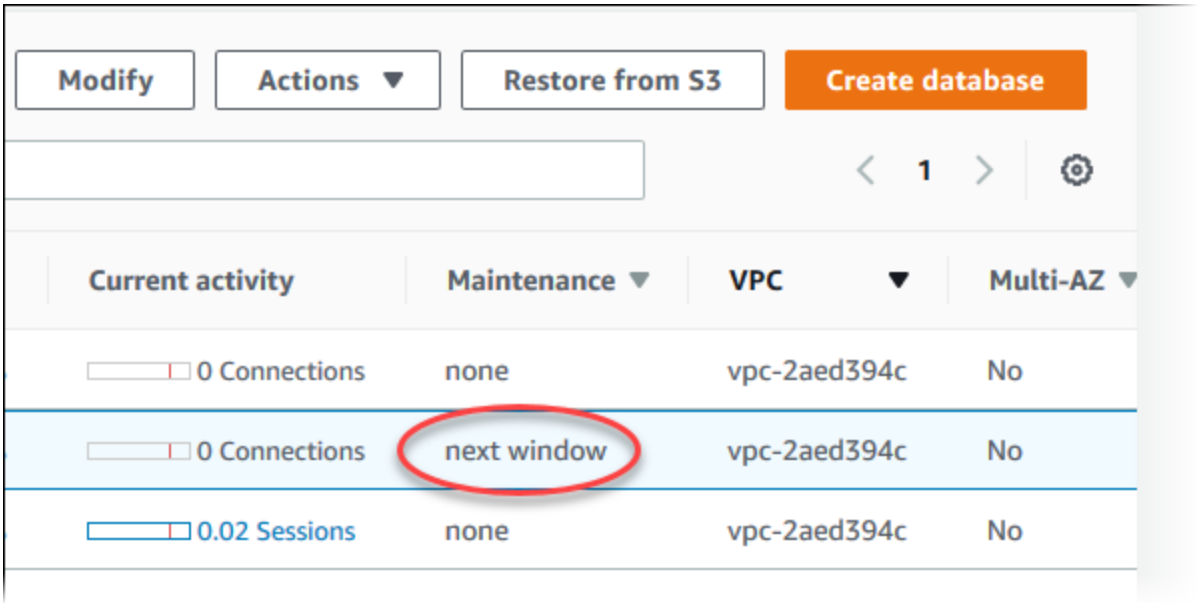
Eine eventuelle Konsistenz kann sich auf die Art und Weise auswirken, wie Sie Ihre Wartungsupdates verwaltet haben. Wenn Sie beispielsweise den `ApplyPendingMaintenanceActions` Befehl ausführen, um die Version der Datenbank-Engine für einen zu `DescribePendingMaintenanceActions` aktualisieren, wird dieser irgendwann für Sie sichtbar sein. In diesem Szenario `DescribePendingMaintenanceActions` könnte sich zeigen, dass die Wartungsaktion nicht angewendet wurde, obwohl sie angewendet wurde.

Um die eventuelle Konsistenz zu gewährleisten, können Sie wie folgt vorgehen:

- Bestätigen Sie den Status Ihres , bevor Sie einen Befehl ausführen, um ihn zu ändern. Führen Sie den entsprechenden `DescribePendingMaintenanceActions` Befehl mit einem exponentiellen Backoff-Algorithmus aus, um sicherzustellen, dass Sie genügend Zeit für die Ausbreitung des vorherigen Befehls im System einplanen. Führen Sie dazu den `DescribePendingMaintenanceActions` Befehl wiederholt aus, wobei Sie mit einer Wartezeit von einigen Sekunden beginnen und die Wartezeit schrittweise auf bis zu fünf Minuten erhöhen.
- Fügen Sie Wartezeiten zwischen aufeinanderfolgenden Befehlen hinzu, auch wenn ein `DescribePendingMaintenanceActions` Befehl eine genaue Antwort zurückgibt. Wenden Sie einen exponentiellen Backoff-Algorithmus an, der mit einer Wartezeit von einigen Sekunden beginnt, und erhöhen Sie die Wartezeit schrittweise auf etwa fünf Minuten.

Ausstehende Wartungsupdates anzeigen

Prüfen Sie mithilfe der RDS-Konsole, der oder der RDS-API, ob ein Wartungsupdate für Ihren verfügbar ist. AWS CLI Wenn ein Update verfügbar ist, wird es in der Spalte Wartung für den auf der Amazon RDS-Konsole angezeigt, wie in dieser Abbildung dargestellt.



Wenn für eine DB-Instance keine Aktualisierung verfügbar ist, lautet ihr bzw. sein Spaltenwert none (keine).

Wenn eine Wartungsaktualisierung für eine DB-Instance verfügbar ist, sind die folgenden Spaltenwerte möglich:

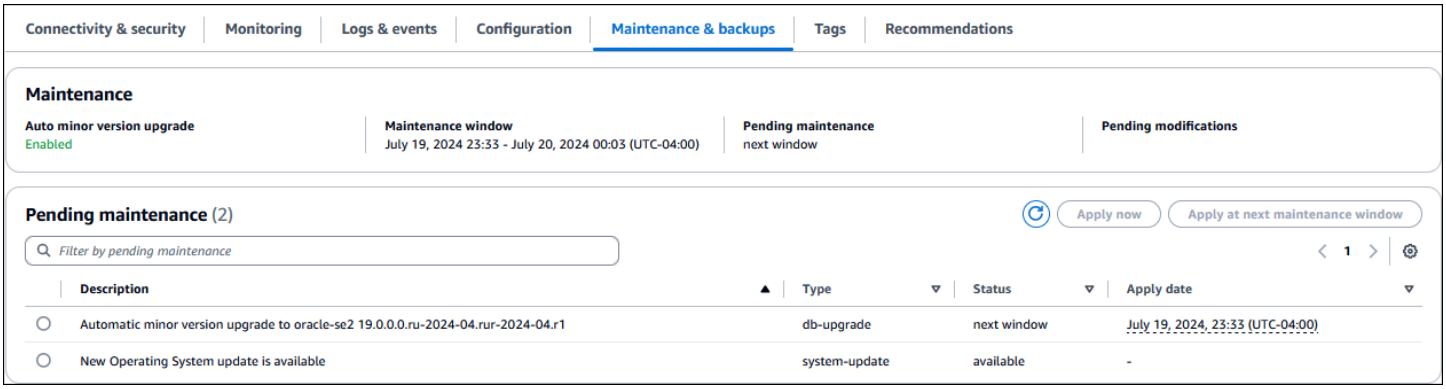
- required (erforderlich) – Die Wartungsaktion wird auf die Ressource angewendet und kann nicht unbegrenzt aufgeschoben werden.
- available (verfügbar) – Die Wartungsaktion ist verfügbar, wird aber nicht automatisch auf die Ressource angewandt. Sie können sie manuell anwenden.
- next window (nächstes Fenster) – Die Wartungsmaßnahme wird im nächsten Wartungsfenster auf die Ressource angewandt.
- In Bearbeitung — Die Wartungsaktion wird auf die Ressource angewendet.

Wenn ein Update verfügbar ist, können Sie eine der folgenden Aktionen durchführen:

- Wenn der Wartungswert „Nächstes Fenster“ lautet, verschieben Sie die Wartungsaktionen, indem Sie unter Aktionen die Option Upgrade aufschieben wählen. Sie können eine Wartungsaktion, die bereits gestartet wurde, nicht verschieben.
- Sofortiges Durchführen der Wartungsaktivitäten-
- Wenden Sie die Wartungsmaßnahmen während Ihres nächsten Wartungsfensters an.
- Keine Aktion.

Um eine Maßnahme zu ergreifen, verwenden Sie AWS Management Console

1. Wählen Sie die DB-Instance aus, um die zugehörigen Details anzuzeigen.
2. Wählen Sie Maintenance & backups (Wartung und Sicherungen). Die ausstehenden Wartungsaktionen werden angezeigt.
3. Wählen Sie die Aktion aus, die Sie ergreifen möchten, und wählen Sie dann aus, wann sie angewendet werden soll.



Der Wartungszeitraum legt fest, wann die ausstehenden Operationen gestartet werden, gibt aber keine Gesamtlaufzeit für diese Operationen vor. Wartungsarbeiten werden nicht zwingend vor Ende des Wartungszeitraums abgeschlossen. Sie können daher über die angegebene Endzeit hinaus fortgesetzt werden. Weitere Informationen finden Sie unter Amazon RDS-Wartungsfenster.

Sie können auch überprüfen, ob ein Wartungsupdate für Ihren verfügbar ist, indem Sie den describe-pending-maintenance-actions AWS CLI Befehl ausführen.

Hinweise zum Anwenden von Wartungsupdates finden Sie unterUpdates auf einen anwenden.

Wartungsmaßnahmen für Amazon RDS

Die folgenden Wartungsmaßnahmen gelten für RDS-DB-Instances:

- `ca-certificate-rotation`— Aktualisieren Sie das Zertifikat der Amazon RDS-Zertifizierungsstelle für die DB-Instance.
- `db-upgrade`— Aktualisieren Sie die DB-Engine-Version für die DB-Instance.
- `hardware-maintenance`— Führen Sie Wartungsarbeiten an der zugrunde liegenden Hardware für die DB-Instance durch.
- `system-update`— Aktualisieren Sie das Betriebssystem für die DB-Instance.

Warten der Multi-AZ-Bereitstellungen

Das Ausführen einer DB-Instance als Multi-AZ-Bereitstellung kann die Auswirkungen eines Wartungsereignisses weiter reduzieren. Dieses Ergebnis ist darauf zurückzuführen, dass Amazon RDS Betriebssystemupdates mit folgenden Schritten anwendet:

1. Durchführen der Wartung im Standby.
2. Erwägen des Standby zu Primär.
3. Durchführung der Wartung in der alten primären Instance, die zur neuen Standby-Instance wird.

Wenn Sie die Datenbank-Engine für Ihre DB-Instance in einer Multi-AZ-Bereitstellung aktualisieren, ändert Amazon RDS gleichzeitig die primären und sekundären DB-Instances. In diesem Fall sind während des Upgrades sowohl die primären als auch die sekundären DB-Instances in der Multi-AZ-Bereitstellung nicht verfügbar. Dieser Vorgang führt zu Ausfallzeiten, bis das Upgrade abgeschlossen ist. Die Dauer des Nutzungsausfalls ist von der Größe Ihrer DB-Instance abhängig.

Wenn zugrunde liegende Betriebssystem-Patches installiert werden müssen, ist ein kurzes Multi-AZ-Failover erforderlich, um die Patches auf die primäre DB-Instance anzuwenden. Dieser Failover dauert in der Regel weniger als eine Minute.

Wenn auf Ihrer DB-Instance RDS für MySQL, RDS für PostgreSQL oder RDS für MariaDB ausgeführt wird, können Sie die für ein Upgrade erforderlichen Ausfallzeiten minimieren, indem Sie eine blaue/grüne Bereitstellung verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates. Wenn Sie eine RDS for SQL Server- oder RDS Custom for SQL Server-DB-Instance in einer Multi-AZ-Bereitstellung aktualisieren, führt Amazon RDS fortlaufende Upgrades durch, sodass Sie nur für die Dauer eines Failovers einen Ausfall haben. Weitere Informationen finden Sie unter Überlegungen zu Multi-AZ.

Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Amazon RDS-Wartungsfenster

Das Wartungsfenster ist ein wöchentliches Zeitintervall, in dem alle Systemänderungen vorgenommen werden. Jeder hat ein wöchentliches Wartungsfenster. Das Wartungsfenster bietet die Möglichkeit, zu kontrollieren, wann Änderungen und Software-Patches vorgenommen werden. Weitere Informationen zur Anpassung des Wartungsfensters finden Sie unter Anpassen des bevorzugten DB-Instance-Wartungsfensters.

RDS verbraucht während der Wartung einige Ressourcen auf Ihrer DB-Instance. Sie können einen minimalen Einfluss auf die Leistung beobachten. Bei einer DB-Instance kann in seltenen Fällen ein Multi-AZ-Failover erforderlich sein, damit ein Wartungs-Update abgeschlossen werden kann.

Wenn ein Wartungsereignis für eine bestimmte Woche geplant ist, wird es während des 30-minütigen Wartungsfensters eingeleitet, das Sie festlegen. Die meisten Wartungsereignisse werden auch während des 30-minütigen Wartungsfensters abgeschlossen, obwohl größere Wartungsereignisse länger als 30 Minuten dauern können. Das Wartungsfenster wird angehalten, wenn der gestoppt wird.

Das 30-minütige Wartungsfenster wird zufällig aus einem 8-Stunden-Zeitraum pro Region ausgewählt. Wenn Sie beim Erstellen der DB-Instance kein Wartungsfenster angeben, legt RDS ein 30-minütiges Wartungsfenster an einem zufällig ausgewählten Wochentag fest.

Die folgende Tabelle zeigt die Zeitblöcke für die einzelnen AWS-Region Zeitblöcke, denen die Standard-Wartungsfenster zugewiesen sind.

Name der Region	Region	Zeitblock
USA Ost (Nord-Virginia)	us-east-1	03:00 bis 11:00 Uhr UTC
USA Ost (Ohio)	us-east-2	03:00 - 11:00 UTC
USA West (Nordkalifornien)	us-west-1	06:00 - 14:00 UTC
USA West (Oregon)	us-west-2	06:00 - 14:00 UTC
Africa (Cape Town)	af-south-1	03:00 - 11:00 UTC

Name der Region	Region	Zeitblock
Asia Pacific (Hong Kong)	ap-east-1	06:00 - 14:00 UTC
Asien-Pazifik (Hyderabad)	ap-south-2	06:30 – 14:30 Uhr UTC
Asien-Pazifik (Jakarta)	ap-southeast-3	08:00–16:00 Uhr UTC
Asien-Pazifik (Malaysia)	ap-southeast-5	09:00 — 17:00 UHR UTC
Asien-Pazifik (Melbourne)	ap-southeast-4	11:00–19:00 Uhr UTC
Asien-Pazifik (Mumbai)	ap-south-1	06:00 - 14:00 UTC
Asia Pacific (Osaka)	ap-northeast-3	22:00 - 23:59 UTC
Asia Pacific (Seoul)	ap-northeast-2	13:00 - 21:00 UTC
Asien-Pazifik (Singapur)	ap-southeast-1	14:00 - 22:00 UTC
Asien-Pazifik (Sydney)	ap-southeast-2	12:00 - 20:00 UTC
Asien-Pazifik (Taipeh)	ap-east-2	9:00 — 17:00 UTC
Asien-Pazifik (Thailand)	ap-southeast-7	8:00 — 16:00 UHR UTC
Asien-Pazifik (Tokio)	ap-northeast-1	13:00 - 21:00 UTC
Canada (Central)	ca-central-1	03:00 bis 11:00 Uhr UTC

Name der Region	Region	Zeitblock
Kanada West (Calgary)	ca-west-1	18:00 - 02:00 UTC
China (Beijing)	cn-north-1	06:00 - 14:00 UTC
China (Ningxia)	cn-northwest-1	06:00 - 14:00 UTC
Europe (Frankfurt)	eu-central-1	21:00 - 05:00 UTC
Europa (Irland)	eu-west-1	22:00 - 06:00 UTC
Europe (London)	eu-west-2	22:00 bis 06:00 Uhr UTC
Europa (Mailand)	eu-south-1	02:00 - 10:00 UTC
Europa (Paris)	eu-west-3	23:59 - 07:29 UTC
Europa (Spanien)	eu-south-2	02:00 - 10:00 UTC
Europe (Stockholm)	eu-north-1	23:00 - 07:00 UTC
Europa (Zürich)	eu-central-2	02:00 - 10:00 UTC
Israel (Tel Aviv)	il-central-1	03:00 bis 11:00 Uhr UTC
Mexiko (Zentral)	mx-central-1	19:00 — 3:00 UHR UTC
Naher Osten (Bahrain)	me-south-1	06:00 - 14:00 UTC
Naher Osten (VAE)	me-central-1	05:00–13:00 UHR UTC
Südamerika (São Paulo)	sa-east-1	00:00 - 08:00 UTC
AWS GovCloud (US-Ost)	us-gov-east-1	17:00 - 01:00 UTC

Name der Region	Region	Zeitblock
AWS GovCloud (US-West)	us-gov-west-1	06:00 - 14:00 UTC

Themen

- Anpassen des bevorzugten DB-Instance-Wartungsfensters

Anpassen des bevorzugten DB-Instance-Wartungsfensters

Das Wartungsfenster sollte in den Zeitraum mit der geringsten Nutzung fallen und daher unter Umständen von Zeit zu Zeit geändert werden. Ihre DB-Instance ist während dieser Zeit nur dann nicht verfügbar, wenn Systemänderungen (z. B. eine Änderung der DB-Instance-Klasse) durchgeführt werden und einen Ausfall erforderlich machen. Ihre DB-Instance ist nur für die minimale Zeitspanne nicht verfügbar, die für die notwendigen Änderungen benötigt wird.

Im folgenden Beispiel passen Sie das bevorzugte Wartungsfenster für eine DB-Instance an.

Für dieses Beispiel nehmen wir an, dass die DB-Instance mit dem Namen mydbinstance existiert und ein bevorzugtes Wartungsfenster von „Sun:05:00-Sun:06:00“ UTC hat.

Konsole

So passen Sie das bevorzugte Wartungsfenster an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die zu ändernde DB-Instance aus.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Aktualisieren Sie im Bereich Wartung den Wartungszeitraum.

> **Note**
>
> Das Wartungsfenster und das Sicherungsfenster für die DB-Instance können sich nicht überschneiden. Wenn Sie einen Wert für das Wartungsfenster eingeben, das sich mit dem Sicherungsfenster überschneidet, wird eine Fehlermeldung angezeigt.

5. Klicken Sie auf Weiter.

   Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen.

6. Um die Änderungen am Wartungszeitraum sofort zu übernehmen, klicken Sie auf Sofort anwenden.
7. Klicken Sie auf DB-Instance ändern, um Ihre Änderungen zu speichern.

   Klicken Sie anderenfalls auf Zurück, um Ihre Änderungen zu bearbeiten, oder klicken Sie auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Verwenden Sie den AWS CLI `modify-db-instance`Befehl mit den folgenden Parametern, um das bevorzugte Wartungsfenster anzupassen:

- `--db-instance-identifier`
- `--preferred-maintenance-window`

Example

Im folgenden Codebeispiel wird das Wartungsfenster Dienstags von 4:00 – 4:30 Uhr UTC festgelegt.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

RDS-API

Verwenden Sie die Amazon-RDS-API-Operation `ModifyDBInstance` mit den folgenden Parametern, um das bevorzugte Wartungsfenster einzustellen:

- `DBInstanceIdentifier`
- `PreferredMaintenanceWindow`

Updates auf einen anwenden

Mit Amazon RDS können Sie auswählen, zu welchem Zeitpunkt Wartungsoperationen angewendet werden sollen. Mithilfe der, oder RDS-API können Sie entscheiden AWS Management Console AWS CLI, wann Amazon RDS Updates einführt.

Konsole

Verwalten eines Update für eine DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die DB-Instance aus, für die/den ein erforderliches Update angegeben ist.
4. Wählen Sie unter Aktionen eine der folgenden Optionen:

   - Jetzt patchen
   - Im nächsten Fenster patchen

   > **Note**
   >
   > Wenn Sie „Im nächsten Fenster patchen“ wählen und das Update später verzögern möchten, können Sie „Upgrade aufschieben“ wählen. Sie können eine Wartungsaktion nicht verschieben, wenn sie bereits gestartet wurde.
   > Um eine Wartungsaktion abzubrechen, ändern Sie die DB-Instance und deaktivieren Sie Auto minor version upgrade (Automatisches Upgrade einer Unterversion).

AWS CLI

Verwenden Sie den apply-pending-maintenance-action AWS CLI Befehl, um ein ausstehendes Update auf einen anzuwenden.

Example

Für LinuxmacOS, oderUnix:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db \
```

```
    --apply-action system-update \
    --opt-in-type immediate
```

Windows:

```
aws rds apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db ^
    --apply-action system-update ^
    --opt-in-type immediate
```

> **Note**
>
> Um eine Wartungsaktion zu verschieben, geben Sie `undo-opt-in` für `--opt-in-type` an. Sie können `undo-opt-in` nicht für `--opt-in-type` angeben, wenn die Wartungsaktion bereits gestartet wurde.
>
> Um eine Wartungsaktion abzubrechen, führen Sie den modify-db-instance AWS CLI Befehl aus und geben Sie Folgendes an`--no-auto-minor-version-upgrade`.

Verwenden Sie den describe-pending-maintenance-actions AWS CLI Befehl, um eine Liste der Ressourcen zurückzugeben, für die mindestens ein Update aussteht.

Example

Für LinuxmacOS, oderUnix:

```
aws rds describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

Windows:

```
aws rds describe-pending-maintenance-actions ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

Sie können auch eine Liste von Ressourcen für einen zurückgeben, indem Sie den `--filters` Parameter des `describe-pending-maintenance-actions` AWS CLI Befehls angeben. Der Befehl hat das folgende Format für `--filters`: `Name=`*`filter-name`*`,Value=`*`resource-id`*`,...`.

Für den `Name`-Parameter eines Filters sind folgende Werte gültig:

- `db-instance-id`— Akzeptiert eine Liste von DB-Instance-Identifikatoren oder Amazon-Ressourcennamen (ARNs). Die zurückgegebene Liste enthält nur ausstehende Wartungsaktionen für die DB-Instances, die durch diese Identifikatoren oder identifiziert wurden. ARNs
- `db-cluster-id`— Akzeptiert eine Liste von DB-Cluster-Identifikatoren oder ARNs für Amazon Aurora. Die zurückgegebene Liste enthält nur ausstehende Wartungsaktionen für die DB-Cluster, die anhand dieser Kennungen oder identifiziert wurden. ARNs

In dem folgenden Beispiel wird eine Liste der aussehenden Wartungsaktionen für die DB-Instances `sample-instance1` und `sample-instance2` zurückgegeben.

Example

Für LinuxmacOS, oderUnix:

```
aws rds describe-pending-maintenance-actions \
 --filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

Windows:

```
aws rds describe-pending-maintenance-actions ^
 --filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

RDS-API

Rufen Sie die Amazon-RDS-API-Operation `ApplyPendingMaintenanceAction` auf, um ein Update auf einer DB-Instance zu installieren.

Rufen Sie die Amazon RDS-API-Operation `DescribePendingMaintenanceActions` auf, um eine Liste der Ressourcen zurückzugeben, für die mindestens ein Update aussteht.

Betriebssystem-Updates für RDS-DB-Instances

RDS für Db2, RDS für MariaDB, RDS für MySQL, RDS für PostgreSQL, RDS für SQL Server und RDS für Oracle DB-Instances erfordern gelegentlich Betriebssystemupdates. Amazon RDS aktualisiert das Betriebssystem auf eine neuere Version, um die Datenbankleistung und der allgemeine Sicherheitsstatus der Kunden zu verbessern. In der Regel dauern die Updates etwa 10 Minuten. Betriebssystem-Updates ändern nicht die DB-Engine-Version oder die DB-Instance-Klasse einer DB-Instance.

Betriebssystem-Updates können entweder optional oder obligatorisch sein:

- Ein optionales Update kann jederzeit angewendet werden. Obwohl diese Updates optional sind, empfehlen wir Ihnen, sie regelmäßig anzuwenden, um Ihre RDS-Flotte auf dem neuesten Stand zu halten. RDS wendet diese Updates nicht automatisch an.

  Wenn Sie benachrichtigt werden möchten, sobald ein neuer optionaler System-Patch verfügbar ist, können Sie RDS-EVENT-0230 in der Kategorie „Sicherheitspatch-Ereignis“ abonnieren. Informationen zum Abonnieren von RDS-Ereignissen finden Sie unter RDSAmazon-Veranstaltungsbenachrichtigung abonnieren.

  > **Note**
  >
  > `RDS-EVENT-0230` gilt nicht für Upgrades der Betriebssystemdistribution.

- Ein obligatorisches Update ist erforderlich und hat ein Anwendungsdatum. Planen Sie Ihr Update vor diesem Anwendungsdatum. Nach dem angegebenen Anwendungsdatum aktualisiert Amazon RDS das Betriebssystem für Ihre DB-Instance während eines Ihrer zugewiesenen Wartungsfenster automatisch auf die neueste Version.

> **Note**
>
> Es ist möglicherweise erforderlich, im Hinblick auf alle optionalen und obligatorischen Updates auf dem Laufenden zu bleiben, um verschiedene Compliance-Auflagen zu erfüllen. Wir empfehlen Ihnen, alle von RDS zur Verfügung gestellten Updates während Ihrer Wartungsfenster routinemäßig anzuwenden.

Sie können das AWS Management Console oder das verwenden, AWS CLI um Informationen über die Art des Betriebssystem-Upgrades zu erhalten.

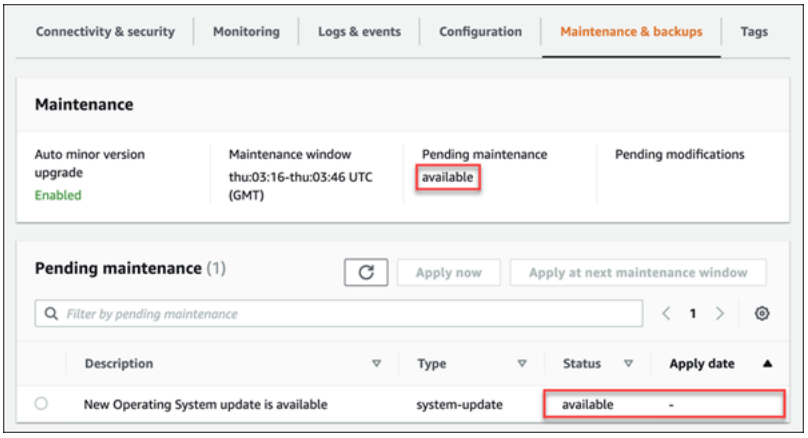
Konsole

Um Aktualisierungsinformationen zu erhalten, verwenden Sie den AWS Management Console

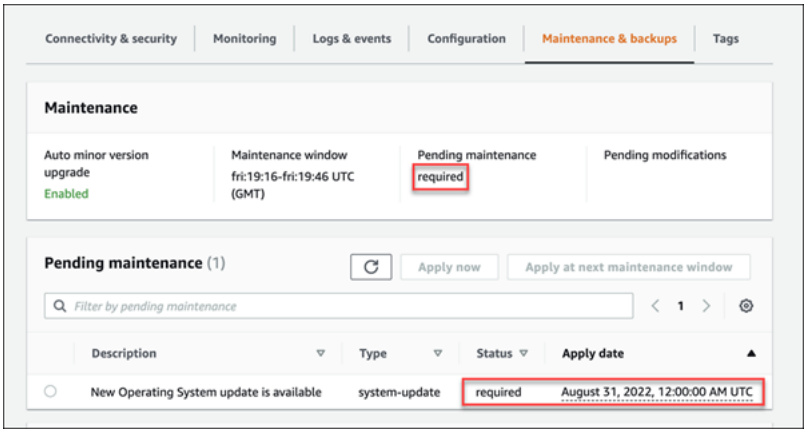
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und anschließend die MySQL-DB-Instance aus.
3. Wählen Sie Maintenance & backups (Wartung und Sicherungen).

4. Suchen Sie im Abschnitt Pending maintenance (Ausstehende Wartungen) das Betriebssystem-Update und überprüfen Sie den Wert Status.

Bei einem optionalen Update ist der AWS Management ConsoleWartungsstatus auf verfügbar gesetzt und es gibt kein Anwendungsdatum, wie in der folgenden Abbildung dargestellt.



Bei einem obligatorischen Update ist der Status für die Wartung auf required (erforderlich) festgelegt und ein Apply date (Datum übernehmen) angegeben, wie in der folgenden Abbildung gezeigt.



AWS CLI

Verwenden Sie den describe-pending-maintenance-actionsBefehl AWS CLI, um Aktualisierungsinformationen von zu erhalten.

```
aws rds describe-pending-maintenance-actions
```

Ein obligatorisches Betriebssystem-Update enthält einen `AutoAppliedAfterDate`- und einen `CurrentApplyDate`-Wert. Ein optionales Betriebssystem-Update enthält diese Werte nicht.

Die folgende Ausgabe zeigt ein obligatorisches Betriebssystem-Update.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Die folgende Ausgabe zeigt ein optionales Betriebssystem-Update.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Verfügbarkeit von Betriebssystem-Updates

Betriebssystem-Updates sind spezifisch für die DB-Engine-Version und die DB-Instance-Klasse. Daher erhalten oder erfordern DB-Instances Updates zu verschiedenen Zeiten. Wenn für Ihre DB-Instance basierend auf der Engine-Version und der Instance-Klasse ein Betriebssystemupdate erforderlich ist, wird das erforderliche Update in der Konsole angezeigt. Es kann auch angezeigt werden, indem describe-pending-maintenance-actions AWS CLI Sie den Befehl ausführen oder den DescribePendingMaintenanceActionsRDS-API-Vorgang aufrufen. Wenn für Ihre Instance ein Update verfügbar ist, können Sie Ihr Betriebssystem aktualisieren, indem Sie den Anweisungen unter Updates auf einen anwenden folgen.

Upgrade der Engine-Version für eine DB-Instance

Amazon RDS bietet neuere Versionen jeder unterstützten Datenbank-Engine, sodass Sie Ihren behalten können up-to-date. Neuere Versionen können Korrekturen, Sicherheitsverbesserungen und andere Verbesserungen der Datenbank-Engine beinhalten. Wenn von Amazon RDS eine neue Version einer Datenbank-Engine unterstützt wird, können Sie festlegen, wie und wann ein Upgrade für die DB-Instances Ihrer Datenbank durchgeführt wird.

Es gibt zwei Arten von Upgrades: Upgrades von Hauptversionen und Upgrades von Nebenversionen. Generell können in Engine-Hauptversions-Upgrades Änderungen enthalten sein, die nicht mit vorhandenen Anwendungen kompatibel sind. Im Gegensatz hierzu enthalten Nebenversions-Upgrades nur Änderungen, die abwärtskompatibel mit bestehenden Anwendungen sind.

Bei Multi-AZ-DB-Clustern werden Hauptversions-Upgrades nur für RDS für PostgreSQL unterstützt. Nebenversions-Upgrades werden für alle Engines unterstützt, die Multi-AZ-DB-Cluster unterstützen. Weitere Informationen finden Sie unter the section called “Aktualisierung eines Multi-AZ-DB-Clusters”.

Die Reihenfolge der Versionsnummerierung ist für jede Datenbank-Engine spezifisch festgelegt. Beispielsweise sind RDS für MySQL 5.7 und 8.0 Engine-Hauptversionen, sodass das Upgrade von einer beliebigen 5.7-Version auf eine beliebige 8.0-Version ein Hauptversions-Upgrade darstellt. RDS für MySQL Version 5.7.22 und 5.7.23 sind Engine-Unterversionen, wodurch es sich beim Upgrade von Version 5.7.22 auf 5.7.23 um ein Unterversions-Upgrade handelt.

> **Important**
>
> Sie können eine DB-Instance nicht ändern, wenn sie aktualisiert wird. Während eines Upgrades lautet der Status der DB-Instance `upgrading`.

Weitere Informationen zu Haupt- und Unterversions-Upgrades für eine bestimmte DB-Engine finden Sie in der folgenden Dokumentation für Ihre DB-Engine:

- Upgrades der MariaDB-DB-Engine
- Upgrades der Microsoft SQL Server-DB-Engine
- Upgrades der DB-Engine RDS für MySQL
- Aktualisieren der DB-Engine von RDS für Oracle
- Upgrades der RDS for PostgreSQL-DB-Engine

Bei größeren Versions-Upgrades müssen Sie die DB-Engine-Version manuell über die AWS Management Console, AWS CLI, oder RDS-API ändern. Bei Unterversions-Upgrades besteht die Möglichkeit, die Engine-Version manuell zu ändern oder die Option Automatisches Unterversion-Upgrade zu aktivieren.

> **Note**
>
> Datenbank-Engine-Upgrades erfordern Ausfallzeiten. Sie können die für das DB-Instance-Upgrade erforderlichen Ausfallzeiten minimieren, indem Sie eine Blau/Grün-Bereitstellung verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Themen

- Manuelles Upgraden der Engine-Version
- Automatisches Upgraden der Engine-Unterversion

Manuelles Upgraden der Engine-Version

Um die Engine-Version einer DB-Instance manuell zu aktualisieren, können Sie die AWS Management Console AWS CLI, oder die RDS-API verwenden.

Konsole

So upgraden Sie die Engine-Version einer DB-Instance über die Konsole:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance aus, die Sie upgraden möchten.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Wählen Sie unter DB Engine Version die neue Version aus.
5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.
6. Entscheiden Sie, wann Sie Ihr Upgrade planen möchten:

- Um die Änderungen in die Warteschlange für ausstehende Änderungen zu stellen, wählen Sie im nächsten geplanten Wartungsfenster die Option Anwenden. Während des nächsten Wartungsfensters wendet RDS alle ausstehenden Änderungen in der Warteschlange an.
- Wählen Sie Apply immediately, um die Änderungen sofort anzuwenden. Die Auswahl dieser Option kann in einigen Fällen einen Ausfall verursachen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

7. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern.

   Klicken Sie anderenfalls auf Zurück, um Ihre Änderungen zu bearbeiten, oder klicken Sie auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Verwenden Sie den modify-db-instanceCLI-Befehl, um die Engine-Version einer DB-Instance zu aktualisieren. Geben Sie die folgenden Parameter an:

- `--db-instance-identifier`: der Name der DB-Instance
- `--engine-version`: die Versionsnummer der Datenbank-Engine, auf die das Upgrade durchgeführt wird

  Verwenden Sie den AWS CLI describe-db-engine-versionsBefehl, um Informationen zu gültigen Engine-Versionen zu erhalten.
- `--allow-major-version-upgrade`, um ein Upgrade der Hauptversion durchführen.
- `--no-apply-immediately`, um Änderungen im nächsten Wartungszeitraum anwenden. Verwenden Sie , um Änderungen sofort anzuwende `--apply-immediately`.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version new_version \
    --allow-major-version-upgrade \
    --no-apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine-version new_version ^
    --allow-major-version-upgrade ^
    --no-apply-immediately
```

RDS-API

Verwenden Sie die DBInstance Aktion Ändern, um die Engine-Version einer DB-Instance zu aktualisieren. Geben Sie die folgenden Parameter an:

- `DBInstanceIdentifier` – der Name der DB-Instance, z. B *`mydbinstance`*.
- `EngineVersion`: die Versionsnummer der Datenbank-Engine, auf die das Upgrade durchgeführt wird Informationen zu gültigen Engine-Versionen erhalten Sie mit dem Vorgang DBEngineVersionen beschreiben.
- `AllowMajorVersionUpgrade`, um festzulegen, ob ein Hauptversions-Upgrade zugelassen wird. Setzen Sie hierzu den Wert auf `true`.
- `ApplyImmediately`: Änderungen sofort oder während des nächsten Wartungszeitraums anwenden Legen Sie den Wert auf fest, um Änderungen sofort anzuwende `true`. Legen Sie den Wert auf fest, um Änderungen im nächsten Wartungszeitraum durchzuführe `false`.

Automatisches Upgraden der Engine-Unterversion

Durch automatische Upgrades kleinerer Versionen wird Ihre Datenbank regelmäßig auf die neuesten Versionen der Datenbank-Engine aktualisiert. Das Upgrade beinhaltet jedoch möglicherweise nicht immer die neueste Version der Datenbank-Engine. Wenn Sie Ihre Datenbanken zu bestimmten Zeiten auf bestimmten Versionen belassen müssen, empfehlen wir Ihnen, ein manuelles Upgrade auf die Datenbankversionen durchzuführen, die Sie gemäß Ihrem erforderlichen Zeitplan benötigen. Bei kritischen Sicherheitsproblemen oder wenn eine Version ihr end-of-support Datum erreicht, führt Amazon RDS Amazon möglicherweise ein Upgrade einer Nebenversion durch, auch wenn Sie die Option Automatisches Upgrade der Nebenversion nicht aktiviert haben. Weitere Informationen finden Sie in der Upgrade-Dokumentation für Ihre spezifische Datenbank-Engine.

- Automatische Upgrades für Nebenversionen RDS für Postgre SQL
- Automatische Upgrades für kleinere Versionen für RDS für My SQL

- Automatische Upgrades für Nebenversionen RDS für MariaDB
- Oracle-Unterversion-Upgrades
- Upgrades der Microsoft SQL Server-DB-Engine
- Versionen von Db2 auf Amazon RDS

Themen

- Funktionsweise von automatischen Nebenversion-Upgrades
- Einschalten von automatischen Nebenversions-Upgrades
- Ermitteln der Verfügbarkeit von Wartungs-Updates
- Suchen von Zielen für automatische Nebenversion-Upgrades

Funktionsweise von automatischen Nebenversion-Upgrades

Das Upgrade-Ziel ist die DB-Engine-Version, auf die Amazon RDS Ihre Datenbank aktualisiert. Eine kleinere Engine-Version wird als Upgrade-Ziel festgelegt, wenn die folgenden Bedingungen erfüllt sind:

- In der Datenbank wird eine Nebenversion der DB-Engine ausgeführt, die niedriger ist als die Zielversion der Nebenengine.

  Sie finden die aktuelle Engine-Version für Ihre DB-Instance, indem Sie auf der Datenbankdetailseite auf der Registerkarte Konfiguration nachsehen oder den CLI-Befehl ausführen`describe-db-instances`.
- Für die Datenbank ist das automatische Upgrade der Nebenversion aktiviert.

RDS plant, dass das Upgrade automatisch im Wartungsfenster ausgeführt wird. Während des Upgrades geht RDS wie folgt vor:

1. Führt eine Systemvorabprüfung durch, um sicherzustellen, dass die Datenbank fehlerfrei und bereit für ein Upgrade ist
2. Führt ein Upgrade der DB-Engine auf die Zielversion der Nebenengine durch
3. Führt Prüfungen nach dem Upgrade durch
4. Markiert das Datenbank-Upgrade als abgeschlossen

Automatische Upgrades führen zu Ausfallzeiten. Die Dauer der Ausfallzeit hängt von verschiedenen Faktoren ab, einschließlich dem Typ der DB-Engine und der Größe der Datenbank.

Einschalten von automatischen Nebenversions-Upgrades

Beim Ausführen der folgenden Aktionen können Sie entscheiden, ob automatische Unterversions-Upgrades für eine DB-Instance zugelassen werden:

- Erstellen einer DB-Instance
- Ändern einer DB-Instance
- Erstellen eines Lesereplikats
- Wiederherstellen einer DB-Instance aus einem Snapshot
- Wiederherstellen einer DB-Instance zu einem bestimmten Zeitpunkt
- Importieren einer DB-Instance aus Amazon S3 (für eine MySQL-Sicherung in Amazon S3)

Beim Ausführen dieser Aktionen können Sie entscheiden, ob automatische Unterversions-Upgrades für die DB-Instance aktiviert sind. Sie haben hierbei die folgenden Möglichkeiten:

- Aktivieren Sie über die Konsole die Option Auto minor version upgrade (Automatisches Unterversions-Upgrade).
- Stellen Sie mit dem AWS CLI die `--auto-minor-version-upgrade|--no-auto-minor-version-upgrade` Option ein.
- Legen Sie mit der RDS-API den Parameter `AutoMinorVersionUpgrade` fest.

Ermitteln der Verfügbarkeit von Wartungs-Updates

Um festzustellen, ob ein Wartungsupdate, z. B. ein DB-Engine-Versionsupgrade, für Ihren verfügbar ist, können Sie die Konsole oder die RDS-API verwenden. AWS CLI Es besteht auch die Möglichkeit, ein Upgrade für die DB-Engine-Version manuell durchzuführen und den Wartungszeitraum anzupassen. Weitere Informationen finden Sie unter Warten einer DB-Instance.

Suchen von Zielen für automatische Nebenversion-Upgrades

Sie können den folgenden AWS CLI Befehl verwenden, um die aktuelle Zielversion für das automatische kleinere Upgrade für eine bestimmte DB-Engine-Unterversion in einem bestimmten Bereich zu ermitteln AWS-Region. Die möglichen `--engine` Werte für diesen Befehl finden Sie in der Beschreibung des `Engine` Parameters in Create DBInstance.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine engine \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine engine ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Der folgende AWS CLI Befehl bestimmt beispielsweise das automatische kleinere Upgrade-Ziel für die MySQL-Nebenversion 8.0.11 in der AWS Region USA Ost (Ohio) (us-east-2).

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
```

```
--output table
```

Ihre Ausgabe sieht Folgendem ähnlich.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

In diesem Beispiel ist der `AutoUpgrade`-Wert `True` für MySQL-Version 8.0.23. Das automatische Nebenversions-Upgrade-Ziel ist daher die MySQL-Version 8.0.23, die in der Ausgabe hervorgehoben wird.

> **Important**
>
> Wenn Sie planen, eine DB-Instance von RDS für PostgreSQL in naher Zukunft auf einen DB-Cluster von Aurora PostgreSQL zu migrieren, empfehlen wir Ihnen dringend, Upgrades von automatischen Nebenversionen für die DB-Instance zu Beginn der Migrationsplanungsphase zu deaktivieren. Die Migration nach Aurora PostgreSQL kann sich verzögern, wenn die RDS for PostgreSQL-Version von Aurora PostgreSQL noch nicht unterstützt wird. Informationen zu Aurora PostgreSQL-Versionen finden Sie unter Engine-Versionen für Amazon Aurora-PostgreSQL.

Umbenennen einer DB-Instance

Sie können eine DB-Instance umbenennen, indem Sie die AWS Management Console, den AWS CLI `modify-db-instance` Befehl oder die Amazon `ModifyDBInstance` RDS-API-Aktion verwenden. Das Umbenennen einer DB-Instance kann weitgehende Auswirkungen haben. Im Folgenden finden Sie eine Liste von Überlegungen, bevor Sie eine DB-Instance umbenennen.

- Wenn Sie eine DB-Instance umbenennen, ändert sich der Endpunkt der DB-Instance, da die URL den der DB-Instance zugewiesenen Namen beinhaltet. Sie sollten den Datenverkehr immer von der alten URL auf die neue umleiten.
- Wenn Sie eine DB-Instance umbenennen, wird der alte DNS-Name der DB-Instance sofort gelöscht (obwohl er noch einige Minuten im Cache verbleiben könnte). Der neue DNS-Name für die umbenannte DB-Instance wird nach etwa 10 Minuten übernommen. Die umbenannte DB-Instance ist erst verfügbar, wenn der neue Name übernommen wurde.
- Sie können keinen bestehenden DB-Instance-Namen verwenden, wenn Sie eine Instance umbenennen.
- Alle Read Replicas, die der DB-Instance zugeordnet sind, bleiben auch nach der Umbenennung zugeordnet. Angenommen, Sie nutzen eine DB-Instance als Produktionsdatenbank und haben der Instance mehrere Read Replicas zugeordnet. Wenn Sie die DB-Instance umbenennen und diese anschließend in der Produktionsumgebung durch einen DB-Snapshot ersetzen, bleiben die Read Replicas der umbenannten DB-Instance weiterhin zugeordnet.
- Metriken und Ereignisse, die dem Namen der DB-Instance zugeordnet sind, bleiben aufrecht, wenn Sie einen DB-Instance-Namen erneut verwenden. Wenn Sie beispielsweise ein Lesereplikat hochstufen und es in den Namen der vorherigen primären DB-Instance umbenennen, werden die der primären DB-Instance zugeordneten Ereignisse und Metriken der umbenannten Instance zugeordnet.
- DB-Instance-Tags für die DB-Instance bleiben erhalten, ungeachtet einer Umbenennung.
- DB-Snapshots werden für eine umbenannte DB-Instance aufbewahrt.

> **Note**
>
> Eine DB-Instance ist eine isolierte Datenbankumgebung, die in der Cloud ausgeführt wird. Eine DB-Instance kann mehrere Datenbanken hosten, oder auch eine Oracle-Datenbank mit mehreren Schemas. Informationen zum Ändern eines Datenbanknamens finden Sie in der Dokumentation für Ihre DB-Engine.

Umbenennen einer bestehenden DB-Instance, um diese zu ersetzen

Die häufigsten Gründe für das Umbenennen einer DB-Instance sind, dass Sie eine Read Replica heraufstufen oder Daten aus einem DB-Snapshot oder einer point-in-time Wiederherstellung (PITR) wiederherstellen. Durch das Umbenennen der Datenbank können Sie die DB-Instance ersetzen, ohne Anwendungscode ändern zu müssen, der die DB-Instance referenziert. In solchen Fällen gehen Sie wie folgt vor:

1. Stoppen Sie den gesamten Datenverkehr zur primären DB-Instance. Dazu können Sie den an die Datenbanken auf der DB-Instance gerichteten Datenverkehr entweder umleiten oder mithilfe einer anderen Methode verhindern, dass auf die Datenbanken auf der DB-Instance zugegriffen wird.
2. Geben Sie der primären DB-Instance einen neuen Namen, der angibt, dass sie nicht mehr die primäre DB-Instance ist, wie später in diesem Thema beschrieben.
3. Erstellen Sie eine neue primäre DB-Instance, indem Sie eine Wiederherstellung aus einem DB-Snapshot durchführen oder ein Lesereplikat hochstufen. Geben Sie der neuen Instance dann den Namen der vorherigen primären DB-Instance.
4. Ordnen Sie der neuen primären DB-Instance alle Lesereplikate zu.

Wenn Sie die alte primäre DB-Instance löschen, sollten Sie auch die nicht länger benötigten DB-Snapshots der alten primären DB-Instance löschen.

Weitere Informationen zum Hochstufen eines Lesereplikats finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

> **Important**
>
> Die DB-Instance wird neu gestartet, wenn sie umbenannt wird. Bei RDS for SQL Server-Multi-AZ-Instances mit aktivierter AlwaysOn Spiegelungsoption ist ein Failover zu erwarten, wenn die Instanz nach dem Umbenennungsvorgang neu gestartet wird.

Konsole

So benennen Sie eine DB-Instance um

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie umbenennen möchten.
4. Wählen Sie Ändern aus.
5. Geben Sie unter Einstellungen einen neuen Namen für die DB-Instance-Kennung ein.
6. Klicken Sie auf Weiter.
7. Wählen Sie Apply immediately, um die Änderungen sofort anzuwenden. Die Auswahl dieser Option kann in einigen Fällen einen Ausfall verursachen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
8. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern.

   Klicken Sie anderenfalls auf Zurück, um Ihre Änderungen zu bearbeiten, oder klicken Sie auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Verwenden Sie den AWS CLI Befehl, um eine DB-Instance umzubenennen `modify-db-instance`. Geben Sie für den derzeitigen `--db-instance-identifier`-Wert und den `--new-db-instance-identifier`-Parameter den neuen Namen der DB-Instance an.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier DBInstanceIdentifier \
    --new-db-instance-identifier NewDBInstanceIdentifier
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier DBInstanceIdentifier ^
    --new-db-instance-identifier NewDBInstanceIdentifier
```

RDS-API

Rufen Sie die Amazon-RDS-API-Funktion `ModifyDBInstance` mit den folgenden Parametern auf, um eine DB-Instance umzubenennen:

- `DBInstanceIdentifier` – bestehender Name für die Instance
- `NewDBInstanceIdentifier` – neuer Name für die Instance

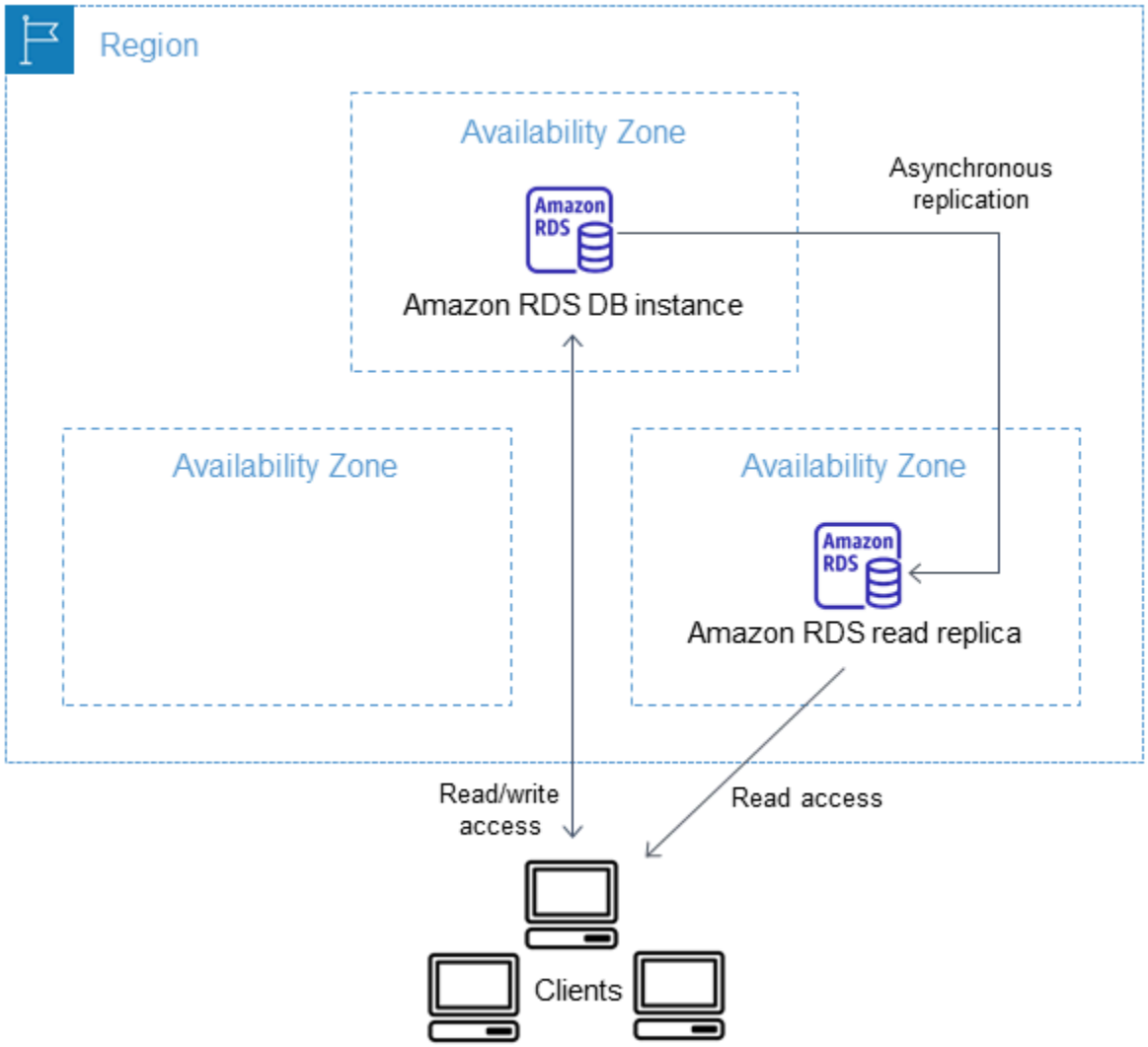
Arbeiten mit DB-Instance-Lesereplikaten

Ein Lesereplikat ist eine schreibgeschützte Kopie einer DB-Instance. Sie können die Arbeitslast Ihrer primären DB-Instance reduzieren, indem Sie Abfragen aus Ihren Anwendungen an das Lesereplikat weiterleiten. Dies ermöglicht eine elastische Aufskalierung über die Kapazitätseinschränkungen einer einzelnen DB-Instance für leseintensive Datenbank-Workloads hinaus.

Amazon RDS nutzt die integrierten Replikationsfunktionen der DB-Engine, um ein Lesereplikat aus einer Quell-DB-Instance zu erstellen. Weitere Informationen zur Verwendung dieser Lesereplikate mit einer bestimmten Engine finden Sie in den folgenden Abschnitten:

- Arbeiten mit Replikaten für Amazon RDS for Db2
- Arbeiten mit MariaDB Read Replicas
- Arbeiten mit Read Replicas für Microsoft SQL Server in Amazon RDS
- Mit My SQL Read Replicas arbeiten
- Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle
- Arbeiten mit Read Replicas in Amazon RDS for PostgreSQL

Nachdem Sie ein Lesereplikat aus einer Quell-DB-Instance erstellt haben, wird die Quelle zur primären DB-Instance. Wenn Sie Updates an der primären DB-Instance vornehmen, kopiert Amazon RDS sie asynchron in das Lesereplikat. Das folgende Diagramm zeigt eine Quell-DB-Instance, die auf ein Lesereplikat in einer anderen Availability Zone (AZ) repliziert. Clients haben read/write Zugriff auf die primäre DB-Instance und schreibgeschützten Zugriff auf das Replikat.



Read Replicas werden als Standard-DB-Instances zu den gleichen Tarifen abgerechnet wie die DB-Instance-Klasse, die für das Replikat verwendet wird. Die Datenübertragung, die bei der Replikation von Daten zwischen der Quell-DB-Instance und einer Read Replica innerhalb derselben entsteht, wird Ihnen nicht in Rechnung gestellt. AWS-Region Weitere Informationen erhalten Sie unter Kosten für regionsübergreifende Replikationen und DB-Instance-Abrechnung für Amazon RDS.

Themen

- Übersicht über Amazon RDS-Lesereplikate
- Unterschiede zwischen Read Replicas für DB-Engines
- Erstellen eines Lesereplikats
- Hochstufen eines Lesereplikats zur eigenständigen DB-Instance
- Überwachen der Lesereplikation

- Erstellen eines Lesereplikats in einer anderen AWS-Region

Übersicht über Amazon RDS-Lesereplikate

In den folgenden Abschnitten werden DB-Instance-Lesereplikate behandelt. Informationen zu Lesereplikaten von Multi-AZ-DB-Clustern finden Sie unter the section called “Arbeiten mit Multi-AZ-DB-Cluster-Lesereplikaten”.

Themen

- Anwendungsfälle für Lesereplikate
- Funktionsweise von Lesereplikaten
- Lesereplikate in einer Multi-AZ-Bereitstellung
- Regionsübergreifende Lesereplikate
- Lesereplikat-Speichertypen
- Einschränkungen beim Erstellen eines Replikats aus einem Replikat
- Überlegungen zum Löschen von Replikaten

Anwendungsfälle für Lesereplikate

Die Bereitstellung eines oder mehrerer Lesereplikate für eine bestehende Quell-DB-Instance kann in einer Vielfalt von Szenarien sinnvoll sein, einschließlich der Folgenden:

- Skalierung über die Rechenleistung oder I/O Kapazität einer einzelnen DB-Instance hinaus für leseintensive Datenbank-Workloads. Sie können diesen übermäßigen Datenverkehr an Lesevorgängen einem oder mehreren Lesereplikaten zuweisen.
- Unterstützung des Lesedatenverkehrs bei einer nicht verfügbaren Quell-DB-Instance. In einigen Fällen ist Ihre Quell-DB-Instance möglicherweise nicht in der Lage, I/O Anfragen anzunehmen, z. B. aufgrund einer I/O-Unterbrechung für Backups oder aufgrund von geplanten Wartungsarbeiten. In diesen Fällen können Sie den Lesedatenverkehr an Ihre Lesereplikate leiten. Denken Sie bei diesem Anwendungsfall daran, dass die Daten im Lesereplikat "veraltet" sein können, da die Quell-DB-Instance nicht verfügbar ist.
- Szenarien mit Geschäftsberichten oder Data-Warehousing, bei denen es sich eventuell empfiehlt, Abfragen zu Geschäftsberichten über ein Lesereplikat und nicht über Ihre Produktions-DB-Instance laufen zu lassen.

- Implementieren der Notfallwiederherstellung Sie können ein Lesereplikat als Lösung zur Notfallwiederherstellung auf eine eigenständige Instance hochstufen, wenn die primäre DB-Instance ausfällt.

Funktionsweise von Lesereplikaten

Wenn Sie eine Read Replica erstellen, geben Sie eine vorhandene DB-Instance als Quelle an. Dann erstellt Amazon RDS einen Snapshot von der Quell-Instance und erstellt eine schreibgeschützte Instance aus diesem Snapshot. Dann verwendet Amazon RDS die asynchrone Replikationsmethode für die DB-Engine, um das Lesereplikat bei jeder Änderung an der primären DB-Instance zu aktualisieren.

Die Read Replica arbeitet als DB-Instance, die nur schreibgeschützte Verbindungen zulässt. Ausnahmen sind die DB-Engines RDS für Db2 und RDS für Oracle, die Replikatdatenbanken im Standby-Modus bzw. im Mount-Modus unterstützen. Ein Standby-Replikat und ein bereitgestelltes Replikat akzeptieren keine Benutzerverbindungen und können daher keine schreibgeschützten Workloads verarbeiten. Die primäre Verwendung von Standby-Replikaten und bereitgestellten Replikaten ist die regionsübergreifende Notfallwiederherstellung. Weitere Informationen erhalten Sie unter Arbeiten mit Replikaten für Amazon RDS for Db2 und Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle.

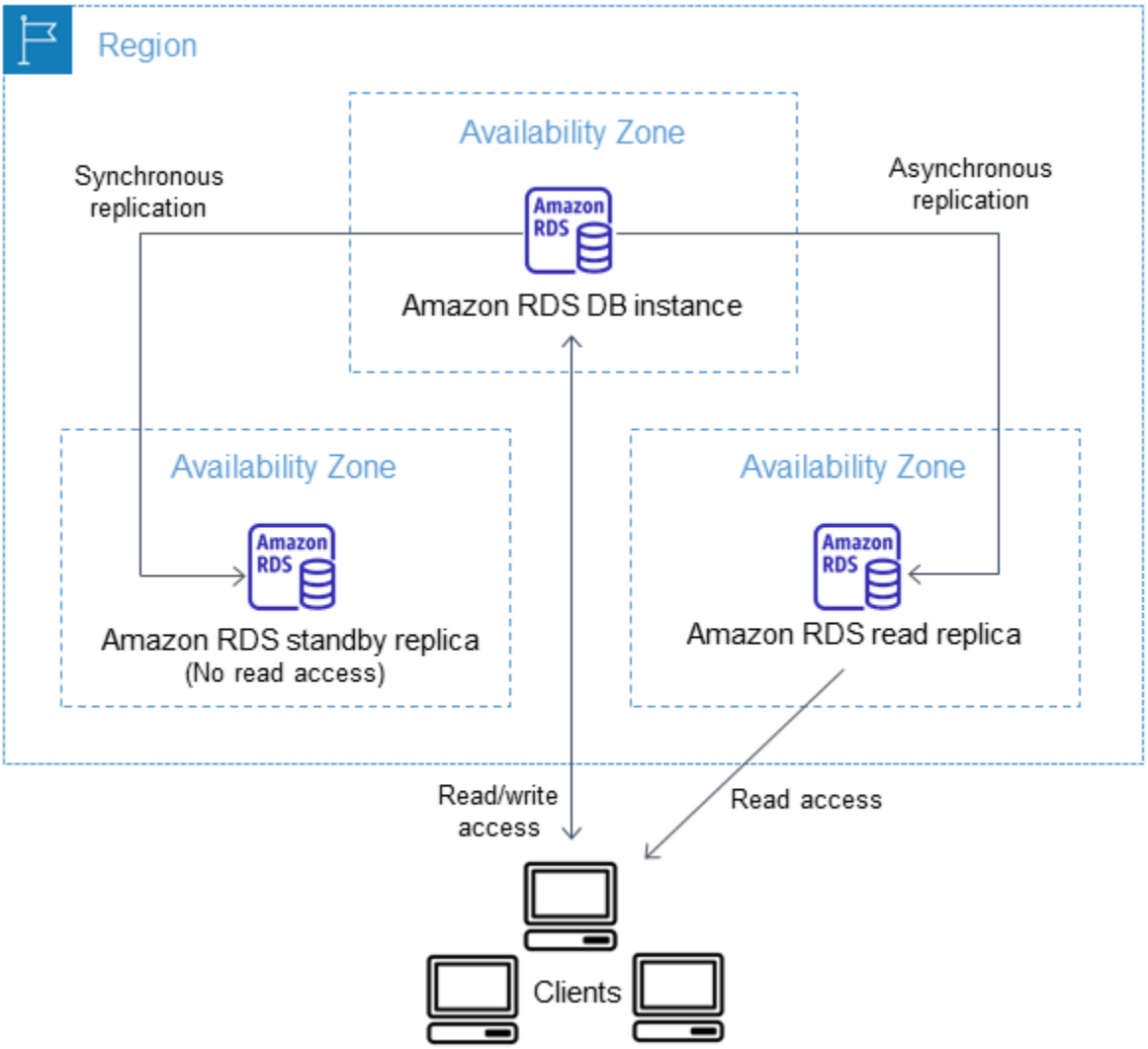
Anwendungen verbinden sich mit dem Lesereplikat so wie mit einer DB-Instance. Amazon RDS repliziert alle Datenbanken in der Quell-DB-Instance.

Sie müssen Read Replicas manuell erstellen. RDS unterstützt keine automatische Skalierung von Read Replicas, d. h. das automatische Hinzufügen oder Entfernen von Read Replicas, wenn sich der Lesebedarf ändert.

Lesereplikate in einer Multi-AZ-Bereitstellung

Sie können ein Lesereplikat für eine DB-Instance konfigurieren, für die auch ein Standby-Replikat für hohe Verfügbarkeit in einer Multi-AZ-Bereitstellung konfiguriert ist. Die Replikation mit dem Standby-Replikat erfolgt synchron. Im Gegensatz zu Lesereplikaten kann ein Standby-Replikat keinen Lesedatenverkehr bereitstellen.

Im folgenden Szenario haben Clients read/write Zugriff auf eine primäre DB-Instance in einer AZ. Die primäre Instance kopiert Updates asynchron auf ein Lesereplikat in einer zweiten AZ und kopiert sie auch synchron auf ein Standby-Replikat in einer dritten AZ. Clients haben nur Lesezugriff auf das Lesereplikat.



Weitere Informationen zu Hochverfügbarkeits- und Standby-Replikaten finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Regionsübergreifende Lesereplikate

In einigen Fällen befindet sich eine Read Replica in einer anderen AWS-Region als ihrer primären DB-Instance. In diesen Fällen richtet Amazon RDS einen sicheren Kommunikationskanal zwischen der primären DB-Instance und dem Lesereplikat ein. Amazon RDS richtet alle AWS Sicherheitskonfigurationen ein, die zur Aktivierung des sicheren Kanals erforderlich sind, z. B. das Hinzufügen von Sicherheitsgruppeneinträgen. Weitere Informationen zu regionsübergreifenden Read Replicas finden Sie unter Erstellen eines Lesereplikats in einer anderen AWS-Region.

Die Informationen in diesem Kapitel beziehen sich auf die Erstellung von Amazon RDS-Read Replicas entweder in derselben AWS-Region wie die Quell-DB-Instance oder in einer separaten

AWS-Region. Die folgenden Informationen gelten nicht für die Einrichtung der Replikation mit einer Instance, die auf einer EC2 Amazon-Instance läuft oder lokal ist.

Lesereplikat-Speichertypen

Ein Lesereplikat wird standardmäßig mit dem selben Speichertyp erstellt wie die Quell-DB-Instance. Jedoch können Sie ein Lesereplikat erstellen, das einen anderen Speicherchip aufweist als die Quell-DB-Instance, basierend auf den in der folgenden Tabelle gelisteten Optionen.

Quell-DB-Instance-Speichertyp	Quell-DB-Instance-Speicherzuteilung	Optionen für den Lesereplikat-Speichertyp
Bereitgestellte IOPS	100 GiB–64 TiB	Bereitgestellte IOPS, Allgemeine Zwecke, magnetisch
Allgemeine Zwecke	100 GiB–64 TiB	Bereitgestellte IOPS, Allgemeine Zwecke, magnetisch
Allgemeine Zwecke	<100 GiB	Allgemeine Zwecke, magnetisch
Magnetic	100 GiB–6 TiB	Bereitgestellte IOPS, Allgemeine Zwecke, magnetisch
Magnetic	<100 GiB	Allgemeine Zwecke, magnetisch

> **Note**
>
> Wenn Sie den zugewiesenen Speicher eines Lesereplikats erhöhen, muss er um mindestens 10 Prozent erhöht werden. Wenn Sie versuchen, den Wert um weniger als 10 Prozent zu erhöhen, erhalten Sie einen Fehler.

Einschränkungen beim Erstellen eines Replikats aus einem Replikat

Amazon RDS unterstützt keine zirkulierende Replikation. Sie können eine DB-Instance nicht so konfigurieren, dass Sie als Replikationsquelle für eine vorhandene DB-Instance dient. Sie können ein neues Lesereplikat nur aus einer vorhandenen DB-Instance erstellen. Wenn beispielsweise **`MySourceDBInstance`** in **`ReadReplica1`** repliziert wird, können Sie **`ReadReplica1`** nicht für eine Rückreplikation auf **`MySourceDBInstance`** konfigurieren.

Bei RDS für MariaDB und RDS für MySQL und bei bestimmten Versionen von RDS für PostgreSQL ist es möglich, ein Lesereplikat aus einem vorhandenen Lesereplikat zu erstellen. Sie können beispielsweise ein neues Lesereplikat **`ReadReplica2`** aus einem vorhandenen Replikat **`ReadReplica1`** erstellen. Für RDS for Db2, RDS for Oracle und RDS for SQL Server können Sie keine Read Replica aus einer vorhandenen Read Replica erstellen.

Überlegungen zum Löschen von Replikaten

RDS unterstützt die automatische Skalierung von Read Replicas nicht. Daher erhöht RDS nicht die Anzahl der Replikate, wenn die Nachfrage steigt, und verringert die Anzahl der Replikate nicht, wenn die Nachfrage sinkt. Wenn Sie Read Replicas nicht mehr benötigen, löschen Sie sie manuell. Verwenden Sie dabei dieselben Mechanismen wie beim Löschen einer DB-Instance. Wenn Sie eine Quell-DB-Instance löschen, ohne ihre Read Replicas in derselben zu löschen AWS-Region, wird jedes Replikat zu einer eigenständigen DB-Instance heraufgestuft.

Weitere Informationen zum Löschen einer DB-Instance finden Sie unter Löschen einer DB-Instance. Weitere Informationen zum Hochstufen von Read Replicas finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance. Informationen zum Löschen der Quell-DB-Instance für eine regionsübergreifende Read Replica finden Sie unter. Überlegungen zur regionsübergreifenden Replikation

Unterschiede zwischen Read Replicas für DB-Engines

Da Amazon RDS-DB-Engines die Replikation unterschiedlich implementieren, gibt es mehrere signifikante Unterschiede, die Sie kennen sollten.

Db2

Replicas for RDS for Db2 haben die folgenden Funktionen und Verhaltensweisen:

- Replikationsmethode — Physische Replikation.

- Löschen von Transaktionsprotokollen — RDS for Db2 löscht die Protokolle aus der primären DB-Instance, wenn die folgenden Bedingungen erfüllt sind:
  - Die Protokolle sind mindestens zwei Stunden alt.
  - Die Einstellung für die Aufbewahrungszeit von Archivprotokollen ist abgelaufen.
  - RDS for Db2 hat die Protokolle erfolgreich auf alle Replikat-DB-Instances repliziert.

  Dies gilt sowohl für dieselben AWS-Region DB-Instances als auch für regionsübergreifende DB-Instances. Weitere Informationen zum Festlegen von Aufbewahrungszeiten für Archivprotokolle finden Sie unter rdsadmin.set_archive_log_retention.
- Schreibbare Replikate — Ein Db2-Replikat ist eine physische Kopie, und Db2 erlaubt keine Schreibvorgänge in einem Replikat. Sie können das Replikat heraufstufen, um es schreibbar zu machen. Das heraufgestufte Replikat verfügt über die replizierten Daten bis zu dem Zeitpunkt, an dem die Anfrage zur Heraufstufung gestellt wurde.
- Backups — Automatische Backups und manuelle Snapshots werden auf RDS für Db2-Replikate unterstützt.
- Parallele Replikation — Archivprotokolldaten werden immer parallel von der Primärdatenbank an alle ihre Replikate übertragen.
- Standby-Status — Die primäre Verwendung von Standby-Replikaten ist die regionsübergreifende Notfallwiederherstellung. Weitere Informationen finden Sie unter Arbeiten mit Replikaten für Amazon RDS for Db2.

MariaDB und MySQL

Read Replicas für RDS für MariaDB und RDS für MySQL haben die folgenden Funktionen und Verhaltensweisen:

- Replikationsmethode — Logische Replikation.
- Löschen von Transaktionsprotokollen — RDS für MariaDB und RDS für MySQL speichern alle Binärprotokolle, die nicht angewendet wurden.
- Schreibbare Replikate — Sie können die MariaDB- oder MySQL-Lesereplik so einrichten, dass sie beschreibbar ist.
- Backups — Automatische Backups und manuelle Snapshots werden auf RDS for MariaDB oder RDS for MySQL Read Replicas unterstützt.
- Parallele Replikation — Alle unterstützten MariaDB- und MySQL-Versionen ermöglichen parallel Replikations-Threads.

- Bereitgestellter Status — Nicht unterstützt.

Oracle

Read Replicas für RDS for Oracle verfügen über die folgenden Funktionen und Verhaltensweisen:

- Replikationsmethode — Physische Replikation.
- Löschen von Transaktionsprotokollen — Wenn eine primäre DB-Instance keine regionsübergreifenden Read Replicas hat, speichert Amazon RDS for Oracle mindestens zwei Stunden an Transaktionsprotokollen auf der Quell-DB-Instance. Protokolle werden nach zwei Stunden oder nach Ablauf der Einstellung für die Aufbewahrungszeit der Archivprotokolle, je nachdem, welcher Zeitraum länger dauert, aus der Quell-DB-Instance bereinigt. Protokolle werden nach Ablauf der Einstellung für die Aufbewahrungszeit der Archivprotokolle nur dann aus dem Lesereplikat bereinigt, wenn sie erfolgreich auf die Datenbank angewendet wurden.

  In manchen Fällen kann eine primäre DB-Instance ein oder mehrere regionsübergreifende Lesereplikate haben. Wenn dies der Fall ist, behält Amazon RDS for Oracle die Transaktionsprotokolle auf der Quell-DB-Instance bei, bis diese übertragen und auf alle regionsübergreifenden Lesereplikate angewendet wurden.

  Weitere Informationen zum Festlegen von Aufbewahrungszeiten für Archivprotokolle finden Sie unter Beibehaltung von archivierten Redo-Log-Dateien.
- Schreibbare Replikate — Eine Oracle-Read Replica ist eine physische Kopie, und Oracle erlaubt keine Schreibvorgänge in einer Read Replica. Sie können das Lesereplikat hochstufen, um es beschreibbar zu machen. Das hochgestufte Lesereplikat weist die replizierten Daten bis zu dem Punkt auf, an dem die Anforderung zum Hochstufen ausgegeben wurde.
- Backups — Automatische Backups und manuelle Snapshots werden auf RDS für Oracle-Read Replicas unterstützt.
- Parallele Replikation — Redo-Log-Daten werden immer parallel von der Primärdatenbank an alle ihre Read Replicas übertragen.
- Bereitgestellter Zustand — Bereitgestellte Replikate werden hauptsächlich für die regionsübergreifende Notfallwiederherstellung verwendet. Für aufgespielte Replikate ist keine Active Data Guard-Lizenz erforderlich. Weitere Informationen finden Sie unter Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle.

PostgreSQL

Read Replicas für RDS for PostgreSQL haben die folgenden Funktionen und Verhaltensweisen:

- Replikationsmethode — Physische Replikation.
- Löschen von Transaktionsprotokollen — PostgreSQL hat den Parameter`wal_keep_segments`, der festlegt, wie viele Write-Ahead-Log-Dateien (WAL) aufbewahrt werden, um Daten für die Read Replicas bereitzustellen. Der Parameterwert gibt die Anzahl an beizubehaltenden Protokollen an.
- Schreibbare Replikate — Eine PostgreSQL-Read Replica ist eine physische Kopie, und PostgreSQL erlaubt es nicht, eine Read Replica schreibbar zu machen.
- Backups — Manuelle Snapshots werden für RDS for PostgreSQL Read Replicas unterstützt. Automatisierte Backups für Lesereplikate werden nur für RDS for PostgreSQL 14.1 und höhere Versionen unterstützt. Sie können keine automatisierten Backups für PostgreSQL-Lesereplikate für RDS for PostgreSQL Versionen vor 14.1 aktivieren. Erstellen Sie für RDS for PostgreSQL 13 und frühere Versionen einen Snapshot aus einem Lesereplikat, wenn Sie eine Sicherungskopie davon wünschen.
- Parallele Replikation — PostgreSQL hat einen einzigen Prozess, der die Replikation abwickelt.
- Bereitgestellter Status — Nicht unterstützt.

SQL Server

Read Replicas für RDS for SQL Server verfügen über die folgenden Funktionen und Verhaltensweisen:

- Replikationsmethode — Physische Replikation.
- Löschen von Transaktionsprotokollen — Die virtuelle Protokolldatei (VLF) der Transaktionsprotokolldatei auf dem primären Replikat kann gekürzt werden, wenn sie für die sekundären Replikate nicht mehr benötigt wird.

  Die VLF kann nur dann als inaktiv markiert werden, wenn die Protokolldatensätze in den Replikaten gehärtet wurden. Unabhängig davon, wie schnell die Festplattensubsysteme im primären Replikat sind, wird das Transaktionsprotokoll so lange gespeichert, VLFs bis das langsamste Replikat es gehärtet hat.
- Schreibbare Replikate — Eine SQL Server-Lesereplik ist eine physische Kopie und erlaubt auch keine Schreibvorgänge. Sie können das Lesereplikat hochstufen, um es beschreibbar zu machen.

Das hochgestufte Lesereplikat weist die replizierten Daten bis zu dem Punkt auf, an dem die Anforderung zum Hochstufen ausgegeben wurde.

- Backups — Automatische Backups und manuelle Snapshots werden auf RDS for SQL Server Read Replicas nicht unterstützt.
- Parallele Replikation — Redo-Log-Daten werden immer parallel von der Primärdatenbank an alle ihre Read Replicas übertragen.
- Bereitgestellter Status — Nicht unterstützt.

Erstellen eines Lesereplikats

Sie können mit der, oder RDS-API eine Read Replica aus einer vorhandenen DB-Instance erstellen. AWS Management Console AWS CLI Sie erstellen ein Lesereplikat, indem Sie `SourceDBInstanceIdentifier` angeben, die eine DB-Instance-Kennung der Quell-DB-Instance ist, aus der Sie replizieren möchten.

Wenn Sie ein Lesereplikat erstellen, fertigt Amazon RDS einen DB-Snapshot Ihrer Quell-DB-Instance an und beginnt mit der Replikation. Die Quell-DB-Instance wird sehr kurzzeitig unterbrochen, I/O wenn der DB-Snapshot-Vorgang beginnt. Die I/O Unterbrechung dauert in der Regel etwa eine Sekunde. Sie können die I/O Aussetzung vermeiden, wenn es sich bei der Quell-DB-Instance um eine Multi-AZ-Bereitstellung handelt, da in diesem Fall der Snapshot von der sekundären DB-Instance stammt.

Eine aktive, langlaufende Transaktion kann den Prozess der Erstellung des Lesereplikats verlangsamen. Wir empfehlen Ihnen zu warten, bis langlaufende Transaktionen abgeschlossen sind, bevor ein Lesereplikat erstellt wird. Wenn Sie mehrere Lesereplikate parallel aus derselben Quell-DB-Instance erstellen, macht Amazon RDS nur einen Snapshot zu Beginn der ersten Erstellungsaktion.

Beim Erstellen eines Lesereplikats müssen einige Dinge beachtet werden. Als Erstes müssen Sie automatische Backups für die DB-Instance aktivieren, indem sie den Aufbewahrungszeitraum für Backups auf einen anderen Wert als 0 festlegen. Diese Erfordernis gilt auch für ein Lesereplikat, das die Quell-DB-Instance für ein anderes Lesereplikat ist. Wenn Sie automatische Backups auf einem Lesereplikat für RDS for MySQL aktivieren möchten, erstellen Sie zuerst das Lesereplikat und ändern Sie es dann, um automatische Backups zu aktivieren.

> **Note**
>
> Innerhalb eines empfehlen wir dringend AWS-Region, alle Read Replicas in derselben Virtual Private Cloud (VPC) auf Basis von Amazon VPC als Quell-DB-Instance zu erstellen.

Wenn Sie ein Lesereplikat in einer anderen VPC als der Quell-DB-Instance erstellen, können Classless Inter-Domain Routing (CIDR)-Bereiche zwischen dem Replikat und dem RDS-System einander überlappen. Die CIDR-Überlappung macht das Replikat instabil, was sich negativ auf Anwendungen auswirken kann, die eine Verbindung herstellen. Wenn beim Erstellen des Lesereplikats eine Fehlermeldung angezeigt wird, wählen Sie eine andere Ziel-DB-Subnetzgruppe aus. Weitere Informationen finden Sie unter Arbeiten mit einer DB-Instance in einer VPC.
Es gibt keine direkte Möglichkeit, AWS-Konto mithilfe der Konsole oder eine Read Replica in einer anderen zu erstellen. AWS CLI

Konsole

Zum Erstellen einer Read Replica aus einer Quell-DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie als Quelle für eine Read Replica verwenden möchten.
4. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.
5. Geben Sie unter DB instance identifier (DB-Instance-Kennung) einen Namen für das Lesereplikat ein.
6. Wählen Sie Ihre Instance-Konfiguration. Wir empfehlen Ihnen, dieselbe oder eine größere DB-Instance-Klasse und denselben Speichertyp wie bei der Quell-DB-Instance für das Lesereplikat zu verwenden.
7. Legen Sie unter AWS-Region die Zielregion für das Lesereplikat fest.
8. Legen Sie unter Speicher die Größe des zugewiesenen Speichers fest und geben Sie an, ob Sie die automatische Speicherskalierung verwenden möchten.

   Wenn Ihre Quell-DB-Instance nicht die neueste Speicherkonfiguration aufweist, ist die Option Speicherdateisystemkonfiguration aktualisieren verfügbar. Sie können diese Einstellung aktivieren, um das Speicherdateisystem des Lesereplikats auf die bevorzugte Konfiguration zu aktualisieren. Weitere Informationen finden Sie unter the section called “Upgrade des Speicherdateisystems”.
9. Wählen Sie unter Verfügbarkeit aus, ob eine Standby-Version des Replikats in einer anderen Availability Zone als Failover-Support für das Replikat erstellt werden soll.

> **Note**
>
> Das Erstellen Ihres Lesereplikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist.

10. Legen Sie weitere DB-Instance-Einstellungen fest. Weitere Informationen zu den verfügbaren Einstellungen finden Sie unter Einstellungen für DB-Instances.
11. Um ein verschlüsseltes Lesereplikat zu erstellen, erweitern Sie Zusätzliche Konfiguration und geben Sie die folgenden Einstellungen an:

    a. Wählen Sie Enable encryption (Verschlüsselung aktivieren) aus.

    b. FürAWS KMS key, wählen Sie die AWS KMS key -Bezeichner des KMS-Schlüssels.

> **Note**
>
> Die DB-Quell-Instance muss verschlüsselt werden. Weitere Informationen über das Verschlüsseln der Quell-DB-Instance finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.

12. Wählen Sie Read Replica erstellen aus.

Nachdem die Read Replica erstellt wurde, können Sie sie auf der Seite „ Datenbanken “ in der RDS-Konsole sehen. Es zeigt Replica in der Spalte Rolle.

AWS CLI

Um eine Read Replica aus einer Quell-DB-Instance zu erstellen, verwenden Sie den AWS CLI Befehl create-db-instance-read-replica. In diesem Beispiel wird außerdem die Größe des zugewiesenen Speichers festgelegt, die automatische Speicherskalierung aktiviert und das Dateisystem auf die bevorzugte Konfiguration aktualisiert.

Sie können andere Einstellungen festlegen. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

> **Note**
>
> Um ein RDS für Db2-Standby-Replikat zu erstellen, fügen Sie im folgenden Befehl Folgendes ein. AWS CLI `--replica-mode mounted`

Example

FürLinux, odermacOS: Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --upgrade-storage-config
```

Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --upgrade-storage-config
```

RDS-API

Um eine Read Replica aus einer Quell-MySQL-, MariaDB-, Oracle-, PostgreSQL- oder SQL Server-DB-Instance zu erstellen, rufen Sie den Amazon DBInstance ReadReplica RDS-API-Vorgang Create mit den folgenden erforderlichen Parametern auf:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

> **Note**
>
> Um ein Standby-Replikat von RDS for Db2 zu erstellen, setzen Sie den optionalen Vorgang auf. `ReplicaMode` `mounted`

Hochstufen eines Lesereplikats zur eigenständigen DB-Instance

Sie können eine Read Replica in eine eigenständige DB-Instance hochstufen. Wenn eine Quell-DB-Instance über mehrere Lesereplikate verfügt, hat das Hochstufen eines der Lesereplikate zu einer DB-Instance keine Auswirkung auf die anderen Replikate.

Wenn Sie eine Read Replica hochstufen, startet RDS die DB-Instance neu, bevor sie verfügbar gemacht wird. Das Hochstufen kann einige Minuten oder mehr in Anspruch nehmen, abhängig von der Größe des Lesereplikats.



Anwendungsfälle für das Heraufstufen einer Read Replica

Möglicherweise möchten Sie eine Read Replica aus einem der folgenden Gründe zu einer eigenständigen DB-Instance heraufstufen:

- Implementierung einer Failover-Wiederherstellung – Sie können die Hochstufung des Lesereplikats als Datenwiederherstellungsschema verwenden, wenn die primäre DB-Instance fehlschlägt. Dieser Ansatz ergänzt die synchrone Replikation, die automatische Fehlererkennung und das Failover.

Wenn Sie sich der Auswirkungen und Beschränkungen von asynchroner Replikation bewusst sind und Sie dennoch Lesereplikathochstufung für die Datenwiederherstellung verwenden möchten, können Sie das tun. Erstellen Sie dazu zuerst ein Lesereplikat und überwachen Sie anschließend die primäre DB-Instance auf Fehler. Im Fall eines Ausfalls machen Sie Folgendes:

1. Stufen Sie das Lesereplikat hoch.
2. Leiten Sie den Datenverkehr der Datenbank an die hochgestufte DB-Instance weiter.
3. Erstellen Sie ein Ersatzlesereplikat mit der hochgestuften DB-Instance als Quelle.

- Aktualisieren der Speicherkonfiguration – Wenn Ihre Quell-DB-Instance nicht die bevorzugte Speicherkonfiguration aufweist, können Sie ein Lesereplikat der Instance erstellen und die Konfiguration des Speicherdateisystems aktualisieren. Mit dieser Option wird das Dateisystem des Lesereplikats auf die bevorzugte Konfiguration migriert. Sie können dann das Lesereplikat zu einer eigenständigen Instance hochstufen.

  Sie können diese Option verwenden, um die Skalierungsbeschränkungen in Bezug auf Speicher und Dateigröße für ältere 32-Bit-Dateisysteme zu überwinden. Weitere Informationen finden Sie unter the section called “Upgrade des Speicherdateisystems”.

  Diese Option ist nur verfügbar, wenn Ihre Quell-DB-Instance nicht die neueste Speicherkonfiguration aufweist oder wenn Sie die DB-Instance-Klasse in derselben Anfrage ändern.

- Sharding: Sharding verkörpert die Shared-Nothing-Architektur. Ihr wesentliches Prinzip besteht darin, eine große Datenbank in mehrere kleinere Datenbanken aufzuteilen. Normalerweise erfolgt die Aufteilung einer Datenbank, indem die Tabellen, die nicht in derselben Abfrage enthalten sind, auf verschiedene Hosts aufgeteilt werden. Ein weitere Methode besteht darin, die Tabelle auf mehrere Hosts zu duplizieren und dann einen Hash-Algorithmus zu verwenden, um festzustellen, welcher Host eine bestimmte Aktualisierung erhalten hat. Sie können Lesereplikate erstellen, die Ihren einzelnen Shards (kleineren Datenbanken) entsprechen, und sie hochstufen, wenn Sie beschließen, sie zu unabhängigen Shards zu konvertieren. Sie können anschließend je nach Ihren Erfordernissen den Schlüsselraum (wenn Sie Zeilen aufteilen) oder die Verteilung von Tabellen für jedes der Shards aufteilen.

- Ausführen von DDL-Operations (ausschließlich MySQL und MariaDB): DDL-Operationen wie das Erstellen oder Wiederaufbauen von Indizes sind zeitaufwendig und sorgen für eine beträchtliche Leistungsminderung der DB-Instance. Sie können diese Operationen in einem MySQL- oder MariaDB-Lesereplikat ausführen, sobald das Lesereplikat mit seiner primären DB-Instance

synchronisiert ist. Anschließend können Sie das Lesereplikat hochstufen und Ihre Anwendungen für die Nutzung der hochgestuften Instance weiterleiten.

> **Note**
>
> Wenn es sich bei Ihrer Read Replica um eine RDS für Oracle-DB-Instance handelt, können Sie statt einer Heraufstufung einen Switchover durchführen. Bei einem Switchover wird die Quell-DB-Instance zum neuen Replikat und das Replikat zur neuen Quell-DB-Instance. Weitere Informationen finden Sie unter So führen Sie eine Oracle Data Guard-Umschaltung aus.

Eigenschaften einer hochgestuften Read Replica

Nachdem Sie die Read Replica heraufgestuft haben, funktioniert sie nicht mehr als Read Replica und wird zu einer eigenständigen DB-Instance. Die neue eigenständige DB-Instance weist die folgenden Eigenschaften auf:

- Die eigenständige DB-Instance behält die Optionsgruppe und die Parametergruppe der Read Replica vor der Promotion bei.
- Sie können Read Replicas von der eigenständigen DB-Instance aus erstellen und Wiederherstellungsvorgänge durchführen point-in-time.
- Sie können die DB-Instance nicht als Replikationsziel verwenden, da es sich nicht mehr um eine Read Replica handelt.

Voraussetzungen für das Heraufstufen einer Read Replica

Gehen Sie wie folgt vor, bevor Sie ein Read Replica heraufstufen:

- Überprüfen Sie Ihre Backup-Strategie:
  - Wir empfehlen, dass Sie Backups aktivieren und mindestens ein Backup durchführen. Die Sicherungsdauer ist eine Funktion der Anzahl von Änderungen an der Datenbank seit der letzten Backup.
  - Wenn Sie Backups für Ihre Lesereplikate aktiviert haben, konfigurieren Sie das automatische Sicherungsfenster so, dass die täglichen Backups das Hochstufen des Lesereplikats nicht beeinträchtigen.

  - Stellen Sie sicher, dass Ihre Read Replica nicht den `backing-up` Status hat. Sie können eine Read Replica nicht heraufstufen, wenn sie sich in diesem Status befindet.
- Beenden Sie das Schreiben aller Transaktionen in die primäre DB-Instance und warten Sie dann, bis RDS alle Aktualisierungen auf die Read Replica angewendet hat.

  Datenbank-Updates werden im Lesereplikat durchgeführt, nachdem sie in der primären DB-Instance vorgenommen wurden. Die Verzögerung bei der Replikation kann erheblich variieren. Verwenden Sie die Metrik `Replica Lag`, um zu bestimmen, wann alle Aktualisierungen am Lesereplikat vorgenommen wurden.
- (Nur MySQL und MariaDB) Um Änderungen an einer MySQL- oder MariaDB-Read Replica vorzunehmen, bevor Sie sie heraufstufen, setzen Sie den Parameter `0` in der `read_only` DB-Parametergruppe für die Read Replica auf. Sie können anschließend alle nötigen DDL-Operationen, wie das Erstellen von Indizes, im Lesereplikat ausführen. Im Lesereplikat vorgenommene Aktionen haben keine Auswirkung auf die Leistungen der primären DB-Instance.

Eine Read Replica heraufstufen: grundlegende Schritte

Die folgenden Schritte zeigen den allgemeinen Vorgang für das Hochstufen eines Lesereplikats zu einer DB-Instance:

1. Bewerben Sie die Read Replica, indem Sie die Option Promote auf der Amazon RDS-Konsole, den AWS CLI Befehl `promote-read-replica`oder den `PromoteReadReplica`Amazon RDS-API-Vorgang verwenden.

   > **Note**
   >
   > Das Hochstufen kann einige Minuten in Anspruch nehmen. Wenn Sie eine Read Replica heraufstufen, stoppt RDS die Replikation und startet die Read Replica neu. Sobald der Neustart abgeschlossen ist, steht das Lesereplikat als neue DB-Instance zur Verfügung.

2. (Optional) Ändern Sie die neue DB-Instance in eine Multi-AZ-Bereitstellung. Weitere Informationen erhalten Sie unter Ändern einer Amazon RDS DB-Instance und Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Konsole

Hochstufen einer Read Replica zu einer eigenständigen DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der Amazon RDS-Konsole Databases (Datenbanken) aus.

   Der Bereich Databases (Datenbanken) wird angezeigt. Jedes Lesereplikat zeigt Replica (Replikat) in der Spalte Role (Rolle) an.
3. Wählen Sie das Lesereplikat aus, das Sie hochstufen möchten.
4. Wählen Sie für Actions (Aktionen) Promote (Hochstufen) aus.
5. Geben Sie auf der Seite Read Replica hochstufen den Aufbewahrungszeitraum und das Sicherungsfenster für die neu hochgestufte DB-Instance an.
6. Wenn die Optionen nach Ihrem Bedarf eingestellt sind, wählen Sie Weiter aus.
7. Wählen Sie auf der Bestätigungsseite Read Replica hochstufen aus.

AWS CLI

Verwenden Sie den AWS CLI `promote-read-replica`Befehl, um eine Read Replica zu einer eigenständigen DB-Instance hochzustufen.

Example

Für LinuxmacOS, oderUnix:

```
aws rds promote-read-replica \
    --db-instance-identifier myreadreplica
```

Windows:

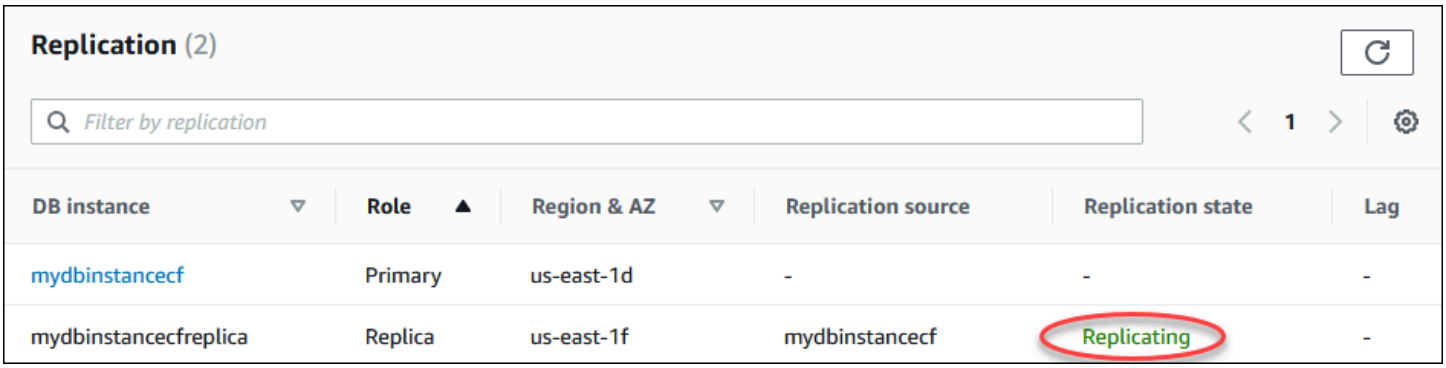
```
aws rds promote-read-replica ^
    --db-instance-identifier myreadreplica
```

RDS-API

Um ein Lesereplikat auf eine eigenständige DB-Instance hochzustufen, rufen Sie die Amazon-RDS-API-Operation `PromoteReadReplica` mit dem erforderlichen Parametern `DBInstanceIdentifier` auf.

Überwachen der Lesereplikation

Es gibt mehrere Arten, den Status eines Lesereplikats zu überwachen. Die Amazon-RDS-Konsole zeigt den Status eines Lesereplikats in den Lesereplikat-Details im Abschnitt Replication (Replikation) der Registerkarte Connectivity & security (Konnektivität & Sicherheit) an. Um die Details für ein Lesereplikat anzuzeigen, klicken Sie in der Liste der DB-Instances in der Amazon-RDS-Konsole auf den Namen des Lesereplikats.



Sie können den Status einer Read Replica auch mithilfe des AWS CLI `describe-db-instances` Befehls oder des Amazon `DescribeDBInstances` RDS-API-Vorgangs anzeigen.

Der Status eines Lesereplikats kann einer der folgenden sein:

- replicating (replizierend)– Das Lesereplikat wird erfolgreich repliziert.
- Replikation beeinträchtigt (nur SQL Server und PostgreSQL) – Replikate erhalten Daten von der primären Instance, doch mindestens eine Datenbank erhält möglicherweise keine Aktualisierungen. Dies kann beispielsweise auftreten, wenn ein Replikat gerade neu erstellte Datenbanken einrichtet. Er kann auch auftreten, wenn DDL oder große Objektänderungen in der blauen Umgebung einer Bereitstellung vorgenommen werden, die nicht unterstützt werden. blue/green

  Der Status wechselt nicht von `replication degraded` zu `error`, es sei denn, während des beeinträchtigten Zustands tritt ein Fehler auf.
- error (Fehler) – Während des Replikationsvorgangs ist ein Fehler aufgetreten. Überprüfen Sie das Feld Replikationsfehler in der Amazon RDS-Konsole oder im Ereignisprotokoll, um den genauen Fehler zu bestimmen. Weitere Informationen über Fehlerbehebung eines Replikationsfehlers finden Sie unter Behebung eines My SQL Read Replica-Problems.
- terminated (beendet) (nur MariaDB, MySQL oder PostgreSQL) – Die Replikation ist beendet. Dies tritt auf, wenn die Replikation für mehr als 30 aufeinanderfolgende Tage entweder manuell oder aufgrund eines Replikationsfehlers angehalten wurde. In diesem Fall beendet Amazon RDS die Replikation zwischen der primären DB-Instance und allen Lesereplikaten. Amazon RDS tut dies,

um erhöhten Speicheranforderungen in der Quell-DB-Instance vorzubeugen und lange Failover-Zeiten zu vermeiden.

Unterbrochene Replikation kann sich auf den Speicher auswirken, da die Protokolle aufgrund des hohen Volumens an Fehlermeldungen, die in das Protokoll geschrieben werden, an Größe und Anzahl zunehmen können. Unterbrochene Replikation kann sich auch auf die Ausfallwiederherstellung auswirken, bedingt durch den Zeitaufwand von Amazon RDS für das Warten und Verarbeiten einer großen Anzahl an Protokollen während der Wiederherstellung.

- terminated (beendet) (nur Oracle) – Die Replikation wurde beendet. Dies tritt auf, wenn die Replikation länger als 8 Stunden angehalten wurde, weil auf dem Lesereplikat nicht mehr genügend Speicherplatz vorhanden ist. In diesem Fall beendet Amazon RDS die Replikation zwischen der primären DB-Instance und den betroffenen Lesereplikaten. Dieser Status ist ein Terminalstatus und das Lesereplikat muss neu erstellt werden.
- stopped (angehalten) (nur für MySQL oder MariaDB) – Die Replikation wurde aufgrund einer benutzerseitigen Anfrage angehalten.
- replication stop point set (Stopppunkt für das Beenden der Replikation festgelegt) (nur MySQL) – Ein vom Benutzer initiierter Stopppunkt wurde mit der gespeicherten Prozedur festgelegt und die Replikation ist im Gange.
- replication stop point reached (Stopppunkt für das Beenden der Replikation erreicht) (nur MySQL) – Ein vom Benutzer initiierter Stopppunkt wurde mit der gespeicherten Prozedur festgelegt und die Replikation wurde gestoppt, da der Stopppunkt erreicht wurde.

Sie können sehen, wo eine DB-Instance repliziert wird, und wenn ja, ihren Replizierungsstatus überprüfen. Auf der Seite „ Datenbanken “ in der RDS-Konsole wird „ Primär “ in der Spalte „ Rolle “ angezeigt. Wählen Sie seinen DB-Instance-Namen. Auf der Detailseite auf der Registerkarte Connectivity & Security befindet sich der Replikationsstatus unter Replikation.

Überwachen einer Replikationsverzögerung

Sie können die Replikationsverzögerung in Amazon überwachen, CloudWatch indem Sie sich die Amazon `ReplicaLag` RDS-Metrik ansehen.

Für Db2 ist die `ReplicaLag` Metrik die maximale Verzögerung von Datenbanken, die ins Hintertreffen geraten sind, in Sekunden. Wenn beispielsweise zwei Datenbanken 5 Sekunden bzw. 10 Sekunden hinterherhinken, dann sind `ReplicaLag` es 10 Sekunden. Datenbanken ohne verfügbaren HADR-Status (High Availability Disaster Recovery) werden nicht in die Berechnung einbezogen.

Für MariaDB und MySQL gibt die`ReplicaLag` Metrik den Wert des`Seconds_Behind_Master` Feldes des`SHOW REPLICA STATUS` Befehls an. Häufige Ursachen für Replikationsverzögerungen in MySQL und MariaDB sind die Folgenden:

- Ein Netzwerkausfall.
- Schreibvorgänge auf Tabellen mit Indizes auf einem Lesereplikat. Wenn der Parameter `read_only` im Lesereplikat nicht auf 0 gesetzt ist, kann es die Replikation unterbrechen.
- Die Verwendung einer nicht-transaktionalen Speicher-Engine wie MyISAM: Replikation wird nur für die InnoDB-Speicher-Engine in MySQL und die XtraDB-Speicher-Engine in MariaDB unterstützt.

> **Note**
>
> Frühere Versionen von MariaDB wurden`SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie eine MariaDB-Version unter 10.5 verwenden, verwenden Sie. `SHOW SLAVE STATUS`

Wenn die Metrik `ReplicaLag` den Wert 0 erreicht, hat das Replikat den Stand der primären DB-Instance erreicht. Wenn die `ReplicaLag`-Metrik `-1` zurückgibt, ist die Replikation aktuell nicht aktiv. `ReplicaLag = -1` ist gleich `Seconds_Behind_Master = NULL`.

Für Oracle ist die Metrik `ReplicaLag` die Summe des `Apply Lag`-Wertes und der Differenz zwischen der aktuellen Zeit und dem `DATUM_TIME`-Wert der zeitlichen Verzögerung. Der `DATUM_TIME`-Wert ist der Zeitpunkt, an dem das Lesereplikat zuletzt Daten von seiner Quell-DB-Instance erhalten hat. Weitere Informationen finden Sie unter V$DATAGUARD_STATS in der Oracle-Dokumentation.

Für SQL Server ist die `ReplicaLag`-Metrik die maximale Verzögerung von Datenbanken, die zurückgefallen sind, in Sekunden. Wenn Sie beispielsweise zwei Datenbanken haben, die eine Verzögerung von 5 bzw. 10 Sekunden aufweisen, dann beträgt `ReplicaLag` 10 Sekunden. Die `ReplicaLag`-Metrik gibt den Wert der folgenden Abfrage zurück.

```
SELECT MAX(secondary_lag_seconds) max_lag FROM sys.dm_hadr_database_replica_states;
```

Weitere Informationen finden Sie unter secondary_lag_seconds in der Microsoft-Dokumentation.

`ReplicaLag` gibt `-1` zurück, wenn RDS die Verzögerung nicht ermitteln kann, z. B. während der Replikateinrichtung oder wenn sich das Lesereplikat im Status `error` befindet.

> **Note**
>
> Neue Datenbanken werden erst dann in die Verzögerungsberechnung einbezogen, wenn auf dem Lesereplikat auf sie zugegriffen werden kann.

Für PostgreSQL gibt die `ReplicaLag`-Metrik den Wert der folgenden Abfrage zurück.

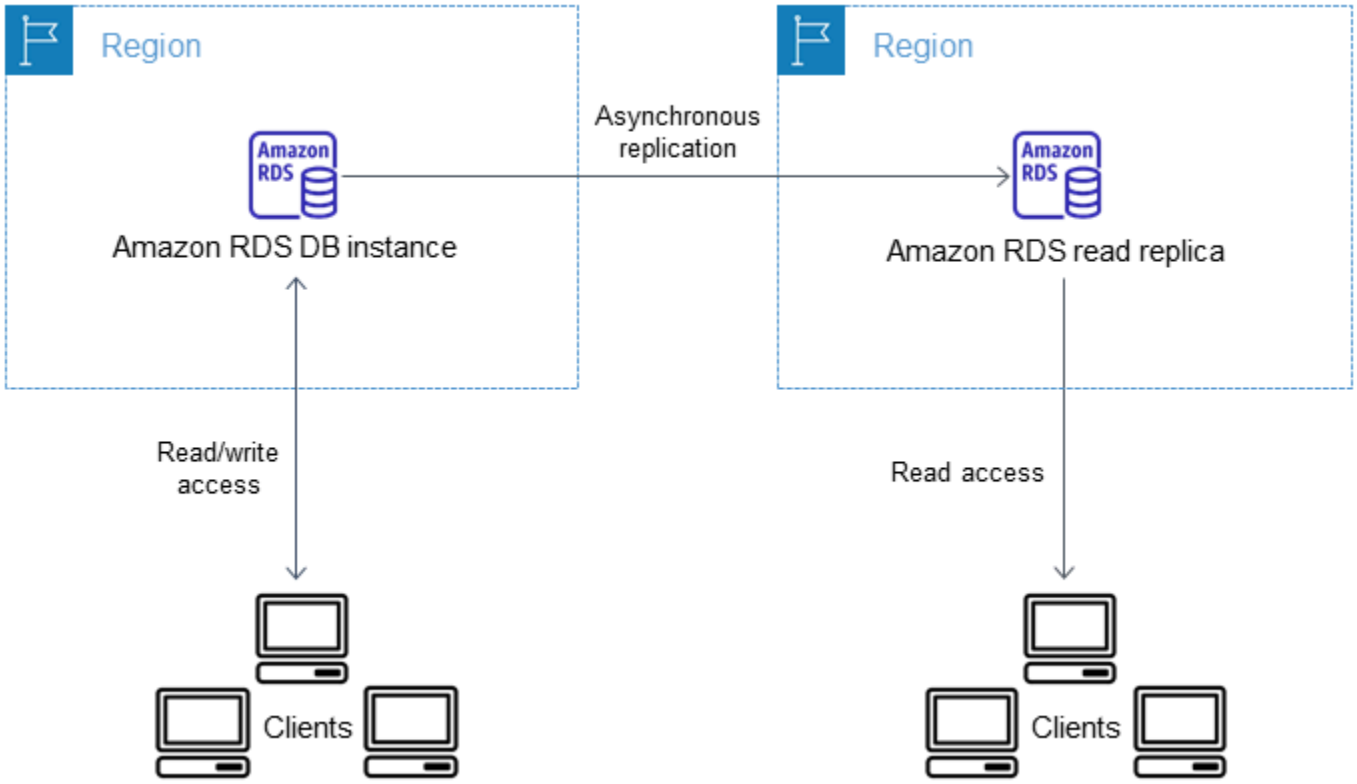
```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS reader_lag
```

Die PostgreSQL-Versionen 9.5.2 und höher verwenden physische Replikations-Slots, um die Write-Ahead-Log-Aufbewahrung (WAL) in der Quell-Instance zu verwalten. Für jede regionsübergreifende Lesereplikat-Instance erstellt Amazon RDS einen physischen Replikations-Slot und weist diesen der Instance zu. Zwei CloudWatch Amazon-Metriken `Oldest Replication Slot Lag` und zeigen`Transaction Logs Disk Usage`, wie weit das verzögerteste Replikat in Bezug auf die empfangenen WAL-Daten zurückliegt und wie viel Speicher für WAL-Daten verwendet wird. Der Wert `Transaction Logs Disk Usage` kann erheblich ansteigen, wenn ein regionsübergreifendes Lesereplikat signifikant verzögert ist.

Weitere Informationen zur Überwachung einer DB-Instance mit finden Sie CloudWatch unter. Überwachung von Amazon RDS mit Amazon CloudWatch

Erstellen eines Lesereplikats in einer anderen AWS-Region

Mit Amazon RDS können Sie eine Read Replica in einer anderen AWS-Region als der Quell-DB-Instance erstellen.



Sie erstellen eine Read Replica in einer anderen AWS-Region , um Folgendes zu tun:

- Verbessern Ihrer Notfallwiederherstellungsfähigkeiten.
- Skalieren Sie Lesevorgänge so, dass sie Ihren Benutzern AWS-Region näher kommen.
- Vereinfachen Sie die Migration von einem Rechenzentrum in einem AWS-Region Rechenzentrum in einem anderen AWS-Region.

Das Erstellen eines Read Replicas in einer AWS-Region anderen als der Quellinstanz ähnelt dem Erstellen eines Replikats in derselben. AWS-Region Sie können den API-Vorgang verwenden AWS Management Console, den `create-db-instance-read-replica`Befehl ausführen oder den `CreateDBInstanceReadReplica`API-Vorgang aufrufen.

> **Note**
>
> Um eine verschlüsselte Read Replica in einer anderen AWS-Region als der Quell-DB-Instance zu erstellen, muss die Quell-DB-Instance verschlüsselt sein.

Themen

- Verfügbarkeit von Regionen und Versionen
- Erstellen einer regionsübergreifenden Read Replica
- Wie Amazon RDS regionsübergreifende Replikationen durchführt
- Überlegungen zur regionsübergreifenden Replikation
- Kosten für regionsübergreifende Replikationen

Verfügbarkeit von Regionen und Versionen

Verfügbarkeit von Funktionen und Support variiert zwischen bestimmten Versionen der einzelnen Datenbank-Engines und über alle AWS-Regionen hinweg. Weitere Informationen zur Version und Region mit regionsübergreifender Replikation finden Sie unter Unterstützte Regionen und DB-Engines für regionsübergreifende Read Replicas in Amazon RDS.

Erstellen einer regionsübergreifenden Read Replica

Die folgenden Verfahren zeigen das Erstellen eines Lesereplikats aus einer Quell-DB-Instance von MariaDB, Microsoft SQL Server, MySQL, Oracle oder PostgreSQL in einer anderen AWS-Region.

Konsole

Sie können eine Read Replica erstellen, indem Sie den AWS-Regionen verwenden. AWS Management Console

Um eine Read Replica auf der anderen Seite der Konsole AWS-Regionen zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die DB-Instance von MariaDB, Microsoft SQL Server, MySQL, Oracle oder PostgreSQL aus, die Sie als Quelle für ein Lesereplikat verwenden möchten.
4. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.
5. Geben Sie unter DB instance identifier (DB-Instance-Kennung) einen Namen für das Lesereplikat ein.
6. Wählen Sie die Zielregion aus.
7. Wählen Sie die Instance-Spezifikationen aus, die Sie verwenden möchten. Wir empfehlen Ihnen, dieselbe oder eine größere DB-Instance-Klasse und denselben Speichertyp für das Lesereplikat zu verwenden.

8. Um eine verschlüsselte Read Replica in einem anderen AWS-Region zu erstellen:

   a. Wählen Sie Enable encryption (Verschlüsselung aktivieren) aus.

   b. Für AWS KMS key wählen Sie den AWS KMS key Bezeichner des KMS-Schlüssels in der Ziel- AWS-Region aus.

> **Note**
>
> Die Quell-DB-Instance muss verschlüsselt sein, um ein verschlüsseltes Lesereplikat zu erstellen. Weitere Informationen über das Verschlüsseln der Quell-DB-Instance finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.

9. Wählen Sie andere Optionen wie Autoscaling von Speicher.
10. Wählen Sie Read Replica erstellen aus.

AWS CLI

Um eine Read Replica aus einer Quell-MariaDB-, Microsoft SQL Server-, MySQL-, Oracle- oder PostgreSQL-DB-Instance in einer anderen zu erstellen AWS-Region, können Sie den Befehl verwenden. `create-db-instance-read-replica` In diesem Fall verwenden Sie `create-db-instance-read-replica` aus der AWS-Region , in der Sie Ihre Read Replica (Zielregion) haben möchten, und geben den Amazon-Ressourcennamen (ARN) für die Quell-DB-Instance an. Ein ARN bezeichnet eindeutig eine in Amazon Web Services erstellte Quelle.

Wenn sich beispielsweise Ihre Quell-DB-Instance in der Region US East (N. Virginia) befindet, sieht der ARN ähnlich wie in diesem Beispiel aus:

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Hinweise dazu finden Sie unter. ARNs Amazon-Ressourcennamen (ARNs) in Amazon RDS

Um eine Read Replica in einer anderen AWS-Region als der Quell-DB-Instance zu erstellen, können Sie den AWS CLI `create-db-instance-read-replica`Befehl vom Ziel AWS-Region aus verwenden. Die folgenden Parameter sind für die Erstellung einer Read Replica in einer anderen AWS-Region erforderlich:

- `--region`— Das Ziel AWS-Region , an dem die Read Replica erstellt wird.

- `--source-db-instance-identifier` – Die DB-Instance-Kennung für die Quell-DB-Instance. Dieser Bezeichner muss im ARN-Format der Quell- AWS-Region angegeben werden.
- `--db-instance-identifier` – Die Kennung für die Read Replica in der Ziel- AWS-Region.

Example eine regionsübergreifende Read Replica

Der folgende Code erstellt ein Read Replica in der USA West (Oregon)-Region von einer Quell-DB-Instance in der US East (N. Virginia)-Region.

Für LinuxmacOS, oderUnix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Die folgenden Parameter sind auch für die Erstellung einer verschlüsselten Read Replica in einer anderen AWS-Region erforderlich:

- `--kms-key-id`— Die AWS KMS key Kennung des KMS-Schlüssels, der zum Verschlüsseln der Read Replica im Ziel verwendet werden soll. AWS-Region

Example einer verschlüsselten regionsübergreifenden Read Replica

Der folgende Code erstellt ein verschlüsseltes Read Replica in der USA West (Oregon)-Region von einer Quell-DB-Instance in der US East (N. Virginia)-Region.

FürLinux, odermacOS: Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
```

```
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 \
    --kms-key-id my-us-west-2-key
```

Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 ^
    --kms-key-id my-us-west-2-key
```

Die `--source-region` Option ist erforderlich, wenn Sie eine verschlüsselte Lesereplik zwischen den Regionen AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) erstellen. Geben Sie für `--source-region` die AWS-Region der Quell-DB-Instance an.

Wenn `--source-region` nicht festgelegt ist, geben Sie einen `--pre-signed-url`-Wert an. Eine vorsignierte URL ist eine URL, die eine mit der Signaturversion 4 signierte Anforderung für den Befehl `create-db-instance-read-replica` enthält, die in der Quell- AWS-Region aufgerufen wird. Weitere Informationen zu dieser *`pre-signed-url`* Option finden Sie unter create-db-instance-read-replica in der Befehlsreferenz.AWS CLI

RDS-API

Um eine Read Replica aus einer Quell-MariaDB-, MySQL-, Microsoft SQL Server-, Oracle- oder PostgreSQL-DB-Instance in einer anderen zu erstellen AWS-Region, können Sie die Amazon RDS-API-Operation Create aufrufen. DBInstance ReadReplica In diesem Fall rufen Sie Create DBInstance ReadReplica von der AWS-Region Stelle aus auf, an der Sie die Read Replica haben möchten (Zielregion), und geben den Amazon-Ressourcennamen (ARN) für die Quell-DB-Instance an. Ein ARN bezeichnet eindeutig eine in Amazon Web Services erstellte Quelle.

Um eine verschlüsselte Read Replica in einer anderen AWS-Region als der Quell-DB-Instance zu erstellen, können Sie den Amazon `CreateDBInstanceReadReplica`RDS-API-Vorgang vom Ziel AWS-Region aus verwenden. Um eine verschlüsselte Read Replica in einer anderen zu erstellen AWS-Region, müssen Sie einen Wert für angeben. `PreSignedURL` `PreSignedURL`sollte eine Anforderung für den `CreateDBInstanceReadReplica`Vorgang enthalten, um die Quelle

AWS-Region aufzurufen, in der das Lesereplikat erstellt wurde. Weitere Informationen über `PreSignedUrl` finden Sie unter `CreateDBInstanceReadReplica`.

Wenn sich beispielsweise Ihre Quell-DB-Instance in der Region US East (N. Virginia) befindet, sieht der ARN ähnlich wie der folgende aus.

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Hinweise zu finden Sie ARNs unterAmazon-Ressourcennamen (ARNs) in Amazon RDS.

Example

```
https://us-west-2.rds.amazonaws.com/
    ?Action=CreateDBInstanceReadReplica
    &KmsKeyId=my-us-east-1-key
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCreateDBInstanceReadReplica
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBInstanceIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%123456789012%25253Adb%25253Amydbinstance
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4%2526SourceDBInstanceIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amydbinstance
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &DBInstanceIdentifier=myreadreplica
    &SourceDBInstanceIdentifier=&region-arn;rds:us-east-1:123456789012:db:mydbinstance
    &Version=2012-01-15
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2012-01-20T22%3A06%3A23.624Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Wie Amazon RDS regionsübergreifende Replikationen durchführt

Amazon RDS verwendet den folgenden Vorgang, um ein regionsübergreifendes Lesereplikat zu erstellen. Je nach den AWS-Regionen beteiligten Daten und der Menge der Daten in den Datenbanken kann dieser Vorgang Stunden in Anspruch nehmen. Sie können diese Informationen verwenden, um zu bestimmen, wie weit fortgeschritten der Vorgang ist, wenn Sie ein regionsübergreifendes Lesereplikat erstellen:

1. Amazon RDS beginnt mit der Konfiguration der Quell-DB-Instance als Replikationsquelle und setzt den Status auf Wird geändert ….
2. Amazon RDS beginnt mit dem Einrichten des bestimmten Lesereplikats in der Ziel- AWS-Region und setzt den Status auf creating.
3. Amazon RDS erstellt einen automatisierten DB-Snapshot der Quell-DB-Instance in der Quell-AWS-Region. Das Format des DB-Snapshot-Namens ist `rds:<InstanceID>-<timestamp>`, wobei `<InstanceID>` die Kennung der Quell-Instance ist und `<timestamp>` Startdatum und -zeit der Kopie sind. Beispielsweise wurde `rds:mysourceinstance-2013-11-14-09-24` aus der Instance `mysourceinstance` am `2013-11-14-09-24` erstellt. Während der Erstellung eines automatischen DB-Snapshots bleibt der Status der Quell-DB-Instance modifying (wird geändert), der Status des Lesereplikats creating (wird erstellt) und der Status des DB-Snapshots creating (wird erstellt). Die Fortschrittsspalte auf der DB-Snapshot-Seite in der Konsole meldet, wie weit die Erstellung des DB-Snapshots fortgeschritten ist. Wenn der DB-Snapshot fertiggestellt ist, wird sowohl der Status des DB-Snapshots als auch der Quell-DB-Instance auf Verfügbar gesetzt.
4. Amazon RDS beginnt mit einer regionsübergreifenden Snapshot-Kopie für die erste Datenübertragung. Die Snapshot-Kopie wird als ein automatischer Snapshot in der Ziel- AWS-Region mit dem Status creating aufgelistet. Sie hat denselben Namen wie der Quell-DB-Snapshot. Die Fortschrittspalte in der DB-Snapshot-Anzeige in der Konsole gibt an, wie weit die Kopie fortgeschritten ist. Wenn die Kopie fertiggestellt ist, wird der Status der DB-Snapshot-Kopie auf Verfügbar gesetzt.
5. Amazon RDS verwendet dann den kopierten DB-Snapshot für den ersten Datenladevorgang in das Lesereplikat. Während dieser Phase wird das Lesereplikat mit dem Status creating (wird erstellt) in der Liste von DB-Instances im Ziel aufgelistet. Wenn der Ladevorgang abgeschlossen ist, wird der Status des Lesereplikats auf available (verfügbar) gesetzt und die DB-Snapshot-Kopie wird gelöscht.
6. Wenn das Lesereplikat den Status "available (verfügbar)" erreicht, startet Amazon RDS mit der Replikation der Änderungen, die in der Quell-Instance seit dem Start der Operation zum Erstellen

des Lesereplikats vorgenommen wurden. Während dieser Phase ist die Verzögerungszeit der Replikation für das Lesereplikat größer als 0.

Weitere Informationen zur zeitlichen Verzögerung bei der Replikation finden Sie unter Überwachen der Lesereplikation.

Überlegungen zur regionsübergreifenden Replikation

Alle Überlegungen zur Durchführung einer Replikation innerhalb einer AWS-Region gelten für die regionsübergreifende Replikation. Die folgenden zusätzlichen Überlegungen gelten, wenn zwischen AWS-Regionen repliziert wird:

- Eine Quell-DB-Instance kann über regionsübergreifende Read Replicas in mehreren AWS-Regionen verfügen. Aufgrund der Beschränkung der Anzahl der ACL-Einträge (Access Control List) für die Quell-VPC kann RDS nicht mehr als fünf regionsübergreifende Read Replica-DB-Instances garantieren.
- Sie können zwischen den Regionen GovCloud (USA-Ost) und GovCloud (US-West) replizieren, jedoch nicht innerhalb oder außerhalb von (USA). GovCloud
- Für die folgenden RDS-Engines können Sie nur dann eine regionsübergreifende Amazon RDS-Read Replica aus einer Amazon RDS-DB-Quell-Instance erstellen, wenn es sich nicht um eine Read Replica einer anderen Amazon RDS-DB-Instance handelt:
  - RDS für SQL Server
  - RDS für Oracle
  - RDS für PostgreSQL-Versionen unter 14.1

  Diese Einschränkung gilt nicht für DB-Instances, auf denen RDS für PostgreSQL Version 14.1 und höher, RDS für MariaDB und RDS für MySQL ausgeführt wird.
- Sie können mit einer höheren Verzögerungszeit für jede Read Replica rechnen, die sich in einer anderen als der Quell-Instance befindet. AWS-Region Die zeitliche Verzögerung lässt sich auf längere Netzwerkkanäle zwischen regionalen Rechenzentren zurückführen.
- Für regionsübergreifende Lesereplikate muss jeder der Befehle zum Erstellen eines Lesereplikats, der den Parameter `--db-subnet-group-name` angibt, eine DB-Subnetzgruppe aus derselben VPC angeben.
- In den meisten Fällen verwendet das Lesereplikat die Standard-DB-Parametergruppe und DB-Optionsgruppe für die angegebene DB-Engine.

Für die Db2-DB-Engine müssen Sie eine benutzerdefinierte Parametergruppe für das Replikat angeben. Für Replikate, die das BYOL-Modell verwenden, muss diese benutzerdefinierte Parametergruppe Ihr und enthalten. IBM Site ID IBM Customer ID Sie können diese benutzerdefinierte Parametergruppe für das Replikat mithilfe der AWS Management Console, der oder der AWS CLI RDS-API angeben. Weitere Informationen finden Sie unter IBM IDs für Bring Your Own License (BYOL) für Db2.

Für die MySQL- und Oracle-DB-Engines können Sie in der `--db-parameter-group-name` Option des AWS CLI Befehls create-db-instance-read-replica eine benutzerdefinierte Parametergruppe für die Read Replica angeben. Sie können keine benutzerdefinierte Parametergruppe angeben, wenn Sie die AWS Management Console verwenden.

- Das Lesereplikat verwendet die Standardsicherheitsgruppe.
- Wenn bei DB-Instances von MariaDB, Microsoft SQL Server, MySQL und Oracle die Quell-DB-Instance für ein nicht-regionsübergreifendes Lesereplikat gelöscht wird, wird das Lesereplikat hochgestuft.
- Für PostgreSQL-DB-Instances wird der Replikationsstatus des Lesereplikats auf `terminated` gesetzt, wenn die Quell-DB-Instance für ein regionsübergreifendes Lesereplikat gelöscht wird. Das Lesereplikat wird nicht hochgestuft.

  Sie müssen das Lesereplikat manuell hochstufen oder löschen.

Anfordern einer regionsübergreifenden Read Replica

Um mit der Quellregion zu kommunizieren und die Erstellung einer regionsübergreifenden Read Replica anzufordern, muss der Anforderer (IAM-Rolle oder IAM-Benutzer) Zugriff auf die Quell-DB-Instance und die Quellregion haben.

Bestimmte Bedingungen in der IAM-Richtlinie des Anforderers können dazu führen, dass die Anfrage fehlschlägt. Die folgenden Beispiele gehen davon aus, dass sich die Quell-DB-Instance in USA Ost (Ohio) befindet, und die Read Replica in US East (N. Virginia) erstellt wird. Diese Beispiele zeigen die Bedingungen in der IAM-Richtlinie des Anforderers, die dazu führen, dass die Anfrage fehlschlägt:

- Die Richtlinie des Anforderers hat eine Bedingung für `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
```

```
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

Die Anfrage schlägt fehl, weil die Richtlinie den Zugriff auf die Quellregion nicht zulässt. Für eine erfolgreiche Anfrage geben Sie sowohl die Quell- als auch die Zielregion an.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- Die Richtlinie des Anforderers erlaubt keinen Zugriff auf die Quell-DB-Instance.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "arn:aws:rds:us-east-1:123456789012:db:myreadreplica"
...
```

Geben Sie für eine erfolgreiche Anfrage sowohl die Quell-Instance als auch das Replikat an.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:db:myreadreplica",
    "arn:aws:rds:us-east-2:123456789012:db:mydbinstance"
]
...
```

- Die Richtlinie des Anforderers lehnt `aws:ViaAWSService` ab.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

  Die Kommunikation mit der Quellregion erfolgt über RDS im Namen des Anforderers. Für eine erfolgreiche Anfrage lehnen Sie keine Aufrufe von AWS -Services ab.

- Die Richtlinie des Anforderers hat eine Bedingung für `aws:SourceVpc` oder `aws:SourceVpce`.

  Diese Anforderungen können fehlschlagen, da RDS den Aufruf an die entfernte Region nicht vom angegebenen VPC- oder VPC-Endpunkt ausführt.

Wenn Sie eine der vorherigen Bedingungen verwenden müssen, die dazu führen würde, dass eine Anfrage fehlschlägt, können Sie eine zweite Anweisung mit `aws:CalledVia` in Ihrer Richtlinie aufnehmen, um die Anfrage erfolgreich zu machen. Zum Beispiel können Sie `aws:CalledVia` mit `aws:SourceVpce` wie hier gezeigt verwenden:

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CreateDBInstanceReadReplica"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
```

```
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Weitere Informationen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

Autorisieren des Read Replica

Nachdem eine regionsübergreifende DB-Anforderung zum Erstellung der Read Replica `success` zurückgibt, startet RDS die Replica-Erstellung im Hintergrund. Es wird eine Berechtigung für RDS für den Zugriff auf die Quell-DB-Instance erstellt. Diese Autorisierung verknüpft die Quell-DB-Instance mit der Read Replica und ermöglicht es RDS, nur in die angegebene Read Replica zu kopieren.

Die Autorisierung wird von RDS unter Verwendung der `rds:CrossRegionCommunication`-Berechtigung in der serviceverknüpfte IAM-Rolle verifiziert. Wenn das Replikat autorisiert ist, kommuniziert RDS mit der Quellregion und schließt die Erstellung der Read Replica ab.

RDS hat keinen Zugriff auf DB-Instances, die zuvor nicht von einer `CreateDBInstanceReadReplica`-Anfrage authorisiert wurden. Die Autorisierung wird widerrufen, wenn die Erstellung der Read Replica abgeschlossen ist.

RDS verwendet die serviceverknüpfte Rolle, um die Autorisierung in der Quellregion zu überprüfen. Wenn Sie die serviceverknüpfende Rolle während des Replizierungserstellungsprozesses löschen, schlägt die Erstellung fehl.

Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen im -IAM-Benutzerhandbuch.

Anmeldeinformationen verwenden AWS Security Token Service

Sitzungstoken vom Endpunkt global AWS Security Token Service (AWS STS) sind nur in Regionen gültig AWS-Regionen , die standardmäßig aktiviert sind (kommerzielle Regionen). Wenn Sie Anmeldeinformationen aus dem `assumeRole` API-Vorgang in verwenden AWS STS, verwenden Sie den regionalen Endpunkt, wenn es sich bei der Quellregion um eine Opt-in-Region handelt. Andernfalls schlägt die Anforderung fehl. Dies liegt daran, dass Ihre Anmeldeinformationen in beiden Regionen gültig sein müssen. Dies gilt nur für Opt-in-Regionen, wenn der regionale AWS STS Endpunkt verwendet wird.

Um den globalen Endpunkt zu verwenden, stellen Sie sicher, dass er für beide Regionen in den Vorgängen aktiviert ist. Stellen Sie den globalen Endpunkt `Valid in all AWS-Regionen` in den AWS STS Kontoeinstellungen auf ein.

Die gleiche Regel gilt für Anmeldeinformationen im vorsignierten URL-Parameter.

Weitere Informationen finden Sie unter Verwaltung AWS STS in an AWS-Region im IAM-Benutzerhandbuch.

Kosten für regionsübergreifende Replikationen

Für Daten, die für regionsübergreifende Replikation übermittelt werden, fallen Amazon RDS-Datenübertragungskosten an. Bei diesen regionsübergreifenden Replikationsaktionen fallen Gebühren für die übermittelten Daten aus der Quell- AWS-Region an:

- Wenn Sie ein Lesereplikat erstellen, erstelllt Amazon RDS einen Snapshot der Quell-Instance und leitet den Snapshot an die AWS-Region der Read Replica weiter.
- Für jede in den Quelldatenbanken vorgenommene Datenänderung überträgt Amazon RDS Daten von der Quelle AWS-Region zur Read Replica AWS-Region.

Weitere Informationen zu den Kosten von Datenübertragungen finden Sie unter Amazon RDS – Preise.

Für MySQL- und MariaDB-Instances können Sie Ihre Datenübertragungskosten senken, indem Sie die Anzahl der erstellten regionsübergreifenden Lesereplikate reduzieren. Nehmen wir zum Beispiel an, Sie haben eine Quell-DB-Instance in einer AWS-Region und möchten in einer anderen drei Read Replicas haben. AWS-Region In diesem Fall können Sie nur eines der Lesereplikate aus der Quell-DB-Instance erstellen. Die anderen beiden Lesereplikate erstellen Sie aus dem ersten Lesereplikat anstelle der Quell-DB-Instance.

Wenn Sie beispielsweise `source-instance-1` in einer haben AWS-Region, können Sie Folgendes tun:

- Erstellen Sie `read-replica-1` das Neue AWS-Region und geben Sie `source-instance-1` es als Quelle an.
- Erstellen Sie `read-replica-2` aus `read-replica-1`.
- Erstellen Sie `read-replica-3` aus `read-replica-1`.

In diesem Beispiel werden Ihnen nur die übermittelten Daten von `source-instance-1` nach `read-replica-1` in Rechnung gestellt. Ihnen werden keine Datenübertragungen von `read-replica-1` zu den anderen beiden Replikaten in Rechnung gestellt, weil sie sich in derselben AWS-Region befinden. Wenn Sie alle drei Replikate direkt von einem anderen aus erstellen AWS-Region, werden Ihnen die Datenübertragungen zu allen drei Replikaten `source-instance-1` in Rechnung gestellt.

Verschlagwortung von Amazon-Ressourcen RDS

Ein RDS Amazon-Tag ist ein Name-Wert-Paar, das Sie definieren und einer RDS Amazon-Ressource wie einer DB-Instance oder einem DB-Snapshot zuordnen. Der Name wird als der Schlüssel bezeichnet. Optional können Sie einen Wert für den Schlüssel angeben.

Sie können Amazon AWS Management Console, The oder Amazon verwenden AWS CLI, um Tags RDS API zu RDS Amazon-Ressourcen hinzuzufügen, aufzulisten und zu löschen. Wenn Sie das CLI oder verwendenAPI, stellen Sie sicher, dass Sie den Amazon-Ressourcennamen (ARN) für die RDS Ressource angeben, mit der Sie arbeiten möchten. Weitere Informationen zum Konstruieren einer finden Sie ARN unterErstellen eines ARN für Amazon RDS.

Sie können Tags verwenden, um Metadaten zu Ihren Amazon RDS hinzuzufügen. Sie können die Tags verwenden, um Ihre eigenen Notationen unter anderem zu Datenbank-Instances, Snapshots und Aurora-Clustern hinzuzufügen. Auf diese Weise können Sie Ihre Amazon RDS dokumentieren. Sie können die Tags auch mit automatisierten Wartungsvorgängen verwenden.

Insbesondere können Sie diese Tags mit IAM Richtlinien verwenden. Sie können sie verwenden, um den Zugriff auf Amazon RDS zu verwalten und zu kontrollieren, welche Aktionen auf diese Ressourcen angewendet werden können. Sie können Tags auch verwenden, um Kosten zu verfolgen, indem Ausgaben für ähnlich markierte Ressourcen gruppiert werden.

Sie können die folgenden Amazon RDS taggen:

- DB-Instances
- DB-Cluster
- Globale Aurora-Cluster
- DB-Cluster-Endpunkte
- Read Replicas
- DB-Snapshots
- DB-Cluster-Snapshots
- Reservierte DB-Instances
- Ereignisabonnements
- DB-Optionsgruppen
- DB-Parametergruppen

- DB-Cluster-Parametergruppen
- DB-Subnetzgruppen
- RDSProxys
- RDSProxy-Endpunkte
- Blau/Grün-Bereitstellungen
- Keine Integrationen ETL

> **Note**
>
> Derzeit können Sie RDS Proxys und RDS Proxy-Endpunkte nicht mit dem taggen. AWS Management Console

Themen

- Warum Amazon RDS Resource Tags verwenden?
- So funktionieren RDS Amazon-Ressourcen-Tags
- Bewährte Methoden für das Taggen von Amazon-Ressourcen RDS
- Tags in DB-Snapshots kopieren
- Hinzufügen und Löschen von Tags in Amazon RDS
- Tutorial: Geben Sie mithilfe von Tags an, welche DB-Instances gestoppt werden sollen

Warum Amazon RDS Resource Tags verwenden?

Sie können Tags verwenden, um Folgendes zu tun:

- Kategorisieren Sie Ihre RDS Ressourcen nach Anwendung, Projekt, Abteilung, Umgebung usw. Sie könnten beispielsweise einen Tag-Schlüssel verwenden, um eine Kategorie zu definieren, wobei der Tag-Wert ein Element in dieser Kategorie ist. Sie könnten das Tag erstellen`environment=prod`. Oder Sie können einen Tag-Schlüssel `project` und einen Tag-Wert von definieren`Salix`, was darauf hinweist, dass dem Salix-Projekt eine RDS Amazon-Ressource zugewiesen ist.
- Automatisieren Sie Aufgaben im Ressourcenmanagement. Sie könnten beispielsweise ein Wartungsfenster für markierte Instanzen erstellen`environment=prod`, das sich von dem Fenster

für markierte Instanzen unterscheidetenvironment=test. Sie könnten auch automatische DB-Snapshots für markierte environment=prod Instances konfigurieren.

- Steuern Sie den Zugriff auf RDS Ressourcen innerhalb einer IAM Richtlinie. Hierzu können Sie den globalen Bedingungsschlüssel aws:ResourceTag/*tag-key* verwenden. Eine Richtlinie könnte es beispielsweise nur Benutzern in der DBAdmin Gruppe ermöglichen, DB-Instances zu ändern, die mit gekennzeichnet sindenvironment=prod. Informationen zur Verwaltung des Zugriffs auf markierte Ressourcen mit IAM Richtlinien finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS Steuern des Zugriffs auf AWS Ressourcen im AWS Identity and Access Management-Benutzerhandbuch.
- Überwachen Sie Ressourcen anhand eines Tags. Sie können beispielsweise ein CloudWatch Amazon-Dashboard für DB-Instances erstellen, die mit gekennzeichnet sindenvironment=prod.
- Verfolgen Sie die Kosten, indem Sie die Ausgaben für Ressourcen mit ähnlichen Tags gruppieren. Wenn Sie beispielsweise RDS Ressourcen, die mit dem Salix-Projekt verknüpft sindproject=Salix, mit taggen, können Sie Kostenberichte für dieses Projekt erstellen und Ausgaben diesem Projekt zuordnen. Weitere Informationen finden Sie unter So funktioniert die AWS Abrechnung mit Tags in Amazon RDS.

So funktionieren RDS Amazon-Ressourcen-Tags

AWS wendet Ihren Tags keine semantische Bedeutung an. Tags werden streng als Zeichenfolgen interpretiert.

Themen

- Tag-Sets bei Amazon RDS
- Tag-Struktur in Amazon RDS
- RDSAmazon-Ressourcen, die für das Tagging in Frage kommen
- So funktioniert die AWS Abrechnung mit Tags in Amazon RDS

Tag-Sets bei Amazon RDS

Jede RDS Amazon-Ressource hat einen Container, der als Tag-Set bezeichnet wird. Der Container enthält alle Tags, die der Ressource zugewiesen sind. Eine Ressource hat genau einen Tagsatz.

Ein Tag-Set enthält 0—50 Tags. Wenn Sie einer RDS Ressource ein Tag mit demselben Schlüssel wie ein vorhandenes Ressourcen-Tag hinzufügen, überschreibt der neue Wert den alten.

Tag-Struktur in Amazon RDS

Die Struktur eines RDS Tags ist wie folgt:

Tag-Schlüssel

Der Schlüssel ist der erforderliche Name des Tags. Der Zeichenkettenwert muss 1—128 Unicode-Zeichen lang sein und darf nicht mit einem `aws:` oder als Präfix versehen werden. `rds:` Die Zeichenfolge kann nur den Satz von Unicode-Buchstaben, Ziffern, Leerzeichen,,_,., : / =+, - und enthalten. @ Der Java-Regex ist. `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$"` Bei Tag-Schlüsseln wird die Groß- und Kleinschreibung beachtet. Somit sind die Schlüssel `project` und B `Project` unterschiedlich.

Ein Schlüssel ist für einen Tagsatz eindeutig. Sie können beispielsweise kein Schlüsselpaar in einem Tag-Set haben, bei dem der Schlüssel zwar derselbe, aber unterschiedliche Werte hat, wie `project=Trinity` z. B. und. `project=Xanadu`

Tag-Wert

Der Wert ist ein optionaler Zeichenkettenwert des Tags. Der Zeichenkettenwert muss 1—256 Unicode-Zeichen lang sein. Die Zeichenfolge kann nur den Satz von Unicode-Buchstaben, Ziffern, Leerzeichen,_,.,:, / = +-, und enthalten. @ Der Java-Regex ist. `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$"` Bei Tag-Werten muss die Groß- und Kleinschreibung beachtet werden. Somit sind die Werte `prod` und 2 `Prod` unterschiedlich.

Werte müssen in einem Tag-Set nicht eindeutig sein und können Null sein. Es ist z. B. ein Schlüssel-Wert-Paar in einem Tag-Satz `project=Trinity` und `cost-center=Trinity` möglich.

RDSAmazon-Ressourcen, die für das Tagging in Frage kommen

Sie können die folgenden RDS Amazon-Ressourcen taggen:

- DB-Instances
- DB-Cluster
- DB-Cluster-Endpunkte
- Read Replicas
- DB-Snapshots

- DB-Cluster-Snapshots
- Reservierte DB-Instances
- Ereignisabonnements
- DB-Optionsgruppen
- DB-Parametergruppen
- DB-Cluster-Parametergruppen
- DB-Subnetzgruppen
- RDSProxys
- RDSProxy-Endpunkte

> **Note**
>
> Derzeit können Sie RDS Proxys und RDS Proxy-Endpunkte nicht mit dem taggen. AWS Management Console

- Blau/Grün-Bereitstellungen
- ETLZero-Integrationen (Vorschau)

So funktioniert die AWS Abrechnung mit Tags in Amazon RDS

Verwenden Sie Tags, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Melden Sie sich dazu an, um Ihre AWS-Konto Rechnung mit den Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können. Weitere Informationen finden Sie unter Verwendung von Kostenzuordnungs-Tags im AWS Billing -Benutzerhandbuch.

So funktionieren Kostenzuweisungs-Tags mit

Sie können einem ein Tag hinzufügen. Diese Gruppierung erscheint jedoch nicht in Ihrer Rechnung. Damit Kostenzuweisungs-Tags auf angewendet werden können, müssen die folgenden Bedingungen erfüllt sein:

- Die Tags müssen an die übergeordnete DB-Instance angehängt werden.

- Die übergeordnete DB-Instance muss genauso vorhanden sein AWS-Konto wie der .
- Die übergeordnete DB-Instance muss genauso vorhanden sein AWS-Region wie der .

gelten als verwaist, wenn sie nicht in derselben Region wie die übergeordnete DB-Instance existieren oder wenn die übergeordnete DB-Instance gelöscht wurde. Verwaiste DB-Snapshots unterstützen keine Kostenzuweisungs-Tags. Die Kosten für verwaiste Snapshots werden in einer einzigen Zeile ohne Tags zusammengefasst. Kontoübergreifende gelten nicht als verwaist, wenn die folgenden Bedingungen erfüllt sind:

- Sie befinden sich in derselben Region wie die übergeordnete DB-Instance.
- Die übergeordnete DB-Instance gehört dem Quellkonto.

> **Note**
>
> Wenn die übergeordnete DB-Instance einem anderen Konto gehört, gelten die Kostenzuweisungs-Tags nicht für kontenübergreifende Snapshots im Zielkonto.

Bewährte Methoden für das Taggen von Amazon-Ressourcen RDS

Wir empfehlen Ihnen, sich bei der Verwendung von Tags an die folgenden bewährten Methoden zu halten:

- Dokumentieren Sie Konventionen für die Verwendung von Tags, die von allen Teams in Ihrer Organisation befolgt werden. Stellen Sie insbesondere sicher, dass die Namen sowohl aussagekräftig als auch konsistent sind. Standardisieren Sie beispielsweise das Format, `environment:prod` anstatt einige Ressourcen mit zu kennzeichnen. `env:production`

  > **Important**
  >
  > Speichern Sie keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen in Tags.

- Automatisieren Sie das Tagging, um Konsistenz zu gewährleisten. Sie können beispielsweise die folgenden Techniken verwenden:
  - Fügen Sie Tags in eine AWS CloudFormation Vorlage ein. Wenn Sie Ressourcen mit der Vorlage erstellen, werden die Ressourcen automatisch markiert.

  - Definieren und wenden Sie Tags mithilfe von AWS Lambda Funktionen an.
  - Erstellen Sie ein SSM Dokument, das Schritte zum Hinzufügen von Tags zu Ihren RDS Ressourcen enthält.
- Verwenden Sie Tags nur bei Bedarf. Sie können bis zu 50 Tags für eine einzelne RDS Ressource hinzufügen. Eine bewährte Methode besteht jedoch darin, eine unnötige Zunahme und Komplexität von Tags zu vermeiden.
- Überprüfen Sie die Tags regelmäßig auf Relevanz und Richtigkeit. Entfernen oder ändern Sie veraltete Tags nach Bedarf.
- Erwägen Sie das Erstellen von Tags mit dem AWS Tag-Editor in der AWS Management Console. Sie können den Tag-Editor verwenden, um mehreren unterstützten AWS Ressourcen, einschließlich RDS Ressourcen, gleichzeitig Tags hinzuzufügen. Weitere Informationen finden Sie unter Tag Editor im Benutzerhandbuch von AWS Resource Groups.

Tags in DB-Snapshots kopieren

Wenn Sie eine DB-Instance erstellen oder wiederherstellen, können Sie festlegen, dass die Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden. Das Kopieren von Tags stellt sicher, dass die Metadaten für die DB-Snapshots mit denen der Quell-DB-Instance übereinstimmen. Es wird außerdem sichergestellt, dass alle Zugriffsrichtlinien für die DB-Snapshots auch mit denen der Quell-DB-Instance übereinstimmen.

Sie können für die folgenden Aktionen festlegen, dass Tags in DB-Snapshots kopiert werden:

- Erstellen einer DB-Instance
- Wiederherstellung einer DB-Instance
- Erstellen eines Lesereplikats
- Kopieren eines DB-Snapshots

Um Tags für die vorherigen Aktionen zu kopieren, wählen Sie Tags in Snapshots kopieren in der AWS Management Console, oder geben Sie `--copy-tags-to-snapshot` in an AWS CLI.

In den meisten Fällen werden Tags nicht standardmäßig kopiert. Wenn Sie jedoch eine DB-Instance aus einem DB-Snapshot wiederherstellen, RDS prüft es, ob Sie neue Tags angeben. Wenn ja, werden die neuen Tags zur wiederhergestellten DB-Instance hinzugefügt. Wenn keine neuen Tags vorhanden sind, werden die Tags aus der Quell-DB-Instance zum Zeitpunkt der Snapshot-Erstellung zur wiederhergestellten DB-Instance RDS hinzugefügt.

Um zu verhindern, dass Tags von Quell-DB-Instances zu wiederhergestellten DB-Instances hinzugefügt werden, empfehlen wir Ihnen, beim Wiederherstellen einer DB-Instance neue Tags anzugeben.

> **Note**
>
> In einigen Fällen können Sie einen Wert für den `--tags` Parameter des create-db-snapshot AWS CLI Befehls angeben. Oder Sie könnten mindestens ein Tag für die reateDBSnapshotAPIC-Operation angeben. Kopiert in diesen Fällen RDS keine Tags von der Quell-DB-Instance in den neuen DB-Snapshot. Diese Funktionalität gilt sogar, wenn in der Quell-DB-Instance die Option `--copy-tags-to-snapshot` (`CopyTagsToSnapshot`) aktiviert ist.
>
> Wenn Sie diesen Ansatz verwenden, können Sie eine Kopie einer DB-Instance aus einem DB-Snapshot erstellen. Dieser Ansatz vermeidet das Hinzufügen von Tags, die nicht für die neue DB-Instance gelten. Sie erstellen Ihren DB-Snapshot mit dem AWS CLI `create-db-snapshot` Befehl (oder der `CreateDBSnapshot` RDS API Operation). Nachdem Sie Ihren DB-Snapshot erstellt haben, können Sie Tags hinzufügen. Dieser Vorgang wird später in diesem Thema beschrieben.

Hinzufügen und Löschen von Tags in Amazon RDS

Sie haben die folgenden Möglichkeiten:

- Erstellen Sie Tags, wenn Sie eine Ressource erstellen, z. B. wenn Sie den AWS CLI Befehl ausführen`create-db-instance`.
- Fügen Sie mithilfe des Befehls Tags zu einer vorhandenen Ressource hinzu`add-tags-to-resource`.
- Listet mithilfe des Befehls Tags auf, die einer bestimmten Ressource zugeordnet sind`list-tags-for-resource`.
- Aktualisieren Sie die Tags mithilfe des Befehls`add-tags-to-resource`.
- Entfernen Sie mithilfe des Befehls Tags aus einer Ressource`remove-tags-from-resource`.

Die folgenden Verfahren zeigen, wie Sie typische Tagging-Operationen für Ressourcen durchführen können, die sich auf DB-Instances beziehen. Beachten Sie, dass Tags für Autorisierungszwecke

zwischengespeichert werden. Aus diesem Grund können beim Hinzufügen oder Aktualisieren von Tags zu RDS Amazon-Ressourcen mehrere Minuten vergehen, bis die Änderungen verfügbar sind.

Konsole

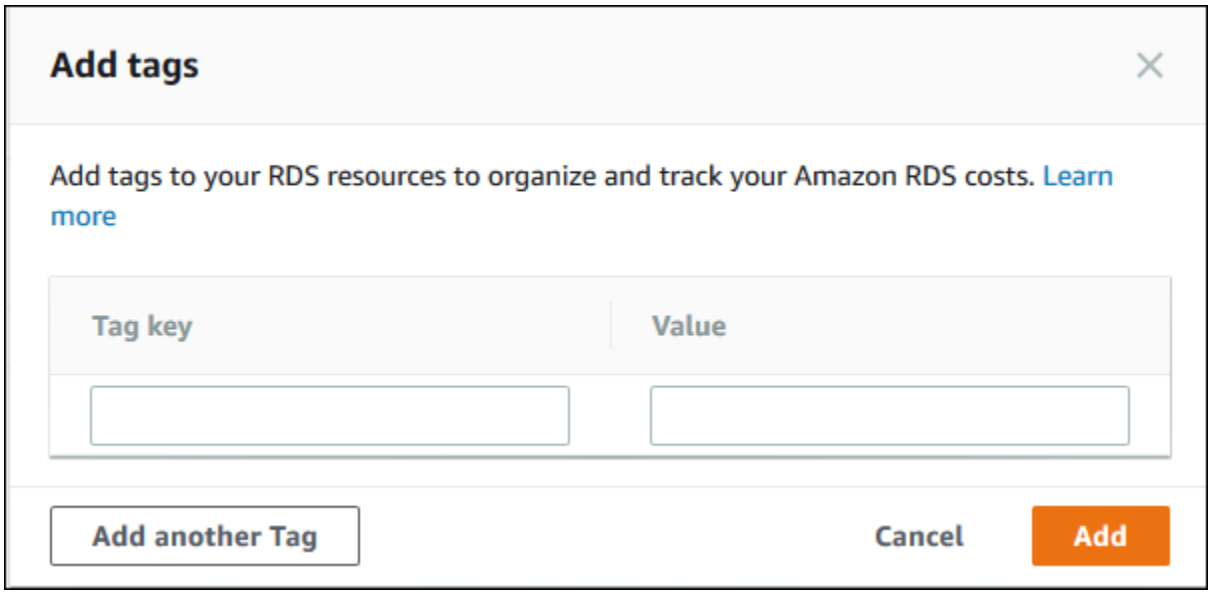
Der Prozess zum Markieren einer RDS Amazon-Ressource ist für alle Ressourcen ähnlich. Das folgende Verfahren zeigt, wie Sie eine RDS Amazon-DB-Instance taggen.

So fügen Sie ein Tag zu einer DB-Instance hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.

> **Note**
>
> Geben Sie unter Filter databases (Datenbanken filtern) eine Textzeichenfolge ein, um die Liste der DB-Instances im Bereich Databases (Datenbanken) zu filtern. Es werden nur DB-Instances angezeigt, welche die Zeichenfolge enthalten.

3. Wählen Sie den Namen der DB-Instance aus, die Sie mit einem Tag versehen möchten, um deren Details anzuzeigen.
4. Scrollen Sie im Detailbereich nach unten zum Bereich Tags.
5. Wählen Sie Add aus. Das Fenster Add tags (Tags hinzufügen) wird angezeigt.



6. Geben Sie einen Wert für Tag key (Tag-Schlüssel) und Wert ein.

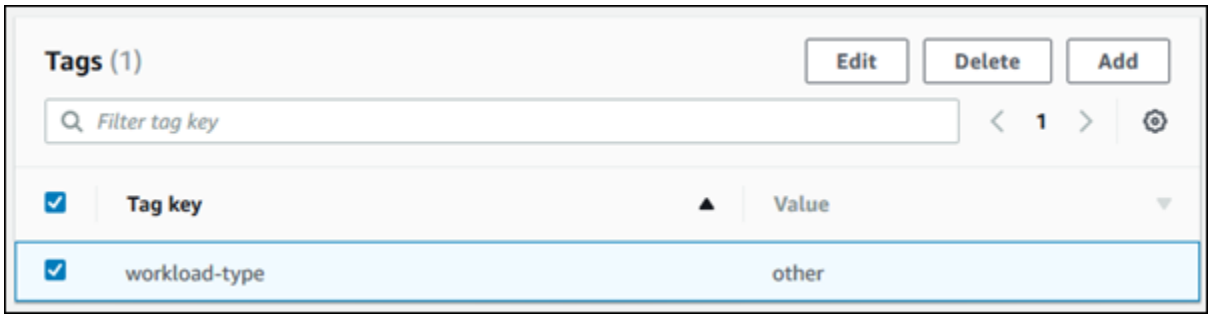
7. Wenn Sie ein weiteres Tag hinzufügen möchten, wählen Sie Add another Tag (Weiteres Tag hinzufügen) aus und geben Sie unter Tag key (Tag-Schlüssel) und Wert einen Wert ein.

   Wiederholen Sie diesen Schritt, bis Sie alle Tags hinzugefügt haben.
8. Wählen Sie Add aus.

So löschen Sie ein Tag aus einer DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.

   > **Note**
   >
   > Geben Sie im Feld Databases (Datenbanken) eine Textzeichenfolge ein, um die Liste der DB-Instances im Bereich Databases (Datenbanken) zu filtern. Es werden nur DB-Instances angezeigt, welche die Zeichenfolge enthalten.
3. Wählen Sie den Namen der entsprechenden DB-Instance aus, um deren Details anzuzeigen.
4. Scrollen Sie im Detailbereich nach unten zum Bereich Tags.
5. Wählen Sie das Tag aus, das Sie löschen möchten.



6. Wählen Sie Löschen und dann im Fenster Löschen von Tags erneut Löschen aus.

AWS CLI

Mithilfe der können Sie Tags für eine DB-Instance hinzufügen, auflisten oder entferne AWS CLI.

- Verwenden Sie den AWS CLI Befehl, um einer RDS Amazon-Ressource ein oder mehrere Tags hinzuzufügen `add-tags-to-resource`.

- Verwenden Sie den AWS CLI Befehl, um die Tags auf einer RDS Amazon-Ressource aufzulisten `list-tags-for-resource`.
- Verwenden Sie den AWS CLI Befehl, um ein oder mehrere Tags aus einer RDS Amazon-Ressource zu entfernen `remove-tags-from-resource`.

Weitere Informationen darüber, wie Sie das erforderliche Objekt erstellenARN, finden Sie unterErstellen eines ARN für Amazon RDS.

RDS API

Sie können Tags für eine DB-Instance mithilfe von Amazon hinzufügen, auflisten oder entfernen RDSAPI.

- Verwenden Sie den `AddTagsToResource`Vorgang, um einer RDS Amazon-Ressource ein Tag hinzuzufügen.
- Um Tags aufzulisten, die einer RDS Amazon-Ressource zugewiesen sind, verwenden Sie den `ListTagsForResource`.
- Verwenden Sie den `RemoveTagsFromResource`Vorgang, um Tags aus einer RDS Amazon-Ressource zu entfernen.

Weitere Informationen darüber, wie Sie die erforderlichen Daten erstellenARN, finden Sie unterErstellen eines ARN für Amazon RDS.

Wenn XML Sie mit Amazon arbeiten RDSAPI, verwenden Tags das folgende Schema:

```
<Tagging>
      <TagSet>
          <Tag>
              <Key>Project</Key>
              <Value>Trinity</Value>
          </Tag>
          <Tag>
              <Key>User</Key>
              <Value>Jones</Value>
          </Tag>
      </TagSet>
  </Tagging>
```

Die folgende Tabelle enthält eine Liste der zulässigen XML Tags und ihrer Eigenschaften. Bei den Werten für `Key` und wird zwischen Groß- und Kleinschreibung `Value` unterschieden. Zum Beispiel `PROJECT=Trinity` sind `project=Trinity` und unterschiedliche Tags.

Markieren von Elementen	Beschreibung
TagSet	Ein Tag-Set ist ein Container für alle Tags, die einer RDS Amazon-Ressource zugewiesen sind. Es ist nur ein Tag-Satz pro Ressource zulässig. Sie arbeiten mit einem TagSet nur über Amazon RDSAPI.
Tag	Ein Tag ist ein benutzerdefiniertes Schlüssel-Wert-Paar. Ein Tag-Satz kann 1 bis 50 Tags enthalten.
Schlüssel	Ein Schlüssel ist der erforderliche Name des Tags. Einschränkungen finden Sie unterTag-Struktur in Amazon RDS. Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein "`aws:`" oder "`rds:`" als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie "_", ".", "/", "=", "+", "-" enthalten (Java-Regex: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)"). Schlüssel müssen in einem Tag-Satz eindeutig sein. Sie können beispielsweise kein Schlüsselpaar in einem Tag-Set haben, bei dem der Schlüssel zwar derselbe, aber unterschiedliche Werte hat, wie z. project/Trinity and project/Xanadu
Wert	Ein Wert ist der optionale Wert des Tags. Einschränkungen finden Sie unter. Tag-Struktur in Amazon RDS Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein "`aws:`" oder "`rds:`" als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie "_", ".", "/", "=", "+", "-" enthalten (Java-Regex: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)"). Die Werte innerhalb eines Tag-Satzes müssen nicht eindeutig und können null sein. Sie können beispielsweise ein Schlüssel-Wert-Paar in einem Tag-Set von haben. project/Trinity and cost-center/Trinity

Tutorial: Geben Sie mithilfe von Tags an, welche DB-Instances gestoppt werden sollen

In diesem Tutorial wird davon ausgegangen, dass Sie mehrere DB-Instances in einer Entwicklungs- oder Testumgebung haben. Sie müssen diese DB-Instances mehrere Tage lang aufbewahren. Einige DB-Instances führen Tests über Nacht durch, während andere über Nacht gestoppt und am nächsten Tag wieder gestartet werden können.

Das folgende Tutorial zeigt, wie Sie DB-Instances, die für einen Stopp über Nacht geeignet sind, ein Tag zuweisen. Das Tutorial zeigt, wie ein Skript erkennen kann, welche DB-Instances das Tag haben, und dann die markierten DB-Instances stoppen kann. In diesem Beispiel spielt der Wertanteil des Schlüssel-Wert-Paares keine Rolle. Das Vorhandensein des `stoppable`-Tags bedeutet, dass die DB-Instance diese benutzerdefinierte Eigenschaft besitzt.

Im folgenden Tutorial werden die Befehle und APIs das Tagging verwendetARNs, sodass Sie RDS nahtlos zwischen AWS Regionen, AWS Konten und verschiedenen Ressourcentypen arbeiten können, die möglicherweise identische Kurznamen haben. Sie können ARN anstelle der DB-Instance-ID in CLI Befehlen, die auf DB-Instances ausgeführt werden, angeben.

Angabe, welche DB-Instances angehalten werden sollen

1. Ermitteln Sie die Nummer ARN einer DB-Instance, die Sie als stopable kennzeichnen möchten.

   Ersetzen Sie im folgenden Beispiel den Namen Ihrer eigenen DB-Instances durch *`dev-test-db-instance`*. Ersetzen Sie in nachfolgenden Befehlen, die ARN ARN Parameter verwenden, durch Ihre eigene DB-Instance. ARNDazu gehören Ihre eigene AWS Konto-ID und der Name der AWS Region, in der sich Ihre DB-Instance befindet.

   ```
   $ aws rds describe-db-instances --db-instance-identifier dev-test-db-instance \
     --query "*[].{DBInstance:DBInstanceArn}" --output text
   arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
   ```

2. Fügen Sie das Tag `stoppable` zu dieser DB-Instance hinzu.

   Der Name für dieses Tag wird von Ihnen ausgewählt. Da das Tag in diesem Beispiel als Attribut behandelt wird, das entweder vorhanden ist oder nicht, wird der `Value=`-Teil des `--tags`-Parameters weggelassen. Dieser Ansatz bedeutet, dass Sie vermeiden können, eine Namenskonvention zu entwickeln, die alle relevanten Informationen in Namen codiert. In einer

solchen Konvention können Sie Informationen im DB-Instance-Namen oder Namen anderer Ressourcen codieren.

```
$ aws rds add-tags-to-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance \
  --tags Key=stoppable
```

3. Bestätigen Sie, dass das Tag in der DB-Instance vorhanden ist.

   Mit den folgenden Befehlen werden die Tag-Informationen für die DB-Instance im JSON Format und in einfachem tabulatorgetrenntem Text abgerufen.

```
$ aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
{
    "TagList": [
        {
            "Key": "stoppable",
            "Value": ""

        }
    ]
}
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance --
output text
TAGLIST stoppable
```

4. Beenden Sie alle DB-Instances, die als gekennzeichnet sind. `stoppable`

   Im folgenden Beispiel wird eine Textdatei erstellt, die alle Ihre DB-Instances auflistet. Der Shell-Befehl durchläuft die Liste und prüft, ob jede DB-Instance mit dem entsprechenden Attribut gekennzeichnet ist, und führt den Befehl `aws rds stop-db-instance` für jede DB-Instance aus.

```
$ aws rds describe-db-instances --query "*[].[DBInstanceArn]" --output text >/tmp/
db_instance_arns.lst
$ for arn in $(cat /tmp/db_instance_arns.lst)
do
  match="$(aws rds list-tags-for-resource --resource-name $arn --output text | grep
 stoppable)"
  if [[ ! -z "$match" ]]
```

```
  then
      echo "DB instance $arn is tagged as stoppable. Stopping it now."
# Note that you need to get the DB instance identifier from the ARN.
      dbid=$(echo $arn | sed -e 's/.*://')
      aws rds stop-db-instance --db-instance-identifier $dbid
  fi
done

DB instance arn:arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance is
 tagged as stoppable. Stopping it now.
{
    "DBInstance": {
        "DBInstanceIdentifier": "dev-test-db-instance",
        "DBInstanceClass": "db.t3.medium",
        ...
```

Sie können am Ende eines jeden Tages ein Skript wie das vorherige ausführen, um sicherzustellen, dass nicht benötigte DB-Instances gestoppt werden. Sie können einen Job auch mit einem Dienstprogramm wie `cron` planen, um jede Nacht eine solche Überprüfung durchzuführen. Sie könnten so beispielsweise vorgehen, wenn einige DB-Instances versehentlich weiter ausgeführt wurden. Hier könnten Sie eine Feinabstimmung des Befehls vornehmen, mit dem die Liste der zu prüfenden DB-Instances vorbereitet wird.

Der folgende Befehl erzeugt eine Liste Ihrer DB-Instances mit dem Status `available`. Das Skript kann DB-Instances ignorieren, die bereits angehalten wurden, da sie unterschiedliche Statuswerte wie `stopped` oder `stopping` haben.

```
$ aws rds describe-db-instances \
  --query '*[].{DBInstanceArn:DBInstanceArn,DBInstanceStatus:DBInstanceStatus}|[?
DBInstanceStatus == `available`]|[].{DBInstanceArn:DBInstanceArn}' \
  --output text
arn:aws:rds:us-east-1:123456789102:db:db-instance-2447
arn:aws:rds:us-east-1:123456789102:db:db-instance-3395
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
arn:aws:rds:us-east-1:123456789102:db:pg2-db-instance
```

Tip

Sie können die Tags zuweisen und DB-Instances mit Hilfe dieser Tags finden, um die Kosten auf andere Weise zu senken. Nehmen wir zum Beispiel dieses Szenario mit DB-Instances,

die für Entwicklungs- und Testzwecke verwendet werden. In diesem Fall können Sie einige DB-Instances festlegen, die am Ende eines jeden Tages gelöscht werden sollen. Oder Sie können sie so festlegen, dass ihre DB-Instances in Zeiten erwarteter geringer Auslastung in kleine DB-Instance-Klassen geändert werden.

Amazon-Ressourcennamen (ARNs) in Amazon RDS

In Amazon Web Services erstellte Ressourcen werden jeweils eindeutig mit einem Amazon-Ressourcennamen (ARN) identifiziert. Für bestimmte RDS Amazon-Operationen müssen Sie eine RDS Amazon-Ressource eindeutig identifizieren, indem Sie sie angebenARN. Wenn Sie beispielsweise eine Read Replica für eine RDS DB-Instance erstellen, müssen Sie die Read Replica ARN für die Quell-DB-Instance angeben.

Informationen zum Erstellen ARN und Abrufen einer vorhandenen ARN finden Sie in den folgenden Themen.

Themen

- Erstellen eines ARN für Amazon RDS
- Ein vorhandenes ARN für Amazon erwerben RDS

Erstellen eines ARN für Amazon RDS

In Amazon Web Services erstellte Ressourcen werden anhand eines Amazon-Ressourcennamens (ARN) eindeutig identifiziert. Sie können mithilfe der folgenden Syntax einen ARN für eine Amazon RDS-Ressource erstellen.

`arn:aws:rds:<region>:<account number>:<resourcetype>:<name>`

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
USA Ost (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS	
		rds-fips.us-east-2.api.aws	HTTPS	
		rds.us-east-2.api.aws	HTTPS	
		rds-fips.us-east-2.amazonaws.com	HTTPS	
USA Ost (Nord-Virginia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS	
		rds-fips.us-east-1.api.aws	HTTPS	

Name der Region	Region	Endpunkt	Protocol (Protokoll)
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
USA West (Nordkalifornien)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
USA West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
Afrika (Kapstadt)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
Asien-Pazifik (Hongkong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS
Asien-Pazifik (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com	HTTPS
		rds.ap-south-2.api.aws	HTTPS
Asien-Pazifik (Jakarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com	HTTPS
		rds.ap-southeast-3.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Asien-Pazifik (Malaysia)	ap-southeast-5	rds.ap-southeast-5.amazonaws.com	HTTPS
Asien-Pazifik (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS
Asien-Pazifik (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS
Asien-Pazifik (Osaka)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS
Asien-Pazifik (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS
Asien-Pazifik (Singapur)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com rds.ap-southeast-1.api.aws	HTTPS HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com rds.ap-southeast-2.api.aws	HTTPS HTTPS
Asien-Pazifik (Taipeh)	ap-east-2	rds.ap-east-2.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Asien-Pazifik (Thailand)	ap-southeast-7	rds.ap-southeast-7.amazonaws.com	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
		rds.ap-northeast-1.api.aws	HTTPS
Kanada (Zentral)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
		rds.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.api.aws	HTTPS
		rds-fips.ca-central-1.amazonaws.com	HTTPS
Kanada West (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS
		rds-fips.ca-west-1.amazonaws.com	HTTPS
Europa (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
		rds.eu-central-1.api.aws	HTTPS
Europa (Irland)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
		rds.eu-west-1.api.aws	HTTPS
Europa (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
		rds.eu-west-2.api.aws	HTTPS
Europa (Mailand)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS
		rds.eu-south-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Europa (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
		rds.eu-west-3.api.aws	HTTPS
Europa (Spanien)	eu-south-2	rds.eu-south-2.amazonaws.com	HTTPS
		rds.eu-south-2.api.aws	HTTPS
Europa (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
		rds.eu-north-1.api.aws	HTTPS
Europa (Zürich)	eu-central-2	rds.eu-central-2.amazonaws.com	HTTPS
		rds.eu-central-2.api.aws	HTTPS
Israel (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com	HTTPS
		rds.il-central-1.api.aws	HTTPS
Mexiko (Zentral)	mx-central-1	rds.mx-central-1.amazonaws.com	HTTPS
Naher Osten (Bahrain)	me-south-1	rds.me-south-1.amazonaws.com	HTTPS
		rds.me-south-1.api.aws	HTTPS
Naher Osten (VAE)	me-central-1	rds.me-central-1.amazonaws.com	HTTPS
		rds.me-central-1.api.aws	HTTPS
Südamerika (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS
		rds.sa-east-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
AWS GovCloud (US-Ost)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (US-West)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS

Die folgende Tabelle zeigt das Format, das Sie beim Erstellen eines ARN für einen bestimmten Amazon RDS-Ressourcentyp verwenden sollten.

Ressourcentyp	ARN-Format
DB-Instance	arn:aws:rds: ::db: *<region>* *<account>* *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :db:my-mysql-instance-1`
DB-Cluster	arn:aws:rds: *<region>* ::cluster: *<account>* *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :cluster: my-aurora-cluster-1`
Ereignisabonnement	arn:aws:rds: *<region>* ::es: *<account>* *<name>* Zum Beispiel:

Ressourcentyp	ARN-Format
	`arn:aws:rds:` *us-east-2* `:`*123456789012* `:es:`*my-subscription*
DB-Optionsgruppe	arn:aws:rds: *<region>* ::og: *<account>* *<name>* Zum Beispiel: `arn:aws:rds:` *us-east-2* `:`*123456789012* `:og:`*my-og*
DB-Parametergruppe	arn:aws:rds: *<region>* ::pg: *<account>* *<name>* Zum Beispiel: `arn:aws:rds:` *us-east-2* `:`*123456789012* `:pg:`*my-param-enable-logs*
DB-Cluster-Parametergruppe	arn:aws:rds: *<region>* :cluster-pg: *<account>* *<name>* Zum Beispiel: `arn:aws:rds:` *us-east-2* `:`*123456789012* `:cluster-pg:` *my-cluster-param-timezone*
Reservierte DB-Instance	arn:aws:rds: *<region>* ::ri: *<account>* *<name>* Zum Beispiel: `arn:aws:rds:` *us-east-2* `:`*123456789012* `:ri:`*my-reserved-postgresql*
DB-Sicherheitsgruppe	arn:aws:rds: *<region>* ::secgrp: *<account>* *<name>* Zum Beispiel: `arn:aws:rds:` *us-east-2* `:`*123456789012* `:secgrp:`*my-public*

Ressourcentyp	ARN-Format
Automatisierter DB-Snapshot	arn:aws:rds: *<region>* :snapshot:rds: *<account>* *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :snapshot:rds: my-mysql-db-2019-07-22-07-23`
Automatisierter DB-Cluster-Snapshot	arn:aws:rds: *<region>* *<account>* :cluster-snapshot:rds: *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot:rds: my-aurora-cluster-2019-07-22-16-16`
Manueller DB-Snapshot	arn:aws:rds: *<region>* ::snapshot: *<account>* *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :snapshot: my-mysql-db-snap`
Manueller DB-Cluster-Snapshot	arn:aws:rds: *<region>* :cluster-snapshot: *<account>* *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot: my-aurora-cluster-snap`
DB-Subnetzgruppe	arn:aws:rds: *<region>* ::subgrp: *<account>* *<name>* Zum Beispiel: `arn:aws:rds: us-east-2 :123456789012 :subgrp:my-subnet-10`

Ein vorhandenes ARN für Amazon erwerben RDS

Sie können den Wert ARN einer RDS Ressource ermitteln, indem Sie AWS Management Console, AWS Command Line Interface (AWS CLI) oder verwenden RDSAPI.

Konsole

Um eine ARN von der zu erhalten AWS Management Console, navigieren Sie zu der Ressource, für die Sie eine ARN suchen, und sehen Sie sich die Details für diese Ressource an.

Sie können beispielsweise den Wert ARN für eine DB-Instance auf der Registerkarte Konfiguration der DB-Instance-Details abrufen.

AWS CLI

Um eine ARN von einer bestimmten RDS Ressource zu erhalten, verwenden Sie den `describe` Befehl für diese Ressource. AWS CLI Die folgende Tabelle zeigt jeden AWS CLI Befehl und die ARN Eigenschaft, die mit dem Befehl verwendet wird, um einen zu erhaltenARN.

AWS CLI Befehl	ARNEigentum
describe-event-subscriptions	EventSubscriptionArn
describe-certificates	CertificateArn
describe-db-parameter-groups	DBParameterGroupArn
describe-db-cluster-parameter-Gruppen	DBClusterParameterGroupArn
describe-db-instances	DBInstanceArn
describe-db-security-groups	DBSecurityGroupArn
describe-db-snapshots	DBSnapshotArn
describe-events	SourceArn
describe-reserved-db-instances	R eservedDBInstance Arn
describe-db-subnet-groups	DBSubnetGroupArn

AWS CLI Befehl	ARNEigentum
describe-option-groups	OptionGroupArn
describe-db-clusters	DBClusterArn
describe-db-cluster-snapshots	DBClusterSnapshotArn

Mit dem folgenden AWS CLI Befehl wird beispielsweise der ARN für eine DB-Instance abgerufen.

Example

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

Windows:

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

Die Ausgabe dieses Befehls sieht wie folgt aus:

```
[
    {
        "DBInstanceArn": "arn:aws:rds:us-west-2:account_id:db:instance_id",
        "DBInstanceIdentifier": "instance_id"
    }
]
```

RDS API

Um eine ARN für eine bestimmte RDS Ressource abzurufen, können Sie die folgenden RDS API Operationen aufrufen und die folgenden ARN Eigenschaften verwenden.

RDSAPIOperation	ARNEigentum
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn
D escribeDBParameter Gruppen	DBParameterGroupArn
D escribeDBCluster Parameter Groups	DBClusterParameterGroupArn
D escribeDBInstances	DBInstanceArn
D escribeDBSecurity Gruppen	DBSecurityGroupArn
D escribeDBSnapshots	DBSnapshotArn
DescribeEvents	SourceArn
DescribeReservedDBInstances	R eservedDBInstance Arn
D escribeDBSubnet Gruppen	DBSubnetGroupArn
DescribeOptionGroups	OptionGroupArn
D escribeDBClusters	DBClusterArn
D escribeDBCluster Schnappschüsse	DBClusterSnapshotArn

Arbeiten mit Speicher für Amazon RDS DB-Instances

Um anzugeben, wie Ihre Daten in Amazon gespeichert werden sollenRDS, wählen Sie einen Speichertyp und geben Sie eine Speichergröße an, wenn Sie eine DB-Instance erstellen oder ändern. Später können Sie die Speichermenge erhöhen oder den Speichertyp ändern, indem Sie die DB-Instance bearbeiten. Weitere Informationen darüber, welcher Speichertyp für Ihr Workload verwenden wird, finden Sie unter Amazon RDS-Speichertypen.

Themen

- Steigern der DB-Instance-Speicherkapazität
- Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung
- Upgrade des Speicherdateisystems für eine DB-Instance
- Einstellungen für bereitgestellten Speicher IOPS SSD ändern
- E/A-intensive Speichermodifikationen
- Ändern von Einstellungen für Allzweck-SSD-Speicher (gp3)
- Verwendung eines dedizierten Protokoll-Volumes (DLV)

Steigern der DB-Instance-Speicherkapazität

Wenn Sie Platz für zusätzliche Daten benötigen, können Sie den Speicher auf einer vorhandenen DB-Instance skalieren. Dazu können Sie die Amazon-RDS-Managementkonsole, die Amazon-RDS-API oder die AWS Command Line Interface (AWS CLI) verwenden. Informationen zu den Speicherlimits finden Sie unter Amazon RDS-DB-Instance-Speicher.

> **Note**
>
> Sie können die Speichermenge für eine DB-Instance nach der Speicherzuweisung nicht reduzieren. Wenn Sie den zugewiesenen Speicherplatz erhöhen, muss er um mindestens 10 Prozent erhöht werden. Wenn Sie versuchen, den Wert um weniger als 10 Prozent zu erhöhen, erhalten Sie einen Fehler.
> Die Skalierung von Speicher für RDS für SQL Server-DB-Instances wird nur für die Speichertypen Allzweck-SSD und Provisioned IOPS SSD unterstützt.

Um die Menge an freiem Speicherplatz für Ihre DB-Instance zu überwachen, damit Sie bei Bedarf reagieren können, empfehlen wir Ihnen, einen CloudWatch Amazon-Alarm zu erstellen. Weitere

Informationen zum Einstellen von CloudWatch Alarmen finden Sie unter CloudWatch Alarme verwenden.

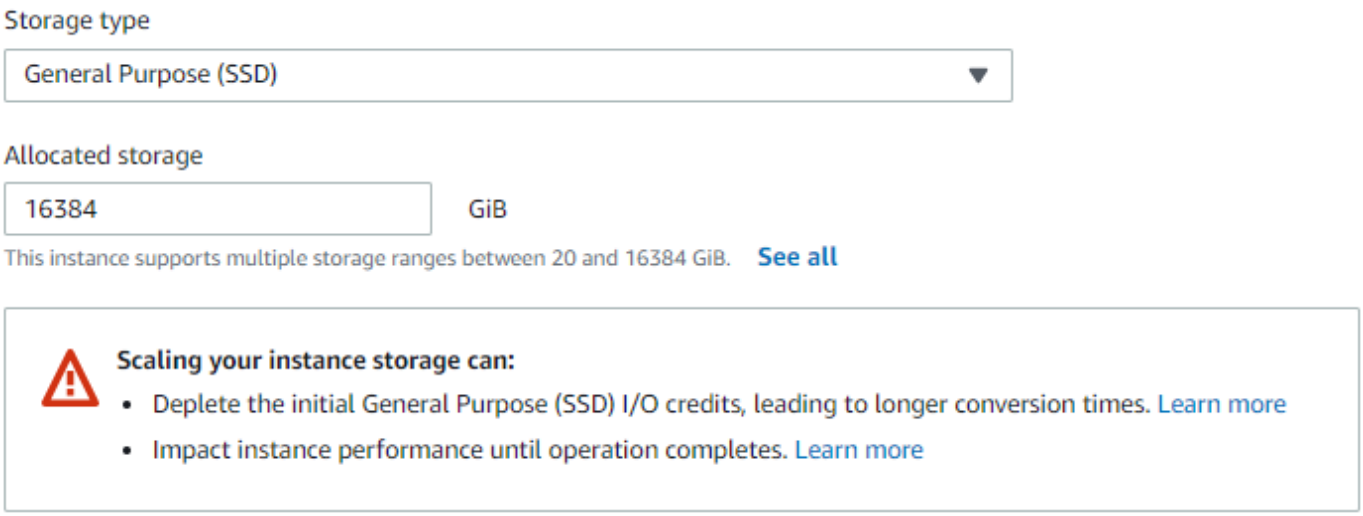
Die Skalierung des Speichers verursacht normalerweise keine Ausfälle oder Leistungseinbußen der DB-Instance. Nachdem Sie die Speichergröße für eine DB-Instance geändert haben, lautet der Status der DB-Instance Speicheroptimierung.

> **Note**
>
> Die Speicheroptimierung kann mehrere Stunden dauern. Sie können keine weiteren Speicheränderungen für sechs (6) Stunden vornehmen oder bis die Speicheroptimierung auf der Instance abgeschlossen ist, je nachdem, welcher Zeitraum länger ist. Sie können den Fortschritt der Speicheroptimierung im AWS Management Console oder mithilfe des describe-db-instances AWS CLI Befehls anzeigen.

Konsole

So erhöhen Sie den Speicher für eine DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Geben Sie einen neuen Wert für Allocated storage (Zugewiesener Speicherplatz) ein. Er muss größer als der aktuelle Wert sein.



6. Wählen Sie Weiter aus, um auf den nächsten Bildschirm zu wechseln.
7. Wählen Sie Apply immediately (Sofort anwenden) im Abschnitt Scheduling of modifications (Planung von Änderungen) aus, um die Speicheränderungen sofort auf die DB-Instance anzuwenden.

   Oder wählen Sie Apply during the next scheduled maintenance window (Anwenden während des nächsten geplanten Wartungsfensters) aus, um die Änderungen im nächsten Wartungsfenster zu übernehmen.
8. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Modify DB instance (DB-Instance ändern).

AWS CLI

Verwenden Sie den AWS CLI Befehl, um den Speicherplatz für eine DB-Instance zu erhöhen `modify-db-instance`. Legen Sie die folgenden Parameter fest:

- `--allocated-storage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `--apply-immediately` – Verwenden Sie `--apply-immediately`, um die Speicheränderungen sofort anzuwenden.

  Oder verwenden Sie `--no-apply-immediately` (Standardeinstellung), um die Änderungen während des nächsten Wartungsfensters anzuwenden. Ein sofortiger Ausfall tritt ein, wenn die Änderungen übernommen werden.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

RDS-API

Um den Speicherplatz für eine DB-Instance zu erhöhen, verwenden Sie die Amazon RDS API-Operation `ModifyDBInstance`. Legen Sie die folgenden Parameter fest:

- `AllocatedStorage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `ApplyImmediately` – Setzen Sie diese Option auf `True`, um die Speicheränderungen sofort anzuwenden. Setzen Sie diese Option auf `False` (Standardeinstellung), um die Änderungen während des nächsten Wartungsfensters zu übernehmen. Ein sofortiger Ausfall tritt ein, wenn die Änderungen übernommen werden.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung

Wenn Ihr Workload unvorhersehbar ist, können Sie die automatische Speicherskalierung für eine Amazon RDS-DB-Instance aktivieren. Dazu können Sie die Amazon-RDS-Konsole, die Amazon-RDS-API oder die AWS CLI verwenden.

Sie können diese Funktion beispielsweise für eine neue Mobile-Gaming-Anwendung verwenden, die von Benutzern schnell angenommen wird. In diesem Fall kann eine schnell steigende Systemlast den verfügbaren Datenbankspeicher übersteigen. Um zu vermeiden, dass Sie den Datenbankspeicher manuell vergrößern müssen, können Sie die automatische Skalierung des Amazon RDS-Speichers verwenden.

Wenn die automatische Speicherskalierung aktiviert ist, wird Ihr Speicher automatisch vergrößert, wenn Amazon RDS feststellt, dass Ihnen der freie Speicherplatz ausgeht. Wenn die Speicheranpassung für eine autoskalierbare DB-Instance erfolgt, startet Amazon RDS, wenn die folgenden Faktoren zutreffen:

- Der verfügbare freie Speicherplatz ist gleich oder kleiner als 10 Prozent des zugewiesenen Speichers.
- Der Zustand mit weniger Speicherplatz dauert mindestens fünf Minuten.
- Mindestens sechs Stunden sind seit der letzten Speicheränderung vergangen oder die Speicheroptimierung auf der Instance abgeschlossen ist, je nachdem, welcher Zeitraum länger ist.

Der zusätzliche Speicher wird in Schritten der folgenden Einheiten angegeben, je nach dem, was größer ist:

- 10 GiB
- 10 Prozent des aktuell zugewiesenen Speichers
- Prognostiziertes Speicherwachstum, das die aktuell zugewiesene Speichergröße in den nächsten 7 Stunden übersteigt, basierend auf den `FreeStorageSpace`-Metriken der letzten Stunde. Weitere Informationen zu Metriken finden Sie unter Monitoring with Amazon CloudWatch.

Der maximale Speicherschwellenwert ist die Grenze, die Sie für das automatische Skalieren der DB-Instance festgelegt haben. Es gelten die folgenden Einschränkungen:

- Sie müssen den maximalen Speicherschwellenwert auf mindestens 10 % mehr als den aktuell zugewiesenen Speicher festlegen. Wir empfehlen, es auf mindestens 26% mehr einzustellen, um zu vermeiden, dass Sie eine Ereignisbenachrichtigung erhalten, dass sich die Speichergröße dem maximalen Speicherschwellenwert nähert.

  Wenn Sie beispielsweise eine DB-Instance mit 1.000 GiB zugewiesenem Speicher haben, legen Sie den maximalen Speicherschwellenwert auf mindestens 1.100 GiB fest. Wenn Sie dies nicht tun, erhalten Sie eine Fehlermeldung wie Ungültige maximale Speichergröße für *`engine_name`*. Es wird jedoch empfohlen, den maximalen Speicherschwellenwert auf mindestens 1 260 GiB einzustellen, um eine Ereignisbenachrichtigung zu vermeiden.
- Für eine DB-Instance, die bereitgestellte IOPS-Speicher (io1 oder io2 Block Express) verwendet, muss das Verhältnis von IOPS zum maximalen Speicherschwellenwert (in GiB) innerhalb eines bestimmten Bereichs liegen. Weitere Informationen finden Sie unter Bereitgestellter IOPS SSD-Speicher.
- Sie können den Schwellenwert für den maximalen Speicherplatz für autoscaling-aktivierte Instanzen nicht auf einen Wert setzen, der größer ist als der maximal zugewiesene Speicherplatz für die Datenbank-Engine und die DB-Instance-Klasse.

  Zum Beispiel: Die SQL Server Standard Edition auf db.m5.xlarge hat standardmäßig einen zugewiesenen Speicher für die Instance von 20 GiB (Minimum) und einen maximal zugewiesenen Speicher von 16.384 GiB. Der Standard-Maximalspeicherschwellenwert für die automatische Skalierung ist 1.000 GiB. Wenn Sie diesen Standardwert verwenden, wird die Instance nicht automatisch über 1.000 GiB skaliert. Dies gilt auch dann, wenn der zugewiesene Maximalspeicher für die Instance 16.384 GiB beträgt.

> **Note**
>
> Wir empfehlen Ihnen, den maximalen Speicherschwellenwert basierend auf Nutzungsmustern und Kundenanforderungen sorgfältig auszuwählen. Wenn es Abweichungen in den Verwendungsmustern gibt, kann der maximale Speicherschwellenwert verhindern, dass der Speicher auf einen unerwartet hohen Wert skaliert wird, wenn die automatische Skalierung einen sehr hohen Schwellenwert prognostiziert. Nachdem eine DB-Instance automatisch skaliert wurde, kann der ihr zugewiesene Speicher nicht reduziert werden.

Themen

- Einschränkungen der automatischen Speicherskalierung
- Aktivieren der automatischen Speicherskalierung für eine neue DB-Instance
- Ändern der Einstellungen zur automatischen Speicherskalierung für eine DB-Instance
- Manuelles Skalieren Ihrer DB-Instance nach unten oder unten
- Deaktivieren der automatischen Speicherskalierung für eine DB-Instance

Einschränkungen der automatischen Speicherskalierung

Für die automatische Speicherskalierung gelten folgende Einschränkungen:

- Die automatische Skalierung erfolgt nicht, wenn der maximale Speicherschwellenwert durch die Speichererhöhung überschritten werden würde.
- Beim Autoscaling prognostiziert RDS die Speichergröße für nachfolgende Autoscaling-Vorgänge. Wenn prognostiziert wird, dass ein nachfolgender Vorgang den maximalen Speicherschwellenwert überschreiten wird, skaliert RDS automatisch auf den maximalen Speicherschwellenwert.
- Die automatische Skalierung kann Speicher-Situationen bei großen Datenlasten nicht vollständig verhindern. Dies liegt daran, dass weitere Speicheränderungen erst nach sechs (6) Stunden oder nach Abschluss der Speicheroptimierung auf der Instanz vorgenommen werden können, je nachdem, welcher Zeitraum länger ist.

  Wenn Sie eine große Datenmenge laden und die automatische Skalierung nicht genügend Speicherplatz zur Verfügung stellt, bleibt die Datenbank möglicherweise mehrere Stunden lang im Speicher-Voll-Status. Dies kann die Datenbank schädigen.

- Wenn Sie einen Speicherskalierungsvorgang gleichzeitig mit Amazon RDS und einem Autoskalierungsvorgang starten, hat Ihre Speicheranpassung Vorrang. Der Autoskalierungsvorgang wird abgebrochen.
- Autoscaling kann den zugewiesenen Speicherplatz nicht verringern. Sie können die Speichermenge für eine DB-Instance nach der Speicherzuweisung nicht reduzieren.
- Die automatische Skalierung kann nicht mit magnetischem Speicher verwendet werden.
- Die automatische Skalierung kann nicht mit den folgenden Instance-Klassen der vorherigen Generation verwendet werden, die weniger als 6 TiB bestellbaren Speicher haben: db.m3.large, db.m3.xlarge und db.m3.2xlarge.
- Autoscaling-Operationen werden nicht protokolliert. AWS CloudTrail Weitere Informationen zu finden Sie CloudTrail unter überwachen Amazon RDS API ruft an AWS CloudTrail.

Obwohl die automatische Skalierung Ihnen hilft, den Speicherplatz Ihrer Amazon RDS DB-Instance dynamisch zu vergrößern, sollten Sie den anfänglichen Speicherplatz für Ihre DB-Instance dennoch auf eine für Ihren Workload typische angemessene Größe konfigurieren.

Aktivieren der automatischen Speicherskalierung für eine neue DB-Instance

Wenn Sie eine neue Amazon RDS DB-Instance erstellen, können Sie auswählen, ob die automatische Skalierung des Speichers aktiviert werden soll. Sie können außerdem eine Obergrenze für den Speicher festlegen, den Amazon RDS für die DB-Instance zuweisen kann.

Die Aktivierung von Storage Autoscaling erfordert keinen Neustart der Datenbank und verursacht keine Ausfallzeiten. Die Funktion wird sofort wirksam, ohne den Datenbankbetrieb zu unterbrechen.

> **Note**
>
> Wenn Sie eine Amazon RDS DB-Instance klonen, für die die automatische Speicherskalierung aktiviert ist, wird diese Einstellung nicht automatisch an die geklonte Instance vererbt. Die neue DB-Instance hat die gleiche Menge an zugewiesenem Speicherplatz wie die ursprüngliche Instance. Sie können die automatische Speicherskalierung für die neue Instance wieder aktivieren, wenn die geklonte Instance ihren Speicherbedarf weiter erhöht.

Konsole

So aktivieren Sie die automatische Speicherskalierung für eine neue DB-Instance:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die AWS Region aus, in der Sie die DB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus. Wählen Sie auf der Seite Select engine (Engine auswählen) Ihre Datenbank-Engine und geben Sie Ihre DB-Instance-Informationen wie unter Erste Schritte mit Amazon RDS beschrieben an.
5. Legen Sie im Abschnitt Storage autoscaling (Automatische Speicherskalierung) den Wert Maximum storage threshold (Maximaler Speicherschwellenwert) für die DB-Instance fest.
6. Geben Sie den Rest Ihrer DB-Instance-Informationen wie unter beschrieben a Erste Schritte mit Amazon RDS.

AWS CLI

Verwenden Sie den Befehl, um die automatische Speicherskalierung für eine neue DB-Instance zu aktivieren. AWS CLI `create-db-instance` Legen Sie die folgenden Parameter fest:

- `--max-allocated-storage` Schaltet die automatische Skalierung des -Speichers ein und legt die Obergrenze der Speichergröße in Gibibytes fest.

Verwenden Sie den AWS CLI `describe-valid-db-instance-modifications`Befehl, um zu überprüfen, ob Amazon RDS-Speicher-Autoscaling für Ihre DB-Instance verfügbar ist. Um vor dem Erstellen einer Instance anhand der Instance-Klasse zu prüfen, verwenden Sie den Befehl `describe-orderable-db-instance-options`. Überprüfen Sie das folgende Feld im Rückgabewert:

- `SupportsStorageAutoscaling` Gibt an, ob die DB-Instance oder die Instance-Klasse die automatische Skalierung des Speichers unterstützt.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

RDS-API

Um die automatische Speicherskalierung für eine neue DB-Instance zu aktivieren, verwenden Sie die Amazon RDS-API-Operation `CreateDBInstance`. Legen Sie die folgenden Parameter fest:

- `MaxAllocatedStorage` Schaltet die automatische Skalierung des Amazon-RDS-Speichers ein und legt die Obergrenze der Speichergröße in Gibibytes fest.

Um zu überprüfen, ob die automatische Skalierung des Amazon RDS-Speichers für Ihre DB-Instance verfügbar ist, verwenden Sie die Amazon RDS-API-Operation `DescribeValidDbInstanceModifications` für eine vorhandene Instance oder die Operation `DescribeOrderableDBInstanceOptions`, bevor Sie eine Instance erstellen. Überprüfen Sie das folgende Feld im Rückgabewert:

- `SupportsStorageAutoscaling` Gibt an, ob die DB-Instance die automatische Skalierung des Speichers unterstützt.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

Ändern der Einstellungen zur automatischen Speicherskalierung für eine DB-Instance

Sie können die automatische Speicherskalierung für eine vorhandene Amazon RDS DB-Instance aktivieren. Sie können die Obergrenze für den Speicherplatz, den Amazon RDS für die DB-Instance zuweisen kann.

Das Ändern der Einstellungen für die automatische Speicherskalierung erfordert keinen Neustart der Datenbank und verursacht keine Ausfallzeiten. Die Änderungen werden sofort wirksam, ohne den Datenbankbetrieb zu unterbrechen.

Konsole

So ändern Sie die Einstellungen für die automatische Speicherskalierung einer DB-Instance:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die zu ändernde DB-Instance und anschließend Modify (Ändern) aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.

4. Ändern Sie die Speichergrenze im Abschnitt Autoscaling (Autoskalierung). Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
5. Wenn alle Änderungen Ihren Wünschen entsprechen, wählen Sie Continue (Weiter) aus und überprüfen Sie Ihre Änderungen.
6. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance, um Ihre Änderungen zu speichern. Wenn sie nicht korrekt sind, wählen Sie Back (Zurück) aus, um Ihre Änderungen zu bearbeiten, oder Cancel (Abbrechen), um Ihre Änderungen zu verwerfen.

   Die Änderung des Grenzwert für die automatische Speicherskalierung erfolgt sofort. Diese Einstellung beachtet nicht die Einstellung Apply immediately.

AWS CLI

Verwenden Sie den AWS CLI Befehl `modify-db-instance`, um die Einstellungen für die automatische Speicherskalierung für eine DB-Instance zu ändern. Legen Sie die folgenden Parameter fest:

- `--max-allocated-storage` Legt die Obergrenze der Speichergröße in Gibibytes fest. Wenn der Wert größer als der `--allocated-storage`-Parameter ist, wird die automatische Speicherskalierung aktiviert. Wenn der Wert mit dem `--allocated-storage`-Parameter übereinstimmt, wird die automatische Skalierung des Speichers deaktiviert.

Verwenden Sie den AWS CLI `describe-valid-db-instance-modifications`Befehl, um zu überprüfen, ob Amazon RDS-Speicher-Autoscaling für Ihre DB-Instance verfügbar ist. Um vor dem Erstellen einer Instance anhand der Instance-Klasse zu prüfen, verwenden Sie den Befehl `describe-orderable-db-instance-options`. Überprüfen Sie das folgende Feld im Rückgabewert:

- `SupportsStorageAutoscaling` Gibt an, ob die DB-Instance die automatische Skalierung des Speichers unterstützt.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

RDS-API

Um die Einstellungen für die automatische Speicherskalierung für eine DB-Instance zu ändern, verwenden Sie die Amazon RDS-API-Operation `ModifyDBInstance`. Legen Sie die folgenden Parameter fest:

- `MaxAllocatedStorage` Legt die Obergrenze der Speichergröße in Gibibytes fest.

Um zu überprüfen, ob die automatische Skalierung des Amazon RDS-Speichers für Ihre DB-Instance verfügbar ist, verwenden Sie die Amazon RDS-API-Operation `DescribeValidDbInstanceModifications` für eine vorhandene Instance oder die Operation `DescribeOrderableDBInstanceOptions`, bevor Sie eine Instance erstellen. Überprüfen Sie das folgende Feld im Rückgabewert:

- `SupportsStorageAutoscaling` Gibt an, ob die DB-Instance die automatische Skalierung des Speichers unterstützt.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

Manuelles Skalieren Ihrer DB-Instance nach unten oder unten

Amazon RDS bietet automatische Speicherskalierung, um der wachsenden Nachfrage gerecht zu werden. Es gibt jedoch Einschränkungen bei der Skalierung nach unten und in:

- RDS-Speicher — RDS unterstützt zwar die automatische Skalierung des Speichers bei steigendem Bedarf, aber nicht automatisch nach unten.
- Read Replicas — RDS unterstützt kein automatisches Aus- (Hinzufügen) oder Skalieren (Löschen) von Read Replicas. Sie müssen Read Replicas je nach Ihren Ladeanforderungen manuell hinzufügen oder entfernen.

Um Ihre RDS-Ressourcen zu verkleinern, führen Sie die folgenden manuellen Aktionen aus:

- Beim Speichern können Sie den zugewiesenen Speicherplatz einer DB-Instance mit dem `modify-db-instance` Befehl nicht manuell reduzieren. Wählen Sie stattdessen eine der folgenden Techniken:
  - Verwenden Sie eine blue/green Bereitstellung, wenn Ihre DB-Engine sie unterstützt. Erstellen Sie eine grüne Datenbank mit einer geringeren Speichergröße und machen Sie dann Ihre grüne

Datenbank zu Ihrer blauen Datenbank. Weitere Informationen finden Sie unter Ändern Sie die Speicher- und Leistungseinstellungen.

- Erstellen Sie eine neue DB-Instance mit weniger zugewiesenem Speicherplatz, migrieren Sie die Daten manuell von Ihrer aktuellen Datenbank zur neu erstellten Datenbank-Instance und wechseln Sie Ihre Datenbank-Endpunkte.

- Löschen Sie bei Read Replicas alle ungenutzten Replikate manuell über die RDS-Konsole oder. AWS CLI

Deaktivieren der automatischen Speicherskalierung für eine DB-Instance

Wenn Sie Amazon RDS nicht mehr benötigen, um den Speicherplatz für eine Amazon RDS DB-Instance automatisch zu vergrößern, können Sie die automatische Speicherskalierung deaktivieren. Auch nach diesem Schritt können Sie den Speicherplatz für Ihre DB-Instance manuell erhöhen.

Konsole

So deaktivieren Sie die automatische Speicherskalierung für eine DB-Instance:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die zu ändernde DB-Instance und anschließend Modify (Ändern) aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Legen Sie im Abschnitt Storage autoscaling (Automatische Speicherskalierung) den Wert Maximum storage threshold (Maximaler Speicherschwellenwert) für die DB-Instance fest. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
5. Wenn alle Änderungen wie gewünscht sind, wählen Sie Continue (Weiter) aus und überprüfen Sie die Änderungen.
6. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance, um Ihre Änderungen zu speichern. Wenn sie nicht korrekt sind, wählen Sie Back (Zurück) aus, um Ihre Änderungen zu bearbeiten, oder Cancel (Abbrechen), um Ihre Änderungen zu verwerfen.

Die Änderung des Grenzwert für die automatische Speicherskalierung erfolgt sofort. Diese Einstellung beachtet nicht die Einstellung Apply immediately.

AWS CLI

Verwenden Sie den AWS CLI Befehl `modify-db-instance`und den folgenden Parameter, um die automatische Speicherskalierung für eine DB-Instance zu deaktivieren:

- `--max-allocated-storage` – Geben Sie einen Wert an, der gleich der Einstellung `--allocated-storage` ist, um eine weitere automatische Skalierung des Amazon RDS-Speichers für die angegebene DB-Instance zu verhindern.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

RDS-API

Um die automatische Speicherskalierung für eine DB-Instance zu deaktivieren, verwenden Sie die Amazon RDS-API-Operation `ModifyDBInstance`. Legen Sie die folgenden Parameter fest:

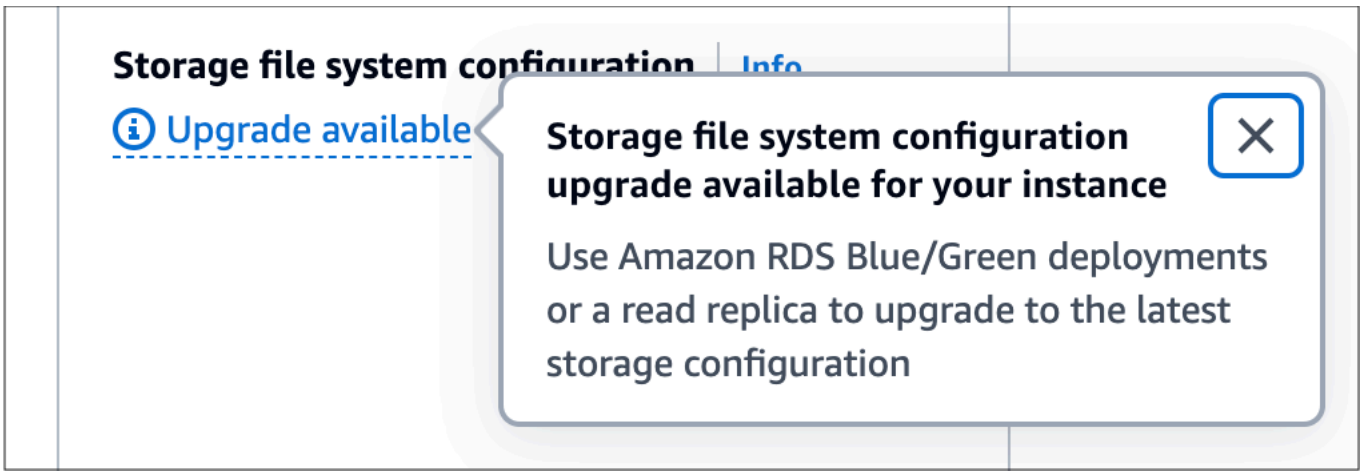
- `MaxAllocatedStorage` – Geben Sie einen Wert an, der gleich der Einstellung `AllocatedStorage` ist, um eine weitere automatische Skalierung des Amazon RDS-Speichers für die angegebene DB-Instance zu verhindern.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

Upgrade des Speicherdateisystems für eine DB-Instance

Die meisten RDS-DB-Instances bieten eine maximale Speichergröße von 64 TiB für RDS für MariaDB-, MySQL- und PostgreSQL-Datenbanken. Einige ältere 32-Bit-Dateisysteme haben möglicherweise geringere Speicherkapazitäten. Verwenden Sie den Befehl -modifications, um die Speicherkapazität Ihrer DB-Instance zu ermitteln. describe-valid-db-instance AWS CLI

RDS prüft, ob Ihr Speichersystem eine Speichergröße von 16 TiB, eine Dateigrößenbeschränkung von 2 TiB oder nicht optimierte Schreibvorgänge hat. Wenn Ihre DB-Instances diese Bedingungen erfüllen, weist RDS Sie darauf hin, dass Ihre Dateisystemkonfiguration für ein Upgrade in Frage kommt. Sie können die Upgrade-Eignung einer DB-Instance im Bereich Speicher auf der Seite mit den DB-Instance-Details überprüfen.



Wenn Ihre DB-Instance für ein Dateisystem-Upgrade in Frage kommt, verwenden Sie eine der folgenden Methoden:

- Erstellen Sie eine Blau/Grün-Bereitstellung und geben Sie Upgrade der Speicherdatei-Systemkonfiguration an.

  Mit dieser Option stufen Sie das Dateisystem in der grünen Umgebung auf die bevorzugte Konfiguration hoch. Sie können dann die Blau/Grün-Bereitstellung umstellen, wodurch die grüne Umgebung zur neuen Produktionsumgebung hochgestuft wird. Detaillierte Anweisungen finden Sie unter the section called “Erstellen einer Blau/Grün-Bereitstellung”.
- Erstellen Sie ein Lesereplikat der DB-Instance und geben Sie Upgrade der Speicherdatei-Systemkonfiguration an.

  Mit dieser Option stufen Sie das Dateisystem des Lesereplikats auf die bevorzugte Konfiguration hoch. Sie können dann das Lesereplikat auf eine eigenständige Instance hochstufen. Detaillierte Anweisungen finden Sie unter the section called “Erstellen eines Lesereplikats”.

Während des Speicher-Upgrades ist Ihre Datenbank-Engine nicht verfügbar. Die Aktualisierung der Speicherkonfiguration ist ein E/A-intensiver Vorgang und führt zu längeren Erstellungszeiten für Lesereplikate und Blau/Grün-Bereitstellungen. Das Speicher-Upgrade ist schneller, wenn die beiden folgenden Bedingungen erfüllt sind:

- Die Quell-DB-Instance verwendet bereitgestellten IOPS-SSD-Speicher (io1 oder io2 Block Express).
- Sie haben die grüne Umgebung oder das Read Replica mit einer Instance-Größe von 4xlarge oder mehr bereitgestellt.

Speicher-Upgrades mit Allzweck-SSD-Speicher (gp2) können Ihr E/A-Guthaben verbrauchen, wodurch es zu längeren Upgrade-Zeiten kommt. Weitere Informationen finden Sie unter the section called “DB-Instance-Speicher”.

Während eines Speicher-Upgrades erhöht RDS die zugewiesene Speichergröße für die grüne Instance oder das Read Replica um 10%, wenn die beiden folgenden Bedingungen erfüllt sind:

- Der Speicherverbrauch auf Ihrer Quell-DB-Instance ist größer oder gleich 90% der zugewiesenen Speichergröße.
- Die automatische Speicherskalierung ist aktiviert.

RDS deaktiviert Autoscaling, wenn die neue Speichergröße größer oder gleich dem maximal zugewiesenen Speicher ist, der für die Instance festgelegt wurde. Wenn die automatische Speicherskalierung vor Beginn des Speicher-Upgrades deaktiviert ist, erhöht sich die Speichergröße während des Upgrades nicht.

Einstellungen für bereitgestellten Speicher IOPS SSD ändern

Sie können die Einstellungen für eine DB-Instance ändern, die bereitgestellten IOPS SSD Speicher verwendet AWS CLI, indem Sie die RDS Amazon-Konsole oder Amazon RDS API verwenden. Geben Sie den Speichertyp, den zugewiesenen Speicher und die Menge an bereitgestelltem Speicherplatz anIOPS, die Sie benötigen. Der Bereich hängt von Ihrer Datenbank-Engine und Ihrem Instance-Typ ab.

Sie können zwar die für Ihre Instanz IOPS bereitgestellte Menge reduzieren, aber nicht die Speichergröße.

Meistens sind für die Speicherskalierung keine Ausfallzeiten erforderlich. Auch wird die Leistung des Servers nicht beeinträchtigt. Nachdem Sie den Speicher IOPS für eine DB-Instance geändert haben, lautet der Status der DB-Instance Speicheroptimierung.

> **Note**
>
> Die Speicheroptimierung kann mehrere Stunden dauern. Sie können keine weiteren Speicheränderungen für sechs (6) Stunden vornehmen oder bis die Speicheroptimierung auf der Instance abgeschlossen ist, je nachdem, welcher Zeitraum länger ist.

Informationen zu den Bereichen zugewiesener und bereitgestellter Speicher, die für jede Datenbank-Engine IOPS verfügbar sind, finden Sie unter. Bereitgestellter IOPS SSD-Speicher

Konsole

So ändern Sie die IOPS Provisioned-Einstellungen für eine DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.

   Um die Liste der DB-Instances zu filtern, geben Sie für Datenbanken filtern eine Textzeichenfolge ein, die Amazon RDS zum Filtern der Ergebnisse verwenden soll. Nur DB-Instances in deren Namen die Zeichenfolge vorkommt werden gelistet.
3. Wählen Sie die DB-Instance mit Provisioned ausIOPS, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie auf der Seite DB-Instance ändern als Speichertyp die Option IOPS SSD Bereitgestellt (io1) oder Bereitgestellt IOPS SSD (io2) aus.
6. Geben Sie für Bereitgestellt einen Wert ein. IOPS

   Wenn der Wert, den Sie entweder für Zugewiesener Speicher oder Bereitgestellt angeben, außerhalb der vom anderen Parameter unterstützten Grenzwerte IOPS liegt, wird eine Warnmeldung angezeigt. Diese Meldung gibt den für den anderen Parameter erforderlichen Wertebereich an.
7. Klicken Sie auf Weiter.
8. Wählen Sie Apply immediately (Sofort anwenden) im Abschnitt Scheduling of modifications (Planung von Änderungen) aus, um die Änderungen sofort auf die DB-Instance anzuwenden. Oder wählen Sie Apply during the next scheduled maintenance window (Anwenden während des nächsten geplanten Wartungsfensters) aus, um die Änderungen im nächsten Wartungsfenster zu übernehmen.
9. Überprüfen Sie die Parameter, die geändert werden sollen und wählen Sie Modify DB instance (DB-Instance ändern) aus, um die Änderung abzuschließen.

   Der neue Wert für zugewiesenen Speicher oder für Bereitgestellt IOPS wird in der Spalte Status angezeigt.

AWS CLI

Verwenden Sie den Befehl, um die IOPS Einstellung „Bereitgestellt“ für eine DB-Instance zu ändern. AWS CLI `modify-db-instance` Legen Sie die folgenden Parameter fest:

- `--storage-type`— Auf `io1` oder `io2` für IOPS Bereitgestellt eingestellt.
- `--allocated-storage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `--iops`— Die neue Menge an Bereitstellungen IOPS für die DB-Instance, ausgedrückt in I/O-Vorgängen pro Sekunde.
- `--apply-immediately`Verwenden Sie `--apply-immediately`, um Änderungen sofort anzuwenden. Verwenden Sie `--no-apply-immediately` (Standard), um Änderungen im nächsten Wartungsfenster zu übernehmen.

RDS API

Verwenden Sie den RDS API Amazon-Vorgang `ModifyDBInstance`, um die bereitgestellten IOPS Einstellungen für eine DB-Instance zu ändern. Legen Sie die folgenden Parameter fest:

- `StorageType`— Auf `io1` oder `io2` für IOPS Bereitgestellt eingestellt.
- `AllocatedStorage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `Iops`— Die neue IOPS Rate für die DB-Instance, ausgedrückt in I/O-Vorgängen pro Sekunde.
- `ApplyImmediately` – Legen Sie diese Option auf `True` fest, um Änderungen sofort zu übernehmen. Legen Sie diese Option auf `False` (Standard) fest, um Änderungen im nächsten Wartungsfenster anzuwenden.

E/A-intensive Speichermodifikationen

Amazon RDS DB-Instances verwenden Amazon Elastic Block Store (EBS) -Volumes für die Datenbank- und Protokollspeicherung. Abhängig von der Menge des angeforderten Speichers RDS (außer RDS für SQL Server) wird automatisch ein Striping auf mehrere EBS Amazon-Volumes verteilt, um die Leistung zu verbessern. RDSDB-Instances mit SSD Speichertypen werden entweder von einem oder vier gestreiften EBS Amazon-Volumes in einer RAID 0-Konfiguration unterstützt. Standardmäßig haben Speicheränderungsvorgänge für eine RDS DB-Instance nur minimale Auswirkungen auf den laufenden Datenbankbetrieb.

In den meisten Fällen werden Änderungen an der Speicherskalierung vollständig auf die EBS Amazon-Ebene übertragen und sind für die Datenbank transparent. Dieser Vorgang ist in der Regel

innerhalb weniger Minuten abgeschlossen. Bei einigen älteren RDS Speichervolumes ist jedoch ein anderer Prozess erforderlich, um die Größe, die Bereitstellung oder den IOPS Speichertyp zu ändern. Dies beinhaltet das Erstellen einer vollständigen Kopie der Daten unter Verwendung eines potenziell I/O-intensiven Vorgangs.

Bei der Speichermodifikation wird ein I/O-intensiver Vorgang verwendet, wenn einer der folgenden Faktoren zutrifft:

- Der Quellspeichertyp ist magnetisch. Der magnetische Speicher unterstützt keine elastische Volumenmodifikation.
- Die RDS DB-Instance befindet sich nicht auf einem EBS Amazon-Layout mit einem oder vier Volumes. Sie können die Anzahl der EBS Amazon-Volumes, die in Ihren RDS DB-Instances verwendet werden, mithilfe von Enhanced Monitoring-Metriken anzeigen. Weitere Informationen finden Sie unter Betriebssystemmetriken in der RDS Konsole anzeigen.
- Die Zielgröße der Änderungsanforderung erhöht den zugewiesenen Speicher auf über 400 GiB RDS für MariaDB-SQL, My- und SQL Postgre-Instances und 200 GiB für RDS Oracle. Autoscaling-Vorgänge für Speicher haben den gleichen Effekt, wenn sie die zugewiesene Speichergröße Ihrer DB-Instance über diese Schwellenwerte erhöhen.

Wenn Ihre Speicheränderung einen I/O-intensiven Vorgang umfasst, verbraucht sie I/O-Ressourcen und erhöht die Last Ihrer DB-Instance. Speicheränderungen bei I/O-intensiven Vorgängen mit Allzweckspeicher SSD (GP2) können Ihr I/O-Guthaben aufbrauchen, was zu längeren Konvertierungszeiten führt.

Wir empfehlen als Best Practice, diese Speicheränderungsanfragen außerhalb der Spitzenzeiten zu planen, um den Zeitaufwand für die Durchführung des Speicheränderungsvorgangs zu reduzieren. Alternativ können Sie ein Lesereplikat der DB-Instance erstellen und die Speicheränderung am Lesereplikat durchführen. Stufen Sie dann das Lesereplikat zur primären DB-Instance hoch. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Weitere Informationen finden Sie unter Warum bleibt eine RDS Amazon-DB-Instance im Änderungsstatus hängen, wenn ich versuche, den zugewiesenen Speicherplatz zu erhöhen?

Ändern von Einstellungen für Allzweck-SSD-Speicher (gp3)

Sie können die Einstellungen für eine DB-Instance ändern, die General Purpose SSD (gp3) -Speicher verwendet, AWS CLI indem Sie die Amazon RDS-Konsole oder die Amazon RDS-API verwenden.

Geben Sie den Speichertyp, den zugewiesenen Speicher, die Menge der bereitgestellten IOPS und den Speicherdurchsatz Ihren Anforderungen entsprechend an.

Sie können zwar die Anzahl der bereitgestellten IOPS und den Speicherdurchsatz für Ihre DB-Instance reduzieren, aber Sie können die Speichergröße nicht reduzieren.

In den meisten Fällen ist für die Speicherskalierung keine Unterbrechung erforderlich. Nachdem Sie die Speicher-IOPS für eine DB-Instance geändert haben, lautet der Status der DB-Instance Speicheroptimierung. Während der Speicheroptimierung können Sie mit erhöhten Latenzen rechnen, die jedoch immer noch im einstelligen Millisekundenbereich liegen. Die DB-Instance ist nach einer Speichermodifikation voll funktionsfähig.

> **Note**
>
> Sie können erst sechs (6) Stunden nach Abschluss der Speicheroptimierung auf der Instance weitere Speicheränderungen vornehmen.

Informationen zu den Bereichen des zugewiesenen Speichers, der bereitgestellten IOPS und des Speicherdurchsatzes, die für jede Datenbank-Engine verfügbar sind, finden Sie unter GP3-Speicher (empfohlen).

Konsole

So ändern Sie die Einstellungen für die Speicherleistung einer DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.

   Geben Sie unter Filter databases (Datenbanken filtern) eine Textzeichenfolge für Amazon RDS ein, damit die Ergebnisse gefiltert werden. Nur DB-Instances in deren Namen die Zeichenfolge vorkommt werden gelistet.
3. Wählen Sie die DB-Instance mit gp3-Speicher aus, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie auf der Seite DB-Instance modify General Purpose SSD (gp3) als Speichertyp aus und gehen Sie dann wie folgt vor:

   a. Wählen Sie für Provisioned IOPS (Bereitgestellte IOPS) einen Wert aus.

Wenn sich der Wert, den Sie für Allocated storage (Zugewiesener Speicher) oder für Provisioned IOPS (Bereitgestellte IOPS) eingeben, außerhalb der vom anderen Parameter unterstützten Grenze befindet, wird eine Warnmeldung angezeigt. Diese Meldung gibt den für den anderen Parameter erforderlichen Wertebereich an.

b. Wählen Sie für Storage throughput (Speicherdurchsatz) einen Wert aus.

Wenn sich der Wert, den Sie entweder für Provisioned IOPS (Bereitgestellte IOPS) oder für Storage throughput (Speicherdurchsatz) eingeben, außerhalb der vom anderen Parameter unterstützten Grenze befindet, wird eine Warnmeldung angezeigt. Diese Meldung gibt den für den anderen Parameter erforderlichen Wertebereich an.

6. Klicken Sie auf Weiter.
7. Wählen Sie Apply immediately (Sofort anwenden) im Abschnitt Scheduling of modifications (Planung von Änderungen) aus, um die Änderungen sofort auf die DB-Instance anzuwenden. Oder wählen Sie Apply during the next scheduled maintenance window (Anwenden während des nächsten geplanten Wartungsfensters) aus, um die Änderungen im nächsten Wartungsfenster zu übernehmen.
8. Überprüfen Sie die Parameter, die geändert werden sollen und wählen Sie Modify DB instance (DB-Instance ändern) aus, um die Änderung abzuschließen.

   Der neue Wert für bereitgestellte IOPS erscheint in der Spalte Status.

AWS CLI

Verwenden Sie den AWS CLI Befehl `modify-db-instance`, um die Speicherleistungseinstellungen für eine DB-Instance zu ändern. Legen Sie die folgenden Parameter fest:

- `--storage-type` – Legen Sie `gp3` für Allzweck-SSD (gp3) fest.
- `--allocated-storage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `--iops`: die neue Menge von bereitgestellten IOPS für die DB-Instance, ausgedrückt in I/O-Operationen pro Sekunde
- `--storage-throughput`— Der neue Speicherdurchsatz für die DB-Instance, ausgedrückt in MiBps.
- `--apply-immediately`Verwenden Sie `--apply-immediately`, um Änderungen sofort anzuwenden. Verwenden Sie `--no-apply-immediately` (Standard), um Änderungen im nächsten Wartungsfenster zu übernehmen.

RDS-API

Verwenden Sie die API-Operation `ModifyDBInstance` von Amazon RDS, um die Einstellungen für die Speicherleistung für eine DB-Instance zu ändern. Legen Sie die folgenden Parameter fest:

- `StorageType` – Legen Sie `gp3` für Allzweck-SSD (gp3) fest.
- `AllocatedStorage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `Iops`: das neue IOPS-Verhältnis für die DB-Instance, ausgedrückt in I/O-Operationen pro Sekunde
- `StorageThroughput`— Der neue Speicherdurchsatz für die DB-Instance, ausgedrückt in MiBps.
- `ApplyImmediately` – Legen Sie diese Option auf `True` fest, um Änderungen sofort zu übernehmen. Legen Sie diese Option auf `False` (Standard) fest, um Änderungen im nächsten Wartungsfenster anzuwenden.

Verwendung eines dedizierten Protokoll-Volumes (DLV)

Sie können ein dediziertes Protokollvolumen (DLV) für eine DB-Instance verwenden, die Provisioned IOPS (PIOPS) -Speicher verwendet. Ein DLV verschiebt Transaktionsprotokolle und Anforderungen MySQL/MariaDB redo logs and binary logs to a storage volume that's separate from the volume containing the database tables. A DLV makes transaction write logging more efficient and consistent. DLVs are ideal for databases with large allocated storage, high I/O pro Sekunde (IOPS) für PostgreSQL-Datenbanken oder latenzempfindliche Workloads.

DLVs werden für PIOPS-Speicher (io1 und io2 Block Express) unterstützt und werden mit einer festen Größe von 1.024 GiB und 3.000 bereitgestellten IOPS erstellt.

Amazon RDS unterstützt DLVs AWS-Regionen insgesamt die folgenden Versionen:

- MariaDB 10.6.7 und höhere 10-Versionen
- MySQL 8.0.28 und höhere 8.0-Versionen, MySQL 8.4.3 und höhere 8.4-Versionen
- PostgreSQL 13.10 und höhere 13-Versionen, 14.7 und höhere 14-Versionen sowie 15.2 und höhere 15-Versionen

RDS unterstützt DLVs Multi-AZ-Bereitstellungen. Wenn Sie eine Multi-AZ-Instance ändern oder erstellen, wird ein DLV sowohl für die primäre als auch für die sekundäre Instanz erstellt.

RDS unterstützt DLVs mit Read Replicas. Wenn für die primäre DB-Instance ein DLV aktiviert ist, verfügen alle Lesereplikate, die nach der Aktivierung des DLV erstellt wurden, auch über ein DLV. Für

alle Lesereplikate, die vor dem Wechsel zu DLV erstellt wurden, wird diese Funktion nicht aktiviert, sofern sie nicht ausdrücklich entsprechend geändert wurde. Wir empfehlen, dass alle Lesereplikate, die vor der Aktivierung von DLV einer primären Instance angefügt wurden, ebenfalls manuell so geändert werden, dass sie über ein DLV verfügen.

> **Note**
>
> Wir empfehlen DLVs für Datenbankkonfigurationen von 5 TiB oder mehr.

Weitere Informationen zu den Vorteilen von DLVs finden Sie in den folgenden Blogbeiträgen:

- Verbessern Sie die Datenbankleistung mit dedizierten Protokollvolumes von Amazon RDS
- Benchmark Amazon RDS for PostgreSQL mit dedizierten Protokollvolumes
- Maximierung der Leistung von AWS RDS for MySQL mit dedizierten Log-Volumes in der Percona-Dokumentation

Informationen zu den Bereichen des zugewiesenen Speichers, der bereitgestellten IOPS und des Speicherdurchsatzes, die für jede Datenbank-Engine verfügbar sind, finden Sie unter Bereitgestellter IOPS SSD-Speicher.

Themen

- Überlegungen zur Aktivierung und Deaktivierung von DLV
- DLV aktivieren, wenn Sie eine DB-Instance erstellen
- DLV auf einer vorhandenen DB-Instance aktivieren
- Überwachung des DLV-Speichers

Überlegungen zur Aktivierung und Deaktivierung von DLV

Das Aktivieren und Deaktivieren von DLV kann zeitaufwändig sein und zu Ausfallzeiten führen. Bei diesem Vorgang werden alle Transaktionsprotokolle oder Redo- und Binärprotokolle (je nach Datenbank-Engine) bei der Aktivierung auf das neue Volume oder bei der Deaktivierung zurück in den ursprünglichen Speicher kopiert. Die Dauer dieses Vorgangs wird von mehreren Faktoren beeinflusst:

- Anzahl der Transaktionsprotokolle:

  - Größere Datenbanken mit mehr Transaktionen generieren mehr Protokolle, was den Zeitaufwand für das Kopieren erhöht.
  - Transaktionsprotokolle können sich auf der primären DB-Instance ansammeln, wenn die Replikationssteckplätze inaktiv sind oder wenn die Replikation verzögert wird, wodurch sich die für das Kopieren benötigte Zeit erhöht. Stellen Sie sicher, dass die Replikation aktuell ist, und entfernen Sie alle nicht benötigten Steckplätze.
- Speicherkonfiguration:
  - EBS-Bandbreite der DB-Instance — Eine höhere Bandbreite ermöglicht eine schnellere Datenübertragung.
  - Anzahl der bereitgestellten IOPS — Mehr Eingabe-/Ausgabevorgänge pro Sekunde (IOPS) können den Kopiervorgang beschleunigen.
- Datenbankaktivität — Ein hohes Maß an Datenbankaktivität während der Konfiguration kann den Vorgang verlangsamen.

Um Ausfallzeiten zu minimieren, empfehlen wir Ihnen, in Zeiten geringer Aktivität oder in Wartungsfenstern zu planen und zu planen.

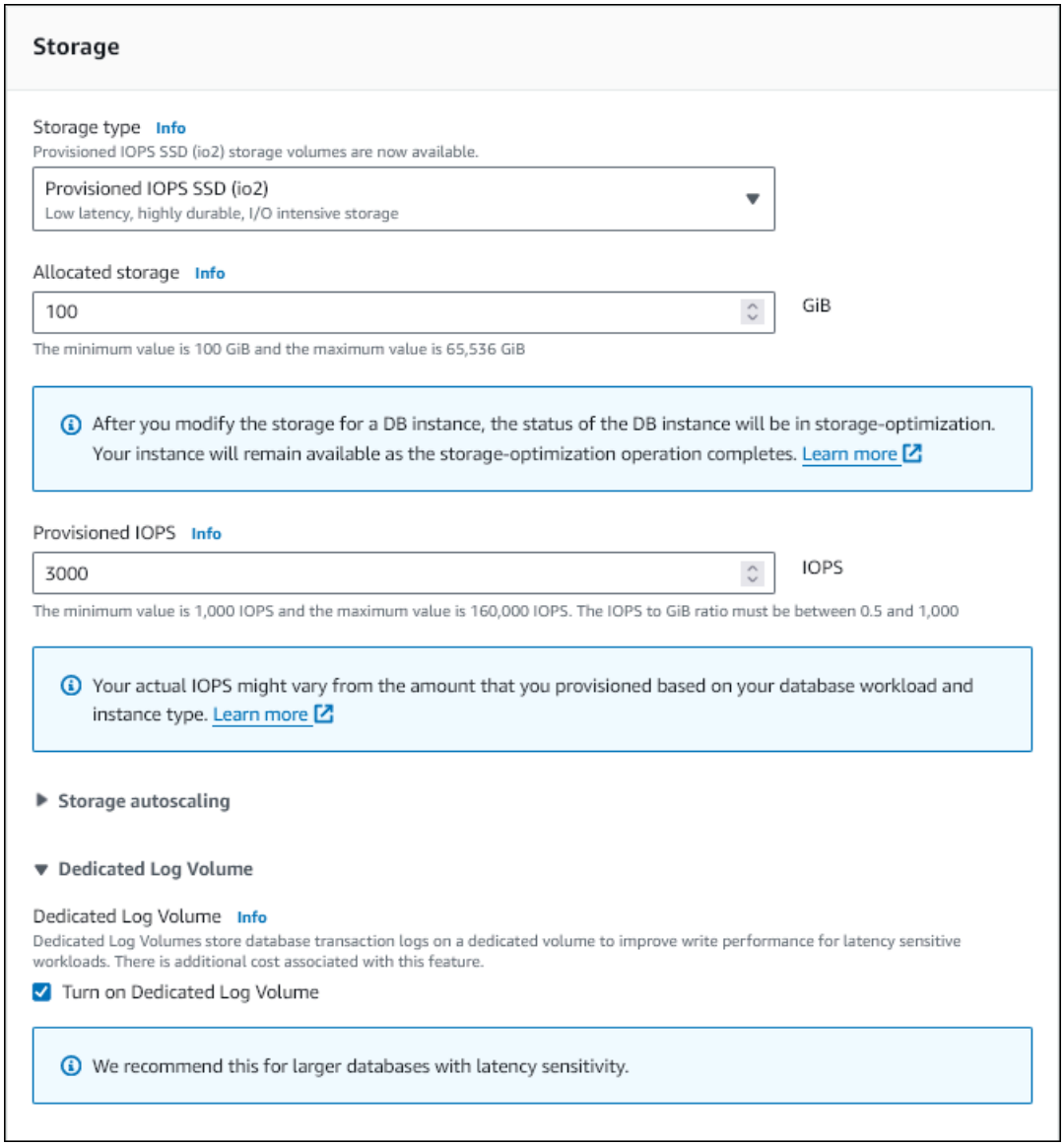
DLV aktivieren, wenn Sie eine DB-Instance erstellen

Sie können die AWS Management Console, oder RDS-API verwenden AWS CLI, um eine DB-Instance mit aktiviertem DLV zu erstellen.

Konsole

Um DLV auf einer neuen DB-Instance zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbank erstellen aus.
3. Wählen Sie auf der Seite DB-Instance erstellen eine DB-Engine aus, die DLV unterstützt.
4. Für Speicher:
   a. Wählen Sie entweder Provisioned IOPS SSD (io1) oder Provisioned IOPS SSD (io2).
   b. Geben Sie den gewünschten zugewiesenen Speicher und die bereitgestellten IOPS ein.
   c. Erweitern Sie Dedicated Log Volume und wählen Sie dann Dedicated Log Volume einschalten aus.



5. Wählen Sie nach Bedarf andere Einstellungen.
6. Wählen Sie Datenbank erstellen aus.

Nachdem die Datenbank erstellt wurde, wird der Wert für Dedicated Log Volume auf der Registerkarte Konfiguration der Datenbankdetailseite angezeigt.

CLI

Verwenden Sie den Befehl, um DLV zu aktivieren, wenn Sie eine DB-Instance mit bereitgestelltem IOPS-Speicher erstellen. AWS CLI create-db-instance Legen Sie die folgenden Parameter fest:

- `--dedicated-log-volume`— Aktiviert ein dediziertes Log-Volume.
- `--storage-type`— Auf `io1` oder `io2` für Provisioned IOPS eingestellt.
- `--allocated-storage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `--iops`— Die Menge der bereitgestellten IOPS für die DB-Instance, ausgedrückt in I/O-Vorgängen pro Sekunde.

RDS-API

Um DLV zu aktivieren, wenn Sie eine DB-Instance mit bereitgestelltem IOPS-Speicher erstellen, verwenden Sie den Amazon RDS-API-Vorgang Create. DBInstance Legen Sie die folgenden Parameter fest:

- `DedicatedLogVolume`— Auf einstellen, `true` um ein dediziertes Log-Volume zu aktivieren.
- `StorageType`— Auf `io1` oder `io2` für Provisioned IOPS eingestellt.
- `AllocatedStorage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes.
- `Iops`— Die IOPS-Rate für die DB-Instance, ausgedrückt in I/O-Vorgängen pro Sekunde.

DLV auf einer vorhandenen DB-Instance aktivieren

Sie können die AWS Management Console, oder RDS-API verwenden AWS CLI, um eine DB-Instance so zu ändern, dass DLV aktiviert wird.

Nachdem Sie die DLV-Einstellung für eine DB-Instance geändert haben, müssen Sie die DB-Instance neu starten.

Konsole

Um DLV auf einer vorhandenen DB-Instance zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.

   Geben Sie unter Filter databases (Datenbanken filtern) eine Textzeichenfolge für Amazon RDS ein, damit die Ergebnisse gefiltert werden. Nur DB-Instances in deren Namen die Zeichenfolge vorkommt werden gelistet.
3. Wählen Sie die DB-Instance mit bereitgestelltem IOPS-Speicher aus, die Sie ändern möchten.

4. Wählen Sie Ändern aus.
5. Gehen Sie auf der Seite DB-Instance modifizieren wie folgt vor:
   - Erweitern Sie unter Speicher den Eintrag Dedicated Log Volume und wählen Sie dann Dedicated Log Volume einschalten aus.
6. Klicken Sie auf Weiter.
7. Wählen Sie Sofort anwenden, um die Änderungen sofort auf die DB-Instance anzuwenden. Oder wählen Sie Apply during the next scheduled maintenance window (Anwenden während des nächsten geplanten Wartungsfensters) aus, um die Änderungen im nächsten Wartungsfenster zu übernehmen.
8. Überprüfen Sie die Parameter, die geändert werden sollen und wählen Sie Modify DB instance (DB-Instance ändern) aus, um die Änderung abzuschließen.

Der neue Wert für Dedicated Log Volume wird auf der Registerkarte Konfiguration der Datenbankdetailseite angezeigt.

CLI

Verwenden Sie den Befehl, um DLV auf einer vorhandenen DB-Instance mithilfe des bereitgestellten IOPS-Speichers zu aktivieren oder zu deaktivieren. AWS CLI `modify-db-instance` Legen Sie die folgenden Parameter fest:

- `--dedicated-log-volume`— Aktiviert ein dediziertes Log-Volume.

  Verwenden Sie `--no-dedicated-log-volume` (Standardeinstellung), um ein dediziertes Protokollvolume zu deaktivieren.
- `--apply-immediately`Verwenden Sie `--apply-immediately`, um Änderungen sofort anzuwenden.

  Verwenden Sie `--no-apply-immediately` (Standard), um Änderungen im nächsten Wartungsfenster zu übernehmen.

RDS-API

Verwenden Sie die Amazon-RDS-API-Operation `ModifyDBInstance`, um DLV auf einer vorhandenen DB-Instance mithilfe von bereitgestelltem IOPS-Speicher zu aktivieren oder zu deaktivieren. Legen Sie die folgenden Parameter fest:

- `DedicatedLogVolume`— Stellen Sie diese Option auf ein, `true` um ein dediziertes Log-Volume zu aktivieren.

  Stellen Sie diese Option auf ein, `false` um ein dediziertes Log-Volume zu deaktivieren. Dies ist der Standardwert.
- `ApplyImmediately` – Legen Sie diese Option auf `True` fest, um Änderungen sofort zu übernehmen.

  Legen Sie diese Option auf `False` (Standard) fest, um Änderungen im nächsten Wartungsfenster anzuwenden.

Überwachung des DLV-Speichers

Sie können die DLV-Speichernutzung überwachen, indem Sie die `FreeStorageSpaceLogVolume` Metrik in verwenden. CloudWatch

Sie können die folgende Abfrage für RDS for PostgreSQL verwenden, um die von Transaktionsprotokollen belegte Größe zu ermitteln:

```
SELECT pg_size_pretty(COALESCE(sum(size), 0)) AS total_wal_generated_size
FROM pg_catalog.pg_ls_waldir();
```

Wenn dem DLV der Speicherplatz ausgeht, wechselt die DB-Instance in den `storage-full` Status, was zu Ausfallzeiten führt.

Löschen einer DB-Instance

Sie können eine DB-Instance mithilfe der AWS Management Console AWS CLI, der oder der RDS-API löschen. Wenn Sie eine DB-Instance in einem Aurora-DB-Cluster löschen möchten, finden Sie weitere Informationen unter Löschen von Aurora-DB-Clustern und -DB-Instances.

Themen

- Voraussetzungen für das Löschen einer DB-Instance
- Überlegungen beim Löschen einer DB-Instance
- Löschen einer DB-Instance

Voraussetzungen für das Löschen einer DB-Instance

Stellen Sie vor dem Löschen einer DB-Instance sicher, dass der Löschschutz ausgeschaltet ist. Standardmäßig ist der Löschschutz für eine DB-Instance aktiviert, die mit der Konsole erstellt wurde.

Wenn für Ihre DB-Instance der Löschschutz aktiviert ist, können Sie ihn deaktivieren, indem Sie Ihre Instance-Einstellungen ändern. Wählen Sie auf der Seite mit den Datenbankdetails die Option Ändern oder rufen Sie den modify-db-instanceBefehl auf. Dieser Vorgang verursacht keinen Ausfall. Weitere Informationen finden Sie unter Einstellungen für DB-Instances.

Überlegungen beim Löschen einer DB-Instance

Das Löschen einer DB-Instance wirkt sich auf die Wiederherstellbarkeit der Instance, die Verfügbarkeit von Backups und den Status der Lesereplikate aus. Betrachten Sie die folgenden Punkte:

- Sie können wählen, ob ein endgültiger DB-Snapshot erstellt werden soll. Ihnen stehen folgende Optionen zur Verfügung:
  - Wenn Sie einen endgültigen Snapshot erstellen, können Sie ihn verwenden, um Ihre gelöschte DB-Instance wiederherzustellen. RDS speichert sowohl den endgültigen Snapshot als auch alle manuellen Snapshots, die Sie zuvor aufgenommen haben. Sie können keinen endgültigen DB-Snapshot Ihrer DB-Instance erstellen, wenn sich diese nicht im Status `Available` befindet. Weitere Informationen finden Sie unter Anzeigen von Amazon RDS DB-Instance-Status.
  - Wenn Sie keinen endgültigen Snapshot erstellen, ist das Löschen Ihrer Instance schneller. Der Nachteil ist, dass es keinen endgültigen Snapshot gibt, den Sie später wiederherstellen können. Wenn Sie sich entscheiden, Ihre gelöschte DB-Instance wiederherzustellen, behalten Sie

entweder automatische Backups bei oder verwenden Sie einen früheren manuellen Snapshot, um Ihre DB-Instance auf den Zeitpunkt des früheren Snapshots zurückzusetzen.

- Sie können wählen, ob automatisierte Backups aufbewahrt werden sollen. Ihnen stehen folgende Optionen zur Verfügung:
  - Wenn Sie automatisierte Backups beibehalten, speichert RDS diese für den Aufbewahrungszeitraum, der zu dem Zeitpunkt festgelegt war, als Sie die DB-Instance gelöscht haben. Sie können automatisierte Backups verwenden, um Ihre DB-Instance auf einen Zeitpunkt während des Aufbewahrungszeitraums, aber nicht danach, wiederherzustellen. Der Aufbewahrungszeitraum gilt unabhängig davon, ob Sie einen endgültigen DB-Snapshot erstellen. Informationen zum Löschen eines beibehaltenen automatisierten Backups finden Sie unter Löschen aufbewahrter automatisierter Backups.
  - Bei Aufbewahrung automatisierter Backups und manueller Snapshots fallen Gebühren an, bis sie gelöscht werden. Weitere Informationen finden Sie unter Aufbewahrungskosten.
  - Wenn Sie keine automatisierten Backups aufbewahren, löscht RDS die automatisierten Backups, die sich in derselben Datenbank AWS-Region wie Ihre DB-Instance befinden. Sie können diese Backups nicht wiederherstellen. Wenn Ihre automatisierten Backups in eine andere AWS-Region repliziert wurden, bewahrt RDS diese auch dann auf, wenn Sie sich nicht dafür entscheiden, automatisierte Backups beizubehalten. Weitere Informationen finden Sie unter Automatisierte Backups auf ein anderes replizieren AWS-Region.

    > **Note**
    >
    > Wenn Sie einen endgültigen DB-Snapshot erstellen, müssen Sie in der Regel keine automatisierten Backups beibehalten.

- Wenn Sie Ihre DB-Instance löschen, löscht RDS keine manuellen DB-Snapshots. Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
- Wenn Sie alle RDS-Ressourcen löschen möchten, beachten Sie, dass für die folgenden Ressourcen Abrechnungsgebühren anfallen:
  - DB-Instances
  - DB-Snapshots
  - DB-Cluster

  Wenn Sie Reserved Instances gekauft haben, werden diese gemäß dem Vertrag in Rechnung gestellt, dem Sie beim Kauf der Instance zugestimmt haben. Weitere Informationen finden Sie

unter Reservierte DB-Instances für Amazon RDS. Sie können Abrechnungsinformationen für all Ihre AWS Ressourcen abrufen, indem Sie den verwenden. AWS Cost Explorer Weitere Informationen finden Sie unter Analysieren Ihrer Kosten mit AWS Cost Explorer.

- Wenn Sie eine DB-Instance löschen, in der sich Read Replicas befinden AWS-Region, wird jede Read Replica automatisch zu einer eigenständigen DB-Instance heraufgestuft. Weitere Informationen finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance. Wenn Ihre DB-Instance Read Replicas in verschiedenen Versionen hat AWS-Regionen, finden Sie Informationen Überlegungen zur regionsübergreifenden Replikation zum Löschen der Quell-DB-Instance für eine regionsübergreifende Read Replica unter.
- Wenn der Status einer DB-Instance lautet`deleting`, erscheint ihr CA-Zertifikatswert nicht in der RDS-Konsole oder in der Ausgabe für AWS CLI Befehle oder RDS-API-Operationen. Weitere Informationen zu CA-Zertifikaten finden Sie unter .
- Die zum Löschen einer DB-Instance erforderliche Zeit variiert je nach Aufbewahrungszeitraum für Backups (d. h. wie viele Backups gelöscht werden sollen), wie viele Daten gelöscht werden und ob ein endgültiger Snapshot erstellt wird.

Löschen einer DB-Instance

Sie können eine DB-Instance mithilfe der AWS Management Console AWS CLI, der oder der RDS-API löschen. Sie müssen die folgenden Schritte ausführen:

- Geben Sie den Namen der DB-Instance an.
- Aktivieren oder deaktivieren Sie die Option, einen endgültigen DB-Snapshot der Instance zu erstellen.
- Aktivieren oder deaktivieren Sie die Option zum Speichern automatisierter Sicherungen.

> **Note**
>
> Sie können eine DB-Instance nicht löschen, wenn der Löschschutz aktiviert ist. Weitere Informationen finden Sie unter Voraussetzungen für das Löschen einer DB-Instance.

Konsole

So löschen Sie eine DB-Instance

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance aus, die Sie löschen möchten.
3. Klicken Sie bei Actions auf Delete.
4. Um einen endgültigen DB-Snapshot für die DB-Instance zu erstellen, aktivieren Sie Create final snapshot? (Endgültigen Snapshot erstellen?).
5. Wenn Sie einen endgültigen Snapshot erstellen möchten, geben Sie den Final snapshot name (Name des endgültigen Snapshots) ein.
6. Wählen Sie Retain automated backups (Automatisierte Sicherungen aufbewahren), um automatisierte Sicherungen aufzubewahren.
7. Geben Sie **`delete me`** in das Feld ein.
8. Wählen Sie Löschen aus.

AWS CLI

Rufen Sie den describe-db-instancesfolgenden Befehl auf, um die Instance-IDs der DB-Instances in Ihrem Konto zu finden:

```
aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier]' --output
 text
```

Um eine DB-Instance mithilfe von zu löschen AWS CLI, rufen Sie den delete-db-instanceBefehl mit den folgenden Optionen auf:

- `--db-instance-identifier`
- `--final-db-snapshot-identifier` oder `--skip-final-snapshot`

Example Mit einem endgültigen Snapshot und ohne aufbewahrte automatisierte Sicherungen

Für LinuxmacOS, oderUnix:

```
aws rds delete-db-instance \
```

```
    --db-instance-identifier mydbinstance \
    --final-db-snapshot-identifier mydbinstancefinalsnapshot \
    --delete-automated-backups
```

Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --final-db-snapshot-identifier mydbinstancefinalsnapshot ^
    --delete-automated-backups
```

Example Mit aufbewahrten automatisierten Sicherungen und ohne einen endgültigen Snapshot

Für LinuxmacOS, oderUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier mydbinstance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

RDS-API

Zum Löschen einer DB-Instance mit der Amazon RDS-API rufen Sie die Operation `DeleteDBInstance` mit den folgenden Parametern auf:

- `DBInstanceIdentifier`
- `FinalDBSnapshotIdentifier` oder `SkipFinalSnapshot`

Tutorial: Verwaltung einer My SQL DB-Instance-Umgebung von der Entwicklung bis zur Produktion

Einführung

Ein gängiges Szenario bei der Verwaltung einer Amazon RDS DB-Instance beinhaltet die Überwachung ihres Lebenszyklus von der ersten Entwicklung bis zur Produktionsbereitstellung. Dieses Tutorial bietet Anleitungen zur Bewältigung wichtiger Aufgaben, um sicherzustellen, dass Ihre Datenbank optimal funktioniert und sich an Ihre sich ändernden betrieblichen Anforderungen anpasst. Darüber hinaus werden Optionen zur Synchronisierung von Änderungen zwischen Ihrer Entwicklungs- und Produktionsumgebung beschrieben, um Konsistenz und Zuverlässigkeit zu gewährleisten.

Wenn Sie diese Schritte ausführen, lernen Sie Folgendes:

- So führen Sie bestimmte Aufgaben mit My SQL DB-Instances aus, z. B. das Hinzufügen und Aktualisieren von RDS Amazon-Tags, das Erweitern des Speichers, das Erstellen von Read Replicas und das Löschen von Ressourcen.
- So synchronisieren Sie Updates aus einer Produktionsumgebung mit einer Entwicklungsumgebung für umfassende Tests und Validierungen.

Führen Sie die folgenden Aufgaben aus, um dieses Tutorial abzuschließen:

1. Erstellen Sie eine My SQL DB-Instance.
2. Fügen Sie RDS Amazon-Tags hinzu, um Ihre DB-Instance als Entwicklungsumgebung zu kategorisieren.
3. Erhöhen Sie die Speicherkapazität Ihrer DB-Instance, um den erhöhten Workloads gerecht zu werden.
4. Erstellen Sie Read Replicas, um die Widerstandsfähigkeit und Verfügbarkeit Ihrer DB-Instance zu verbessern.
5. Aktualisieren Sie RDS Amazon-Tags, um Ihre DB-Instance als Produktionsumgebung zu kategorisieren.
6. Löschen Sie DB-Instances, die Sie nicht mehr benötigen, damit für sie keine zusätzlichen Kosten anfallen.

7. Nächste Schritte: Synchronisieren Sie Ihre Entwicklungsinstanz mit der Produktionsinstanz, um Konsistenz in allen Umgebungen zu gewährleisten

Voraussetzungen

Bevor Sie die Schritte in diesem Abschnitt abschließen, stellen Sie sicher, dass Sie folgende Voraussetzungen erfüllen:

- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Fügen Sie Tags hinzu, um Ihre DB-Instance als Entwicklungsumgebung zu kategorisieren

Um die DB-Instance als Entwicklungsumgebung zu kategorisieren, fügen Sie der von Ihnen erstellten Instance ein RDS Amazon-Tag hinzu. Ein RDS Amazon-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und mit Ihrer RDS Instance verknüpfen. Durch die Kennzeichnung Ihrer AWS Ressourcen können Sie zwischen Ihren Entwicklungs- und Produktionsressourcen unterscheiden. AWS Weitere Informationen zu RDS Amazon-Tags finden Sie unterVerschlagwortung von Amazon-Ressourcen RDS.

1. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.
2. Wählen Sie die DB-Instance aus, die Sie taggen möchten.
3. Scrollen Sie im Detailbereich zum Abschnitt Tags.
4. Wähle „Tags verwalten“ und dann „Neues Tag hinzufügen“.
5. Geben Sie einen Wert für Tag key (Tag-Schlüssel) und Wert ein. Sie können beispielsweise das Tag key environment mit dem Wert dev verwenden, um anzugeben, dass die Datenbankinstanz Teil der Entwicklungsumgebung ist.
6. Wählen Sie Neues Tag hinzufügen und Änderungen speichern.

Ihre DB-Instance ist jetzt als Entwicklungsumgebung gekennzeichnet. Dies erleichtert die Identifizierung der DB-Instance und die Verwaltung der mit dieser Ressource verbundenen Kosten.

Erhöhen Sie die Speicherkapazität einer DB-Instance, um den wachsenden Datenanforderungen gerecht zu werden

Ändern Sie als Nächstes die Speicherkapazität der My SQL DB-Instance, um zusätzliche Daten aufzunehmen. Zunächst wird die Speicherkapazität Ihrer DB-Instance so eingestellt, dass sie den unmittelbaren Anforderungen Ihrer Anwendung entspricht. Bei wachsenden Datenmengen kann es jedoch erforderlich sein, die Speichereinstellungen anzupassen, um die kontinuierliche Leistung und Stabilität der Datenbank sicherzustellen. Dieser Prozess beinhaltet die Erhöhung des zugewiesenen Speichers Ihrer DB-Instance. Weitere Informationen zum Ändern der Speicherkapazität Ihrer DB-Instance finden Sie unterArbeiten mit Speicher für Amazon RDS DB-Instances.

1. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.
2. Wählen Sie die DB-Instance aus, die Sie ändern möchten.
3. Wählen Sie Ändern aus.
4. Erhöhen Sie unter Speicher den Wert für zugewiesenen Speicher. Der geänderte Speicherwert muss größer als der aktuelle sein.
5. Klicken Sie auf Weiter.
6. Unter Planung von Änderungen können Sie entweder „Sofort anwenden“ wählen, um die Speicheränderungen sofort auf die DB-Instance anzuwenden, oder „Während des nächsten geplanten Wartungsfensters anwenden“ wählen, um die Änderungen im nächsten Wartungsfenster zu übernehmen.
7. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Modify DB instance (DB-Instance ändern).

Die Speicherkapazität Ihrer DB-Instance ist jetzt erhöht. Dadurch kann sie größere Datenmengen effektiv verarbeiten und gewährleistet kontinuierliche Leistung und Stabilität, wenn die Datenanforderungen Ihrer Anwendung steigen.

Erstellen Sie Read Replicas, um die Widerstandsfähigkeit und Verfügbarkeit einer DB-Instance zu verbessern

Erstellen Sie eine Read Replica der My SQL DB-Instance. Read Replicas verbessern die Widerstandsfähigkeit und Verfügbarkeit Ihrer DB-Instance. Um den Lesetraffic auf Ihrer primären DB-Instance zu reduzieren, erstellen Sie eine Read Replica Ihrer DB-Instance. Dadurch werden Abfragen an die Read Replica weitergeleitet, was dazu beitragen kann, die Last zu verteilen und die allgemeine

Datenbankleistung zu verbessern. Weitere Informationen zu Read Replicas von DB-Instances finden Sie unter. Arbeiten mit DB-Instance-Lesereplikaten

Bevor eine My SQL DB-Instance als Replikationsquelle dienen kann, müssen automatische Backups auf der Quell-DB-Instance aktiviert werden. Dies kann erreicht werden, indem die Aufbewahrungsfrist für Backups auf einen anderen Wert als 0 festgelegt wird. Weitere Informationen zu My SQL Read Replicas finden Sie unterMit My SQL Read Replicas arbeiten.

1. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.
2. Wählen Sie die DB-Instance aus, die Sie als Quelle für die Read Replica verwenden möchten.
3. Wählen Sie unter Aktionen die Option Read Replica erstellen aus.
4. Geben Sie als DB-Instance-ID einen Namen für die Read Replica in Kleinbuchstaben ein.
5. Wählen Sie Ihre Instance-Konfiguration. Wir empfehlen Ihnen, dieselbe oder eine größere DB-Instance-Klasse und denselben Speichertyp wie bei der Quell-DB-Instance für das Lesereplikat zu verwenden.
6. Legen Sie unter AWS-Region die Zielregion für das Lesereplikat fest.
7. Behalten Sie die Standardeinstellungen bei oder ändern Sie sie nach Bedarf.
8. Wählen Sie Read Replica erstellen aus.

Die Read Replica wird unter Ihrer Quell-DB-Instance auf der Datenbankseite in der RDS Konsole angezeigt. In der Spalte Rolle wird Replica angezeigt.

Aktualisieren Sie Tags, um eine DB-Instance als Produktionsumgebung zu kategorisieren

Wenn Ihre DB-Instance bereit ist, von der Entwicklungsphase zur Produktionsphase überzugehen, ist es wichtig, ihre Tags zu aktualisieren, um dem Übergang Rechnung zu tragen. Um Ihre DB-Instance an Ihren Betriebs- und Überwachungsstrategien auszurichten, aktualisieren Sie die ersten Tags so, dass sie darauf hinweisen, dass die DB-Instance jetzt Teil der Produktionsumgebung ist. Dies gewährleistet eine bessere Sichtbarkeit und Verwaltung der Datenbank.

1. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.
2. Wählen Sie die DB-Instance aus, die Sie aktualisieren möchten
3. Scrollen Sie im Detailbereich zum Abschnitt Tags.
4. Wählen Sie Tags verwalten aus.

5. Entfernen Sie Ihr erstes Tag, das auf eine Entwicklungsumgebung hinweist.
6. Wählen Sie Neues Tag hinzufügen aus.
7. Geben Sie einen neuen Wert für Tag-Schlüssel und Wert ein. Sie können beispielsweise das Tag key environment mit dem Wert prod verwenden, um anzugeben, dass die DB-Instance Teil der Produktionsumgebung ist.
8. Wählen Sie Neues Tag hinzufügen und Änderungen speichern.

Das Tag auf Ihrer DB-Instance wird aktualisiert, um den Übergang der Datenbank in eine Produktionsumgebung anzuzeigen.

Löschen Sie eine DB-Instance, wenn sie nicht mehr benötigt wird, um zusätzliche Kosten zu vermeiden

Vor dem Ende dieses Tutorials ist es wichtig, sich mit der Verwaltung Ihrer Ressourcen zu befassen. Wenn Sie über Ressourcen verfügen, die nicht mehr benötigt werden, sollten Sie diese löschen, um zusätzliche Kosten zu vermeiden und Ihre Cloud-Umgebung zu optimieren.

1. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.
2. Wählen Sie die DB-Instance aus, die Sie löschen möchten
3. Wählen Sie unter Aktionen die Option Löschen aus. Durch das Löschen einer DB-Instance wird die Instance mit all ihren Inhalten und zugehörigen Ressourcen dauerhaft gelöscht.
4. Bestätigen Sie das Löschen der DB-Instance und wählen Sie Löschen aus.

Wenn Sie Ihre DB-Instance für die future Verwendung beibehalten möchten, können Sie sie alternativ weiterhin als Teil Ihrer Produktionsumgebung verwalten. Dazu gehört die Aufrechterhaltung einer synchronisierten Entwicklungsumgebung, um umfassende Tests und Validierungen zu ermöglichen. Weitere Informationen finden Sie unter Nächste Schritte: Synchronisieren Sie Ihre Entwicklungsinstanz mit der Produktionsinstanz, um Konsistenz in allen Umgebungen zu gewährleisten.

Nächste Schritte: Synchronisieren Sie Ihre Entwicklungsinstanz mit der Produktionsinstanz, um Konsistenz in allen Umgebungen zu gewährleisten

Erstellen einer Entwicklungsumgebung

Um eine Produktionsumgebung zu verwalten, ist es wichtig, eine synchronisierte Entwicklungsumgebung für umfassende Tests und Validierungen aufrechtzuerhalten. Um eine

neue Entwicklungsumgebung zu erstellen, erstellen Sie zunächst einen DB-Snapshot der aktuellen Produktions-DB-Instance. Ein DB-Snapshot erfasst die gesamte DB-Instance, indem er einen Speicher-Volume-Snapshot erstellt. Anweisungen zum Erstellen eines DB-Snapshots auf der RDS Amazon-Konsole finden Sie unterErstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Nachdem Sie den DB-Snapshot Ihrer Produktionsumgebung erstellt haben, erstellen Sie eine neue DB-Instance für Ihre Entwicklungsumgebung, indem Sie einen DB-Snapshot wiederherstellen. Wiederhergestellte DB-Instances werden automatisch dem Standard-DB-Parameter und Optionsgruppen zugeordnet. Sie können jedoch eine benutzerdefinierte Parametergruppe und eine Optionsgruppe anwenden, indem Sie diese während einer Wiederherstellung angeben. Anweisungen zum Wiederherstellen eines DB-Snapshots finden Sie unterTutorial: Eine RDS Amazon-DB-Instance aus einem DB-Snapshot wiederherstellen.

Definieren Sie abschließend die neue DB-Instance als Ihre neue Entwicklungsumgebung, indem Sie ihre RDS Amazon-Tags aktualisieren. Hinweise zur Aktualisierung von RDS Amazon-Tags an diese Änderung finden Sie im vorherigen AbschnittAktualisieren Sie Tags, um eine DB-Instance als Produktionsumgebung zu kategorisieren.

Sie verfügen jetzt über eine neue Entwicklungsumgebung, die die Datenbankkonfiguration Ihrer Produktionsumgebung widerspiegelt.

Synchronisieren Sie eine Entwicklungsumgebung mit der Produktionsumgebung

Sobald Ihre neue Entwicklungsumgebung eingerichtet ist, muss sie mit allen Änderungen in der Produktionsumgebung synchronisiert werden. Dadurch wird sichergestellt, dass Ihre Entwicklungsumgebung den aktuellen Produktionsstatus genau wiedergibt, was für effektive Tests, Validierungen und Problembehebungen unerlässlich ist. Amazon RDS bietet eine Vielzahl verschiedener Möglichkeiten, um Ihre Entwicklungsumgebung mit Ihrer Produktionsumgebung auf dem neuesten Stand zu halten. Weitere Informationen zu diesen Optionen finden Sie unter Orchestrieren von Datenbank-Aktualisierungen für Amazon RDS und Amazon Aurora.

Eine der wichtigsten Möglichkeiten, Ihre Entwicklungs- und Produktionsumgebungen zu synchronisieren, ist die Erstellung und Wiederherstellung von DB-Snapshots. Mit einem DB-Snapshot können Sie eine Entwicklungsumgebung erstellen, die die Datenbankkonfiguration der Produktionsumgebung zum Zeitpunkt der Erstellung des Snapshots widerspiegelt. Weitere Informationen zu DB-Snapshots finden Sie unterVerwaltung manueller Backups. Weitere Informationen zum Wiederherstellen einer DB-Instance finden Sie unterWiederherstellung auf einer DB-Instance.

DB-Snapshots sind für die folgenden Anwendungsfälle besonders wertvoll.

- Ersteinrichtung einer Entwicklungsumgebung: DB-Snapshots sind nützlich, um die erste Entwicklungsumgebung für Tests zu erstellen, da sie eine konsistente Ausgangsbasis bieten, die den genauen Zustand der Produktionsumgebung zum Zeitpunkt des Snapshots widerspiegelt.
- Anwendungen mit hohem Datenaufkommen: In Produktionsumgebungen, in denen ein kontinuierlicher Betrieb entscheidend ist, vermeidet die Verwendung von Multi-AZ-Bereitstellungen für Snapshots die I/O-Unterbrechung der Primärdatenbank und gewährleistet so eine unterbrechungsfreie Leistung und hohe Verfügbarkeit.
- Gemeinsame Nutzung von Daten zwischen verschiedenen RDS Konten: DB-Snapshots können von verschiedenen Konten gemeinsam genutzt werden AWS-Konten, was die Übertragung von Daten zwischen Konten oder Regionen erleichtert. Dies ist nützlich für Kooperationsprojekte oder Szenarien, in denen Daten für verschiedene Zwecke gemeinsam genutzt werden müssen. Weitere Informationen finden Sie unter Einen DB-Snapshot für Amazon teilen RDS.

In diesem Tutorial haben Sie grundlegende Aufgaben für die Verwaltung Ihrer DB-Instance während ihres gesamten Lebenszyklus untersucht. Sie haben gelernt, wie Sie eine DB-Instance erstellen, RDS Amazon-Tags hinzufügen und aktualisieren, den Speicher erweitern und Read Replicas erstellen. Sie haben auch gelernt, wie Sie auf diesen grundlegenden Vorgängen aufbauen und Ihre Produktionsumgebung effektiv verwalten können. Dazu gehörte die Einrichtung einer Entwicklungsumgebung zum Testen und deren Synchronisation mit der Produktionsumgebung, um Konsistenz zu gewährleisten. Diese Aufgaben tragen dazu bei, eine belastbare und skalierbare Datenbankinfrastruktur aufrechtzuerhalten und sicherzustellen, dass Ihre RDS Amazon-Umgebung effizient funktioniert.

Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS

Multi-AZ-Bereitstellungen können eine Standby-oder zwei Standby-DB-Instances haben. Wenn die Bereitstellung über eine Standby-DB-Instance verfügt, wird sie als Multi-AZ-DB-Instance-Bereitstellung bezeichnet. Eine Multi-AZ DB-Instance-Bereitstellung verfügt über eine Standby-DB-Instance, die Failover-Unterstützung bietet, aber keinen Lesedatenverkehr bereitstellt. Wenn die Bereitstellung über zwei Standby-DB-Instances verfügt, wird sie als Multi-AZ-DB-Cluster-Bereitstellung bezeichnet. Eine Multi-AZ-DB-Cluster-Bereitstellung verfügt über Standby-DB-Instances, die Failover-Unterstützung bieten und auch Leseverkehr bedienen können.

Sie können den verwenden AWS Management Console , um festzustellen, ob es sich bei einer Multi-AZ-Bereitstellung um eine Multi-AZ-DB-Instance-Bereitstellung oder um eine Multi-AZ-DB-Cluster-Bereitstellung handelt. Wählen Sie im Navigationsbereich die Option Databases (Datenbanken) und anschließend eine DB identifier (DB-Kennung).

- Eine Multi-AZ-DB-Instance-Bereitstellung hat folgende Eigenschaften:
  - Es gibt nur eine Zeile für die DB-Instance.
  - Der Wert von Rolle ist Instance oder Primär.
  - Der Wert von Multi-AZ ist Yes (Ja).
- Eine Multi-AZ-DB-Cluster-Bereitstellung hat folgende Eigenschaften:
  - Es gibt eine Zeile auf Clusterebene mit drei DB-Instance-Zeilen darunter.
  - Für die Zeile auf Clusterebene ist der Wert von Rolle Multi-AZ-DB-Cluster.
  - Für jede Zeile auf Instance-Ebene ist der Wert von Rolle Reader oder Reader-Instance.
  - Für jede Zeile auf Instance-Ebene ist der Wert von -Multi-AZ 3 Zonen.

Themen

- Multi-AZ-DB-Instance-Bereitstellungen für Amazon RDS
- Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS

Darüber hinaus gelten die folgenden Themen sowohl für DB-Instances als auch für Multi-AZ-DB-Cluster.

- the section called “ RDS Ressourcen taggen”

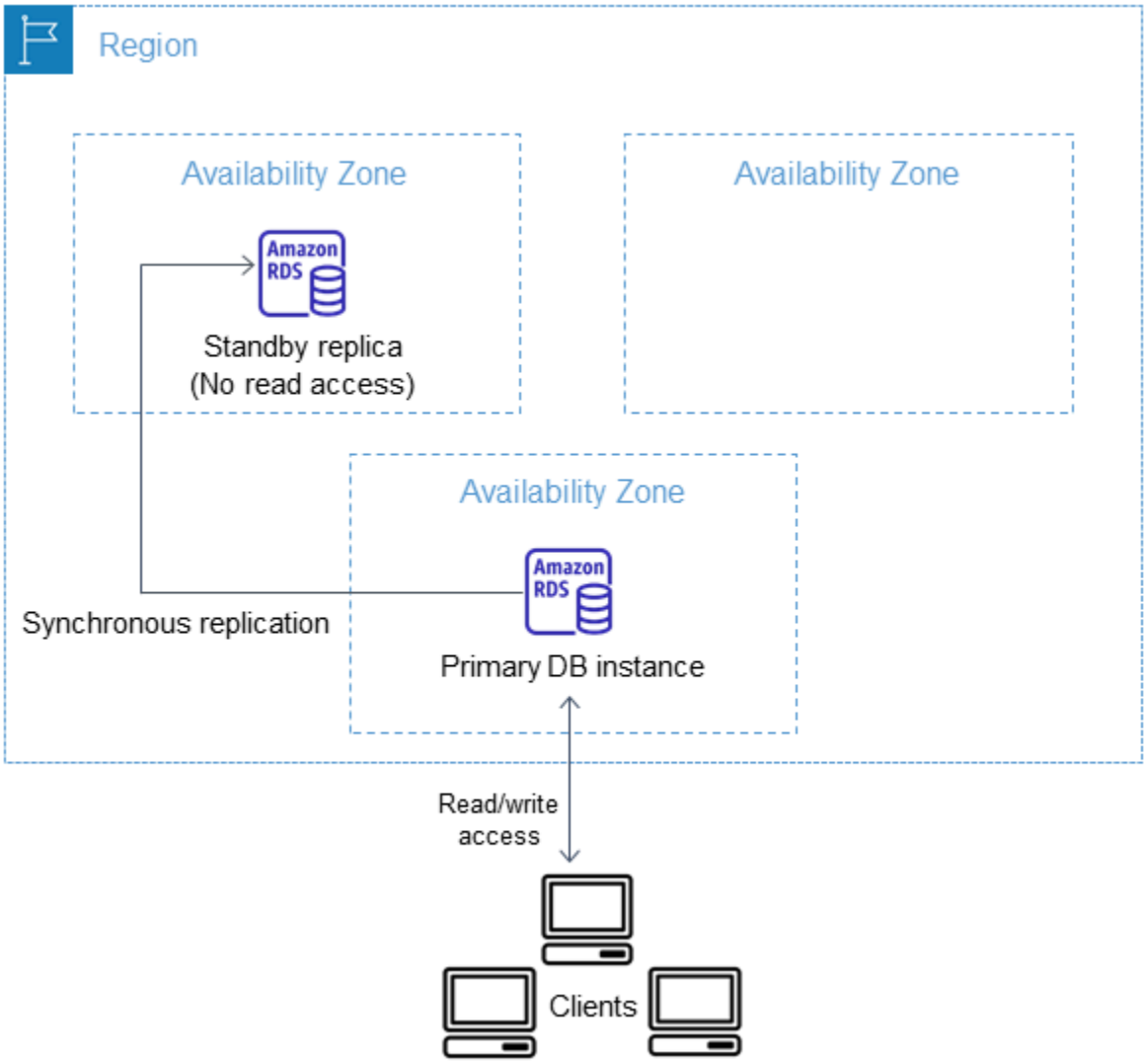
- the section called “ARNsbei Amazon RDS”
- the section called “Arbeiten mit Speicher”
- the section called “Warten einer DB-Instance”
- the section called “Upgrade der Engine-Version”

Multi-AZ-DB-Instance-Bereitstellungen für Amazon RDS

Amazon RDS bietet Hochverfügbarkeit und Failover-Unterstützung für DB-Instances, die Multi-AZ-Bereitstellungen mit einer einzigen Standby-DB-Instance verwenden. Diese Art der Bereitstellung wird als Multi-AZ-DB-Instance-Bereitstellung bezeichnet. Amazon RDS verwendet verschiedene Technologien, um diese Failover-Unterstützung bereitzustellen. Multi-AZ-Bereitstellungen für MariaDB-, My-, Oracle-SQL, Postgre SQL - und RDS Custom for SQL Server-DB-Instances verwenden die Amazon-Failover-Technologie. Microsoft SQL Server-DB-Instances verwenden SQL Server Database Mirroring (DBM) oder AlwaysOn-Verfügbarkeitsgruppen (AGs). Informationen zur Unterstützung von SQL Serverversionen für Multi-AZ finden Sie unter. Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL Informationen zur Arbeit mit RDS Custom for SQL Server für Multi-AZ finden Sie unter. Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Server SQL

In einer Multi-AZ-DB-Instance-Bereitstellung stellt Amazon RDS automatisch ein synchrones Standby-Replikat in einer anderen Availability Zone bereit und verwaltet es. Die primäre DB-Instance wird synchron über Availability Zones hinweg auf ein Standby-Replikat repliziert, um Datenredundanz bereitzustellen und Latenzspitzen während Systemsicherungen zu minimieren. Wenn Sie eine DB-Instance mit hoher Verfügbarkeit ausführen, kann dies die Verfügbarkeit bei geplanten Systemwartungen verbessern. Sie kann auch Ihre Datenbanken bei Ausfällen der DB-Instance und bei Nichtverfügbarkeit von Availability Zones schützen. Weitere Informationen über Availability Zones finden Sie unter Regionen, Availability Zones und Local Zones.

> **Note**
>
> Die Option für hohe Verfügbarkeit ist keine Skalierungslösung für schreibgeschützte Szenarien. Sie können kein Standby-Replikat verwenden, um Leseverkehr bereitzustellen. Um schreibgeschützten Datenverkehr bereitzustellen, verwenden Sie stattdessen einen Multi-AZ-DB-Cluster oder ein Lesereplikat. Weitere Informationen zu Multi-AZ-DB-Clustern finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS. Weitere Informationen über Lesereplikate finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.



Mithilfe der RDS Konsole können Sie eine Multi-AZ-DB-Instance-Bereitstellung erstellen, indem Sie bei der Erstellung einer DB-Instance einfach Multi-AZ angeben. Sie können über die Konsole bestehende DB-Instances in Multi-AZ-Bereitstellungen für DB-Instances umwandeln, indem Sie die DB-Instance ändern und die Option „Multi-AZ“ angeben. Sie können auch eine Multi-AZ-DB-Instance-Bereitstellung mit dem AWS CLI oder Amazon RDS API angeben. Verwenden Sie den modify-db-instanceCLIBefehl create-db-instanceoder oder die odifyDBInstance API Operation CreateDBInstance oder M.

In der RDS Konsole wird die Availability Zone des Standby-Replikats (die so genannte sekundäre AZ) angezeigt. Sie können auch den describe-db-instancesCLIBefehl oder die escribeDBInstancesAPID-Operation verwenden, um die sekundäre AZ zu finden.

DB-Instances, die Multi-AZ-DB-Instance-Bereitstellungen verwenden, können im Vergleich zu einer Single-AZ-Bereitstellung eine höhere Schreib- und Commit-Latenz aufweisen. Dies kann aufgrund

der auftretenden synchronen Datenreplikation geschehen. Es kann zu einer Änderung der Latenz kommen, wenn bei Ihrer Bereitstellung ein Failover auf das Standby-Replikat erfolgt, obwohl es AWS für Netzwerkverbindungen mit niedriger Latenz zwischen Availability Zones konzipiert wurde. Für Produktionsworkloads empfehlen wir, Provisioned IOPS (Eingabe-/Ausgabevorgänge pro Sekunde) zu verwenden, um eine schnelle, konsistente Leistung zu erzielen. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Konvertierung einer DB-Instance in eine Multi-AZ-Bereitstellung für Amazon RDS

Die Änderung einer DB-Instance in eine Multi-AZ-Bereitstellung verbessert die Verfügbarkeit, indem eine Standby-Instance in einer anderen Availability Zone hinzugefügt wird. Dieser Prozess ist mit minimalen Ausfallzeiten verbunden und erfordert eine sorgfältige Planung im Hinblick auf Speicher- und Leistungseinbußen. Diese Änderung verbessert die Fehlertoleranz und reduziert die Wiederherstellungszeit bei Ausfällen, sodass sie sich ideal für Umgebungen mit hoher Verfügbarkeit eignet.

Wenn Sie eine DB-Instance in einer Single-AZ-Bereitstellung haben und diese zu einer Multi-AZ-DB-Instance-Bereitstellung ändern, RDS führt Amazon die folgenden Aktionen durch:

1. Erstellt einen Snapshot der Amazon Elastic Block Store (EBS) -Volumes der primären DB-Instance.
2. Erstellt neue Volumes für das Standby-Replikat aus dem Snapshot. Diese Volumes werden im Hintergrund initialisiert, und die maximale Volume-Leistung wird erreicht, nachdem die Daten vollständig initialisiert wurden.
3. Aktiviert die synchrone Replikation auf Blockebene zwischen den Volumes der primären und Standby-Replikate.

> **Important**
>
> Das Erstellen einer Standby-DB-Instance aus einem Snapshot während einer Single-AZ- zu Multi-AZ-Konvertierung vermeidet Ausfallzeiten, kann jedoch die Leistung beeinträchtigen, insbesondere bei schreibsensitiven Workloads. Die synchrone Replikation kann die I/O-Latenz erhöhen, was sich auf die Datenbankleistung auswirkt. Es hat sich bewährt, die Konvertierung einer Produktions-DB-Instance in eine Multi-AZ-DB-Instance zu vermeiden. Erstellen Sie stattdessen eine Read Replica, aktivieren Sie Backups darauf, konvertieren Sie sie in Multi-AZ, laden Sie Daten in ihre Volumes und stufen Sie sie dann zur primären

DB-Instance herauf. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Es gibt zwei Möglichkeiten, eine DB-Instance in eine Multi-AZ-DB-Instance-Bereitstellung zu ändern:

Themen

- Mit der Konsole in eine Multi-AZ-DB-Instance-Bereitstellung konvertieren RDS
- Ändern einer DB-Instance zu einer Multi-AZ-DB-Instance-Bereitstellung

Mit der Konsole in eine Multi-AZ-DB-Instance-Bereitstellung konvertieren RDS

Sie können die RDS Konsole verwenden, um eine DB-Instance in eine Multi-AZ-DB-Instance-Bereitstellung zu konvertieren.

Sie können die Konvertierung nur mit der Konsole abschließen. Um das AWS CLI oder zu verwenden RDSAPI, folgen Sie den Anweisungen unterÄndern einer DB-Instance zu einer Multi-AZ-DB-Instance-Bereitstellung.

Um mit der Konsole zu einer Multi-AZ-DB-Instance-Bereitstellung zu konvertieren RDS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie unter Actions (Aktionen) die Option Convert to Multi-AZ deployment In Multi-AZ-Bereitstellung konvertieren aus.
4. Damit Änderungen sofort übernommen werden, wählen Sie die Option Apply Immediately (Sofort anwenden) auf der Bestätigungsseite aus. Die Auswahl dieser Option verursacht keine Ausfallzeiten, kann jedoch zur Beeinträchtigung der Leistung führen. Sie können die Aktualisierung auch im nächsten Wartungsfenster übernehmen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.
5. Wählen Sie Convert to Multi-AZ (In Multi-AZ konvertieren) aus.

Ändern einer DB-Instance zu einer Multi-AZ-DB-Instance-Bereitstellung

Sie können eine DB-Instance auf folgende Weise in eine Multi-AZ-DB-Instance-Bereitstellung umwandeln:

- Ändern Sie die DB-Instance mithilfe der RDS Konsole und setzen Sie die Multi-AZ-Bereitstellung auf Ja.
- Rufen Sie mit dem AWS CLI den modify-db-instanceBefehl auf und legen Sie die `--multi-az` Option fest.
- Rufen Sie mit dem RDS API die odifyDBInstance Operation M auf und setzen Sie den `MultiAZ` Parameter auf`true`.

Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance. Nachdem die Änderung abgeschlossen ist, RDS löst Amazon ein Ereignis aus (RDS-EVENT -0025), das anzeigt, dass der Vorgang abgeschlossen ist. Sie können RDS Amazon-Ereignisse überwachen. Weitere Informationen über -Ereignisse finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.

Failover einer Multi-AZ-DB-Instance für Amazon RDS

Wenn ein geplanter oder ungeplanter Ausfall Ihrer Multi-AZ-DB-Instance auf einen Infrastrukturdefekt zurückzuführen ist, wechselt Amazon RDS automatisch zu einem Standby-Replikat in einer anderen Availability Zone.

Die Dauer, bis der Failover-Prozess für die Instance abgeschlossen ist, hängt von der Datenbankaktivität sowie von anderen Bedingungen zu dem Zeitpunkt ab, an dem die primäre DB-Instance ausgefallen ist. Der Failover-Prozess dauert normalerweise 60–120 Sekunden. Diese Failover-Dauer kann sich verlängern, wenn umfangreiche Transaktionen oder zeitintensive Wiederherstellungsprozesse durchgeführt werden. Wenn der Failover abgeschlossen ist, kann es länger dauern, bis die RDS Konsole die neue Availability Zone wiedergibt.

> **Note**
>
> Sie können ein Failover manuell erzwingen, wenn Sie eine Multi-AZ-DB-Instance neu starten. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Amazon RDS wickelt Failovers automatisch ab, sodass Sie den Datenbankbetrieb so schnell wie möglich ohne administrativen Eingriff wieder aufnehmen können. Die primäre DB-Instance schaltet automatisch auf das Standby-Replikat um, wenn eine der in der folgenden Tabelle beschriebenen Bedingungen eintritt: Sie können diese Failover-Gründe im Ereignisprotokoll einsehen.

Failover-Grund	Beschreibung
Das Betriebssystem, das der RDS Datenbank-Instance zugrunde liegt, wird in einem Offline-Vorgang gepatcht.	Ein Failover wurde während des Wartungsfensters für einen Betriebssystem-Patch oder ein Sicherheitsupdate ausgelöst. Weitere Informationen finden Sie unter Warten einer DB-Instance.
Der primäre Host der RDS Multi-AZ-Instance ist fehlerhaft.	Die Multi-AZ-DB-Instance-Bereitstellung hat eine beeinträchtigte primäre DB-Instance erkannt und ein Failover eingeleitet.
Der primäre Host der RDS Multi-AZ-Instance ist aufgrund eines Verlusts der Netzwerkkonnektivität nicht erreichbar.	RDSBei der Überwachung wurde ein Fehler bei der Netzwerkerreichbarkeit der primären DB-Instance festgestellt und ein Failover ausgelöst.
Die RDS Instanz wurde vom Kunden geändert.	Eine Änderung der RDS DB-Instance löste einen Failover aus. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
Die primäre RDS Multi-AZ-Instance ist ausgelastet und reagiert nicht.	Die primäre DB-Instance reagiert nicht. Wir empfehlen Folgendes: • Untersuchen Sie das Ereignis und die CloudWatch Protokolle auf übermäßige Speicher CPU - oder Auslagerungsspeicherbelegung. Weitere Informationen erhalten Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten und Erstellen einer

Failover-Grund	Beschreibung
	Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird. • Prüfen Sie Ihren Workload, um festzustellen, ob Sie die angemessene DB-Instance-Klasse verwenden. Weitere Informationen finden Sie unter . • Verwenden Sie Enhanced Monitoring (Erweiterte Überwachung) für Betriebss ystemmetriken in Echtzeit. Weitere Informati onen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Übe rwachung). • Verwenden Sie Performance-Insights, um Probleme zu analysieren, die sich auf die Leistung Ihrer DB-Instance auswirken . Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS. Weitere Informationen zu diesen Empfehlun gen finden Sie unter Überwachungstools für Amazon RDS und Bewährte Methoden für Amazon RDS.
Auf dem Speichervolume, das dem primären Host der RDS Multi-AZ-Instance zugrunde liegt, ist ein Fehler aufgetreten.	Die Multi-AZ-DB-Instance-Bereitstellung hat ein Speicherproblem der primären DB-Instance erkannt und ein Failover eingeleitet.

Failover-Grund	Beschreibung
Der Benutzer hat ein Failover der DB-Instance angefordert.	Sie haben die DB-Instance neu gestartet und Neustart mit Failover gewählt. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Um festzustellen, ob Ihre Multi-AZ-DB-Instance erfolgreich ausgeführt wurde, können Sie Folgendes tun:

- Richten Sie DB-Event-Abonnements ein, um Sie per E-Mail oder darüber zu informierenSMS, dass ein Failover eingeleitet wurde. Weitere Informationen über -Ereignisse finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.
- Zeigen Sie Ihre DB-Ereignisse mithilfe der RDS Konsole oder mithilfe von API Operations an.
- Zeigen Sie den aktuellen Status Ihrer Multi-AZ-DB-Instance-Bereitstellung mithilfe der RDS Konsole oder mithilfe von API Operations an.

Informationen darüber, wie Sie auf Failover reagieren, die Wiederherstellungszeit verkürzen können, und weitere bewährte Methoden für Amazon RDS finden Sie unterBewährte Methoden für Amazon RDS.

Einstellung der JVM TTL für DNS Namenssuchvorgänge

Der Failover-Mechanismus ändert den Domain Name System (DNS) -Datensatz der DB-Instance automatisch so, dass er auf die Standby-DB-Instance verweist. Als Ergebnis müssen alle bestehenden Verbindungen zu Ihrer DB-Instance neu hergestellt werden. In einer Java-Virtual-Machine-Umgebung (JVM) müssen Sie aufgrund der Funktionsweise des DNS Java-Caching-Mechanismus möglicherweise die Einstellungen neu JVM konfigurieren.

Die Namen der JVM Caches suchen DNS nach Namen. Wenn der einen Hostnamen in eine IP-Adresse JVM auflöst, speichert er die IP-Adresse für einen bestimmten Zeitraum im Cache, der als () bezeichnet wird. time-to-liveTTL

Da AWS Ressourcen DNS Namenseinträge verwenden, die sich gelegentlich ändern, empfehlen wir, dass Sie Ihren JVM mit einem TTL Wert von nicht mehr als 60 Sekunden konfigurieren. Dadurch wird

sichergestellt, dass Ihre Anwendung, wenn sich die IP-Adresse einer Ressource ändert, die neue IP-Adresse der Ressource empfangen und verwenden kann, indem sie die erneut abfragt. DNS

Bei einigen Java-Konfigurationen TTL ist die JVM Standardeinstellung so eingestellt, dass DNS Einträge erst aktualisiert werden, wenn der JVM neu gestartet wird. Wenn sich also die IP-Adresse für eine AWS Ressource ändert, während Ihre Anwendung noch läuft, kann sie diese Ressource erst verwenden, wenn Sie die manuell neu starten JVM und die zwischengespeicherten IP-Informationen aktualisiert werden. In diesem Fall ist es wichtig, die JVM s TTL so einzustellen, dass die zwischengespeicherten IP-Informationen regelmäßig aktualisiert werden.

Sie können die JVM Standardeinstellung abrufen, TTL indem Sie den Eigenschaftswert abrufen: `networkaddress.cache.ttl`

```
String ttl = java.security.Security.getProperty("networkaddress.cache.ttl");
```

> **Note**
>
> Die Standardeinstellung TTL kann je nach Ihrer Version JVM und je nachdem, ob ein Sicherheitsmanager installiert ist, variieren. Viele JVMs bieten eine Standardeinstellung von TTL weniger als 60 Sekunden. Wenn Sie einen solchen JVM und keinen Sicherheitsmanager verwenden, können Sie den Rest dieses Themas ignorieren. Weitere Informationen zu Sicherheits-Managern in Oracle finden Sie unter The Security Manager in der Oracle-Dokumentation.

Um die JVM s zu ändernTTL, legen Sie den `networkaddress.cache.ttl` Eigenschaftswert fest. Nutzen Sie dazu eine der folgenden Methoden je nach Ihrem Bedarf:

- Um den Eigenschaftswert global für alle Anwendungen festzulegenJVM, die den `networkaddress.cache.ttl` in der `$JAVA_HOME/jre/lib/security/java.security` Datei festgelegten Wert verwenden.

  ```
  networkaddress.cache.ttl=60
  ```

- Um die Eigenschaft nur für Ihre Anwendung lokal festzulegen, legen Sie `networkaddress.cache.ttl` im Initialisierungscode Ihrer Anwendung fest, bevor Netzwerkverbindungen hergestellt werden.

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS

Eine Multi-AZ-DB-Cluster-Bereitstellung ist ein halbsynchroner, hochverfügbarer Bereitstellungsmodus von Amazon RDS mit zwei lesbaren Replikat-DB-Instances. Ein Multi-AZ-DB-Cluster verfügt über eine Schreiber-DB-Instance und zwei Leser-DB-Instances in drei separaten Availability Zones in der selben AWS-Region. Multi-AZ-DB-Cluster bieten hohe Verfügbarkeit, erhöhte Kapazität für Lese-Workloads und eine geringere Schreiblatenz im Vergleich zu Multi-AZ DB-Instance-Bereitstellungen.

Sie können Daten aus einer On-Premises-Datenbank in einen Multi-AZ-DB-Cluster importieren, indem Sie die Anweisungen unter Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit befolgen.

Sie können Reserved DB-Instances für einen Multi-AZ-DB-Cluster erwerben. Weitere Informationen finden Sie unter Reserved DB-Instances für einen Multi-AZ-DB-Cluster.

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Version und regionalen Verfügbarkeit von Amazon RDS mit Multi-AZ-DB-Clustern finden Sie unterUnterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS.

Themen

- Verfügbarkeit der Instance-Klassen für Multi-AZ-DB-Cluster
- Multi-AZ-DB-Cluster-Architektur
- Parametergruppen für Multi-AZ-DB-Cluster
- RDSProxy mit Multi-AZ-DB-Clustern
- Replikatverzögerung und Multi-AZ-DB-Cluster
- Snapshots von Multi-AZ-DB-Clustern
- Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen
- Verbindung zu einem Multi-AZ-DB-Cluster für Amazon herstellen RDS
- Automatisches Verbinden einer AWS Rechenressource und eines Multi-AZ-DB-Clusters für Amazon RDS
- Ändern eines Multi-AZ-DB-Clusters für Amazon RDS
- Aktualisierung der Engine-Version eines Multi-AZ-DB-Clusters für Amazon RDS

- Umbenennen eines Multi-AZ-DB-Clusters für Amazon RDS
- Einen Multi-AZ-DB-Cluster und Reader-DB-Instances für Amazon neu starten RDS
- Failover eines Multi-AZ-DB-Clusters für Amazon RDS
- Einrichtung der SQL logischen Postgre-Replikation mit Multi-AZ-DB-Clustern für Amazon RDS
- Arbeiten mit Multi-AZ-DB-Cluster-Read Replicas für Amazon RDS
- Einrichtung der externen Replikation von Multi-AZ-DB-Clustern für Amazon RDS
- Löschen eines Multi-AZ-DB-Clusters für Amazon RDS
- Einschränkungen von Multi-AZ-DB-Clustern für Amazon RDS

> **Important**
>
> Multi-AZ-DB-Cluster sind nicht dasselbe wie Aurora-DB-Cluster. Informationen zu Aurora-DB-Clustern finden Sie im Amazon-Aurora-Benutzerhandbuch.

Verfügbarkeit der Instance-Klassen für Multi-AZ-DB-Cluster

Multi-AZ-DB-Cluster-Bereitstellungen werden für die folgenden DB-Instance-Klassen unterstützt:`db.m5d`,`db.m6gd`,`db.m6id`,`db.m6idn`,,`db.r5d`,`db.r6gd`, und `db.x2iedn`db.r6id`, und`db.r6idn`. `db.c6gd`

> **Note**
>
> Die c6gd-Instance-Klassen sind die einzigen, die die Instance-Größe unterstützen. `medium`

Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”.

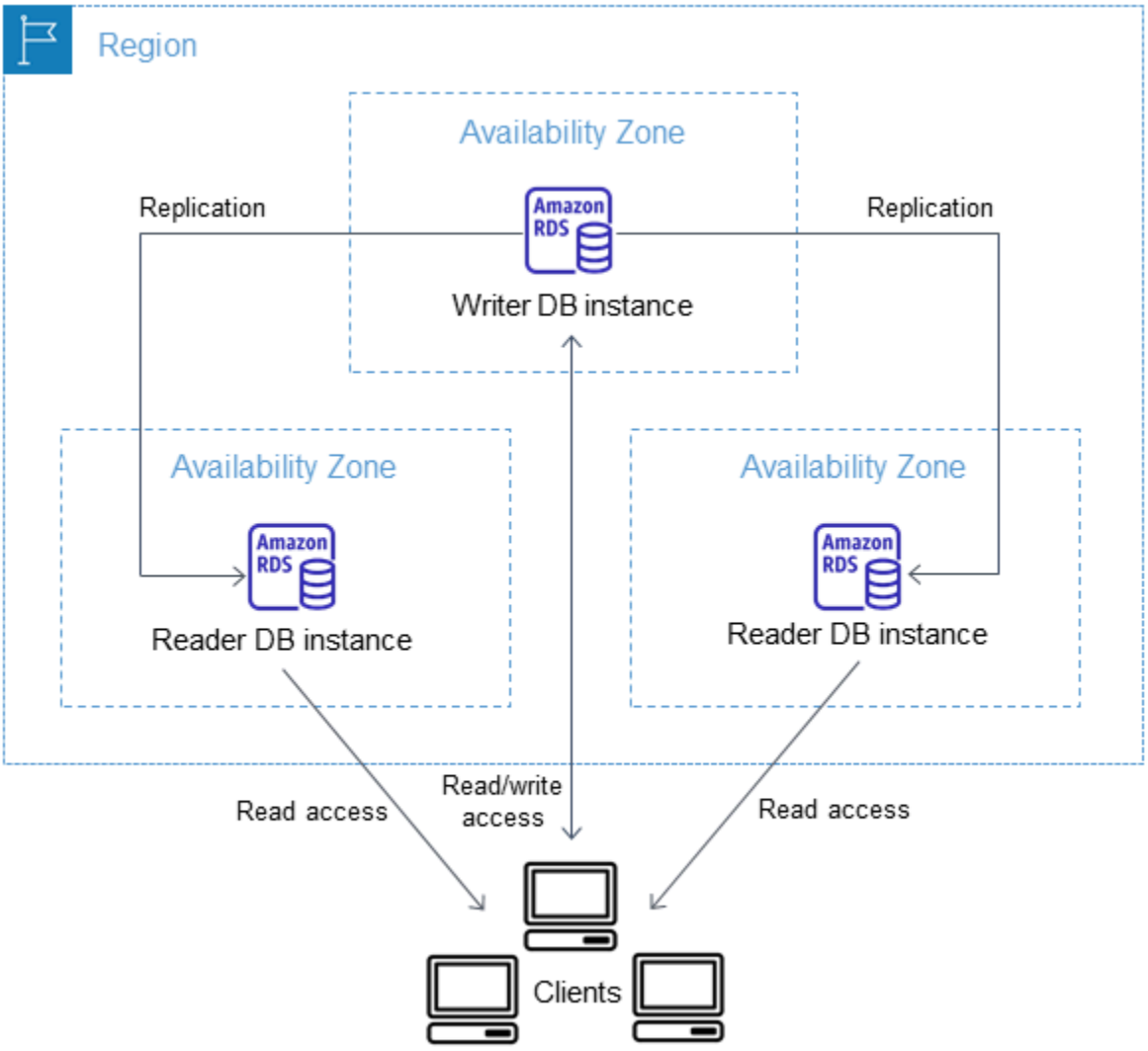
Multi-AZ-DB-Cluster-Architektur

Bei einem Multi-AZ-DB-Cluster RDS repliziert Amazon Daten von der Writer-DB-Instance auf beide Reader-DB-Instances mithilfe der nativen Replikationsfunktionen der DB-Engine. Wenn eine Änderung an der Writer-DB-Instance vorgenommen wird, wird sie an jede Reader-DB-Instance gesendet.

Multi-AZ-DB-Cluster-Bereitstellungen verwenden eine halbsynchrone Replikation, bei der eine Bestätigung von mindestens einer Reader-DB-Instance erforderlich ist, damit eine Änderung festgeschrieben wird. Es ist keine Bestätigung dafür erforderlich, dass die Ereignisse vollständig ausgeführt wurden und ein Commit für alle Replikate ausgeführt wurde.

Reader-DB-Instances fungieren als automatische Failover-Ziele und dienen auch dem Leseverkehr, um den Lesedurchsatz der Anwendung zu erhöhen. Wenn bei Ihrer Writer-DB-Instance ein Ausfall auftritt, RDS verwaltet es den Failover zu einer der Reader-DB-Instances. RDStut dies auf der Grundlage der Reader-DB-Instance, die den letzten Änderungsdatensatz hat.

Das folgende Diagramm zeigt einen Multi-AZ-DB-Cluster.



Multi-AZ-DB-Cluster haben in der Regel eine geringere Schreiblatenz im Vergleich zu Multi-AZ-DB-Instance-Bereitstellungen. Auch schreibgeschützte Workloads dürfen auf Reader-DB-Instances ausgeführt werden. In der RDS Konsole werden die Availability Zone der Writer-DB-Instance und

die Availability Zones der Reader-DB-Instances angezeigt. Sie können auch den describe-db-clustersCLIBefehl oder die escribeDBClustersAPID-Operation verwenden, um diese Informationen zu finden.

> **Important**
>
> Um Replikationsfehler in RDS My SQL Multi-AZ DB-Clustern zu vermeiden, empfehlen wir dringend, dass alle Tabellen über einen Primärschlüssel verfügen.

Parametergruppen für Multi-AZ-DB-Cluster

In einem Multi-AZ-DB-Cluster fungiert eine DB-Cluster-Parametergruppe als Container für Engine-Konfigurationswerte, die auf jede DB-Instance im Multi-AZ-DB-Cluster angewendet werden.

In einem Multi-AZ-DB-Cluster wird eine DB-Parametergruppe auf die Standard-DB-Parametergruppe für die DB-Engine und die DB-Engine-Version festgelegt. Die Einstellungen in der Parametergruppe des DB-Clusters werden für alle DB-Instances im Cluster verwendet.

Informationen zu Parametergruppen finden Sie unter the section called “DB-Cluster-Parametergruppen”.

RDSProxy mit Multi-AZ-DB-Clustern

Sie können Amazon RDS Proxy verwenden, um einen Proxy für Ihre Multi-AZ-DB-Cluster zu erstellen. Durch die Verwendung von RDS Proxy können Ihre Anwendungen Datenbankverbindungen bündeln und gemeinsam nutzen, um ihre Skalierbarkeit zu verbessern. Jeder Proxy führt Verbindungsmultiplexing durch, was auch als Wiederverwendung von Verbindungen bezeichnet wird. Beim Multiplexing führt der RDS Proxy alle Operationen für eine Transaktion mithilfe einer zugrunde liegenden Datenbankverbindung aus. RDSProxy kann auch die Ausfallzeit für ein kleineres Versionsupgrade eines Multi-AZ-DB-Clusters auf eine Sekunde oder weniger reduzieren. Weitere Informationen zu den Vorteilen von RDS Proxy finden Sie unterAmazon-RDS-Proxy.

Um einen Proxy für einen Multi-AZ-DB-Cluster einzurichten, wählen Sie beim Erstellen des Clusters die Option Create an RDS Proxy aus. Anweisungen zum Erstellen und Verwalten von RDS Proxy-Endpunkten finden Sie unter. the section called “Arbeiten mit RDS Proxy-Endpunkten”

Replikatverzögerung und Multi-AZ-DB-Cluster

Die Replikatverzögerung ist der Zeitunterschied zwischen der neuesten Transaktion auf der Writer-DB-Instance und der zuletzt angewendeten Transaktion auf einer Reader-DB-Instance. Die CloudWatch Amazon-Metrik `ReplicaLag` repräsentiert diesen Zeitunterschied. Weitere Informationen zu CloudWatch Metriken finden Sie unterÜberwachung von Amazon RDS mit Amazon CloudWatch.

Obwohl Multi-AZ-DB-Cluster eine hohe Schreibleistung ermöglichen, kann eine Replikatverzögerung aufgrund der Art der Engine-basierten Replikation immer noch auftreten. Da jedes Failover zuerst die Replikatverzögerung auflösen muss, bevor es eine neue Writer-DB-Instance fördert, ist die Überwachung und Verwaltung dieser Replikatverzögerung eine Überlegung wert.

RDSBei My SQL Multi-AZ DB-Clustern hängt die Failover-Zeit von der Replikatverzögerung der beiden verbleibenden Reader-DB-Instances ab. Beide Reader-DB-Instances müssen nicht angewendete Transaktionen anwenden, bevor eine von ihnen auf die neue Writer-DB-Instance befördert wird.

RDSBei SQL Postgre-Multi-AZ-DB-Clustern hängt die Failover-Zeit von der niedrigsten Replikatverzögerung der beiden verbleibenden Reader-DB-Instances ab. Die Reader-DB-Instance mit der kleinsten Replikatverzögerung muss nicht angewendete Transaktionen anwenden, bevor sie auf die neue Writer-DB-Instance befördert wird.

Ein Tutorial, das Ihnen zeigt, wie Sie einen CloudWatch Alarm auslösen, wenn die Replikatverzögerung einen bestimmten Zeitraum überschreitet, finden Sie unter. Tutorial: Erstellen eines CloudWatch Amazon-Alarms für Multi-AZ-DB-Cluster-Replikatverzögerungen für Amazon RDS

Häufige Ursachen für Replikatverzögerung

Im Allgemeinen tritt eine Replikatverzögerung auf, wenn die Write-Workload zu hoch ist, als dass die Reader-DB-Instances die Transaktionen effizient anwenden könnten. Verschiedene Workloads können eine vorübergehende oder kontinuierliche Replikatverzögerung verursachen. Einige gängige Beispiele:

- Hohe Write-Parallelität oder starke Batch-Aktualisierung auf der Writer-DB-Instance, wodurch der Anwendungsprozess auf den Reader-DB-Instances zurückbleibt.
- Starke Read-Workload, die Ressourcen auf einer oder mehreren Reader-DB-Instances verwendet. Das Ausführen langsamer oder großer Abfragen kann sich auf den Anwendungsprozess auswirken und die Replikatverzögerung verursachen.

- Transaktionen, die große Datenmengen oder DDL Anweisungen ändern, können manchmal zu einer vorübergehenden Erhöhung der Replikatverzögerung führen, da die Datenbank die Commit-Reihenfolge beibehalten muss.

Minderung der Replikatverzögerung

Bei Multi-AZ-DB-Clustern RDS für My SQL und RDS für Postgre können Sie die Replikatverzögerung verringernSQL, indem Sie die Belastung Ihrer Writer-DB-Instance reduzieren. Sie können auch die Flusssteuerung verwenden, um Replikatverzögerung zu reduzieren. Die Flusssteuerung funktioniert, indem Schreibvorgänge auf der Writer-DB-Instance gedrosselt werden, wodurch sichergestellt wird, dass die Replikatverzögerung nicht unbegrenzt weiter zunimmt. Die Schreibdrosselung wird erreicht, indem am Ende einer Transaktion eine Verzögerung hinzugefügt wird, wodurch der Schreibdurchsatz auf der Writer-DB-Instance verringert wird. Obwohl die Flusskontrolle nicht garantiert, Verzögerungen zu verhindern, kann sie dazu beitragen, die allgemeine Verzögerung bei vielen Workloads zu reduzieren. Die folgenden Abschnitte enthalten Informationen zur Verwendung der Flusskontrolle mit RDS für My SQL und für Postgre. RDS SQL

Verringerung der Replikatverzögerung durch Flusskontrolle für for My RDS SQL

Wenn Sie RDS für My SQL Multi-AZ DB-Cluster verwenden, ist die Flusskontrolle standardmäßig mithilfe des dynamischen Parameters aktiviert. `rpl_semi_sync_master_target_apply_lag` Dieser Parameter gibt die Obergrenze an, die für die Replikatverzögerung gewünscht wird. Wenn sich die Replikatverzögerung diesem konfigurierten Grenzwert nähert, drosselt die Flusssteuerung die Schreibtransaktionen auf der Writer-DB-Instance, um zu versuchen, die Replikatverzögerung unter den angegebenen Wert zu halten. In einigen Fällen kann die Replikatverzögerung den angegebenen Grenzwert überschreiten. Standardmäßig ist dieser Parameter auf 120 Sekunden eingestellt. Um die Flusskontrolle auszuschalten, setzen Sie diesen Parameter auf seinen Maximalwert von 86.400 Sekunden (ein Tag).

Um die aktuelle Verzögerung anzuzeigen, die von der Flusssteuerung injiziert wird, zeigen Sie den Parameter `Rpl_semi_sync_master_flow_control_current_delay` an, indem Sie die folgende Abfrage ausführen.

```
SHOW GLOBAL STATUS like '%flow_control%';
```

Ihre Ausgabe sollte in etwa wie folgt aussehen.

```
+------------------------------------------------+-------+
| Variable_name                                  | Value |
```

```
+-----------------------------------------------+-------+
| Rpl_semi_sync_master_flow_control_current_delay | 2010  |
+-----------------------------------------------+-------+
1 row in set (0.00 sec)
```

> **Note**
>
> Die Verzögerung wird in Mikrosekunden angezeigt.

Wenn Sie Performance Insights für einen RDS for My SQL Multi-AZ-DB-Cluster aktiviert haben, können Sie das Warteereignis überwachen, das einer SQL Aussage entspricht, die angibt, dass die Abfragen durch eine Flusskontrolle verzögert wurden. Wenn eine Verzögerung durch eine Ablaufsteuerung eingeführt wurde, können Sie das Warteereignis, das der SQL Anweisung `/wait/synch/cond/semisync/semi_sync_flow_control_delay_cond` entspricht, im Performance Insights Insights-Dashboard anzeigen. Stellen Sie sicher, dass das Leistungsschema aktiviert ist, um diese Metriken anzuzeigen. Weitere Informationen zu Performance Insights finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.

Minderung von Replikatverzögerungen mit Flusssteuerung für Postgre RDS SQL

Wenn Sie SQL Multi-AZ-DB-Cluster RDS für Postgre verwenden, wird die Flusskontrolle als Erweiterung bereitgestellt. Sie aktiviert einen Hintergrund-Worker für alle DB-Instances im DB-Cluster. Standardmäßig kommunizieren die Hintergrund-Worker auf den Reader-DB-Instances die aktuelle Replikatverzögerung mit dem Hintergrund-Worker auf der Writer-DB-Instance. Wenn die Verzögerung bei einer Reader-DB-Instance zwei Minuten überschreitet, fügt der Hintergrund-Worker der Writer-DB-Instance am Ende einer Transaktion eine Verzögerung hinzu. Um den Verzögerungsschwellenwert zu steuern, verwenden Sie den Parameter `flow_control.target_standby_apply_lag`.

Wenn eine Ablaufsteuerung einen SQL Postgre-Prozess drosselt, weist das `Extension` Warteereignis in `pg_stat_activity` und Performance Insights darauf hin. Die Funktion `get_flow_control_stats` zeigt Details darüber an, wie viel Verzögerung gerade hinzugefügt wird.

Die Flusssteuerung kann den meisten Workloads zur Online-Transaktionsverarbeitung (OLTP) zugute kommen, bei denen es zu kurzen, aber sehr gleichzeitigen Transaktionen kommt. Wenn die Verzögerung durch lang andauernde Transaktionen wie Batchvorgänge verursacht wird, bietet die Flusskontrolle keinen so starken Vorteil.

Sie können die Flusskontrolle ausschalten, indem Sie die Erweiterung aus `shared_preload_libraries` entfernen und Ihre DB-Instance neu starten.

Snapshots von Multi-AZ-DB-Clustern

Amazon RDS erstellt und speichert automatische Backups Ihres Multi-AZ-DB-Clusters während des konfigurierten Backup-Fensters. RDSerstellt einen Speichervolume-Snapshot Ihres DB-Clusters und sichert dabei den gesamten Cluster und nicht nur einzelne Instances.

Sie können auch manuelle Backups Ihres Multi-AZ-DB-Clusters erstellen. Für sehr langfristige Backups sollten Sie erwägen, die Snapshot-Daten nach Amazon S3 zu exportieren. Weitere Informationen finden Sie unter the section called “Erstellen eines Multi-AZ-DB-Cluster-Snapshots”.

Sie können einen Multi-AZ-DB-Cluster auf einen bestimmten Zeitpunkt wiederherstellen, wodurch ein neuer Multi-AZ-DB-Cluster erstellt wird. Detaillierte Anweisungen finden Sie unter the section called “Wiederherstellen eines Multi-AZ-DB-Clusters zu einer bestimmten Zeit”.

Alternativ können Sie einen Multi-AZ-DB-Cluster-Snapshot in einer Single-AZ-Bereitstellung oder einer Multi-AZ-DB-Instance-Bereitstellung wiederherstellen. Detaillierte Anweisungen finden Sie unter the section called “Wiederherstellen über einen Snapshot eines Multi-AZ-DB-Clusters in einer DB-Instance”.

Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen

Ein Multi-AZ-DB-Cluster verfügt über eine Writer-DB-Instance und zwei Reader-DB-Instances in drei separaten Availability Zones. Multi-AZ-DB-Cluster bieten hohe Verfügbarkeit, erhöhte Kapazität für Lese-Workloads und eine geringere Latenz im Vergleich zu Multi-AZ-Bereitstellungen. Weitere Informationen zu Multi-AZ-DB-Clustern finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS.

> **Note**
>
> Multi-AZ-DB-Cluster werden nur für die MySQL- und PostgreSQL-DB-Engines unterstützt.

Voraussetzungen für DB-Cluster

> **Important**
>
> Bevor Sie einen Multi-AZ DB-Cluster erstellen können, müssen Sie die Aufgaben unter Einrichtung Ihrer Amazon RDS-Umgebung abschließen.

Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie einen Multi-AZ DB-Cluster erstellen.

Themen

- Netzwerk für den DB-Cluster konfigurieren
- Zusätzliche Voraussetzungen

Netzwerk für den DB-Cluster konfigurieren

Sie können eine DB-Instance nur in einer Virtual Private Cloud (VPC) erstellen, die auf dem Amazon-VPC-Service basiert. Es muss sich in einer befinden AWS-Region , die mindestens drei Availability Zones hat. Die DB-Subnetzgruppe, die Sie für das DB-Cluster wählen, muss mindestens drei Availability Zones abdecken. Diese Konfiguration stellt sicher, dass sich jede DB-Instance im DB-Cluster in einer anderen Availability Zone befindet.

Um die Konnektivität zwischen Ihrem neuen DB-Cluster und einer EC2 Amazon-Instance in derselben VPC einzurichten, tun Sie dies, wenn Sie den DB-Cluster erstellen. Um von anderen

Ressourcen als EC2 Instances in derselben VPC aus eine Verbindung zu Ihrem DB-Cluster herzustellen, konfigurieren Sie die Netzwerkverbindungen manuell.

Themen

- Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instanz EC2
- Manuelles Konfigurieren des Netzwerks

Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instanz EC2

Wenn Sie einen Multi-AZ-DB-Cluster erstellen, können Sie den verwenden, AWS Management Console um die Konnektivität zwischen einer EC2 Instance und dem neuen DB-Cluster einzurichten. In diesem Fall konfiguriert RDS Ihre VPC- und Netzwerkeinstellungen automatisch. Der DB-Cluster wird in derselben VPC wie die Instance erstellt, sodass die EC2 EC2 Instance auf den DB-Cluster zugreifen kann.

Die folgenden Anforderungen gelten für die Verbindung einer EC2 Instance mit dem DB-Cluster:

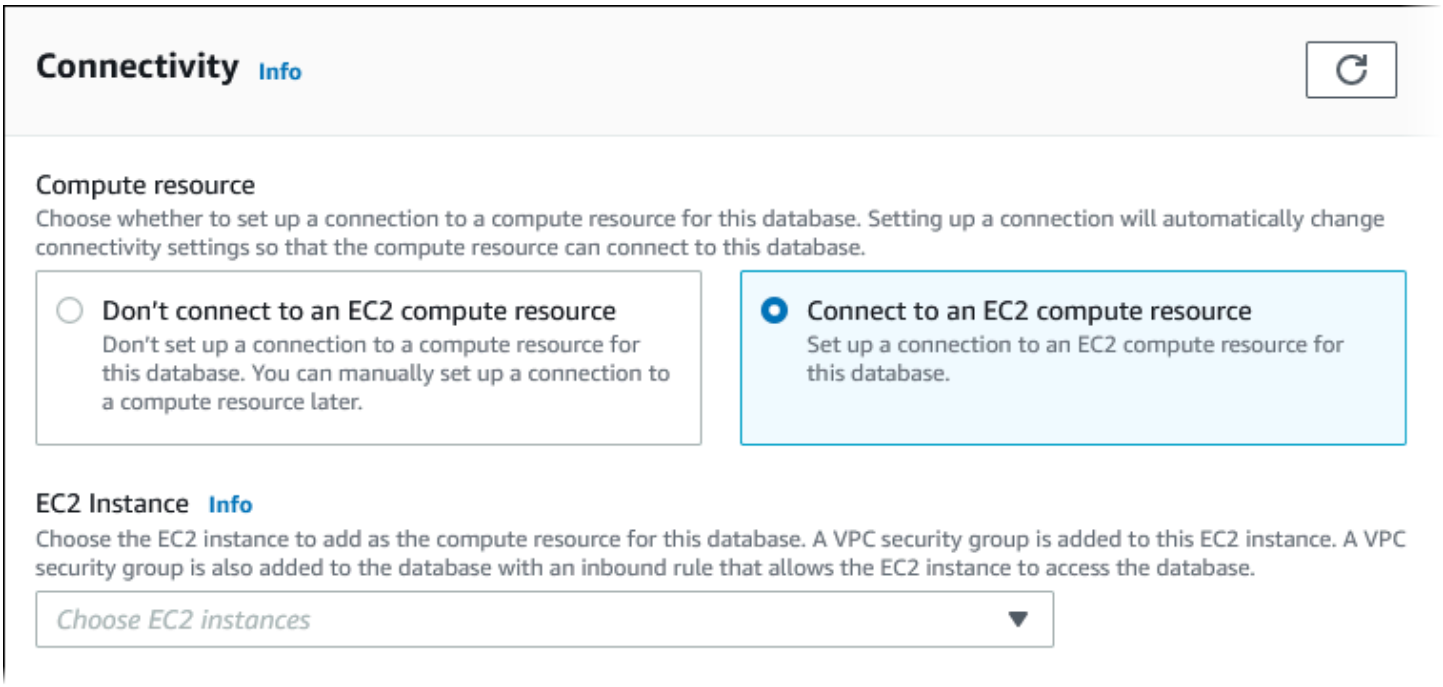
- Die EC2 Instance muss in der vorhanden sein, AWS-Region bevor Sie den DB-Cluster erstellen.

  Wenn im keine EC2 Instanzen vorhanden sind AWS-Region, bietet die Konsole einen Link zum Erstellen einer Instanz.
- Der Benutzer, der den DB-Cluster erstellt, muss über Berechtigungen zum Ausführen der folgenden Vorgänge verfügen:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`

- `ec2:RevokeSecurityGroupEgress`

Mit dieser Option wird ein privater DB-Cluster erstellt. Der DB-Cluster verwendet eine DB-Subnetzgruppe mit ausschließlich privaten Subnetzen, um den Zugriff auf Ressourcen innerhalb der VPC einzuschränken.

Um eine EC2 Instance Connect dem DB-Cluster zu verbinden, wählen Sie im Abschnitt Konnektivität auf der Seite Datenbank erstellen die Option Mit einer EC2 Rechenressource verbinden aus.



Wenn Sie Mit einer EC2 Rechenressource Connect wählen, legt RDS die folgenden Optionen automatisch fest. Sie können diese Einstellungen nur ändern, wenn Sie sich dafür entscheiden, keine Konnektivität mit einer EC2 Instance einzurichten, indem Sie „Keine Verbindung zu einer EC2 Rechenressource herstellen“ wählen.

Konsolenoption	Automatische Einstellung
Virtual Private Cloud (VPC)	RDS setzt die VPC auf die mit der EC2 Instance verknüpfte.
DB subnet group (DB-Subnetzgruppe)	RDS benötigt eine DB-Subnetzgruppe mit einem privaten Subnetz in derselben Availability Zone wie die Instance. EC2 Wenn eine DB-Subnetzgruppe vorhanden ist, die diese Anforderung erfüllt, dann verwendet RDS die vorhandene DB-

Konsolenoption	Automatische Einstellung
	Subnetzgruppe. Standardmäßig ist diese Option auf Automatic setup (Automatische Einrichtung) eingestellt. Wenn Sie Automatic setup (Automatische Einrichtung) auswählen und es keine DB-Subnetzgruppe gibt, die diese Anforderung erfüllt, wird die folgende Aktion ausgeführt. RDS verwendet drei verfügbare private Subnetze in drei Availabil ity Zones, wobei eine der Availability Zones mit der Instance identisch ist. EC2 Wenn kein privates Subnetz in einer Availabil ity Zone verfügbar ist, erstellt RDS ein privates Subnetz in der Availability Zone. Anschließend erstellt RDS die DB-Subnet zgruppe. Wenn ein privates Subnetz verfügbar ist, verwendet RDS die zugehörige Routing-Tabelle und fügt alle Subnetze, die es erstellt, dieser Routing-Tabelle hinzu. Wenn kein privates Subnetz verfügbar ist, erstellt RDS eine Routing-Tabelle ohne Internet-Gateway-Zugriff und fügt die erstellten Subnetze der Routing-Tabelle hinzu. Mit RDS können Sie auch vorhandene DB-Subnetzgruppen verwenden. Wählen Sie Choose existing (Vorhandene wählen) aus, wenn Sie eine vorhandene DB-Subnetzgruppe Ihrer Wahl verwenden möchten.
Öffentlicher Zugriff	RDS wählt Nein, so dass der DB-Cluster nicht öffentlich zugänglich ist. Aus Sicherheitsgründen ist es eine bewährte Methode, die Datenbank privat zu halten und sicherzustellen, dass sie nicht über das Internet zugänglich ist.

Konsolenoption	Automatische Einstellung
VPC-Sicherheitsgruppe (Firewall)	RDS erstellt eine neue Sicherheitsgruppe, die mit dem DB-Cluster verknüpft ist. Die Sicherheitsgruppe heißt `rds-ec2-`*n*, wobei *n* eine Zahl ist. Diese Sicherheitsgruppe enthält eine Regel für eingehenden Datenverkehr mit der EC2 VPC-Siche rheitsgruppe (Firewall) als Quelle. Diese Sicherheitsgruppe, die dem DB-Cluster zugeordnet ist, ermöglicht der EC2 Instance den Zugriff auf den DB-Cluster. RDS erstellt außerdem eine neue Sicherheitsgruppe, die der EC2 Instance zugeordnet ist. Die Sicherheitsgruppe heißt `ec2-rds-`*n*, wobei *n* eine Zahl ist. Diese Sicherheitsgruppe enthält eine ausgehende Regel mit der VPC-Sicherheitsgruppe des DB-Clusters als Quelle. Diese Sicherheitsgruppe ermöglicht es der EC2 Instance, Datenverkehr an den DB-Cluster zu senden. Sie können eine weitere neue Sicherheitsgruppe hinzufügen, indem Sie Neu erstellen wählen und den Namen der neuen Sicherheitsgruppe eingeben. Sie können vorhandene Sicherheitsgruppen hinzufügen, indem Sie Bestehende auswählen und Sicherheitsgruppen auswählen, die hinzugefügt werden sollen.
Availability Zone	RDS wählt die Availability Zone der EC2 Instance für eine DB-Instance in der Multi-AZ-DB-Cluster-Bereitstellung. RDS wählt nach dem Zufallsprinzip eine andere Availability Zone für beide anderen DB-Instances aus. Die Writer-DB-Instance wird in derselben Availability Zone wie die EC2 Instance erstellt. Es besteht die Möglichkeit von Kosten der Availability Zone, wenn ein Failover auftritt und sich die Writer-DB-Instance in einer anderen Availability Zone befindet.

Weitere Informationen zu diesen Einstellungen finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.

Wenn Sie diese Einstellungen ändern, nachdem der DB-Cluster erstellt wurde, können sich die Änderungen auf die Verbindung zwischen der EC2 Instance und dem DB-Cluster auswirken.

Manuelles Konfigurieren des Netzwerks

Um von anderen Ressourcen als EC2 Instances in derselben VPC aus eine Verbindung zu Ihrem DB-Cluster herzustellen, konfigurieren Sie die Netzwerkverbindungen manuell. Wenn Sie den verwenden AWS Management Console , um Ihren Multi-AZ-DB-Cluster zu erstellen, können Sie Amazon RDS automatisch eine VPC für Sie erstellen lassen. Sie können auch eine bestehende VPC verwenden oder eine neue VPC für Ihren Multi-AZ-DB-Cluster erstellen. Ihre VPC muss mindestens ein Subnetz in jeder von mindestens drei Availability Zones haben, damit Sie sie mit einem Multi-AZ-DB-Cluster verwenden können. Informationen dazu finden Sie unter VPCs. Amazon VPC und Amazon RDS

Wenn Sie keine Standard-VPC haben oder keine VPC erstellt haben und die Konsole nicht verwenden möchten, gehen Sie wie folgt vor:

- Erstellen Sie eine VPC mit mindestens einem Subnetz in jeder der mindestens drei Availability Zones in der AWS Region, in der Sie Ihren DB-Cluster bereitstellen möchten. Weitere Informationen finden Sie unter Arbeiten mit einer DB-Instance in einer VPC.
- Legen Sie eine VPC-Sicherheitsgruppe fest, die Verbindungen mit Ihrem DB-Cluster autorisiert. Weitere Informationen erhalten Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe und Zugriffskontrolle mit Sicherheitsgruppen.
- Legen Sie eine RDS-DB-Subnetzgruppe fest, die mindestens drei Subnetze in der VPC definiert, die vom Multi-AZ-DB-Cluster verwendet werden können. Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.

Informationen zu Einschränkungen für Multi-AZ-DB-Cluster finden Sie unter Einschränkungen von Multi-AZ-DB-Clustern für Amazon RDS.

Wenn Sie eine Verbindung mit einer Ressource herstellen möchten, die sich nicht in derselben VPC wie der Multi-AZ-DB-Cluster befindet, sehen Sie sich die entsprechenden Szenarien unter Szenarien für den Zugriff auf einen in einem VPC an.

Zusätzliche Voraussetzungen

Bevor Sie Ihren Multi-AZ-DB-Cluster erstellen, sollten Sie die folgenden zusätzlichen Voraussetzungen berücksichtigen:

- Wenn Sie die Konfigurationsparameter für Ihren DB-Cluster anpassen möchten, müssen Sie eine DB-Cluster-Parametergruppe mit den erforderlichen Parametereinstellungen festlegen. Informationen über das Erstellen oder Ändern einer DB-Cluster-Parametergruppe finden Sie unter Parametergruppen für Multi-AZ-DB-Cluster.
- Ermitteln Sie die TCP/IP Portnummer, die Sie für Ihren DB-Cluster angeben möchten. Die Firewalls einiger Unternehmen blockieren Verbindungen zu diesen Standard-Ports. Wenn die Firewall Ihres Unternehmens den Standard-Port blockiert, wählen Sie einen anderen Port für Ihr DB-Cluster aus. Alle DB-Instances in einem DB-Cluster verwenden denselben Port.
- Wenn die Haupt-Engine-Version für Ihre Datenbank das Ende des Standard-Supports für RDS erreicht hat, müssen Sie die CLI-Option Extended Support oder den RDS-API-Parameter verwenden. Weitere Informationen finden Sie unter RDS Extended Support unterEinstellungen zum Erstellen von Multi-AZ-DB-Clustern.

Erstellen eines DB-Clusters

Sie können einen Multi-AZ-DB-Cluster mithilfe der AWS Management Console AWS CLI, der oder der RDS-API erstellen.

Konsole

Sie können einen Multi-AZ-DB-Cluster erstellen, indem Sie Multi-AZ-DB-Cluster im Abschnitt Verfügbarkeit und Haltbarkeit auswählen.

Erstellen Sie einen Multi-AZ-DB-Cluster mithilfe der Konsole wie folgt:

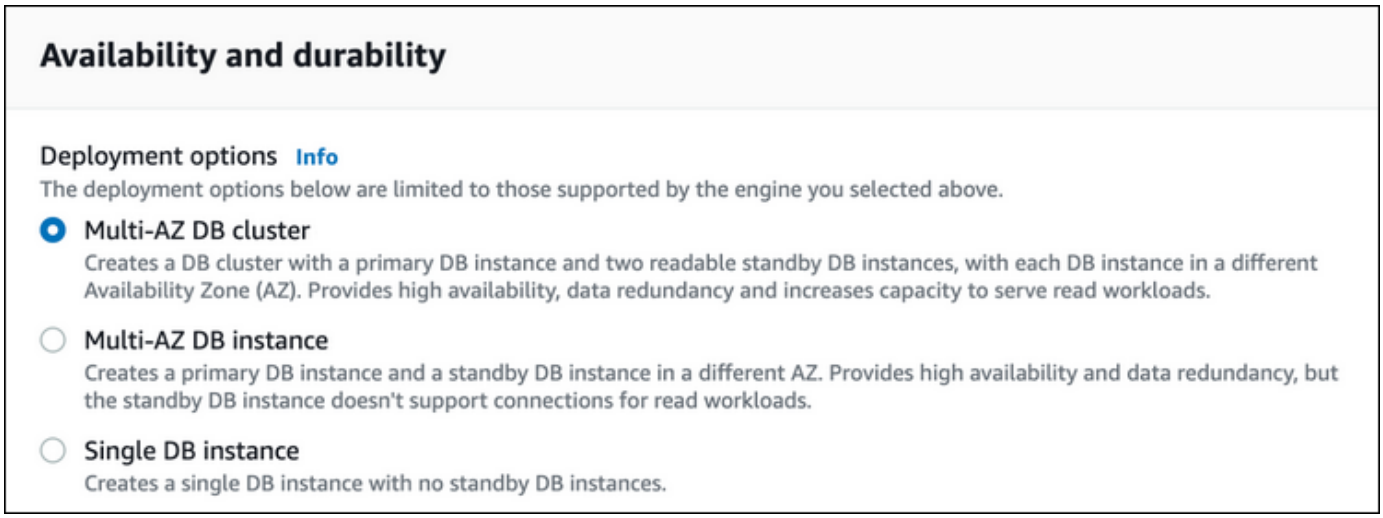
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke des den aus AWS Management Console, AWS-Region in dem Sie den DB-Cluster erstellen möchten.

   Informationen zu den AWS-Regionen , die Multi-AZ-DB-Cluster unterstützen, finden Sie unter. Einschränkungen von Multi-AZ-DB-Clustern für Amazon RDS
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.

   Stellen Sie zum Erstellen eines Multi-AZ-DB-Clusters sicher, dass Standarderstellung ausgewählt ist und Einfache Erstellung nicht.

5. Wählen Sie unter Engine-Typ MySQL oder PostgreSQL aus.
6. Wählen Sie unter Version die DB-Engine-Version aus.

   Weitere Informationen zu DB-Engine-Versionen, die Multi-AZ-DB-Cluster unterstützen, finden Sie unter Einschränkungen von Multi-AZ-DB-Clustern für Amazon RDS.
7. Wählen Sie unter Vorlagen die entsprechende Vorlage für Ihre Bereitstellung aus.
8. Wählen Sie unter Verfügbarkeit und Haltbarkeit die Option Multi-AZ-DB-Cluster aus.



9. Geben Sie unter DB-Cluster-Kennung die Kennung für Ihren DB-Cluster ein.
10. Geben Sie unter Haupt-Benutzername Ihren Haupt-Benutzernamen ein oder behalten Sie die Standardeinstellung bei.
11. Geben Sie Ihr Haupt-Passwort ein:

    a. Öffnen Sie im Abschnitt Settings (Einstellungen) die Option Credential Settings (Einstellungen zu Anmeldeinformationen).

    b. Wenn Sie ein Passwort angeben möchten, deaktivieren Sie das Kontrollkästchen Passwort automatisch generieren, wenn es aktiviert ist.

    c. (Optional) Ändern Sie den Wert des Haupt-Benutzernamens.

    d. Geben Sie das gleiche Passwort in Haupt-Passwort und Passwort bestätigen ein.
12. Wählen Sie unter DB-Instance-Klasse eine DB-Instance-Klasse aus. Eine Liste der unterstützten DB-Instance-Klassen, finden Sie unter the section called “Verfügbarkeit der Instance-Klassen für Multi-AZ-DB-Cluster”.
13. (Optional) Richten Sie eine Verbindung zu einer Rechenressource für diesen DB-Cluster ein.

Sie können die Konnektivität zwischen einer EC2 Amazon-Instance und dem neuen DB-Cluster während der DB-Cluster-Erstellung konfigurieren. Weitere Informationen finden Sie unter Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instanz EC2 .

14. Wenn Sie im Abschnitt Konnektivität unter VPC-Sicherheitsgruppe (Firewall) die Option Neu erstellen auswählen, wird eine VPC-Sicherheitsgruppe mit einer Regel für eingehenden Datenverkehr erstellt, die es der IP-Adresse Ihres lokalen Computers ermöglicht, auf die Datenbank zuzugreifen.
15. In den übrigen Abschnitten geben Sie die Einstellungen für Ihren DB-Cluster an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.
16. Wählen Sie Create database (Datenbank erstellen) aus.

    Wenn Sie ein automatisch generiertes Passwort verwenden, wird auf der Seite Databases (Datenbanken) die Schaltfläche View credential details (Details zu Anmeldeinformationen anzeigen) angezeigt.

    Um den Hauptbenutzernamen und das Passwort für den DB-Cluster anzuzeigen, wählen Sie Anmeldeinformationen anzeigen.

    Verwenden Sie den angezeigten Benutzernamen und das angezeigte Passwort, um sich als Hauptbenutzer mit dem DB-Cluster zu verbinden.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen.

17. Wählen Sie in Databases (Datenbanken) den Namen des neuen DB-Clusters aus.

In der RDS-Konsole werden die Details des neuen DB-Clusters angezeigt. Der DB-Cluster hat den Status Wird erstellt, bis der DB-Cluster erstellt und einsatzbereit ist. Wenn der Status zu Available (Verfügbar) geändert wird, können Sie eine Verbindung zum DB-Cluster herstellen. Je nach Klasse und Speicher des DB-Clusters kann es einige Minuten dauern, bis der neue DB-Cluster verfügbar ist.

AWS CLI

Bevor Sie mit dem einen Multi-AZ-DB-Cluster erstellen AWS CLI, stellen Sie sicher, dass die erforderlichen Voraussetzungen erfüllt sind. Dazu gehören das Erstellen einer VPC und einer RDS-DB-Subnetzgruppe. Weitere Informationen finden Sie unter Voraussetzungen für DB-Cluster.

Rufen Sie den Befehl auf, um mit dem einen Multi-AZ-DB-Cluster zu erstellen. AWS CLIcreate-db-cluster Geben Sie `--db-cluster-identifier` an. Geben Sie für die `--engine`-Option entweder `mysql` oder `postgres` an.

Informationen zu den jeweiligen Optionen finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.

Informationen zu den AWS-Regionen DB-Engines und DB-Engine-Versionen, die Multi-AZ-DB-Cluster unterstützen, finden Sie unter. Einschränkungen von Multi-AZ-DB-Clustern für Amazon RDS

Der Befehl `create-db-cluster` erstellt die Writer-DB-Instance für Ihren DB-Cluster und zwei Reader-DB-Instances. Jede DB-Instance befindet sich in einer anderen Availability Zone.

Der folgende Befehl erstellt beispielsweise einen MySQL 8.0 Multi-AZ-DB-Cluster namens `mysql-multi-az-db-cluster`.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.32  \
   --master-username admin \
   --manage-master-user-password  \
   --port 3306 \
   --backup-retention-period 1  \
   --db-subnet-group-name default \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.m5d.xlarge
```

Windows:

```
aws rds create-db-cluster ^
```

```
    --db-cluster-identifier mysql-multi-az-db-cluster ^
    --engine mysql ^
    --engine-version 8.0.32 ^
    --manage-master-user-password ^
    --master-username admin ^
    --port 3306 ^
    --backup-retention-period 1 ^
    --db-subnet-group-name default ^
    --allocated-storage 4000 ^
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

Der folgende Befehl erstellt einen PostgreSQL 13.4 Multi-AZ DB-Cluster namens `postgresql-multi-az-db-cluster`.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-db-cluster \
    --db-cluster-identifier postgresql-multi-az-db-cluster \
    --engine postgres \
    --engine-version 13.4 \
    --manage-master-user-password \
    --master-username postgres \
    --port 5432 \
    --backup-retention-period 1  \
    --db-subnet-group-name default \
    --allocated-storage 4000 \
    --storage-type io1 \
    --iops 10000 \
    --db-cluster-instance-class db.m5d.xlarge
```

Windows:

```
aws rds create-db-cluster ^
    --db-cluster-identifier postgresql-multi-az-db-cluster ^
    --engine postgres ^
    --engine-version 13.4 ^
    --manage-master-user-password ^
    --master-username postgres ^
    --port 5432 ^
```

```
    --backup-retention-period 1 ^
    --db-subnet-group-name default ^
    --allocated-storage 4000 ^
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

RDS-API

Bevor Sie ein Multi-AZ-DB-Cluster über die RDS-API erstellen können, müssen Sie die Voraussetzungen erfüllen, wie das Erstellen einer VPC und einer RDS-DB-Subnetzgruppe. Weitere Informationen finden Sie unter Voraussetzungen für DB-Cluster.

Rufen Sie den DBCluster Vorgang Create auf, um mithilfe der RDS-API einen Multi-AZ-DB-Cluster zu erstellen. Geben Sie `DBClusterIdentifier` an. Geben Sie für den `Engine`-Parameter entweder `mysql` oder `postgresql` an.

Informationen zu den jeweiligen Optionen finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.

Die Operation `CreateDBCluster` erstellt die Writer-DB-Instance für Ihren DB-Cluster und zwei Reader-DB-Instances. Jede DB-Instance befindet sich in einer anderen Availability Zone.

Einstellungen zum Erstellen von Multi-AZ-DB-Clustern

Ausführliche Informationen zu den Einstellungen, die Sie beim Erstellen eines Multi-AZ-DB-Clusters auswählen, finden Sie in der folgenden Tabelle. Weitere Informationen zu den AWS CLI Optionen finden Sie unter create-db-cluster. Weitere Informationen zu den RDS-API-Parametern finden Sie unter Create DBCluster.

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Allocated storage	Die für jede DB-Instance in Ihrem DB-Cluster zuzuweisende Speichermenge (in Gibibyte). Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.	CLI-Option:
		`--allocated-storage`
		API-Parameter:
		`AllocatedStorage`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Automatische Nebenversions-Updates	Aktivieren Sie das automatische Upgrade der Nebenversion, damit Ihr DB-Cluster automatisch Upgrades der bevorzugten Nebenversion der DB-Engine erhält, wenn diese verfügbar sind. Amazon RDS führt im Wartungsfenster automatische Nebenversionenupgrades durch.	CLI-Option:
		`--auto-minor-version-upgrade`
		`--no-auto-minor-version-upgrade`
		API-Parameter:
		`AutoMinorVersionUpgrade`

Aufbewahrungszeitraum für Backups	Die Anzahl der Tage, für die automatischen Backups der DB-Cluster aufbewahrt werden sollen. Bei einem Multi-AZ-DB-Cluster muss dieser Wert auf **1** oder höher gesetzt werden.
	Weitere Informationen finden Sie unter Einführung in Backups.
		CLI-Option:
		`--backup-retention-period`
		API-Parameter:
		`BackupRetentionPeriod`

Backup window	Der Zeitraum, in dem Amazon RDS automatisch ein Backup der DB-Cluster erstellt. Wenn Sie keine bestimmte Zeit haben, zu der Sie Ihre Datenbank sichern möchten, verwenden Sie den Standardwert No Preference (Keine Präferenz).
	Weitere Informationen finden Sie unter Einführung in Backups.
		CLI-Option:
		`--preferred-backup-window`
		API-Parameter:
		`PreferredBackupWindow`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Zertifizierungsstelle	Die Zertifizierungsstelle (CA) für das vom DB-Cluster verwendete Serverzertifikat.
	
	Weitere Informationen finden Sie unter .
		CLI-Option:
		
		`--ca-certificate-identifier`
		
		RDS-API-Parameter:
		
		`CACertificateIdentifier`

Tags zu Snapshots kopieren	Diese Option kopiert alle DB-Cluster-Tags in einen DB-Snapshot, wenn Sie einen Snapshot erstellen.
	
	Weitere Informationen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.
		CLI-Option:
		
		`-copy-tags-to-snapshot`
		
		`-no-copy-tags-to-snapshot`
		
		RDS-API-Parameter:
		
		`CopyTagsToSnapshot`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Datenbank -Authenti fizierung	Die Datenbankauthentifizierungs option, die Sie verwenden möchten.
	
	Wählen Sie Passwortauthentifi zierung aus, um Datenbank benutzer ausschließlich mit Datenbankpasswörtern zu authentif izieren.
	
	Wählen Sie Password and IAM DB authentication (Passwort- und IAM-DB-Authentifizierung) aus, um Datenbankbenutzer mit Datenbankpasswörtern und Benutzeranmeldeinformatione n über Benutzer und Rollen zu authentifizieren. Weitere Informati onen finden Sie unter IAM-Daten bankauthentifizierung für MariaDB, MySQL und PostgreSQL.
		CLI-Option:
		
		`--enable-iam-database-authe ntication`
		
		`--no-enable-iam-database-au thentication`
		
		RDS-API-Parameter:
		
		`EnableIAMDatabaseAuthentica tion`

Datenbankport	Der Port, über den Sie auf den DB-Cluster zugreifen wollen. Der Standardport wird angezeigt.
	
	Der Port kann nicht geändert werden, nachdem der DB-Cluster erstellt wurde.
	
	Die Firewalls einiger Unternehmen blockieren Verbindungen zu diesen Standard-Ports. Wenn die Firewall Ihres Unternehmens den Standard-Port blockiert, geben Sie einen anderen Port für Ihr DB-Cluster ein.
		CLI-Option:
		
		`--port`
		
		RDS-API-Parameter:
		
		`Port`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

DB-Cluster-Kennung	Der Name für Ihren DB-Cluster. Benennen Sie Ihre DB-Cluster auf die gleiche Weise wie Ihre lokalen Server. Ihre DB-Cluster-ID kann bis zu 63 alphanumerische Zeichen enthalten und muss für Ihr Konto in der von Ihnen ausgewählten AWS Region eindeutig sein.	CLI-Option:
		`--db-cluster-identifier`
		RDS-API-Parameter:
		`DBClusterIdentifier`

DB-Instance-Klasse	Die Rechen- und Arbeitsspeicherkapazität jeder DB-Instance im Multi-AZ-DB-Cluster, zum Beispiel `db.m5d.xlarge` .
	Wählen Sie möglichst eine DB-Instance-Klasse, die groß genug ist, um einen typischer Abfragesatz im Arbeitsspeicher halten zu können. Wenn Arbeitssätze im Arbeitsspeicher gehalten werden, kann das System das Schreiben auf die Festplatte vermeiden, was die Leistung verbessert.
	Eine Liste der unterstützten DB-Instance-Klassen, finden Sie unter the section called “Verfügbarkeit der Instance-Klassen für Multi-AZ-DB-Cluster”.
		CLI-Option:
		`--db-cluster-instance-class`
		RDS-API-Parameter:
		`DBClusterInstanceClass`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

DB-Cluste r-Paramet ergruppe	Die DB-Clusterparametergruppe, die Sie mit dem DB-Cluster verknüpfen möchten.
	
	Weitere Informationen finden Sie unter Parametergruppen für Multi-AZ-DB-Cluster.
		CLI-Option:
		
		`--db-cluster-parameter-grou p-name`
		
		RDS-API-Parameter:
		
		`DBClusterParameterGroupName`

DB-Engine-Version	Die Version der Datenbank-Engine, die Sie verwenden möchten.	CLI-Option:
		
		`--engine-version`
		
		RDS-API-Parameter:
		
		`EngineVersion`

DB-Cluste r-Paramet ergruppe	Die DB-Instance-Parametergruppe , die dem DB-Cluster zugeordnet werden soll. Weitere Informationen finden Sie unter Parametergruppen für Multi-AZ-DB-Cluster.	CLI-Option: `--db-cluster-parameter-grou p-name` RDS-API-Parameter: **`DBClusterParameterGroupName`**

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

DB subnet group (DB-Subnetzgruppe)	Die DB-Subnetzgruppe, die Sie für den DB-Cluster verwenden möchten.
	Wählen Sie Choose existing (Vorhandene wählen) aus, um eine vorhandene DB-Subnetzgruppe zu verwenden. Wählen Sie dann die erforderliche Subnetzgruppe aus der Dropdown-Liste Existing DB subnet groups (Vorhandene DB-Subnetzgruppen) aus.
	Wählen Sie Automatic setup (Automatische Einrichtung) aus, damit RDS eine kompatible DB-Subnetzgruppe auswählen kann. Wenn keine vorhanden ist, erstellt RDS eine neue Subnetzgruppe für Ihren Cluster.
	Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.
		CLI-Option:
		`--db-subnet-group-name`
		RDS-API-Parameter:
		`DBSubnetGroupName`

Löschschutz	Um zu verhindern, dass Ihr DB-Cluster gelöscht wird, können Sie die Option Löschschutz aktivieren aktivieren. Wenn Sie einen Produktions-DB-Cluster über die Konsole erstellen, ist der Löschschutz standardmäßig aktiviert.
	Weitere Informationen finden Sie unter Löschen einer DB-Instance.
		CLI-Option:
		`--deletion-protection`
		`--no-deletion-protection`
		RDS-API-Parameter:
		`DeletionProtection`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Verschlüsselung	Enable Encryption (Verschlüsselung aktivieren), um die Verschlüsselung im Ruhezustand für diesen DB-Cluster zu aktivieren.
	
	Die Verschlüsselung ist für Multi-AZ-DB-Cluster standardmäßig aktiviert.
	
	Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
		CLI-Optionen:
		
		`--kms-key-id`
		
		`--storage-encrypted`
		
		`--no-storage-encrypted`
		
		RDS-API-Parameter:
		
		`KmsKeyId`
		
		`StorageEncrypted`

Verbesserte Überwachung	Wählen Sie Enable enhanced monitoring (Erweiterte Überwachung aktivieren) aus, um die Erfassung von Metriken in Echtzeit für das Betriebssystem zu aktivieren, in dem Ihr DB-Cluster ausgeführt wird.
	
	Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).
		CLI-Optionen:
		
		`--monitoring-interval`
		
		`--monitoring-role-arn`
		
		RDS-API-Parameter:
		
		`MonitoringInterval`
		
		`MonitoringRoleArn`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Anfänglicher Datenbankname	Der Name für die Datenbank in Ihrem DB-Cluster. Wenn Sie keinen Namen angeben, erstellt Amazon RDS keine Datenbank im neuen DB-Cluster für MySQL. Es erstellt jedoch eine Datenbank im DB-Cluster für PostgreSQL. Der Name darf kein von der Datenbank-Engine reserviertes Wort sein. Abhängig von der DB-Engine gibt es weitere Einschränkungen.
	
	MySQL:
	
	• Er muss 1–64 alphanumerische Zeichen enthalten.
	
	PostgreSQL:
	
	• Er muss 1–63 alphanumerische Zeichen enthalten.
	• Er muss mit einem Buchstaben oder einem Unterstrich beginnen. Nachfolgende Zeichen können Groß-, Kleinbuchstaben oder Zahlen (0-9) sein.
	• Der anfängliche Datenbankname lautet `postgres`.
		CLI-Option:
		
		`--database-name`
		
		RDS-API-Parameter:
		
		`DatabaseName`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Protokollexporte	Die Typen von Datenbank-Protokoll dateien, die in Amazon CloudWatch Logs veröffentlicht werden sollen.
	
	Weitere Informationen finden Sie unter Datenbankprotokolle in Amazon CloudWatch Logs veröffentlichen.
		CLI-Option:
		
		`-enable-cloudwatch-logs-exports`
		
		RDS-API-Parameter:
		
		`EnableCloudwatchLogsExports`

Wartungsfenster	Das 30-Minuten-Fenster, in dem anstehende Änderungen an Ihrem DB-Cluster durchgeführt werden. Wählen Sie No Preference (Keine Präferenz) aus, wenn der Zeitraum nicht wichtig ist.
	
	Weitere Informationen finden Sie unter Amazon RDS-Wartungsfenster.
		CLI-Option:
		
		`--preferred-maintenance-window`
		
		RDS-API-Parameter:
		
		`PreferredMaintenanceWindow`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Hauptanmeldedaten verwalten in AWS Secrets Manager	Wählen Sie Master-Anmeldeinformationen verwalten in AWS Secrets Manager aus, um das Hauptbenutzerpasswort in Secrets Manager geheim zu verwalten.
	
	Wählen Sie optional einen KMS-Schlüssel zum Schutz des Secrets aus. Wählen Sie aus den KMS-Schlüsseln in Ihrem Konto oder geben Sie den Schlüssel eines anderen Kontos ein.
	
	Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.
		CLI-Option:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		RDS-API-Parameter:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`

Hauptpasswort	Das Passwort für das Masterbenutzerkonto.	CLI-Option:
		
		`--master-user-password`
		
		RDS-API-Parameter:
		
		`MasterUserPassword`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Master-Be nutzername	Der Name, den Sie als Masterben utzernamen für die Anmeldung beim DB-Cluster mit allen Datenbankrechten verwenden wollen.
	• Er kann 1–16 alphanumerische Zeichen und Unterstriche enthalten.
	• Das erste Zeichen muss ein Buchstabe sein.
	• Es darf kein von der Datenbank-Engine reserviertes Wort sein.
	
	Sie können den Master-Be nutzernamen nicht mehr ändern, nachdem der Multi-AZ-DB-Cluster erstellt wurde.
	
	Weitere Informationen zu Berechtig ungen, die dem Masterbenutzer gewährt werden, finden Sie unter Berechtigungen von Hauptbenu tzerkonten.
		CLI-Option:
		
		`--master-username`
		
		RDS-API-Parameter:
		
		`MasterUsername`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Performance Insights	Aktivieren Sie Performance Insights, um die Last Ihrer DB-Cluster zu überwachen, damit Sie Ihre Datenbankleistung analysieren und Fehler beheben können.
	
	Wählen Sie einen Aufbewahrungszeitraum, um festzulegen, wie viele Performance Insights-Daten aufbewahrt werden sollen. Die Aufbewahrungseinstellung ist Standard (7 Tage). Um Ihre Leistungsdaten länger aufzubewahren, geben Sie 1–24 Monate an. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.
	
	Wählen Sie einen KMS-Schlüssel aus, der zum Schutz des Schlüssels für die Verschlüsselung dieses Datenbankvolumes verwendet wird. Wählen Sie aus den KMS-Schlüsseln in Ihrem Konto oder geben Sie den Schlüssel eines anderen Kontos ein.
	
	Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.
		CLI-Optionen:
		
		`--enable-performance-insights`
		
		`--no-enable-performance-insights`
		
		`--performance-insights-retention-period`
		
		`--performance-insights-kms-key-id`
		
		RDS-API-Parameter:
		
		`EnablePerformanceInsights`
		
		`PerformanceInsightsRetentionPeriod`
		
		`PerformanceInsightsKMSKeyId`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Bereitgestellte IOPS	Die Menge von bereitgestellten IOPS (Ein-/Ausgabeoperationen pro Sekunde), die dem DB-Cluster anfänglich zugewiesen werden soll.	CLI-Option:
		`--iops`
		RDS-API-Parameter:
		`Iops`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Öffentlicher Zugriff	Publicly accessible (Öffentlich zugänglich), um dem DB-Cluste r eine öffentliche IP-Adresse zuzuweisen (was bedeutet, dass sie von außerhalb der VPC aus zugänglich ist). Damit der öffentlic he Zugriff für ein DB-Cluster möglich ist, muss sie sich auch in einem öffentlichen Subnetz der VPC befinden.
	
	Not publicly accessible (Nicht öffentlich zugänglich), um den DB-Cluster nur innerhalb der VPC zugänglich zu machen.
	
	Weitere Informationen finden Sie unter Ausblenden einer DB-Instance in einer VPC vor dem Internet.
	
	Um eine Verbindung zu einem DB-Cluster von außerhalb seiner VPC herzustellen, muss der DB-Cluster öffentlich zugänglich sein. Außerdem muss der Zugriff unter Verwendung der eingehenden Regeln der Sicherheitsgruppe dem DB-Cluster gewährt werden, und andere Anforderungen müssen erfüllt sein. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.
		CLI-Option:
		
		`--publicly-accessible`
		
		`--no-publicly-accessible`
		
		RDS-API-Parameter:
		
		`PubliclyAccessible`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter
	Wenn Ihr DB-Cluster nicht öffentlich zugänglich ist, können Sie eine AWS Site-to-Site VPN-Verbindung oder eine AWS Direct Connect Verbindung verwenden, um von einem privaten Netzwerk aus darauf zuzugreifen. Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.	

Erweiterter RDS-Support	Wählen Sie Enable RDS Extended Support aus, damit unterstützte Engine-Hauptversionen auch nach Ablauf des RDS-Standard-Supports weiter ausgeführt werden können.
	
	Wenn Sie einen DB-Cluster erstellen, verwendet Amazon RDS standardmäßig RDS Extended Support. Um die Erstellung eines neuen DB-Clusters nach dem Ende des Standard-Supports für RDS zu verhindern und Gebühren für RDS Extended Support zu vermeiden, deaktivieren Sie diese Einstellung. Für Ihre vorhandenen DB-Cluster fallen bis zum Startdatum der Preise für RDS Extended Support keine Gebühren an.
	
	Weitere Informationen finden Sie unter Erweiterter Support für Amazon RDS mit Amazon RDS.
		CLI-Option:
		
		`--engine-lifecycle-support`
		
		RDS-API-Parameter:
		
		`EngineLifecycleSupport`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter
Storage throughput (Speicherdurchsatz)	Der Wert für den Speicherdurchsatz für den DB-Cluster. Diese Einstellung ist nur sichtbar, wenn Sie General Purpose SSD (gp3) als Speichertyp wählen. Diese Einstellung ist nicht konfigurierbar und wird automatisch auf der Grundlage der von Ihnen angegebenen IOPS festgelegt. Weitere Informationen finden Sie unter GP3-Speicher (empfohlen).	Dieser Wert wird automatisch berechnet und hat keine CLI-Option.
RDS-Proxy	Wählen Sie Create an RDS Proxy (RDS-Proxy erstellen) aus, um einen Proxy für Ihren DB-Cluster zu erstellen. Amazon RDS erstellt automatisch eine IAM-Rolle und ein Secrets-Manager-Secret für den Proxy.	Nicht verfügbar beim Erstellen eines DB-Clusters

Speichertyp	Der Speichertyp für Ihren DB-Cluster.
	
	Es werden nur Allzweck-SSD-Speicher (gp3), bereitgestellte IOPS-Speicher (io1) und bereitgestellte IOPS-SSD-Speicher (io2) unterstützt.
	
	Weitere Informationen finden Sie unter Amazon RDS-Speichertypen.
		CLI-Option:
		
		`--storage-type`
		
		RDS-API-Parameter:
		
		`StorageType`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter
Virtual Private Cloud (VPC)	Eine VPC, die auf dem Amazon-VPC-Service basiert und mit diesem DB-Cluster verknüpft werden soll. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.	Für die CLI und API geben Sie die VPC-Sicherheitsgruppe IDs an.

VPC security group (firewall) (VPC-Sicherheitsgruppe (Firewall))	Die Sicherheitsgruppen, die dem DB-Cluster zugeordnet werden sollen.
	
	Weitere Informationen finden Sie unter Überblick über VPC-Sicherheitsgruppen.
		CLI-Option:
		
		`--vpc-security-group-ids`
		
		RDS-API-Parameter:
		
		`VpcSecurityGroupIds`

Einstellungen, die beim Erstellen von Multi-AZ-DB-Clustern nicht gelten

Die folgenden Einstellungen im AWS CLI Befehl `create-db-cluster`und im RDS-API-Vorgang gelten `CreateDBCluster`nicht für Multi-AZ-DB-Cluster.

Sie können diese Einstellungen auch für Multi-AZ-DB-Cluster in der Konsole nicht angeben.

AWS CLI Einstellung	RDS-API-Einstellung
`--availability-zones`	`AvailabilityZones`
`--backtrack-window`	`BacktrackWindow`
`--character-set-name`	`CharacterSetName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`

AWS CLI Einstellung	RDS-API-Einstellung
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--global-cluster-identifier`	`GlobalClusterIdentifier`
`--option-group-name`	`OptionGroupName`
`--pre-signed-url`	`PreSignedUrl`
`--replication-source-identifier`	`ReplicationSourceIdentifier`
`--scaling-configuration`	`ScalingConfiguration`

Verbindung zu einem Multi-AZ-DB-Cluster für Amazon herstellen RDS

Ein Multi-AZ-DB-Cluster verfügt über drei DB-Instances anstelle einer einzigen DB-Instance. Jede Verbindung wird von einer bestimmten DB-Instance verarbeitet. Wenn Sie eine Verbindung zu einem Multi-AZ-DB-Cluster herstellen, verweisen der von Ihnen angegebene Hostname und Port auf einen vollqualifizierten Domainnamen, der als Endpunkt bezeichnet wird. Der Multi-AZ-DB-Cluster verwendet den Endpunktmechanismus, um diese Verbindungen zu abstrahieren, so dass Sie nicht genau angeben müssen, zu welcher DB-Instance im DB-Cluster eine Verbindung hergestellt werden soll. Sie müssen daher für das Umleiten von Verbindungen nicht alle Hostnamen fest codieren oder Ihre eigene Logik schreiben, wenn einige DB-Instances nicht verfügbar sind.

Der Writer-Endpunkt stellt eine Verbindung zur Writer-DB-Instance des DB-Clusters her, der sowohl Lese- als auch Schreibvorgänge unterstützt. Der Leser-Endpunkt stellt eine Verbindung zu einer der beiden Reader-DB-Instances her, die nur Leseoperationen unterstützen.

Ihrem Anwendungsfall entsprechend können Sie mit Endpunkten jede Verbindung der entsprechenden DB-Instance oder Gruppe von DB-Instances zuordnen. Um beispielsweise DML AND-Anweisungen auszuführenDDL, können Sie eine Verbindung zu der DB-Instance herstellen, bei der es sich um die Writer-DB-Instance handelt. Um Abfragen durchzuführen, können Sie eine Verbindung zum Leser-Endpunkt herstellen, wobei der Multi-AZ-DB-Cluster automatisch Verbindungen zwischen den Reader-DB-Instances verwaltet. Zur Diagnose und Optimierung können Sie eine Verbindung mit einem spezifischen DB-Instance-Endpunkt herstellen, um die Details einer bestimmten DB-Instance zu untersuchen.

Weitere Information über das Verbinden mit der DB-Instance finden Sie unter Verbindung zu einer Amazon RDS DB-Instance herstellen.

Weitere Informationen zum Herstellen einer Verbindung zu Multi-AZ-DB-Clustern finden Sie in den folgenden Themen.

Themen

- Cluster-Endpunkte
- Reader-Endpunkte
- Instance-Endpunkte
- Verbindungen mit hoher Verfügbarkeit
- Verbindung zu Multi-AZ-DB-Clustern mit den AWS Treibern für Amazon herstellen RDS

Arten von Multi-AZ-DB-Cluster-Endpunkten

Ein Endpunkt wird durch einen eindeutigen Bezeichner dargestellt, der eine Hostadresse enthält. In einem Multi-AZ-DB-Cluster stehen die folgenden Endpunkt-Typen zur Verfügung:

Cluster-Endpunkt

Ein Cluster-Endpunkt (oder Writer-Endpunkt) für einen Multi-AZ-DB-Cluster stellt eine Verbindung mit der aktuellen Writer-DB-Instance für diesen DB-Cluster her. Dieser Endpunkt ist der einzige, der Schreiboperationen wie DDL DML AND-Anweisungen ausführen kann. Dieser Endpunkt kann auch Leseoperationen ausführen.

Jeder Multi-AZ-DB-Cluster verfügt über einen Cluster-Endpunkt und eine Writer-DB-Instance.

Sie verwenden den Cluster-Endpunkt für alle Schreibvorgänge auf dem DB-Cluster, einschließlich Einfügungen, Aktualisierungen, Löschungen und DDL Änderungen. Sie können den Cluster-Endpunkt auch für Lesevorgänge nutzen, beispielsweise Abfragen.

Wenn die aktuelle Writer-DB-Instance eines DB-Clusters ausfällt, wechselt der Multi-AZ-DB-Cluster automatisch zu einer neuen Writer-DB-Instance. Während eines Failovers bedient der DB-Cluster weiterhin Verbindungsanfragen von der neuen Schreib-DB-Instance an den Cluster-Endpunkt mit minimaler Serviceunterbrechung.

Das folgende Beispiel zeigt einen Cluster-Endpunkt für einen Multi-AZ-DB-Cluster.

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com
```

Weitere Informationen zum Herstellen einer Verbindung zu Cluster-Endpunkten finden Sie unter. Cluster-Endpunkte

Leser-Endpunkt

Ein Reader-Endpunkt für einen Multi-AZ-DB-Cluster bietet Unterstützung für schreibgeschützte Verbindungen zum DB-Cluster. Verwenden Sie den Leser-Endpunkt für Lesevorgänge, beispielsweise `SELECT`-Abfragen. Durch die Verarbeitung dieser Anweisungen auf den Reader-DB-Instances reduziert dieser Endpunkt den Overhead auf der Writer-DB-Instance. Es hilft dem Cluster auch, die Kapazität zu skalieren, um gleichzeitige `SELECT`-Abfragen zu verarbeiten. Jeder Multi-AZ-DB-Cluster verfügt über einen Reader-Endpunkt.

Der Leser-Endpunkt sendet jede Verbindungsanforderung an eine der Reader-DB-Instances. Wenn Sie den Reader-Endpunkt für eine Sitzung verwenden, können Sie in dieser Sitzung nur schreibgeschützte Anweisungen wie `SELECT` ausführen.

Das folgende Beispiel zeigt einen Leser-Endpunkt für einen Multi-AZ-DB-Cluster. Die schreibgeschützte Absicht eines Reader-Endpunkts wird durch die `-ro` innerhalb des Namen des Cluster-Endpunkts gekennzeichnet.

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com
```

Weitere Informationen zum Herstellen einer Verbindung mit Leser-Endpunkten finden Sie unter. Reader-Endpunkte

Instance-Endpunkt

Ein Instance-Endpunkt stellt innerhalb eines DB-Instance eine Verbindung zu einer spezifischen Multi-AZ-DB-Cluster her. Jede DB-Instance in einem DB-Cluster hat einen eigenen, spezifischen Instance-Endpunkt. Es gibt also einen Instance-Endpunkt für die aktuelle Writer-DB-Instance des DB-Clusters und einen Instance-Endpunkt für jede der Reader-DB-Instances im DB-Cluster.

Der Instance-Endpunkt bietet direkte Kontrolle über Verbindungen zum DB-Cluster. Dieses Steuerelement kann Ihnen helfen, Szenarien zu beheben, in denen die Verwendung des Cluster-Endpunkts oder des Leser-Endpunkts möglicherweise nicht angemessen ist. Beispiel: Ihre Client-Anwendung erfordert möglicherweise einen detaillierteren Lastausgleich je nach Workload-Typ. In diesem Fall können Sie mehrere Clients so konfigurieren, dass sie sich mit verschiedenen Reader-DB-Instances in einem DB-Cluster verbinden, um Leseworkloads zu verteilen.

Das folgende Beispiel zeigt einen Instance-Endpunkt für eine DB-Instance in einem Multi-AZ-DB-Cluster.

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com
```

Weitere Informationen zum Herstellen einer Verbindung zu Instanzendpunkten finden Sie unter. Instance-Endpunkte

Anzeigen von Endpunkten

Verwenden Sie die Konsole oder Amazon AWS CLI, RDS API um die Cluster-, Reader- und Instance-Endpunkte anzuzeigen.

Console

In der AWS Management Console sehen Sie den Cluster-Endpunkt und den Reader-Endpunkt auf der Detailseite für jeden Multi-AZ-DB-Cluster. Der Instance-Endpunkt wird auf der Detailseite der jeweiligen DB-Instance angezeigt.

AWS CLI

Mit dem AWS CLI sehen Sie die Writer- und Reader-Endpunkte in der Ausgabe des describe-db-clustersBefehls. Der folgende Befehl zeigt beispielsweise die Endpunktattribute für alle Cluster in Ihrer aktuellen AWS Region.

```
aws rds describe-db-cluster-endpoints
```

Amazon RDS API

Mit Amazon rufen Sie die Endpunkte ab RDSAPI, indem Sie die Aktion D escribeDBCluster Endpoints aufrufen. Die Ausgabe zeigt auch Amazon-Aurora-DB-Cluster-Endpunkte an, falls vorhanden.

Cluster-Endpunkte

Jeder Multi-AZ-DB-Cluster hat einen einzigen integrierten Cluster-Endpunkt, dessen Name und andere Attribute von Amazon RDS verwaltet werden. Sie können einen solchen Endpunkt nicht erstellen, löschen oder ändern.

Sie verwenden den Cluster-Endpunkt, wenn Sie Ihren DB-Cluster verwalten, Extraktions-, Transformations-, Load (ETL) -Operationen durchführen oder Anwendungen entwickeln und testen. Der Cluster-Endpunkt stellt eine Verbindung zur Writer-DB-Instance des Clusters her. Die Writer-DB-Instance ist die einzige DB-Instance, in der Sie Tabellen und Indizes erstellen, `INSERT` Anweisungen ausführen DDL und DML andere AND-Operationen ausführen können.

Die physische IP-Adresse, auf die der Cluster-Endpunkt verweist, ändert sich, wenn der Failover-Mechanismus eine neue DB-Instance zur Writer-DB-Instance für den Cluster heraufsetzt. Wenn Sie irgendeine Form von Verbindungspooling oder sonstigem Multiplexing verwenden, sollten Sie darauf vorbereitet sein, den Wert für alle zwischengespeicherten Informationen zu leeren oder zu reduzieren. time-to-live DNS Hierdurch wird sichergestellt, dass Sie keine Lese-Schreib-Verbindung mit einer DB-Instance herstellen, die nicht mehr verfügbar ist oder nach einem Failover schreibgeschützt ist.

Reader-Endpunkte

Sie verwenden den Reader-Endpunkt für schreibgeschützte Verbindungen zu Ihrem Multi-AZ-DB-Cluster. Dieser Endpunkt hilft Ihrem DB-Cluster bei der Handhabung einer abfragenintensiven

Workload. Der Reader-Endpunkt ist der Endpunkt, den Sie Anwendungen zur Verfügung stellen, die Berichterstattung oder andere schreibgeschützte Operationen auf dem Cluster ausführen. Der Leser-Endpunkt sendet Verbindungen zu verfügbaren Reader-DB-Instances in einem Multi-AZ-DB-Cluster.

Jeder Multi-AZ-Cluster hat einen einzigen integrierten Leser-Endpunkt, dessen Name und andere Attribute von Amazon RDS verwaltet werden. Sie können einen solchen Endpunkt nicht erstellen, löschen oder ändern.

Instance-Endpunkte

Jede DB-Instance in einem Multi-AZ-DB-Cluster hat ihren eigenen integrierten Instance-Endpunkt, dessen Name und andere Attribute von Amazon RDS verwaltet werden. Sie können einen solchen Endpunkt nicht erstellen, löschen oder ändern. Bei einem Multi-AZ-DB-Cluster verwenden Sie in der Regel die Writer- und Reader-Endpunkte häufiger als die Instance-Endpunkte.

Im day-to-day operativen Bereich verwenden Sie Instance-Endpunkte hauptsächlich zur Diagnose von Kapazitäts- oder Leistungsproblemen, die sich auf eine bestimmte DB-Instance in einem Multi-AZ-DB-Cluster auswirken. Während eine Verbindung zu einer spezifischen DB-Instance besteht, können Sie unter anderem deren Statusvariablen oder Metriken untersuchen. Hierdurch ist es möglich, Unterschiede zwischen den Aktivitäten verschiedener Cluster-DB-Instances zu ermitteln.

Verbindungen mit hoher Verfügbarkeit

Verwenden Sie im Fall von Multi-AZ-DB-Clustern, bei denen eine hohe Verfügbarkeit von großer Bedeutung ist, den Writer-Endpunkt für Lese-/Schreib- oder Allzweck-Verbindungen und den Leser-Endpunkt für schreibgeschützte Verbindungen. Die Schreiber- und Leser-Endpunkte verwalten das Failover von DB-Instances besser als Instance-Endpunkte. Im Gegensatz zu den Instance-Endpunkten ändern die Schreiber- und Leser-Endpunkte automatisch, mit welcher DB-Instance sie eine Verbindung herstellen, wenn eine DB-Instance in Ihrem Cluster nicht mehr verfügbar ist.

Wenn die Writer-DB-Instance eines DB-Clusters ausfällt, führt Amazon RDS automatisch einen Failover zu einer neuen Writer-DB-Instance durch. Dies geschieht durch die Förderung einer Reader-DB-Instance auf eine neue Writer-DB-Instance. Wenn ein Failover auftritt, können Sie den Schreiber-Endpunkt verwenden, um eine Verbindung zu der neu hochgestuften Writer-DB-Instance wiederherzustellen. Oder Sie können den Leser-Endpunkt verwenden, um eine Verbindung zu einer der Reader-DB-Instances im DB-Cluster wiederherzustellen. Während eines Failovers leitet der Reader-Endpunkt Verbindungen möglicherweise für kurze Zeit an die neue Writer-DB-Instance eines DB-Clusters weiter, nachdem eine Reader-DB-Instance zur neuen Writer-DB-Instance hochgestuft wurde. Wenn Sie Ihre Anwendungslogik so entwickeln, dass Verbindungen zu Instance-Endpunkten

verwaltet werden können, ist es möglich, den daraus resultierenden Satz an verfügbaren DB-Instances im DB-Cluster manuell oder programmgesteuert zu ermitteln.

Verbindung zu Multi-AZ-DB-Clustern mit den AWS Treibern für Amazon herstellen RDS

Die AWS Treibersuite wurde so konzipiert, dass sie schnellere Switchover- und Failover-Zeiten sowie Authentifizierung mit AWS Secrets Manager, AWS Identity and Access Management (IAM) und Federated Identity unterstützt. Die AWS Treiber sind darauf angewiesen, den Status des DB-Clusters zu überwachen und die Clustertopologie zu kennen, um den neuen Writer zu ermitteln. Dieser Ansatz reduziert die Switchover- und Failover-Zeiten auf einstellige Sekunden, im Vergleich zu mehreren zehn Sekunden bei Open-Source-Treibern.

Bei der Einführung neuer Servicefunktionen besteht das Ziel der AWS Treibersuite darin, eine integrierte Unterstützung für diese Servicefunktionen zu bieten.

Mit dem Amazon Web Services (AWS) JDBC -Treiber eine Verbindung zu Multi-AZ-DB-Clustern herstellen

Der Amazon Web Services (AWS) JDBC -Treiber wurde als erweiterter JDBC Wrapper konzipiert, um Anwendungen dabei zu unterstützen, die Funktionen geclusterter Datenbanken zu nutzen. Dieser Wrapper ergänzt einen vorhandenen Treiber und erweitert dessen Funktionalität. JDBC Der Treiber ist Drop-In-kompatibel mit den folgenden Community-Treibern:

- Mein Connector/J SQL
- MariaDB Connector/J
- pg JDBC

Um den AWS JDBC Treiber zu installieren, hängen Sie die AWS JDBC Treiber-.jar-Datei an (befindet sich in der Anwendung`CLASSPATH`) und behalten Sie die Verweise auf den jeweiligen Community-Treiber bei. Aktualisieren Sie das jeweilige URL Verbindungspräfix wie folgt:

- `jdbc:mysql://` auf `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` auf `jdbc:aws-wrapper:mariadb://`
- `jdbc:postgresql://` auf `jdbc:aws-wrapper:postgresql://`

Weitere Informationen zum AWS JDBC Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im JDBC GitHub Treiber-Repository von Amazon Web Services (AWS).

Verbindung zu Multi-AZ-DB-Clustern mit dem Amazon Web Services (AWS) Python-Treiber herstellen

Der Amazon Web Services (AWS) Python-Treiber ist als fortschrittlicher Python-Wrapper konzipiert. Dieser Wrapper ergänzt den Open-Source-Treiber Psycopg und erweitert dessen Funktionalität. Der AWS Python-Treiber unterstützt Python-Versionen 3.8 und höher. Sie können das `aws-advanced-python-wrapper` Paket zusammen mit den `psycopg` Open-Source-Paketen mit dem `pip` Befehl installieren.

Weitere Informationen zum AWS Python-Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im GitHub Python-Treiber-Repository von Amazon Web Services (AWS).

Automatisches Verbinden einer AWS Rechenressource und eines Multi-AZ-DB-Clusters für Amazon RDS

Sie können automatisch einen Multi-AZ-DB-Cluster verbinden und AWS Ressourcen wie Amazon Elastic Compute Cloud (AmazonEC2) -Instances und AWS Lambda -Funktionen berechnen.

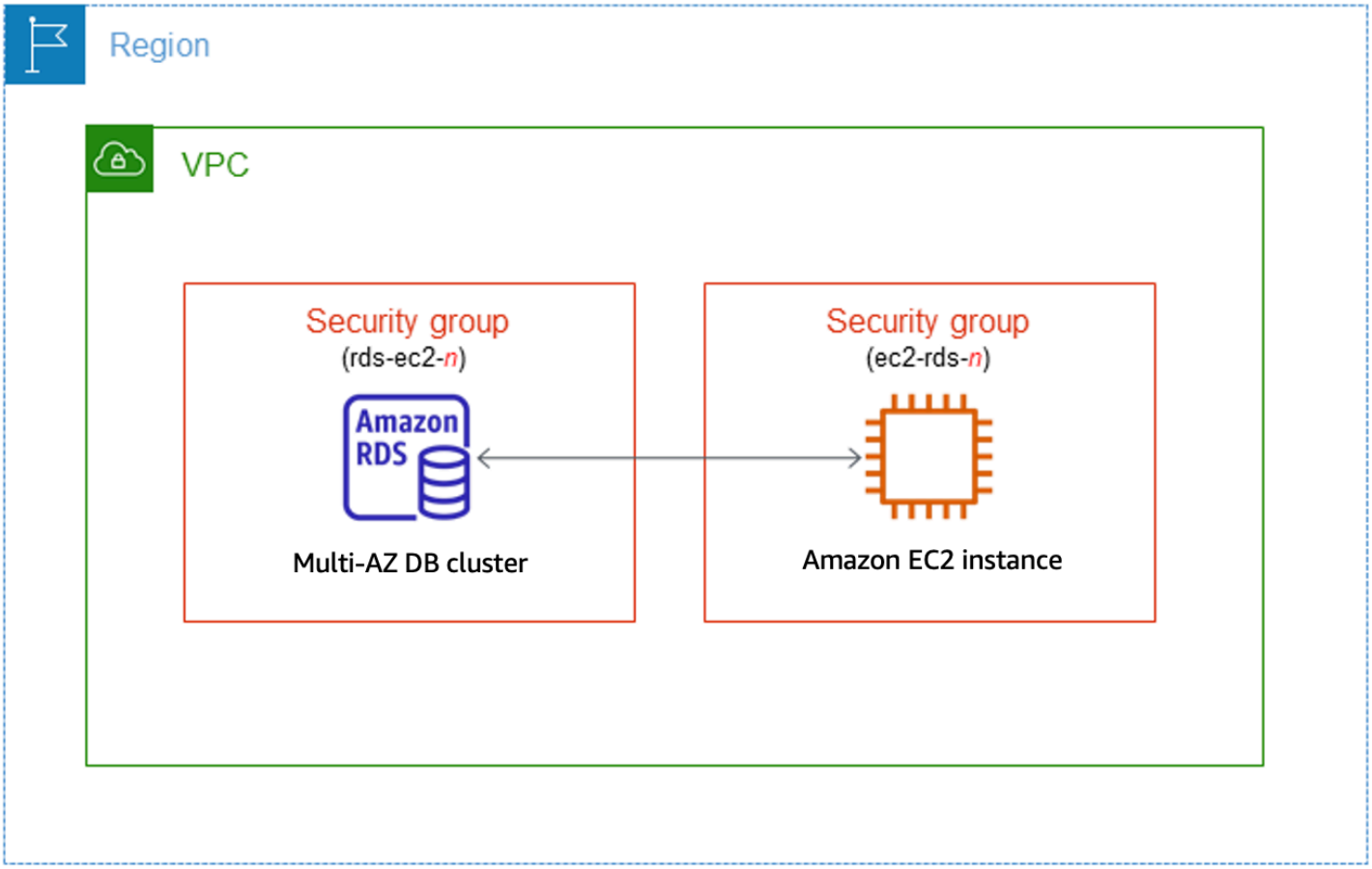
Die folgenden Themen enthalten detaillierte Anweisungen zur Konfiguration von Netzwerkeinstellungen, Sicherheitsgruppen und Verbindungsparametern, um zuverlässige Verbindungen zu RDS Amazon-DB-Instances innerhalb einer Multi-AZ-DB-Cluster-Bereitstellung herzustellen. Sie konzentrieren sich auf die Optimierung der Netzwerkkonnektivität und -leistung für Anwendungen, die mit einem Multi-AZ-DB-Cluster interagieren, um einen sicheren und effizienten Datenbetrieb zu gewährleisten.

Themen

- Automatisches Verbinden einer EC2-Instance und eines Multi-AZ-DB-Clusters
- Automatisches Verbinden einer Lambda-Funktion und eines Multi-AZ-DB-Clusters

Automatisches Verbinden einer EC2-Instance und eines Multi-AZ-DB-Clusters

Sie können die Amazon-RDS-Konsole verwenden, um das Einrichten einer Verbindung zwischen einer Amazon Elastic Compute Cloud (Amazon-EC2)-Instance und einem Multi-AZ-DB-Cluster zu vereinfachen. Häufig befindet sich Ihr Multi-AZ-DB-Cluster in einem privaten Subnetz und Ihre EC2-Instance in einem öffentlichen Subnetz innerhalb einer VPC. Sie können einen SQL-Client auf Ihrer EC2-Instance verwenden, um eine Verbindung mit Ihrem Multi-AZ-DB-Cluster herzustellen. Die EC2-Instance kann auch Webserver oder Anwendungen ausführen, die auf Ihren privaten Multi-AZ-DB-Cluster zugreifen.



Wenn Sie eine Verbindung mit einer EC2-Instance herstellen möchten, die sich nicht in derselben VPC wie der Multi-AZ-DB-Cluster befindet, sehen Sie sich die entsprechenden Szenarien unter the section called “Szenarien für den Zugriff auf einen in einem VPC” an.

Themen

- Übersicht über die automatische Verbindung mit einer EC2-Instance
- Automatisches Verbinden einer EC2-Instance und eines Multi-AZ-DB-Clusters
- Anzeigen verbundener Rechenressourcen

Übersicht über die automatische Verbindung mit einer EC2-Instance

Wenn Sie eine Verbindung zwischen einer EC2-Instance und einem Multi-AZ-DB-Cluster automatisch einrichten, konfiguriert Amazon RDS die VPC-Sicherheitsgruppe für Ihre EC2-Instance und für Ihren DB-Cluster.

Im Folgenden sind die Anforderungen für die Verbindung einer EC2-Instance mit einem Multi-AZ-DB-Cluster aufgeführt:

- Die EC2-Instance muss sich in derselben VPC wie die der Multi-AZ-DB-Cluster befinden.

  Wenn keine EC2-Instances in derselben VPC vorhanden sind, bietet die Konsole einen Link zum Erstellen einer solchen Instance.
- Der Benutzer, der die Verbindung einrichtet, muss über Berechtigungen zum Ausführen der folgenden EC2-Vorgänge verfügen:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Wenn Sie eine Verbindung mit einer EC2-Instance einrichten, führt Amazon RDS eine Aktion aus, die auf der aktuellen Konfiguration der Sicherheitsgruppen basiert, die dem Multi-AZ-DB-Cluster und der EC2-Instance zugeordnet sind, wie in der folgenden Tabelle beschrieben.

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle EC2-Sicherheitsgruppenkonfiguration	RDS-Aktion
Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht (wobei *n* für eine Zahl steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-	Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht (wobei *n* für eine Zahl steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-	Amazon RDS führt keine Aktion aus. Es wurde bereits automatisch eine Verbindung zwischen der EC2-Instance und dem Multi-AZ-DB-Cluster konfiguriert. Da bereits eine Verbindung zwischen der EC2-Instance und der RDS-Datenbank besteht, werden die Sicherheitsgruppen nicht geändert.

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle EC2-Sicherheitsgruppenkonfiguration	RDS-Aktion
Sicherheitsgruppe der EC2-Instance als Quelle.	Sicherheitsgruppe des Multi-AZ-DB-Clusters als Quelle.	

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle EC2-Sicherheitsgruppenkonfiguration	RDS-Aktion
Es gilt eine der folgenden Bedingungen: • Dem Multi-AZ-DB-Cluster sind keine Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. • Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. Keine dieser Sicherheitsgruppen kann jedoch für die Verbindung mit der EC2-Instance verwendet werden. Eine Sicherheitsgruppe kann nicht verwendet werden, wenn sie keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle enthält. Eine Sicherheitsgruppe kann auch nicht verwendet werden, wenn sie geändert wurde. Beispiele für Änderungen sind das Hinzufügen einer Regel	Es gilt eine der folgenden Bedingungen: • Der EC2-Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `ec2-rds-`*n* entspricht. • Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `ec2-rds-`*n* entspricht. Keine dieser Sicherheitsgruppen kann jedoch für die Verbindung mit dem Multi-AZ-DB-Cluster verwendet werden. Eine Sicherheitsgruppe kann nicht verwendet werden, wenn sie keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Quelle enthält. Eine Sicherheitsgruppe kann auch nicht verwendet werden, wenn sie geändert wurde.	RDS action: create new security groups

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle EC2-Sicherheitsgruppenkonfiguration	RDS-Aktion
oder das Ändern des Ports einer vorhandenen Regel.		
Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle.	Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `ec2-rds-`*n* entspricht. Keine dieser Sicherheitsgruppen kann jedoch für die Verbindung mit dem Multi-AZ-DB-Cluster verwendet werden. Eine Sicherheitsgruppe kann nicht verwendet werden, wenn sie keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Quelle enthält. Eine Sicherheitsgruppe kann auch nicht verwendet werden, wenn sie geändert wurde.	RDS action: create new security groups

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle EC2-Sicherheitsgruppenkonfiguration	RDS-Aktion
Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle.	Eine gültige EC2-Sicherheitsgruppe für die Verbindung ist vorhanden, jedoch nicht mit der EC2-Instance verknüpft. Die Sicherheitsgruppe trägt einen Namen, der dem Muster `rds-ec2-`*n* entspricht. Sie wurde nicht geändert. Er enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Quelle.	RDS action: associate EC2 security group

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle EC2-Sicherheitsgruppenkonfiguration	RDS-Aktion
Es gilt eine der folgenden Bedingungen: • Dem Multi-AZ-DB-Cluster sind keine Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. • Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. Keine dieser Sicherheitsgruppen kann jedoch für die Verbindung mit der EC2-Instance verwendet werden. Eine Sicherheitsgruppe kann nicht verwendet werden, wenn sie keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle enthält. Eine Sicherheitsgruppe kann auch nicht verwendet werden, wenn sie geändert wurde.	Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-ec2-`*n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Quelle.	RDS action: create new security groups

RDS-Aktion : neue Sicherheitsgruppen erstellen

Amazon RDS führt die folgenden Aktionen durch:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `rds-ec2-`*n* entspricht. Diese Sicherheitsgruppe enthält eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle. Diese Sicherheitsgruppe, die dem Multi-AZ-DB-Cluster zugeordnet ist, ermöglicht der EC2-Instance den Zugriff auf den Multi-AZ-DB-Cluster.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `ec2-rds-`*n* entspricht. Diese Sicherheitsgruppe enthält eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Quelle. Diese Sicherheitsgruppe ist der EC2-Instance zugeordnet und ermöglicht es der EC2-Instance, Datenverkehr an den Multi-AZ-DB-Cluster zu senden.

RDS-Aktion : EC2-Sicherheitsgruppe zuordnen

Amazon RDS ordnet die gültige, vorhandene EC2-Sicherheitsgruppe der EC2-Instance zu. Diese Sicherheitsgruppe ermöglicht es der EC2-Instance, Datenverkehr an den Multi-AZ-DB-Cluster zu senden.

Automatisches Verbinden einer EC2-Instance und eines Multi-AZ-DB-Clusters

Bevor Sie eine Verbindung zwischen einer EC2-Instance und einer RDS-Datenbank einrichten, stellen Sie sicher, dass Sie die unter Überblick über die automatische Konnektivität mit einer EC2 Instance beschriebenen Anforderungen erfüllen.

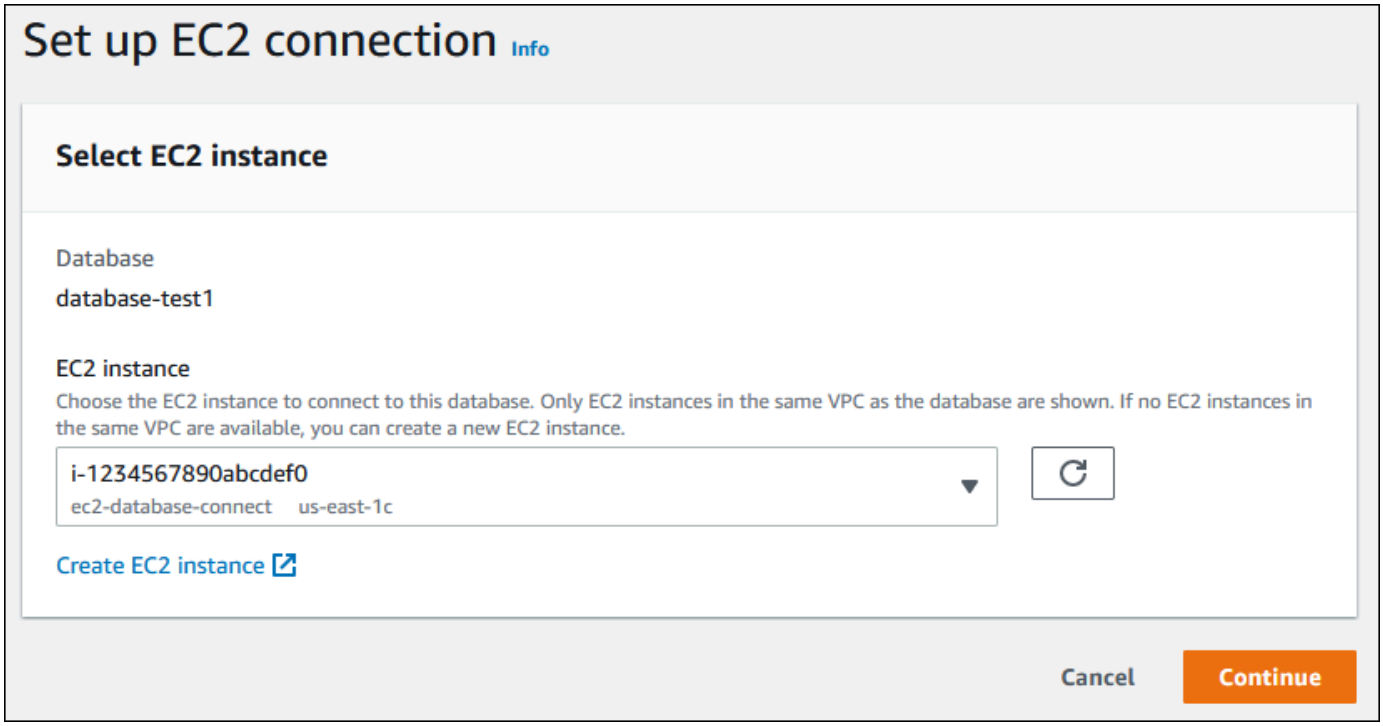
Wenn Sie nach dem Konfigurieren der Verbindung Änderungen an diesen Sicherheitsgruppen vornehmen, können sich die Änderungen auf die Verbindung zwischen der EC2-Instance und der RDS-Datenbank auswirken.

> **Note**
>
> Sie können eine Verbindung zwischen einer EC2-Instance und einer RDS-Datenbank nur automatisch einrichten, indem Sie die AWS Management Console verwenden. Sie können keine automatische Verbindung mit der AWS CLI oder der RDS-API einrichten.

So verbinden Sie eine EC2-Instance und eine RDS-Datenbank automatisch

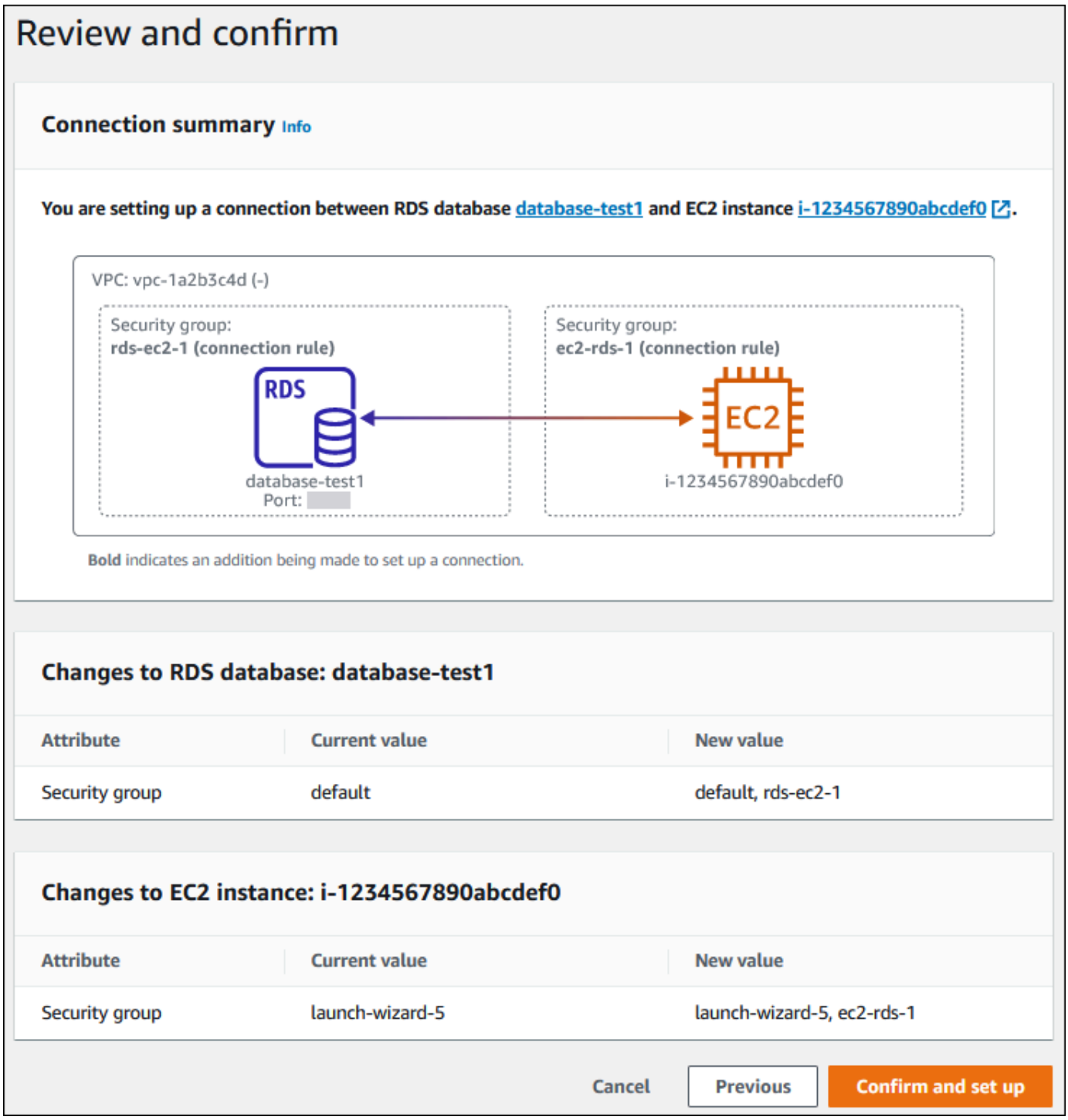
1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann den DB-Cluster aus.
3. Wählen Sie für Aktionen die Option EC2-Verbindung einrichten aus.

   Die Seite Set up EC2 connection (EC2-Verbindung einrichten) wird angezeigt.
4. Wählen Sie auf der Seite Set up EC2 connection (EC2-Verbindung einrichten) die EC2-Instance aus.



   Wenn keine EC2-Instances in derselben VPC vorhanden sind, wählen Sie Create EC2 instance (EC2-Instance erstellen) aus, um eine solche Instance zu erstellen. Stellen Sie in diesem Fall sicher, dass sich die neue EC2-Instance in derselben VPC wie die RDS-Datenbank befindet.
5. Klicken Sie auf Weiter.

   Die Seite Review and confirm (Überprüfen und bestätigen) wird angezeigt.



6. Sehen Sie sich auf der Seite Review and confirm (Überprüfen und bestätigen) die Änderungen an, die RDS beim Einrichten der Verbindung mit der EC2-Instance vornehmen wird.

   Wenn die Änderungen korrekt sind, wählen Sie Bestätigen und einrichten.

   Sind die Änderungen nicht korrekt, wählen Sie Previous (Zurück) oder Cancel (Abbrechen) aus.

Anzeigen verbundener Rechenressourcen

Sie können den verwenden AWS Management Console , um die Rechenressourcen anzuzeigen, die mit einem mit einer RDS-Datenbank verbunden sind. Zu den angezeigten Ressourcen gehören Rechenressourcenverbindungen, die automatisch eingerichtet wurden. Sie können die Konnektivität mit Rechenressourcen auf folgende Weise automatisch einrichten:

- Sie können die Rechenressource auswählen, wenn Sie die Datenbank erstellen.

  Weitere Informationen erhalten Sie unter Erstellen einer Amazon RDS-DB-Instance und Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen.
- Sie können die Konnektivität zwischen einer vorhandenen Datenbank und einer Rechenressource einrichten.

  Weitere Informationen finden Sie unter Automatisches Verbinden einer EC2 Instance und eines RDS.

Die aufgelisteten Rechenressourcen enthalten keine Ressourcen, die manuell mit der Datenbank verbunden wurden. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf eine Datenbank erlauben, indem Sie der VPC-Sicherheitsgruppe, die der Datenbank zugeordnet ist, eine Regel hinzufügen.

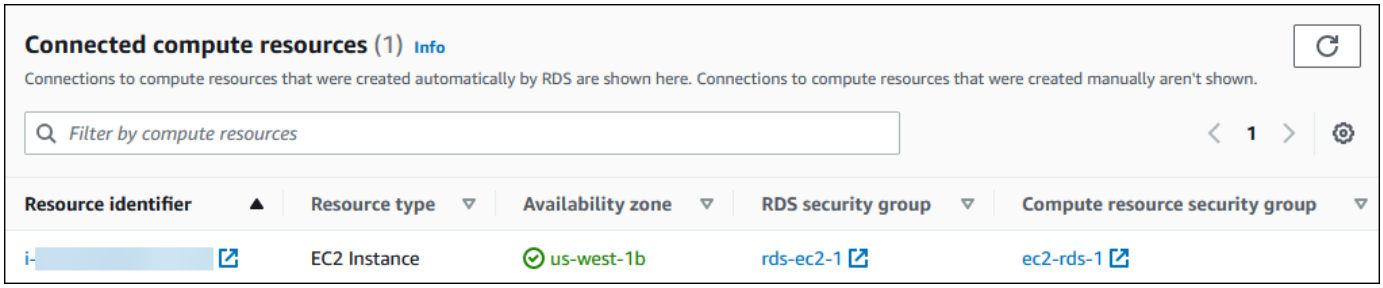
Für die Auflistung einer Rechenressource müssen die folgenden Bedingungen erfüllt sei:

- Der Name der Sicherheitsgruppe, die der Rechenressource zugeordnet ist, entspricht dem Muster `ec2-rds-`*n* (wobei *n* für eine Zahl steht).
- Die Sicherheitsgruppe, die der Rechenressource zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei der Portbereich auf den Port festgelegt ist, den die RDS-Datenbank verwendet.
- Die Sicherheitsgruppe, die der Rechenressource zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei die Quelle auf eine Sicherheitsgruppe festgelegt ist, die der RDS-Datenbank zugeordnet ist.
- Der Name der Sicherheitsgruppe, die der RDS-Datenbank zugeordnet ist, entspricht dem Muster `rds-ec2-`*n* entspricht (wobei *n* für eine Zahl steht).
- Die Sicherheitsgruppe, die der RDS-Datenbank zugeordnet ist, hat eine Regel für eingehenden Datenverkehr, wobei der Portbereich auf den Port festgelegt ist, den die RDS-Datenbank verwendet.

- Die Sicherheitsgruppe, die der RDS-Datenbank zugeordnet ist, hat eine Regel für eingehenden Datenverkehr, wobei die Quelle auf eine Sicherheitsgruppe festgelegt ist, die der Rechenressource zugeordnet ist.

So zeigen Sie Rechenressourcen an, die mit einer RDS-Datenbank verbunden sind

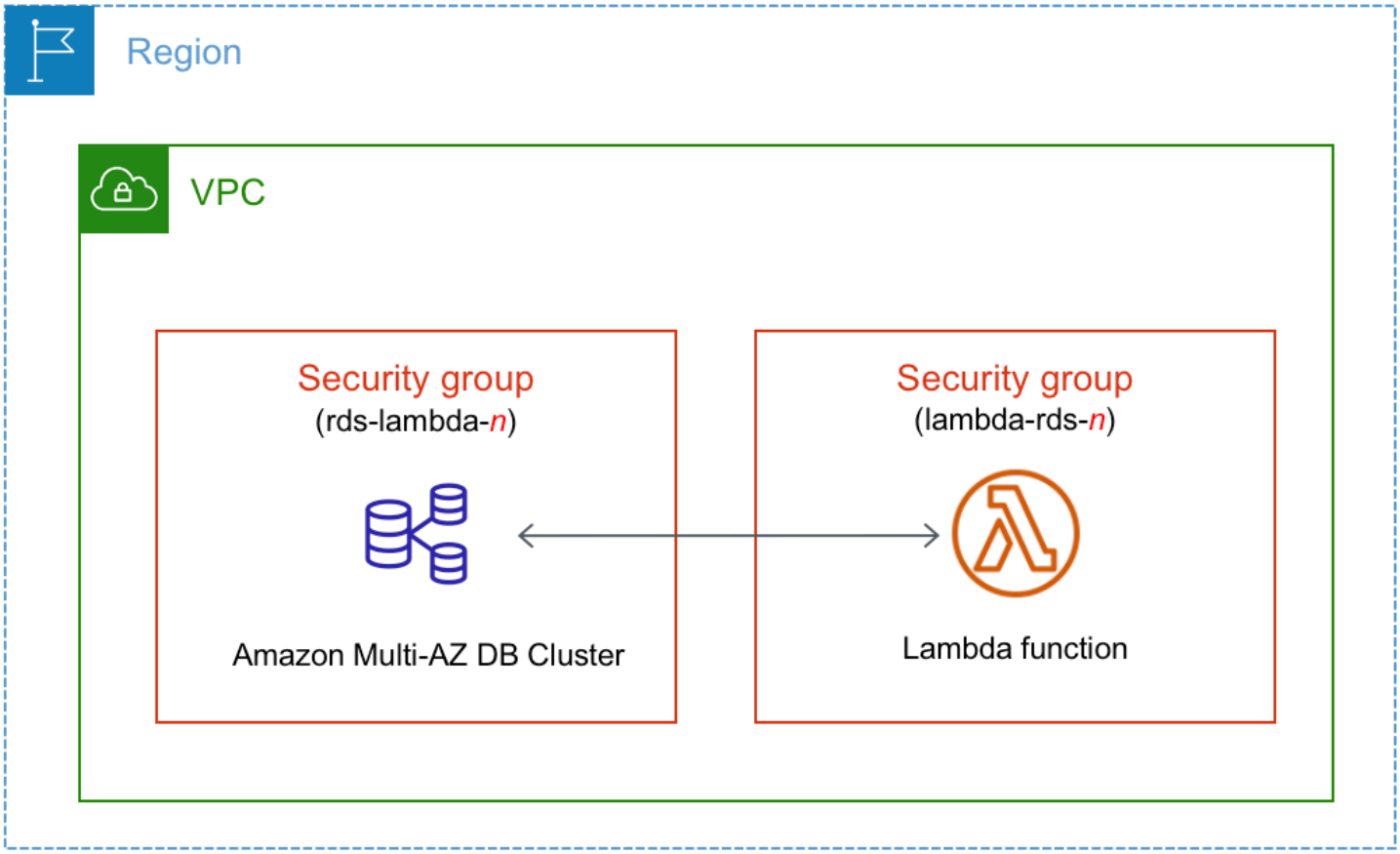
1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann den Namen des DB-Clusters aus.
3. Sehen Sie sich auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) die Rechenressourcen unter Verbundene Rechenressourcen an.



Automatisches Verbinden einer Lambda-Funktion und eines Multi-AZ-DB-Clusters

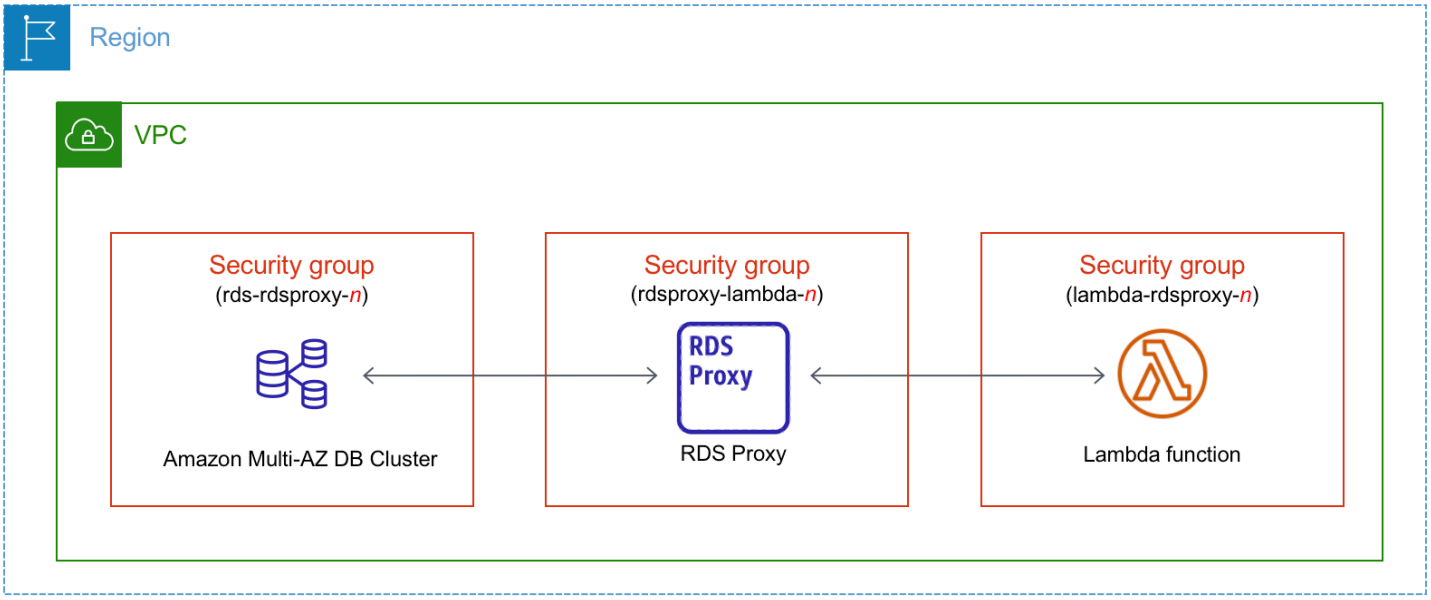
Sie können die RDS-Konsole verwenden, um das Einrichten einer Verbindung zwischen einer Lambda-Funktion und einem Multi-AZ-DB-Cluster zu vereinfachen. Sie können die RDS-Konsole verwenden, um das Einrichten einer Verbindung zwischen einer Lambda-Funktion und einem Multi-AZ-DB-Cluster zu vereinfachen. Häufig befindet sich Ihre Multi-AZ-DB-Cluster in einem privaten Subnetz innerhalb einer VPC. Die Lambda-Funktion kann von Anwendungen verwendet werden, um auf Ihren privaten Multi-AZ-DB-Cluster zuzugreifen.

Das folgende Bild zeigt eine direkte Verbindung zwischen Ihrem Multi-AZ-DB-Cluster und Ihrer Lambda-Funktion.



Sie können die Verbindung zwischen Ihrer Lambda-Funktion und Ihrer Datenbank über RDS-Proxy einrichten, um die Leistung und Stabilität Ihrer Datenbank zu verbessern. Oft stellen Lambda-Funktionen häufige, kurze Datenbankverbindungen her, die von dem von RDS Proxy angebotenen Verbindungspooling profitieren. Sie können eine der IAM-Authentifizierungen nutzen, die Sie bereits für Lambda-Funktionen eingerichtet haben, anstatt Datenbankanmeldeinformationen im Lambda-Anwendungscode zu verwalten. Weitere Informationen finden Sie unter Amazon RDS-Proxy .

Sie können die Konsole verwenden, um automatisch einen Proxy für Ihre Verbindung zu erstellen. Sie können auch vorhandene Proxys auswählen. Die Konsole aktualisiert die Proxy-Sicherheitsgruppe, um Verbindungen von Ihrer Datenbank und der Lambda-Funktion aus zuzulassen. Sie können Ihre Datenbankanmeldeinformationen eingeben oder das Secrets-Manager-Secret auswählen, das Sie für den Zugriff auf die Datenbank benötigen.



Themen

- Überblick über die automatische Konnektivität mit einer Lambda-Funktion
- Automatisches Verbinden einer Lambda-Funktion und eines Multi-AZ-DB-Clusters
- Anzeigen verbundener Rechenressourcen

Überblick über die automatische Konnektivität mit einer Lambda-Funktion

Wenn Sie eine Verbindung zwischen einer Lambda-Funktion und einem Multi-AZ-DB-Cluster automatisch einrichten, konfiguriert Amazon RDS die VPC-Sicherheitsgruppe für Ihre Lambda-Funktion und für Ihren DB-Cluster.

Im Folgenden sind die Anforderungen für die Verbindung einer Lambda-Funktion mit einem Multi-AZ-DB-Cluster aufgeführt:

- Die Lambda-Funktion muss sich in derselben VPC befinden wie der Multi-AZ-DB-Cluster.

  Wenn keine Lambda-Funktion in derselben VPC vorhanden ist, bietet die Konsole einen Link zum Erstellen einer solchen Funktion.
- Der Benutzer, der die Verbindung einrichtet, muss über Berechtigungen zum Ausführen der folgenden Vorgänge von Amazon RDS, Amazon EC2, Lambda, Secrets Manager und IAM verfügen:
  - Amazon RDS
    - `rds:CreateDBProxies`

  - `rds:DescribeDBInstances`
  - `rds:DescribeDBProxies`
  - `rds:ModifyDBInstance`
  - `rds:ModifyDBProxy`
  - `rds:RegisterProxyTargets`
- Amazon EC2
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DeleteSecurityGroup`
  - `ec2:DescribeSecurityGroups`
  - `ec2:RevokeSecurityGroupEgress`
  - `ec2:RevokeSecurityGroupIngress`
- Lambda
  - `lambda:CreateFunctions`
  - `lambda:ListFunctions`
  - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
  - `sercetsmanager:CreateSecret`
  - `secretsmanager:DescribeSecret`
- IAM
  - `iam:AttachPolicy`
  - `iam:CreateRole`
  - `iam:CreatePolicy`
- AWS KMS
  - `kms:describeKey`

Wenn Sie eine Verbindung zwischen einer Lambda-Funktion und einem Multi-AZ-DB-Cluster einrichten, konfiguriert Amazon RDS die VPC-Sicherheitsgruppe für Ihre Funktion und für Ihren Multi-AZ-DB-Cluster. Wenn Sie RDS-Proxy verwenden, konfiguriert Amazon RDS auch die VPC-Sicherheitsgruppe für den Proxy. Amazon RDS handelt gemäß der aktuellen Konfiguration der

Sicherheitsgruppen, die dem Multi-AZ-DB-Cluster und der Lambda-Funktion zugeordnet sind, wie in der folgenden Tabelle beschrieben.

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Amazon RDS führt keine Aktion aus, da Sicherheitsgruppen aller Ressourcen dem richtigen Benennungsmuster folgen und über die richtigen Regeln für ein- und ausgehenden Datenverkehr verfügen.	Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht (wobei *n* für eine Zahl steht), oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE` ist. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Lambda-Funktion oder dem Proxy als Quelle.	Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht (wobei *n* für eine Zahl steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für ausgehenden Datenverkehr, wobei entweder die VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder der Proxy das Ziel ist.	Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht (wobei *n* für eine Zahl steht). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe verfügt über Regeln für ein- und ausgehenden Datenverkehr mit den VPC-Sicherheitsgruppen der Lambda-Funktion und des Multi-AZ-DB-Clusters.

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Es gilt eine der folgenden Bedingungen: • Dem Multi-AZ-DB-Cluster ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE` ist. • Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE`	Es gilt eine der folgenden Bedingungen: • Der Lambda-Funktion ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. • Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem	Es gilt eine der folgenden Bedingungen: • Dem Proxy ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. • Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem Multi-AZ-DB-Cluster oder der Lambda-Funktion verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht	RDS action: create new security groups

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
ist. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Lambda-Funktion verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht verwenden, wenn sie keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Lambda-Funktion oder dem Proxy als Quelle enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde. Beispiele für Änderungen sind das Hinzufügen einer Regel oder das Ändern des Ports einer vorhandenen Regel.	Multi-AZ-DB-Cluster verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht verwenden, wenn diese keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder dem Proxy als Quelle enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	verwenden, wenn diese keine Regel für ein- und ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder der Lambda-Funktion enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE` ist. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Lambda-Funktion oder dem Proxy als Quelle.	Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem Multi-AZ-DB-Cluster verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht verwenden, wenn diese keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Ziel enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem Multi-AZ-DB-Cluster oder der Lambda-Funktion verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht verwenden, wenn diese keine Regel für ein- und ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder der Lambda-Funktion enthält. Amazon RDS kann auch keine Sicherheit	RDS action: create new security groups

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
		tsgruppe verwenden, die geändert wurde.	
Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE` ist. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Lambda-Funktion oder dem Proxy als Quelle.	Eine gültige Lambda-Sicherheitsgruppe für die Verbindung ist vorhanden, jedoch nicht mit der Lambda-Funktion verknüpft. Die Sicherheitsgruppe hat einen Namen, der dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Sie wurde nicht geändert. Sie enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder dem Proxy als Quelle.	Eine gültige Proxy-Sicherheitsgruppe für die Verbindung ist vorhanden, jedoch nicht mit dem Proxy verknüpft. Die Sicherheitsgruppe trägt einen Namen, der dem Muster `rdsproxy-lambda-` *n* entspricht. Sie wurde nicht geändert. Sie verfügt über Regeln für ein- und ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters und der Lambda-Funktion.	RDS action: associate Lambda security group

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Es gilt eine der folgenden Bedingungen: • Dem Multi-AZ-DB-Cluster ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE` ist. • Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppe zugeordnet, deren Name dem Muster `rds-lambda-` *n* entspricht, oder wenn `TargetHealth` eines zugeordneten Proxys `AVAILABLE`	Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine Regel für ausgehenden Datenverkehr, wobei die VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder der Proxy das Ziel ist.	Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe verfügt über Regeln für ein- und ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters und der Lambda-Funktion.	RDS action: create new security groups

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
ist. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Lambda-Funktion oder dem Proxy verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht verwenden, wenn sie keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Lambda-Funktion oder dem Proxy als Quelle enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.			

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
Dem Multi-AZ-DB-Cluster sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rds-rdsproxy-` *n* entspricht (wobei *n* für eine Zahl steht).	Es gilt eine der folgenden Bedingungen: • Der Lambda-Funktion ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. • Der Lambda-Funktion sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `lambda-rds-` *n* oder `lambda-rdsproxy-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem	Es gilt eine der folgenden Bedingungen: • Dem Proxy ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. • Dem Proxy sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `rdsproxy-lambda-` *n* entspricht. Amazon RDS kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit dem Multi-AZ-DB-Cluster oder der Lambda-Funktion verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht	RDS action: create new security groups

Aktuelle RDS-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration der Lambda-Sicherheitsgruppe	Aktuelle Konfiguration der Proxy-Sicherheitsgruppe	RDS-Aktion
	Multi-AZ-DB-Cluster verwenden. Amazon RDS kann eine Sicherheitsgruppe nicht verwenden, wenn diese keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters als Ziel enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	verwenden, wenn diese keine Regel für ein- und ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder der Lambda-Funktion enthält. Amazon RDS kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	

RDS-Aktion : neue Sicherheitsgruppen erstellen

Amazon RDS führt die folgenden Aktionen durch:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `rds-lambda-`*n* entspricht. Diese Sicherheitsgruppe enthält eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Lambda-Funktion oder dem Proxy als Quelle. Diese Sicherheitsgruppe, die dem Multi-AZ-DB-Cluster zugeordnet ist, ermöglicht der Funktion oder dem Proxy den Zugriff auf den Multi-AZ-DB-Cluster.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `lambda-rds-`*n* entspricht. Diese Sicherheitsgruppe hat eine Regel für ausgehenden Datenverkehr, wobei die VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters oder der Proxy das Ziel ist. Diese Sicherheitsgruppe

ist der Lambda-Funktion zugeordnet und ermöglicht es der Lambda-Funktion, Datenverkehr an den Multi-AZ-DB-Cluster zu senden oder Datenverkehr über einen Proxy zu senden.

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `rdsproxy-lambda-`*n* entspricht. Diese Sicherheitsgruppe verfügt über Regeln für ein- und ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe des Multi-AZ-DB-Clusters und der Lambda-Funktion.

RDS-Aktion : Lambda-Sicherheitsgruppe zuordnen

Amazon RDS ordnet die gültige, vorhandene Lambda-Sicherheitsgruppe der Lambda-Funktion zu. Diese Sicherheitsgruppe ermöglicht es der Funktion, Datenverkehr an den Multi-AZ-DB-Cluster zu senden oder Datenverkehr über einen Proxy zu senden.

Automatisches Verbinden einer Lambda-Funktion und eines Multi-AZ-DB-Clusters

Sie können die Amazon-RDS-Konsole verwenden, um eine Lambda-Funktion automatisch mit Ihrem Multi-AZ-DB-Cluster zu verbinden. Dies vereinfacht das Einrichten einer Verbindung zwischen diesen Ressourcen.

Sie können den RDS-Proxy auch verwenden, um einen Proxy in Ihre Verbindung aufzunehmen. Lambda-Funktionen stellen häufige, kurze Datenbankverbindungen her, die von dem von RDS Proxy angebotenen Verbindungspooling profitieren. Sie können auch eine IAM-Authentifizierung nutzen, die Sie bereits für Lambda-Funktionen eingerichtet haben, anstatt Datenbankanmeldeinformationen im Lambda-Anwendungscode zu verwalten.

Sie können einen vorhandenen Multi-AZ-DB-Cluster mit neuen oder bestehenden Lambda-Funktionen unter Verwendung der Seite Lambda-Verbindung einrichten verbinden. Beim Einrichtungsvorgang werden automatisch die erforderlichen Sicherheitsgruppen für Sie eingerichtet.

Vor dem Einrichten einer Verbindung zwischen einer Lambda-Funktion und einem Multi-AZ-DB-Cluster stellen Sie Folgendes sicher:

- Ihre Lambda-Funktion und der Multi-AZ-DB-Cluster befinden sich in derselben VPC.
- Sie verfügen über die richtigen Berechtigungen für Ihr Benutzerkonto. Weitere Informationen zu den Anforderungen finden Sie unter Überblick über die automatische Konnektivität mit einer Lambda-Funktion.

Wenn Sie Sicherheitsgruppen nach dem Konfigurieren der Verbindung ändern, können sich diese Änderungen auf die Verbindung zwischen der Lambda-Funktion und dem Multi-AZ-DB-Cluster auswirken.

> **Note**
>
> Sie können eine Verbindung zwischen einem Multi-AZ-DB-Cluster und einer Lambda-Funktion nur in der AWS Management Console automatisch einrichten. Zum Herstellen einer Verbindung mit einer Lambda-Funktion müssen sich alle Instances im Multi-AZ-DB-Cluster im Status Verfügbar befinden.

So verbinden Sie eine Lambda-Funktion und einen Multi-AZ-DB-Cluster automatisch

<result>

Nachdem Sie die Einrichtung bestätigt haben, beginnt Amazon RDS mit dem Herstellen der Verbindung Ihrer Lambda-Funktion, Ihres RDS-Proxys (falls Sie einen Proxy verwendet haben) und Ihres Multi-AZ-DB-Clusters. Die Konsole zeigt das Dialogfeld Verbindungsdetails an, in dem die Änderungen der Sicherheitsgruppe aufgeführt sind, die Verbindungen zwischen Ihren Ressourcen ermöglichen.
</result>

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den Multi-AZ-DB-Cluster aus, den Sie mit einer Lambda-Funktion verbinden möchten, aus.
3. Wählen Sie für Aktionen die Option Lambda-Verbindung einrichten aus.
4. Führen Sie auf der Seite Lambda-Verbindung einrichten unter Lambda-Funktion auswählen einen der folgenden Schritte aus:
   - Wenn eine Lambda-Funktion in derselben VPC wie Ihr Multi-AZ-DB-Cluster vorhanden ist, wählen Sie Vorhandene Funktion auswählen aus und wählen Sie dann die Funktion aus.
   - Wenn Sie keine Lambda-Funktion in derselben VPC vorhanden ist, wählen Sie Neue Funktion erstellen aus und geben Sie dann einen Namen im Feld Funktionsname ein. Die Standard-Laufzeit ist auf Nodejs.18 festgelegt. Sie können die Einstellungen für Ihre neue Lambda-Funktion in der Lambda-Konsole ändern, nachdem Sie die Verbindungseinrichtung abgeschlossen haben.
5. (Optional) Wählen Sie unter RDS Proxy die Option Über RDS Proxy verbinden aus und führen Sie dann einen der folgenden Schritte aus:
   - Wenn Sie einen vorhandenen Proxy haben, den Sie verwenden möchten, klicken Sie auf Vorhandenen Proxy auswählen und wählen Sie dann den Proxy aus.

- Wenn Sie keinen Proxy haben und möchten, dass Amazon RDS automatisch einen für Sie erstellt, wählen Sie Neuen Proxy erstellen aus. Führen Sie dann für Datenbankanmeldeinformationen einen der folgenden Schritte aus:

  a. Wählen Sie Datenbankbenutzername und Passwort aus und geben Sie dann den Benutzernamen und das Passwort für Ihren Multi-AZ-DB-Cluster ein.

  b. Wählen Sie Secrets-Manager-Secret aus. Wählen Sie dann unter Secret auswählen ein Secret von AWS Secrets Manager aus. Wenn Sie kein Secrets-Manager-Secret haben, wählen Sie Neues Secrets-Manager-Secret erstellen aus, um ein neues Secret zu erstellen. Nachdem Sie das Secret erstellt haben, wählen Sie das neue Secret unter Secret auswählen aus.

  Nachdem Sie den neuen Proxy erstellt haben, wählen Sie Vorhandenen Proxy auswählen aus und wählen Sie dann den Proxy aus. Beachten Sie, dass es einige Zeit dauern kann, bis Ihr Proxy für die Verbindung verfügbar ist.

6. (Optional) Erweitern Sie die Verbindungsübersicht und überprüfen Sie die hervorgehobenen Updates für Ihre Ressourcen.
7. Wählen Sie Set up (Festlegen).

Anzeigen verbundener Rechenressourcen

Sie können die AWS Management Console verwenden, um die Rechenressourcen anzuzeigen, die mit einem Multi-AZ-DB-Cluster verbunden sind. Zu den angezeigten Ressourcen gehören Rechenressourcenverbindungen, die von Amazon RDS automatisch eingerichtet wurden.

Die aufgelisteten Rechenressourcen umfassen keine Ressourcen, die manuell mit dem Multi-AZ-DB-Cluster verbunden sind. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf Ihren Multi-AZ-DB-Cluster erlauben, indem Sie der VPC-Sicherheitsgruppe, die dem Cluster zugeordnet ist, eine Regel hinzufügen.

Damit die Konsole eine Lambda-Funktion auflistet, müssen die folgenden Bedingungen gelten:

- Der Name der Sicherheitsgruppe, die der Rechenressource zugeordnet ist, entspricht dem Muster `lambda-rds-`*n* oder `lambda-rdsproxy-`*n* (wobei *n* für eine Zahl steht).
- Die Sicherheitsgruppe, die der Rechenressource zugeordnet ist, enthält eine Regel für ausgehenden Datenverkehr, wobei der Portbereich auf den Port des Multi-AZ-DB-Clusters oder auf einen zugeordneten Proxy festgelegt ist. Das Ziel für die Regel für ausgehenden

Datenverkehr muss auf eine dem Multi-AZ-DB-Cluster zugeordneten Sicherheitsgruppe oder auf einen zugeordneten Proxy festgelegt werden.

- Der Name der Sicherheitsgruppe, die an den mit Ihrer Datenbank verknüpften Proxy angefügt ist, entspricht dem Muster `rds-rdsproxy-`*n* (wobei *n* für eine Zahl steht).
- Die Sicherheitsgruppe, die der Funktion zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei der Port auf den Port festgelegt ist, den der Multi-AZ-DB-Cluster oder ein zugeordneter Proxy verwendet. Das Ziel muss auf eine dem Multi-AZ-DB-Cluster zugeordneten Sicherheitsgruppe oder auf einen zugeordneten Proxy festgelegt werden.

So zeigen Sie Rechenressourcen an, die automatisch mit einem Multi-AZ-DB-Cluster verbunden sind

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Datenbanken und anschließend den Multi-AZ-DB-Cluster aus.
3. Sehen Sie sich auf der Registerkarte Konnektivität und Sicherheit die Rechenressourcen unter Verbundene Rechenressourcen an.

Ändern eines Multi-AZ-DB-Clusters für Amazon RDS

Ein Multi-AZ-DB-Cluster verfügt über eine Writer-DB-Instance und zwei Reader-DB-Instances in drei separaten Availability Zones. Multi-AZ-DB-Cluster bieten hohe Verfügbarkeit, erhöhte Kapazität für Lese-Workloads und eine geringere Latenz im Vergleich zu Multi-AZ-Bereitstellungen. Weitere Informationen zu Multi-AZ-DB-Clustern finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS.

Sie können einen Multi-AZ-DB-Cluster ändern, um seine Einstellungen zu ändern. Sie können auch Vorgänge an einem Multi-AZ-DB-Cluster ausführen, z. B. einen Snapshot davon machen.

> **Important**
>
> Sie können die DB-Instances innerhalb eines Multi-AZ-DB-Clusters nicht ändern. Alle Änderungen müssen auf DB-Cluster-Ebene vorgenommen werden. Der einzige Vorgang, den Sie auf einer DB-Instance innerhalb eines Multi-AZ-DB-Clusters ausführen können, ist deren Neustart.

Sie können einen Multi-AZ-DB-Cluster mithilfe der AWS Management Console, der oder der AWS CLI RDS-API ändern.

Konsole

Ändern eines Multi-AZ-DB-Clusters

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann den zu ändernden Multi-AZ-DB-Cluster aus.
3. Wählen Sie Ändern aus. Die Seite DB-Cluster ändern wird angezeigt.
4. Ändern Sie alle Einstellungen nach Bedarf. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen zum Ändern von Multi-AZ-DB-Clustern.
5. Nachdem Sie die gewünschten Änderungen vorgenommen haben, wählen Sie Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.
6. (Optional) Klicken Sie auf Apply immediately (Sofort anwenden), um die Änderungen direkt zu übernehmen. Die Auswahl dieser Option kann in einigen Fällen Ausfallzeiten verursachen. Weitere Informationen finden Sie unter Änderungen sofort anwenden.

7. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB cluster (DB-Cluster ändern) aus, um Ihre Änderungen zu speichern.

   Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Um einen Multi-AZ-DB-Cluster mithilfe von zu ändern AWS CLI, rufen Sie den modify-db-clusterBefehl auf. Geben Sie die DB-Cluster-ID und die Werte für die Optionen an, die Sie ändern möchten. Informationen zu den jeweiligen Optionen finden Sie unter Einstellungen zum Ändern von Multi-AZ-DB-Clustern.

Example

Mit folgendem Code wird `my-multi-az-dbcluster` geändert, da der Aufbewahrungszeitraum für Backups auf 1 Woche (7 Tage) festgelegt wird. Der Code ermöglicht den Löschschutz durch Verwendung von `--deletion-protection`. Um den Löschschutz zu deaktivieren, verwenden Sie `--no-deletion-protection`. Die Änderungen werden während des nächsten Wartungsfensters (mit) übernommen `--no-apply-immediately`. Verwenden Sie `--apply-immediately`, damit Änderungen sofort angewendet werden. Weitere Informationen finden Sie unter Änderungen sofort anwenden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier my-multi-az-dbcluster \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier my-multi-az-dbcluster ^
    --backup-retention-period 7 ^
    --deletion-protection ^
    --no-apply-immediately
```

RDS-API

Um einen Multi-AZ-DB-Cluster mithilfe der Amazon RDS-API zu ändern, rufen Sie den DBCluster Vorgang Modify auf. Geben Sie die DB-Cluster-Kennung und die Parameter für die Einstellungen an, die geändert werden sollen. Weitere Informationen zu den einzelnen Parametern finden Sie unter Einstellungen zum Ändern von Multi-AZ-DB-Clustern.

Änderungen sofort anwenden

Wenn Sie einen Multi-AZ-DB-Cluster ändern, können Sie die Änderungen sofort anwenden. Damit Änderungen sofort übernommen werden, wählen Sie die Option Apply Immediately (Sofort anwenden) in der AWS Management Console aus. Oder Sie verwenden die `--apply-immediately` Option beim Aufrufen der AWS CLI oder setzen den `ApplyImmediately` Parameter auf, `true` wenn Sie die Amazon RDS-API verwenden.

Wenn Sie sich entscheiden, die Änderungen nicht sofort zu übernehmen, werden die Änderungen in die Warteschlange für ausstehende Änderungen aufgenommen. Während des nächsten Wartungsfensters, werden alle ausstehenden Änderungen in der Warteschlange angewandt. Wenn Sie sich entscheiden, die Änderungen sofort zu übernehmen, werden alle Ihre neuen Änderungen sowie alle ausstehenden Änderungen in der Warteschlange übernommen.

> **Important**
>
> Wenn es eine der ausstehenden Änderungen erfordert, dass der DB-Cluster vorübergehend nicht verfügbar ist (Ausfallzeiten), kann die Auswahl der Option „Sofort anwenden“ zu unerwarteten Ausfallzeiten führen.
> Wenn Sie sich dafür entscheiden, eine Änderung sofort anzuwenden, werden alle anstehenden Änderungen ebenfalls sofort und nicht erst im nächsten Wartungsfenster übernommen.
> Wenn Sie nicht möchten, dass eine ausstehende Änderung im nächsten Wartungsfenster übernommen wird, können Sie die DB-Instance so abändern, dass die Änderung rückgängig gemacht wird. Sie können dies tun, indem Sie die `--apply-immediately` Option AWS CLI und angeben.

Änderungen an Datenbankeinstellungen werden unmittelbar übernommen, auch wenn Sie sich entscheiden, Ihre Änderungen auf einen späteren Zeitpunkt zu verschieben. Informationen darüber, wie die verschiedenen Datenbankeinstellungen mit der Einstellung „Apply Immediately (Sofort Anwenden)“ interagieren, finden Sie unter Einstellungen zum Ändern von Multi-AZ-DB-Clustern.

Einstellungen zum Ändern von Multi-AZ-DB-Clustern

Weitere Informationen zu Einstellungen, die Sie zum Ändern eines Multi-AZ-DB-Clusters verwenden können, finden Sie in der folgenden Tabelle. Weitere Informationen zu den AWS CLI Optionen finden Sie unter modify-db-cluster. Weitere Informationen zu den RDS-API-Parametern finden Sie unter Ändern DBCluster.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Allocated storage	Die für jede DB-Instance in Ihrem DB-Cluster zuzuweisende Speichermenge (in Gibibyte). Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.	CLI-Option:
		`--allocated-storage`
		RDS-API-Parameter:
		`Allocated Storage`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten keine Ausfallzeiten auf.

Automatische Nebenversions-Updates	Aktivieren Sie das automatische Upgrade der Nebenversion, damit Ihr DB-Cluster automatisch Upgrades der bevorzugten Nebenversion der DB-Engine erhält, wenn diese verfügbar sind. Amazon RDS führt im	CLI-Option:
		`--auto-minor-version-upgrade`
		`--no-auto-minor-version-upgrade`
		RDS-API-Parameter:
			Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
	Wartungsfenster automatische Nebenversionenupgrades durch.	`AutoMinorVersionUpgrade`		

Aufbewahrungszeitraum für Backups	Die Anzahl der Tage, an denen automatische Backups Ihres DB-Clusters aufbewahrt werden. Dieser Wert muss größer als null sein.
	
	Weitere Informationen finden Sie unter Einführung in Backups.
		CLI-Option:
		
		`--backup-retention-period`
		
		RDS-API-Parameter:
		
		`BackupRetentionPeriod`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Backup window	Der Zeitraum, in dem Amazon RDS automatisch ein Backup der DB-Cluster erstellt. Wenn Sie keine bestimmte Zeit haben, zu der Sie Ihre Datenbank sichern möchten, verwenden Sie den Standardwert No Preference (Keine Präferenz).
	
	Weitere Informationen finden Sie unter Einführung in Backups.
		CLI-Option:
		
		`--preferred-backup-window`
		
		RDS-API-Parameter:
		
		`PreferredBackupWindow`
			Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen.	Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Zertifizierungsstelle	Die Zertifizierungsstelle (CA) für das vom DB-Cluster verwendete Serverzertifikat.
	
	Weitere Informationen finden Sie unter .
		CLI-Option:
		
		`--ca-certificate-identifier`
		
		RDS-API-Parameter:
		
		`CACertificateIdentifier`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Ein Ausfall tritt nur auf, wenn die DB-Engine keine Rotation ohne Neustart unterstützt. Sie können den describe-db-engine-versions AWS CLI Befehl verwenden, um festzustellen, ob die DB-Engine die Rotation ohne Neustart unterstützt.

Tags zu Snapshot kopieren	Diese Option kopiert alle DB-Cluster-Tags in einen DB-Snapshot, wenn Sie einen Snapshot erstellen.
	
	Weitere Informationen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.
		CLI-Option:
		
		`-copy-tags-to-snapshot`
		
		`-no-copy-tags-to-snapshot`
		
		RDS-API-Parameter:
		
		`CopyTagsToSnapshot`
			Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
Datenbank-Authentifizierung	Für Multi-AZ-DB-Cluster wird nur die Passwortauthentifizierung unterstützt.	Keine, da die Passwortauthentifizierung der Standard ist.	Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf. Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.	Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

DB-Cluster-Kennung	Die DB-Cluster-ID. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.
	
	Wenn Sie die DB-Cluster-ID ändern, wird der DB-Cluster-Endpunkt geändert. Die IDs und Endpunkte der DB-Instances im DB-Cluster ändern sich ebenfalls. Der neue DB-Clustername muss eindeutig sein. Die maximale Länge beträgt 63 Zeichen.
	
	Die Namen der DB-Instances im DB-Cluster werden so geändert, dass sie dem neuen Namen des DB-Clusters
		CLI-Option:
		
		`--new-db-cluster-identifier`
		
		RDS-API-Parameter:
		
		`NewDBClusterIdentifier`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
	entsprechen. Ein neuer DB-Instance-Name darf nicht mit dem Namen einer vorhandenen DB-Instance identisch sein. Wenn Sie beispielsweise den DB-Clusternamen in `maz` ändern, wird möglicherweise ein DB-Instance-Name in `maz-instance-1` geändert. In diesem Fall darf keine DB-Instance mit dem Namen `maz-instance-1` vorhanden sein. Weitere Informationen finden Sie unter Umbenennen eines Multi-AZ-DB-Clusters für Amazon RDS.			

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

DB-Cluster-Instance-Klasse	Die Rechen- und Arbeitsspeicherkapazität jeder DB-Instance im Multi-AZ-DB-Cluster, zum Beispiel `db.r6gd.xlarge` .
	
	Wählen Sie möglichst eine DB-Instance-Klasse, die groß genug ist, um einen typischer Abfragesatz im Arbeitsspeicher halten zu können. Wenn Arbeitssätze im Arbeitsspeicher gehalten werden, kann das System das Schreiben auf die Festplatte vermeiden, was die Leistung verbessert.
	
	Weitere Informationen finden Sie unter the section
		CLI-Option:
		
		`--db-cluster-instance-class`
		
		RDS-API-Parameter:
		
		`DBClusterInstanceClass`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
	called “Verfügbarkeit der Instance-Klassen für Multi-AZ-DB-Cluster”.			

DB-Cluster-Parametergruppe	Die DB-Clusterparametergruppe, die Sie mit dem DB-Cluster verknüpfen möchten.
	
	Weitere Informationen finden Sie unter Parametergruppen für Multi-AZ-DB-Cluster.
		CLI-Option:
		
		`--db-cluster-parameter-group-name`
		
		RDS-API-Parameter:
		
		`DBClusterParameterGroupName`
			Die Änderung der Parametergruppe wird sofort übernommen.	Während dieser Änderung treten keine Ausfallzeiten auf. Wenn Sie die Parametergruppe ändern, werden die Änderungen an einigen Parametern ohne Neustart sofort auf die DB-Instances im Multi-AZ-DB-Cluster angewendet. Änderungen an anderen Parametern werden erst angewendet, nachdem die DB-Instances neu gestartet wurden.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

DB-Engine-Version	Die Version der Datenbank-Engine, die Sie verwenden möchten.	CLI-Option:
		`--engine-version`
		RDS-API-Parameter:
		`EngineVersion`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten Ausfallzeiten auf.

Löschschutz	Um zu verhindern, dass Ihr DB-Cluster gelöscht wird, können Sie die Option Löschschutz aktivieren aktivieren.
	Weitere Informationen finden Sie unter Löschen einer DB-Instance.
		CLI-Option:
		`--deletion-protection`
		`--no-deletion-protection`
		RDS-API-Parameter:
		`DeletionProtection`
			Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Wartungsfenster	Das 30-Minuten-Fenster, in dem anstehende Änderungen an Ihrem DB-Cluster durchgeführt werden. Wählen Sie No Preference (Keine Präferenz) aus, wenn der Zeitraum nicht wichtig ist.
	
	Weitere Informationen finden Sie unter Amazon RDS-Wartungsfenster.
		CLI-Option:
		
		`--preferred-maintenance-window`
		
		RDS-API-Parameter:
		
		`PreferredMaintenanceWindow`
			Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Sofern eine oder mehrere Aktionen ausstehen, die Ausfallzeiten verursachen, und Sie das Wartungsfenster auf die aktuelle Zeit ändern, werden die ausstehenden Aktionen sofort angewendet und es kommt zu Ausfallzeiten.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Hauptanmeldedaten verwalten in AWS Secrets Manager	Wählen Sie Master-Anmeldeinformationen verwalten in AWS Secrets Manager aus, um das Hauptbenutzerpasswort in Secrets Manager geheim zu verwalten.
	
	Wählen Sie optional einen KMS-Schlüssel zum Schutz des Secrets aus. Wählen Sie aus den KMS-Schlüsseln in Ihrem Konto oder geben Sie den Schlüssel eines anderen Kontos ein.
	
	Wenn RDS bereits das Hauptbenutzerpasswort für den DB-Cluster verwaltet, können Sie dieses
		CLI-Option:
		
		`--manage-master-user-password | --no-manage-master-user-password`
		
		`--master-user-secret-kms-key-id`
		
		`--rotate-master-user-password | --no-rotate-master-user-password`
		
		RDS-API-Parameter:
		
		`ManageMasterUserPassword`
		
		`MasterUserSecretKmsKeyId`
		
		`RotateMasterUserPassword`
			Wenn Sie die automatische Passwortverwaltung für Hauptbenutzer ein- oder ausschalten, erfolgt die Änderung sofort. Bei dieser Änderung wird die Einstellung zum sofortigen Anwenden ignoriert.
			
			Wenn Sie das Hauptbenutzerpasswort ändern, müssen Sie angeben, dass die Änderung sofort übernommen wird.
				Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
	Passwort mit der Option Rotate secret immediately (Sofortige Secret-Drehung) rotieren. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.			

Neues Master-Passwort	Das Passwort für das Masterbenutzerkonto.	CLI-Option:
		
		`--master-user-password`
		
		RDS-API-Parameter:
		
		`MasterUserPassword`
			Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Bereitgestellte IOPS	Die Menge von bereitgestellten IOPS (Ein-/Ausgabeoperationen pro Sekunde), die dem DB-Cluster anfänglich zugewiesen werden soll.	CLI-Option:
		`--iops`
		RDS-API-Parameter:
		`Iops`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
Öffentlicher Zugriff	Öffentlich zugänglich, um dem DB-Cluster eine öffentliche IP-Adresse zuzuweisen, was bedeutet, dass er außerhalb seiner Virtual Private Cloud (VPC) zugänglich ist. Damit der öffentliche Zugriff für ein DB-Cluster möglich ist, muss sie sich auch in einem öffentlichen Subnetz der VPC befinden. Not publicly accessible (Nicht öffentlich zugänglich), um den DB-Cluster nur innerhalb der VPC zugänglich zu machen. Weitere Informationen finden Sie unter Ausblenden einer DB-Instance	Nicht verfügbar beim Ändern eines DB-Clusters.	Die Änderung wird sofort übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
	in einer VPC vor dem Internet. Um eine Verbindung zu einem DB-Cluster von außerhalb seiner VPC herzustellen, muss der DB-Cluster öffentlich zugänglich sein. Außerdem muss der Zugriff unter Verwendung der eingehenden Regeln der Sicherheitsgruppe dem DB-Cluster gewährt werden, und andere Anforderungen müssen erfüllt sein. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.			

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit
	Wenn Ihr DB-Cluster nicht öffentlich zugänglich ist, können Sie eine AWS Site-to-Site VPN-Verbindung oder eine AWS Direct Connect Verbindung verwenden, um von einem privaten Netzwerk aus darauf zuzugreifen. Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.			

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter	Wann wird die Änderung übernommen	Hinweise zur Ausfallzeit

Speichertyp	Der Speichertyp für Ihren DB-Cluster.
	
	Es werden nur Allzweck-SSD-Speicher (gp3), bereitgestellte IOPS-Speicher (io1) und bereitgestellte IOPS-SSD-Speicher (io2) unterstützt.
	
	Weitere Informationen finden Sie unter Amazon RDS-Speichertypen.
		CLI-Option:
		
		`--storage-type`
		
		RDS-API-Parameter:
		
		`StorageType`
			Wenn Sie die Änderung sofort anwenden möchten, tritt sie sofort auf.
			
			Wenn Sie die Änderung nicht sofort anwenden möchten, tritt sie während des nächsten Wartungsfensters auf.
				Während dieser Änderung treten keine Ausfallzeiten auf.

VPC Security Group (VPC-Sicherheitsgruppe)	Die Sicherheitsgruppen, die dem DB-Cluster zugeordnet werden sollen.
	
	Weitere Informationen finden Sie unter Überblick über VPC-Sicherheitsgruppen.
		CLI-Option:
		
		`--vpc-security-group-ids`
		
		RDS-API-Parameter:
		
		`VpcSecurityGroupIds`
			Die Änderung wird asynchron zum nächstmöglichen Zeitpunkt übernommen. Diese Einstellung ignoriert die Einstellung „Apply Immediately (Sofort Anwenden)“.	Während dieser Änderung treten keine Ausfallzeiten auf.

Einstellungen, die beim Ändern von Multi-AZ-DB-Clustern nicht gelten

Die folgenden Einstellungen im AWS CLI Befehl modify-db-clusterund im RDS-API-Vorgang Ändern gelten DBCluster nicht für Multi-AZ-DB-Cluster.

Sie können diese Einstellungen auch für Multi-AZ-DB-Cluster in der Konsole nicht ändern.

AWS CLI Einstellung	RDS-API-Einstellung
`--backtrack-window`	`BacktrackWindow`
`--cloudwatch-logs-export-configuration`	`CloudwatchLogsExportConfiguration`
`--copy-tags-to-snapshot | --no-copy-tags-to-snapshot`	`CopyTagsToSnapshot`
`--db-instance-parameter-group-name`	`DBInstanceParameterGroupName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--option-group-name`	`OptionGroupName`
`--port`	`Port`
`--scaling-configuration`	`ScalingConfiguration`
`--storage-type`	`StorageType`

Aktualisierung der Engine-Version eines Multi-AZ-DB-Clusters für Amazon RDS

Amazon RDS bietet neuere Versionen jeder unterstützten Datenbank-Engine, sodass Sie Ihren Multi-AZ-DB-Cluster auf dem neuesten Stand halten können. In diesem Thema wird der Prozess des Upgrades eines Multi-AZ-DB-Clusters auf neuere Versionen erläutert.

Das Upgrade eines Multi-AZ-DB-Clusters beinhaltet die Auswahl einer neuen kompatiblen Engine-Version und die Planung möglicher Ausfallzeiten. Der Prozess gewährleistet minimale Unterbrechungen, indem die Failover-Funktionen der Multi-AZ-Architektur genutzt werden. Während des Upgrades wird zuerst die primäre Instanz aktualisiert, gefolgt von einem Failover zur Standby-Instance, um die Verfügbarkeit aufrechtzuerhalten. Zu den bewährten Methoden gehören die Durchführung von Upgrades in Zeiten mit geringem Datenverkehr, Tests in Umgebungen außerhalb der Produktion und die Überprüfung der Anwendungskompatibilität mit der neuen Version.

Themen

- Unterversion-Upgrades
- Hauptversions-Upgrades
- Upgrade eines Multi-AZ-DB-Clusters
- Aktualisierung von Read Replicas für Multi-AZ-DB-Cluster

Unterversion-Upgrades

Ein Nebenversion-Upgrade enthalten nur Änderungen, die abwärtskompatibel mit bestehenden Anwendungen sind. Wenn Sie ein Upgrade einer kleineren Version initiieren, aktualisiert Amazon RDS zunächst die Reader-DB-Instances nacheinander. Dann wird eine der Reader-DB-Instances zur neuen Writer-DB-Instance. Amazon RDS aktualisiert dann die alte Writer-Instance (die jetzt eine Reader-Instance ist).

Die Ausfallzeit während des Upgrades ist auf die Zeit beschränkt, die benötigt wird, bis eine der Reader-DB-Instances zur neuen Writer-DB-Instance wird. Diese Ausfallzeit wirkt wie ein automatischer Failover. Weitere Informationen finden Sie unter the section called “Failover eines Multi-AZ-DB-Clusters”. Beachten Sie, dass sich die Replikatverzögerung Ihres Multi-AZ-DB-Clusters auf die Ausfallzeit auswirken kann. Weitere Informationen finden Sie unter the section called “Replikatverzögerung und Multi-AZ-DB-Cluster”.

Für RDS for PostgreSQL Multi-AZ-DB-Cluster-Read Replicas aktualisiert Amazon RDS die Cluster-Mitgliedsinstanzen nacheinander. Die Clusterrollen Reader und Writer wechseln während des Upgrades nicht. Daher kann es in Ihrem DB-Cluster zu Ausfallzeiten kommen, während Amazon RDS die Cluster-Writer-Instance aktualisiert.

> **Note**
>
> Die Ausfallzeit für ein Upgrade der Nebenversion eines Multi-AZ-DB-Clusters beträgt in der Regel 35 Sekunden. Bei Verwendung mit RDS Proxy können Sie die Ausfallzeit weiter auf eine Sekunde oder weniger reduzieren. Weitere Informationen finden Sie unter Amazon-RDS-Proxy. Alternativ können Sie einen Open-Source-Datenbank-Proxy wie ProxySQL oder den AWS Advanced JDBC PgBouncerWrapper Driver verwenden.

Hauptversions-Upgrades

In Hauptversions-Upgrades Änderungen können enthalten sein, die nicht mit vorhandenen Anwendungen kompatibel sind.

Wenn Sie ein Hauptversions-Upgrade eines RDS for PostgreSQL Multi-AZ-DB-Clusters initiieren, aktualisiert Amazon RDS gleichzeitig die Reader- und Writer-Instances. Daher ist Ihr DB-Cluster möglicherweise erst verfügbar, wenn das Upgrade abgeschlossen ist.

Wenn Sie ein Hauptversions-Upgrade eines RDS for MySQL Multi-AZ-DB-Clusters initiieren, aktualisiert Amazon RDS die Cluster-Mitgliedsinstanzen nacheinander, sodass die Replikation von einer niedrigeren Engine-Version auf eine höhere erfolgt. Es ist wichtig sicherzustellen, dass Ihr Workload während eines Hauptversions-Upgrades sowohl mit der Quell- als auch mit der Ziel-Engine-Version kompatibel ist, da Engine-Versionen sich in Syntax und Funktionen unterscheiden können.

> **Note**
>
> Wie bei kleineren Versionsupgrades beträgt die Ausfallzeit für ein Upgrade der Hauptversion von RDS für MySQL in der Regel 35 Sekunden. Bei Verwendung mit RDS Proxy können Sie die Ausfallzeit weiter auf eine Sekunde oder weniger reduzieren. Weitere Informationen finden Sie unter Amazon-RDS-Proxy.

Upgrade eines Multi-AZ-DB-Clusters

Der Vorgang für das Upgrade der Engine-Version eines Multi-AZ-DB-Clusters entspricht dem Vorgang für das Upgrade der Version einer DB-Instance-Engine. Detaillierte Anweisungen finden Sie unter the section called “Upgrade der Engine-Version”. Der einzige Unterschied besteht darin, dass Sie bei der Verwendung von AWS Command Line Interface (AWS CLI) den modify-db-clusterBefehl verwenden und den `--db-cluster-identifier` Parameter (zusammen mit dem `--allow-major-version-upgrade` Parameter) angeben.

Weitere Informationen zu Haupt- und Nebenversions-Upgrades finden Sie in der folgenden Dokumentation für Ihre DB-Engine:

- the section called “Upgrades der PostgreSQL-DB-Engine”
- the section called “Upgrades der MySQL-DB-Engine”

Aktualisierung von Read Replicas für Multi-AZ-DB-Cluster

Amazon RDS aktualisiert Lesereplikate von Multi-AZ-DB-Clustern nicht automatisch. Bei Upgrades kleinerer Versionen müssen Sie zuerst alle Read Replicas manuell aktualisieren und dann den Cluster aktualisieren. Andernfalls wird das Upgrade blockiert. Bei Upgrades der Hauptversion eines Clusters ändert sich der Replikationsstatus aller Lesereplikate in Beendet. Sie müssen die Lesereplikate nach Abschluss des Upgrades löschen und neu erstellen. Weitere Informationen finden Sie unter the section called “Überwachen der Lesereplikation”.

Umbenennen eines Multi-AZ-DB-Clusters für Amazon RDS

Sie können einen Multi-AZ-DB-Cluster umbenennen, indem Sie den AWS Management Console, den AWS CLI `modify-db-cluster` Befehl oder die RDS API `ModifyDBCluster` Amazon-Operation verwenden. Das Umbenennen eines Multi-AZ-DB-Clusters kann erhebliche Auswirkungen haben. Im Folgenden finden Sie eine Liste von Überlegungen, die zu beachten sind, bevor Sie einen Multi-AZ-DB-Cluster umbenennen.

- Wenn Sie einen Multi-AZ-DB-Cluster umbenennen, ändern sich die Cluster-Endpunkte für den Multi-AZ-DB-Cluster. Diese Endpunkte ändern sich, weil sie den Namen enthalten, den Sie dem Multi-AZ-DB-Cluster zugewiesen haben. Sie können Datenverkehr von einem alten Endpunkt auf einen neuen umleiten. Weitere Informationen zu Endpunkten von Multi-AZ-DB-Clustern finden Sie unter Verbindung zu einem Multi-AZ-DB-Cluster für Amazon herstellen RDS.
- Wenn Sie einen Multi-AZ-DB-Cluster umbenennen, wird der alte DNS Name, der vom Multi-AZ-DB-Cluster verwendet wurde, gelöscht, obwohl er einige Minuten zwischengespeichert bleiben könnte. Der neue DNS Name für den umbenannten Multi-AZ-DB-Cluster wird in etwa zwei Minuten wirksam. Der umbenannte Multi-AZ-DB-Cluster ist erst verfügbar, wenn der neue Name wirksam ist.
- Sie können beim Umbenennen eines Clusters keinen vorhandenen Multi-AZ-DB-Clusternamen verwenden.
- Metriken und Ereignisse, die dem Namen eines Multi-AZ-DB-Clusters zugeordnet sind, bleiben erhalten, wenn Sie einen DB-Instance-Namen erneut verwenden.
- Multi-AZ-DB-Cluster-Tags verbleiben unabhängig von der Umbenennung beim Multi-AZ-DB-Cluster.
- DB-Cluster-Snapshots werden für einen umbenannten Multi-AZ-DB-Cluster beibehalten.

> **Note**
>
> Eine DB-Instance ist eine isolierte Datenbankumgebung, die in der Cloud ausgeführt wird. Ein Multi-AZ-DB-Cluster kann mehrere Datenbanken hosten. Informationen zum Ändern eines Datenbanknamens finden Sie in der Dokumentation für Ihre DB-Engine.

Umbenennen, um einen bestehenden Multi-AZ-DB-Cluster zu ersetzen

Zu den häufigsten Szenarien für das Umbenennen eines Multi-AZ-DB-Clusters gehören die Wiederherstellung von Daten aus einem DB-Cluster-Snapshot oder die Durchführung einer point-in-time Wiederherstellung (). PITR Durch das Umbenennen des Multi-AZ-DB-Clusters können Sie den Multi-AZ-DB-Cluster ersetzen, ohne den Anwendungscode zu ändern, der auf den Multi-AZ-DB-Cluster verweist. Führen Sie in diesen Fällen die folgenden Schritte aus:

1. Stoppen Sie den gesamten Datenverkehr an den Multi-AZ-DB-Cluster. Sie können den Datenverkehr umleiten, sodass er nicht auf die Datenbanken des Multi-AZ-DB-Clusters zugreift, oder eine andere Methode auswählen, um zu verhindern, dass Datenverkehr auf Ihre Datenbanken auf dem Multi-AZ-DB-Cluster zugreift.
2. Benennen Sie den bestehenden Multi-AZ-DB-Cluster um.
3. Erstellen Sie einen neuen Multi-AZ-DB-Cluster, indem Sie die Wiederherstellung aus einem DB-Cluster-Snapshot oder eine zeitpunktbezogene Wiederherstellung durchführen. Geben Sie dann dem neuen Multi-AZ-DB-Cluster den Namen des vorherigen Multi-AZ-DB-Clusters.

Wenn Sie den alten Multi-AZ-DB-Cluster löschen, sind Sie für die Löschung aller nicht benötigten DB-Cluster-Snapshots des alten Multi-AZ-DB-Clusters verantwortlich.

Konsole

So benennen Sie einen Multi-AZ-DB-Cluster um

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Multi-AZ-DB-Cluster aus, den Sie umbenennen möchten.
4. Wählen Sie Ändern aus.
5. Geben Sie unter Settings (Einstellungen) einen neuen Namen für DB cluster identifier (DB-Cluster-ID) ein.
6. Klicken Sie auf Weiter.
7. Wählen Sie Apply immediately, um die Änderungen sofort anzuwenden. Die Auswahl dieser Option kann in einigen Fällen einen Ausfall verursachen. Weitere Informationen finden Sie unter Änderungen sofort anwenden.

8. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify cluster (Cluster ändern) aus, um Ihre Änderungen zu speichern.

   Klicken Sie anderenfalls auf Back (Zurück), um Ihre Änderungen zu bearbeiten, oder wählen Sie Cancel (Abbrechen) aus, um Ihre Änderungen zu verwerfen.

AWS CLI

Verwenden Sie den AWS CLI Befehl modify-db-cluster, um einen Multi-AZ-DB-Cluster umzubenennen. Geben Sie für den aktuellen `--db-cluster-identifier`-Wert und den `--new-db-cluster-identifier`-Parameter den neuen Namen des Multi-AZ-DB-Clusters an.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier DBClusterIdentifier \
    --new-db-cluster-identifier NewDBClusterIdentifier
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier DBClusterIdentifier ^
    --new-db-cluster-identifier NewDBClusterIdentifier
```

RDS API

Um einen Multi-AZ-DB-Cluster umzubenennen, rufen Sie die RDS API Amazon-Operation ModifyDBCluster mit den folgenden Parametern auf:

- `DBClusterIdentifier` – der bestehende Name des DB-Clusters.
- `NewDBClusterIdentifier` – der neue Name des DB-Clusters.

Einen Multi-AZ-DB-Cluster und Reader-DB-Instances für Amazon neu starten RDS

Möglicherweise müssen Sie Ihren Multi-AZ-DB-Cluster neu starten, normalerweise aus Wartungsgründen. Wenn Sie beispielsweise bestimmte Änderungen vornehmen oder die einer DB-Cluster zugeordnete DB-Cluster-Parametergruppe ändern, starten Sie den DB-Cluster neu. Dies führt dazu, dass die Änderungen wirksam werden.

Wenn ein DB-Cluster nicht die neuesten Änderungen an der zugehörigen DB-Cluster-Parametergruppe verwendet, wird die DB-Cluster-Parametergruppe mit dem Status „Ausstehender Neustart“ AWS Management Console angezeigt. Die Parametergruppe pending-reboot führt während des nächsten Wartungsfensters nicht zu einem automatischen Neustart. Um die neuesten Parameteränderungen auf diesen DB-Cluster anzuwenden, starten Sie den DB-Cluster manuell neu. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Multi-AZ-DB-Cluster.

Durch das Neustarten eines DB-Clusters, wird der Datenbank-Engine-Service neu gestartet. Der Neustart eines DB-Clusters führt zu einem vorübergehenden Ausfall, während dessen der DB-Cluster-Status auf Neustart gesetzt wird.

Sie können Ihren DB-Cluster nicht neu starten, wenn er sich nicht im Status Verfügbar befindet. Ihre Datenbank kann aus mehreren Gründen nicht verfügbar sein. Zum Beispiel ein laufender Sicherungsvorgang oder eine zuvor von Kunden angefragte Änderung oder eine Aktion im Wartungsfenster.

Die für den Neustart Ihres DB-Clusters erforderliche Zeit hängt vom Wiederherstellungsprozess nach einem Systemabsturz, der Datenbankaktivität zum Zeitpunkt des Neustarts und dem Verhalten Ihres spezifischen DB-Clusters ab. Wir empfehlen, während eines Neustarts die Datenbankaktivitäten möglichst zu minimieren, um die Dauer des Neustarts zu verkürzen. Die Reduzierung der Datenbankaktivität reduziert die Rollback-Aktivität für übertragene Transaktionen.

> **Important**
>
> Multi-AZ-DB-Cluster unterstützen keinen Neustart mit einem Failover. Wenn Sie die Writer-Instance eines Multi-AZ-DB-Clusters neu starten, wirkt sich dies nicht auf die Reader-DB-Instances in diesem DB-Cluster aus und es tritt kein Failover auf. Wenn Sie eine Reader-DB-Instance neu starten, erfolgt kein Failover. Um ein Failover für einen Multi-AZ-DB-Cluster

durchzuführen, wählen Sie in der Konsole Failover aus, rufen Sie den AWS CLI Befehl auf oder rufen Sie den Vorgang `failover-db-cluster`auf. API `FailoverDBCluster`

Konsole

So starten Sie einen DB-Cluster neu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den Multi-AZ-DB-Cluster aus, den Sie neu starten möchten.
3. Wählen Sie unter Aktionen die Option Neustart aus.

   Die Seite Reboot DB cluster (DB-Cluster neu starten) wird angezeigt.
4. Wählen Sie Neustart, um Ihren DB-Cluster neu zu starten.

   Oder klicken Sie auf Abbrechen.

AWS CLI

Rufen Sie den reboot-db-clusterBefehl auf, um einen Multi-AZ-DB-Cluster mithilfe von neu zu starten. AWS CLI

```
aws rds reboot-db-cluster --db-cluster-identifier mymultiazdbcluster
```

RDS API

Um einen Multi-AZ-DB-Cluster mithilfe von Amazon neu zu starten RDSAPI, rufen Sie den ebootDBClusterR-Vorgang auf.

Failover eines Multi-AZ-DB-Clusters für Amazon RDS

Bei einem geplanten oder ungeplanten Ausfall Ihrer Writer-DB-Instance in einem Multi-AZ-DB-Cluster führt Amazon RDS automatisch einen Failover auf eine Reader-DB-Instance in einer anderen Availability Zone durch. Dies gewährleistet eine hohe Verfügbarkeit bei minimalen Unterbrechungen. Failovers können bei Hardwarefehlern, Netzwerkproblemen oder manuellen Anfragen auftreten. In diesem Thema werden die automatische Erkennung von Ausfällen, die Reihenfolge der Ereignisse während eines Failovers und deren Auswirkungen auf Lese- und Schreibvorgänge beschrieben. Außerdem werden bewährte Methoden zur Überwachung und Minimierung von Failover-Zeiten vorgestellt.

Die Zeit bis zum Abschluss des Failovers hängt von der Datenbankaktivität und anderen Bedingungen ab, wenn die Writer-DB-Instance nicht verfügbar war. Der Failover-Prozess dauert normalerweise unter 35 Sekunden. Das Failover wird abgeschlossen, wenn beide Reader-DB-Instances ausstehende Transaktionen vom fehlgeschlagenen Schreiber angewendet haben. Wenn der Failover abgeschlossen ist, kann es länger dauern, bis die RDS Konsole die neue Availability Zone wiedergibt.

Themen

- Automatische Failover
- Manuelles Versagen über einen Multi-AZ-DB-Cluster
- Ermitteln, ob ein Multi-AZ-DB-Cluster ausgefallen ist
- Einstellung der JVM TTL für DNS Namenssuchvorgänge

Automatische Failover

Amazon RDS wickelt Failovers automatisch ab, sodass Sie den Datenbankbetrieb so schnell wie möglich ohne administrativen Eingriff wieder aufnehmen können. Zum Ausfall wechselt die Writer-DB-Instance automatisch zu einer Reader-DB-Instance.

Manuelles Versagen über einen Multi-AZ-DB-Cluster

Wenn Sie ein manuelles Failover für einen Multi-AZ-DB-Cluster durchführen, wird RDS zunächst die primäre DB-Instance beendet. Anschließend erkennt das interne Überwachungssystem, dass die primäre DB-Instance fehlerhaft ist, und stellt eine lesbare Replikat-DB-Instance zur Verfügung. Der Failover-Prozess dauert normalerweise unter 35 Sekunden.

Sie können ein Failover eines Multi-AZ-DB-Clusters manuell durchführen, indem Sie den AWS Management Console, oder den AWS CLI verwenden. RDS API

Konsole

Manuelles Ausfallen eines Multi-AZ-DB-Clusters

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Multi-AZ-DB-Cluster aus, den Sie ausfallen möchten.
4. Wählen Sie für Aktionen die Option Failover aus.

   Die Seite für den Failover-DB-Cluster wird angezeigt.
5. Wählen Sie Failover, um das manuelle Failover zu bestätigen.

AWS CLI

Verwenden Sie den Befehl, um ein manuelles Failover eines Multi-AZ-DB-Clusters durchzuführen. AWS CLI failover-db-cluster

Example

```
aws rds failover-db-cluster --db-cluster-identifier mymultiazdbcluster
```

RDS API

Um ein manuelles Failover eines Multi-AZ-DB-Clusters durchzuführen, rufen Sie Amazon RDS API F auf ailoverDBCluster und geben Sie den `DBClusterIdentifier` an.

Ermitteln, ob ein Multi-AZ-DB-Cluster ausgefallen ist

Um festzustellen, ob Ihr Multi-AZ-DB-Cluster erfolgreich ausgeführt wurde, können Sie Folgendes tun:

- Richten Sie DB-Event-Abonnements ein, um Sie per E-Mail oder darüber zu informierenSMS, dass ein Failover eingeleitet wurde. Weitere Informationen über -Ereignisse finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.

- Zeigen Sie Ihre DB-Ereignisse mithilfe der RDS Amazon-Konsole oder mithilfe von API Operations an.
- Zeigen Sie den aktuellen Status Ihres Multi-AZ-DB-Clusters mithilfe der RDS Amazon-Konsole AWS CLI, der und der RDS API an.

Informationen darüber, wie Sie auf Failover reagieren, die Wiederherstellungszeit verkürzen können, und weitere bewährte Methoden für Amazon RDS finden Sie unterBewährte Methoden für Amazon RDS.

Einstellung der JVM TTL für DNS Namenssuchvorgänge

Der Failover-Mechanismus ändert den Domain Name System (DNS) -Datensatz der DB-Instance automatisch so, dass er auf die Reader-DB-Instance verweist. Als Ergebnis müssen alle bestehenden Verbindungen zu Ihrer DB-Instance neu hergestellt werden. In einer Java-Virtual-Machine-Umgebung (JVM) müssen Sie aufgrund der Funktionsweise des DNS Java-Caching-Mechanismus möglicherweise die Einstellungen neu JVM konfigurieren.

Die Namen der JVM Caches suchen DNS nach Namen. Wenn der einen Hostnamen in eine IP-Adresse JVM auflöst, speichert er die IP-Adresse für einen bestimmten Zeitraum im Cache, der als () bezeichnet wird. time-to-liveTTL

Da AWS Ressourcen DNS Namenseinträge verwenden, die sich gelegentlich ändern, empfehlen wir, dass Sie Ihren JVM mit einem TTL Wert von nicht mehr als 60 Sekunden konfigurieren. Dadurch wird sichergestellt, dass Ihre Anwendung, wenn sich die IP-Adresse einer Ressource ändert, die neue IP-Adresse der Ressource empfangen und verwenden kann, indem sie die erneut abfragt. DNS

Bei einigen Java-Konfigurationen TTL ist die JVM Standardeinstellung so eingestellt, dass DNS Einträge erst aktualisiert werden, wenn der JVM neu gestartet wird. Wenn sich also die IP-Adresse für eine AWS Ressource ändert, während Ihre Anwendung noch läuft, kann sie diese Ressource erst verwenden, wenn Sie die manuell neu starten JVM und die zwischengespeicherten IP-Informationen aktualisiert werden. In diesem Fall ist es wichtig, die JVM s TTL so einzustellen, dass die zwischengespeicherten IP-Informationen regelmäßig aktualisiert werden.

> **Note**
>
> Die Standardeinstellung TTL kann je nach Ihrer Version JVM und je nachdem, ob ein Sicherheitsmanager installiert ist, variieren. Viele JVMs bieten eine Standardeinstellung von TTL weniger als 60 Sekunden. Wenn Sie einen solchen JVM und keinen Sicherheitsmanager

verwenden, können Sie den Rest dieses Themas ignorieren. Weitere Informationen zu Sicherheits-Managern in Oracle finden Sie unter The Security Manager in der Oracle-Dokumentation.

Um die JVM s zu ändernTTL, legen Sie den `networkaddress.cache.ttl`Eigenschaftswert fest. Nutzen Sie dazu eine der folgenden Methoden je nach Ihrem Bedarf:

- Um den Eigenschaftswert global für alle Anwendungen festzulegenJVM, die den `networkaddress.cache.ttl` in der `$JAVA_HOME/jre/lib/security/java.security` Datei festgelegten Wert verwenden.

```
networkaddress.cache.ttl=60
```

- Um die Eigenschaft nur für Ihre Anwendung lokal festzulegen, legen Sie `networkaddress.cache.ttl` im Initialisierungscode Ihrer Anwendung fest, bevor Netzwerkverbindungen hergestellt werden.

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

Einrichtung der SQL logischen Postgre-Replikation mit Multi-AZ-DB-Clustern für Amazon RDS

Wenn Sie die SQL logische Postgre-Replikation mit Ihrem Multi-AZ-DB-Cluster verwenden, können Sie einzelne Tabellen statt der gesamten Datenbank-Instance replizieren und synchronisieren. Für die logische Replikation wird ein Veröffentlichungs- und Abonnementmodell verwendet, um Änderungen aus einer Quelle an einen oder mehrere Empfänger zu replizieren. Es funktioniert mithilfe von Änderungsdatensätzen aus dem SQL Postgre-Write-Ahead-Protokoll (). WAL Weitere Informationen finden Sie unter the section called “Logische Replikation durchführen”.

Wenn Sie einen neuen Slot für die logische Replikation auf der Writer-DB-Instance eines Multi-AZ-DB-Clusters erstellen, wird der Slot asynchron in jede Reader-DB-Instance im Cluster kopiert. Die Slots auf den Reader-DB-Instances werden kontinuierlich mit den Slots auf der Writer-DB-Instance synchronisiert.

Die logische Replikation wird für Multi-AZ-DB-Cluster unterstützt, die RDS für SQL Postgre-Version 14.8-R2 und höher sowie 15.3-R2 und höher ausgeführt werden.

> **Note**
>
> Zusätzlich zur systemeigenen SQL logischen Postgre-Replikationsfunktion unterstützen Multi-AZ-DB-Cluster, die für Postgre ausgeführt werden, auch die Erweiterung. RDS SQL `pglogical`

Weitere Informationen zur SQL logischen Postgre-Replikation finden Sie unter Logische Replikation in der Postgre-Dokumentation. SQL

Themen

- Voraussetzungen
- Einrichten der logischen Replikation
- Einschränkungen und Empfehlungen

Voraussetzungen

Um die SQL logische Postgre-Replikation für Multi-AZ-DB-Cluster zu konfigurieren, müssen Sie die folgenden Voraussetzungen erfüllen.

- Ihr Benutzerkonto muss Mitglied der `rds_superuser`-Gruppe sein und über `rds_superuser`-Rechte verfügen. Weitere Informationen finden Sie unter the section called “Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen”.
- Ihr Multi-AZ-DB-Cluster muss einer benutzerdefinierten DB-Cluster-Parametergruppe zugeordnet sein, damit Sie die im folgenden Verfahren beschriebenen Parameterwerte konfigurieren können. Weitere Informationen finden Sie unter the section called “DB-Cluster-Parametergruppen”.

Einrichten der logischen Replikation

Um eine logische Replikation für einen Multi-AZ-DB-Cluster einzurichten, aktivieren Sie bestimmte Parameter innerhalb der zugehörigen DB-Cluster-Parametergruppe und erstellen dann Slots für die logische Replikation.

> **Note**
>
> Ab Postgre SQL Version 16 können Sie Reader-DB-Instances des Multi-AZ-DB-Clusters für die logische Replikation verwenden.

So richten Sie die logische Replikation für einen RDS Multi-AZ-DB-Cluster für Postgre SQL ein

1. Öffnen Sie die benutzerdefinierte DB-Cluster-Parametergruppe, die Ihrem RDS for Postgre SQL Multi-AZ-DB-Cluster zugeordnet ist.
2. Suchen Sie im Suchfeld Parameter nach dem statischen Parameter `rds.logical_replication` und legen Sie als Wert für den Parameter 1 fest. Diese Parameteränderung kann die WAL Generierung erhöhen. Aktivieren Sie sie daher nur, wenn Sie logische Steckplätze verwenden.
3. Konfigurieren Sie im Rahmen dieser Änderung die folgenden DB-Cluster-Parameter.

   - `max_wal_senders`
   - `max_replication_slots`
   - `max_connections`

   Abhängig von Ihrer erwarteten Auslastung müssen Sie möglicherweise auch die Werte der folgenden Parameter ändern. In vielen Fällen sind die Standardwerte jedoch ausreichend.

   - `max_logical_replication_workers`

- `max_sync_workers_per_subscription`

4. Starten Sie den Multi-AZ-DB-Cluster neu, damit die Parameterwerte wirksam werden. Detaillierte Anweisungen finden Sie unter the section called “Neustarten von Multi-AZ-DB-Clustern”.
5. Erstellen Sie einen Slot für die logische Replikation auf der Writer-DB-Instance des Multi-AZ-DB-Clusters, wie unter the section called “Arbeiten mit logischen Replikations-Slots” beschrieben. Für diesen Prozess ist es erforderlich, dass Sie ein Decodier-Plugin angeben. Derzeit SQL unterstützt RDS for Postgre die `pgoutput` Plug-ins, und `test_decodingwal2json`, die im Lieferumfang von SQL Postgre enthalten sind.

   Der Slot wird asynchron in jede Reader-DB-Instance im Cluster kopiert.
6. Überprüfen Sie den Status des Slots auf allen Reader-DB-Instances des Multi-AZ-DB-Clusters. Überprüfen Sie hierfür die Ansicht `pg_replication_slots` auf allen Reader-DB-Instances und stellen Sie sicher, dass der `confirmed_flush_lsn`-Status voranschreitet, während die Anwendung aktiv logische Änderungen verarbeitet.

   Die folgenden Befehle veranschaulichen, wie der Replikationsstatus auf den Reader-DB-Instances überprüft werden kann.

```
% psql -h test-postgres-instance-2.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D8003628
(1 row)

% psql -h test-postgres-instance-3.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
```

```
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D8003628
(1 row)
```

Beenden Sie nach Abschluss Ihrer Replikationsaufgaben den Replikationsprozess, löschen Sie die Replikationsslots und deaktivieren Sie die logische Replikation. Um die logische Replikation zu deaktivieren, ändern Sie Ihre DB-Cluster-Parametergruppe und setzen Sie den Wert von `rds.logical_replication` auf `0` zurück. Starten Sie den Cluster neu, damit die Parameteränderung in Kraft tritt.

Einschränkungen und Empfehlungen

Die folgenden Einschränkungen und Empfehlungen gelten für die Verwendung der logischen Replikation mit Multi-AZ-DB-Clustern, auf denen SQL Postgre Version 16 ausgeführt wird:

- Sie können nur Writer-DB-Instances verwenden, um logische Replikationsslots zu erstellen oder zu löschen. Beispielsweise muss der `CREATE SUBSCRIPTION` Befehl den Cluster-Writer-Endpunkt in der Host-Verbindungszeichenfolge verwenden.
- Sie müssen den Cluster-Writer-Endpunkt bei jeder Tabellensynchronisation oder Resynchronisierung verwenden. Sie können beispielsweise die folgenden Befehle verwenden, um eine neu hinzugefügte Tabelle erneut zu synchronisieren:

```
Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=writer-endpoint
Postgres=>ALTER SUBSCRIPTION subscription-name REFRESH PUBLICATION
```

- Sie müssen warten, bis die Tabellensynchronisierung abgeschlossen ist, bevor Sie die Reader-DB-Instances für die logische Replikation verwenden können. Sie können die `pg_subscription_rel` Katalogtabelle verwenden, um die Tabellensynchronisation zu überwachen. Die Tabellensynchronisierung ist abgeschlossen, wenn die `srsubstate` Spalte auf ready (`r`) gesetzt ist.
- Es wird empfohlen, Instanzendpunkte für die logische Replikationsverbindung zu verwenden, sobald die erste Tabellensynchronisierung abgeschlossen ist. Der folgende Befehl reduziert

die Belastung der Writer-DB-Instance, indem die Replikation auf eine der Reader-DB-Instances verlagert wird:

```
Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=reader-instance-endpoint
```

Sie können denselben Slot nicht auf mehr als einer DB-Instance gleichzeitig verwenden. Wenn zwei oder mehr Anwendungen logische Änderungen von verschiedenen DB-Instances im Cluster replizieren, können einige Änderungen aufgrund eines Cluster-Failovers oder eines Netzwerkproblems verloren gehen. In diesen Situationen können Sie Instanzendpunkte für die logische Replikation in der Host-Verbindungszeichenfolge verwenden. Die andere Anwendung, die dieselbe Konfiguration verwendet, zeigt die folgende Fehlermeldung an:

```
replication slot slot_name is already active for PID x providing immediate feedback.
```

- Wenn Sie die `pglogical` Erweiterung verwenden, können Sie nur den Cluster-Writer-Endpunkt verwenden. Die Erweiterung hat bekannte Einschränkungen, die dazu führen können, dass bei der Tabellensynchronisierung ungenutzte logische Replikationsslots entstehen. Veraltete Replikationssteckplätze reservieren Write-Ahead-Logdateien (WAL) und können zu Speicherplatzproblemen führen.

Arbeiten mit Multi-AZ-DB-Cluster-Read Replicas für Amazon RDS

Ein DB-Cluster-Lesereplikat ist eine spezielle Art von Cluster, den Sie aus einer Quell-DB-Instance erstellen. Nachdem Sie ein Lesereplikat erstellt haben, werden in der primären DB-Instance ausgeführte Aktualisierungen asynchron in das Lesereplikat des Multi-AZ-DB-Clusters kopiert. Sie können die Arbeitslast für Ihre primären DB-Instance reduzieren, indem Sie Leseabfragen aus Ihren Anwendungen an das Lesereplikat weiterleiten. Mit Lesereplikaten können Sie die Kapazitätseinschränkungen einer einzelnen DB-Instance für leseintensive Datenbank-Workloads elastisch erweitern.

Sie können auch ein oder mehrere DB-Instance-Lesereplikate aus einem Multi-AZ-DB-Cluster erstellen. Mit DB-Instance-Lesereplikaten können Sie über die Rechen- oder I/O-Kapazität des Multi-AZ-DB-Quell-Clusters hinaus skalieren, indem Sie überschüssigen Leseverkehr an die Lesereplikate weiterleiten. Derzeit können Sie kein Lesereplikat eines Multi-AZ-DB-Clusters aus einem vorhandenen Multi-AZ-DB-Cluster erstellen.

Wenn Sie zwischen der Migration zu einem Multi-AZ-DB-Cluster mithilfe einer Read Replica oder der Erstellung einer DB-Instance-Read Replica aus einem Multi-AZ-DB-Cluster wählen, sollten Sie Ihren spezifischen Anwendungsfall und Ihre Leistungsanforderungen berücksichtigen.

Migrieren zu einem Multi-AZ-DB-Cluster mithilfe eines Lesereplikats

> Dieser Ansatz ist ideal, wenn Sie die Verfügbarkeit und Haltbarkeit Ihrer Datenbank verbessern und gleichzeitig Ausfallzeiten minimieren möchten. Durch die Verwendung einer Read Replica für den Übergang zu einem Multi-AZ-DB-Cluster können Sie den kontinuierlichen Betrieb und die Datenkonsistenz sicherstellen. Diese Methode eignet sich besonders für Produktionsumgebungen, in denen die Aufrechterhaltung der Verfügbarkeit und die Reduzierung der Auswirkungen auf Live-Workloads von entscheidender Bedeutung sind.

Erstellen eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster

> Diese Methode eignet sich, wenn Sie Lesevorgänge skalieren oder den Lesetraffic von Ihrer primären Datenbank-Instance auslagern möchten. Durch die Erstellung einer Read Replica aus einem vorhandenen Multi-AZ-DB-Cluster können Sie leseintensive Workloads verteilen und die Leistung verbessern, ohne die Stabilität der primären Instance zu beeinträchtigen.

Die Wahl des richtigen Ansatzes hängt davon ab, ob Ihre Priorität darin besteht, eine hohe Verfügbarkeit und Beständigkeit sicherzustellen oder die Leseleistung zu skalieren. Bewerten Sie Ihre Workload-Merkmale und betrieblichen Anforderungen, um eine fundierte Entscheidung zu treffen.

Themen

- Migrieren zu einem Multi-AZ-DB-Cluster mithilfe eines Lesereplikats
- Erstellen eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster

Migrieren zu einem Multi-AZ-DB-Cluster mithilfe eines Lesereplikats

Wenn Sie eine Single-AZ-Bereitstellung oder Multi-AZ-Bereitstellung einer DB-Instance zu einer Multi-AZ-Bereitstellung eines DB-Clusters mit reduzierter Ausfallzeit migrieren möchten, können Sie ein Lesereplikat des Multi-AZ-DB-Clusters erstellen. Für die Quelle geben Sie die DB-Instance in der Single-AZ-Bereitstellung oder die primäre DB-Instance in der Multi-AZ-Bereitstellung der DB-Instance an. Die DB-Instance kann während der Migration zu einem Multi-AZ-DB-Cluster Schreibtransaktionen verarbeiten.

Beachten Sie die folgenden Überlegungen, bevor Sie ein Multi-AZ-DB-Cluster-Lesereplikat erstellen:

- Die Version der Quell-DB-Instance muss Multi-AZ-DB-Cluster unterstützen. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS.
- Das Lesereplikat des Multi-AZ-DB-Clusters muss dieselbe Hauptversion wie seine Quelle und dieselbe oder eine höhere Nebenversion haben.
- Sie müssen automatische Backups für die Quell-DB-Instance aktivieren, indem Sie den Aufbewahrungszeitraum für Backups auf einen anderen Wert als 0 festlegen.
- Der zugewiesene Speicher der Quell-DB-Instance muss 100 GiB oder mehr betragen.
- RDSFür My SQL müssen `gtid-mode` sowohl die `enforce_gtid_consistency` Parameter als auch `ON` für die Quell-DB-Instance auf gesetzt sein. Sie müssen eine benutzerdefinierte Parametergruppe und keine Standardparametergruppe verwenden. Weitere Informationen finden Sie unter the section called “DB-Parametergruppen”.
- Eine aktive, langlaufende Transaktion kann den Prozess der Erstellung des Lesereplikats verlangsamen. Wir empfehlen Ihnen zu warten, bis langlaufende Transaktionen abgeschlossen sind, bevor ein Lesereplikat erstellt wird.
- Wenn Sie die Quell-DB-Instance für das Lesereplikat eines Multi-AZ-DB-Clusters löschen, wird das Lesereplikat zu einem eigenständigen Multi-AZ-DB-Cluster hochgestuft.

Erstellen und Hochstufen des Lesereplikats eines Multi-AZ-DB-Clusters

Mit dem, oder können Sie eine Multi-AZ-DB-Cluster-Lesereplik erstellen und AWS Management Console heraufstufen AWS CLI. RDS API

> **Note**
>
> Wir empfehlen dringend, dass Sie alle Read Replicas in derselben Virtual Private Cloud (VPC) auf Amazon-Basis VPC der Quell-DB-Instance erstellen.
> Wenn Sie eine Read Replica in einer anderen VPC als der Quell-DB-Instance erstellen, können sich die Bereiche Classless Inter-Domain Routing (CIDR) zwischen dem Replikat und dem Amazon-System überschneiden. RDS CIDRDurch die Überlappung wird das Replikat instabil, was sich negativ auf Anwendungen auswirken kann, die eine Verbindung mit dem Replikat herstellen. Wenn beim Erstellen des Lesereplikats eine Fehlermeldung angezeigt wird, wählen Sie eine andere Ziel-DB-Subnetzgruppe aus. Weitere Informationen finden Sie unter Arbeiten mit einer DB-Instance in einer VPC.

Konsole

Führen Sie die folgenden Schritte unter Verwendung der AWS Management Console aus, um eine Single-AZ-Bereitstellung oder Multi-AZ-Bereitstellung einer DB-Instance mithilfe eines Lesereplikats zu einem Multi-AZ-DB-Cluster zu migrieren.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Erstellen Sie das Lesereplikat des Multi-AZ-DB-Clusters.
    a. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
    b. Wählen Sie die DB-Instance aus, die Sie als Quelle für eine Read Replica verwenden möchten.
    c. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.
    d. Wählen Sie unter Availability and durability (Verfügbarkeit und Beständigkeit) die Option Multi-AZ DB cluster (Multi-AZ-DB-Cluster) aus.
    e. Geben Sie unter DB instance identifier (DB-Instance-Kennung) einen Namen für das Lesereplikat ein.
    f. In den übrigen Abschnitten geben Sie die Einstellungen für Ihren DB-Cluster an. Weitere Informationen zu einer Einstellung finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.
    g. Wählen Sie Read Replica erstellen aus.

3. Wenn Sie bereit sind, stufen Sie das Lesereplikat zu einem eigenständigen Multi-AZ-DB-Cluster hoch:

   a. Halten Sie alle Transaktionen in die Quell-DB-Instance an und warten Sie anschließend, bis alle Updates für das Lesereplikat abgeschlossen wurden.

      Datenbank-Updates werden im Lesereplikat durchgeführt, nachdem sie in der primären DB-Instance vorgenommen wurden. Diese Replikationsverzögerung kann erheblich variieren. Verwenden Sie die Metrik `ReplicaLag`, um zu bestimmen, wann alle Aktualisierungen am Lesereplikat vorgenommen wurden. Weitere Informationen zur Replikationsverzögerung finden Sie unter Überwachen der Lesereplikation.

   b. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

   c. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.

      Der Bereich Databases (Datenbanken) wird angezeigt. Jedes Lesereplikat zeigt Replica (Replikat) in der Spalte Role (Rolle) an.

   d. Wählen Sie das Lesereplikat des Multi-AZ-DB-Clusters aus, das Sie hochstufen möchten.

   e. Wählen Sie für Actions (Aktionen)Promote (Hochstufen) aus.

   f. Geben Sie auf der Seite Promote read replica (Lesereplikat hochstufen) den Aufbewahrungszeitraum und das Backup-Fenster für den neu hochgestuften DB-Cluster an.

   g. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Promote read replica (Lesereplikat hochstufen) aus.

   h. Warten Sie, bis der Status des hochgestuften Multi-AZ-DB-Clusters `Available` lautet.

   i. Weisen Sie Ihre Anwendungen an, den hochgestuften Multi-AZ-DB-Cluster zu verwenden.

   Löschen Sie optional die Single-AZ-Bereitstellung oder die Multi-AZ-Bereitstellung der DB-Instance, wenn sie nicht mehr benötigt wird. Detaillierte Anweisungen finden Sie unter Löschen einer DB-Instance.

AWS CLI

Führen Sie die folgenden Schritte unter Verwendung der AWS CLI aus, um eine Single-AZ-Bereitstellung oder Multi-AZ-Bereitstellung einer DB-Instance mithilfe eines Lesereplikats zu einem Multi-AZ-DB-Cluster zu migrieren.

1. Erstellen Sie das Lesereplikat des Multi-AZ-DB-Clusters.

   Verwenden Sie den AWS CLI Befehl `create-db-cluster`, um eine Read Replica aus der Quell-DB-Instance zu erstellen. Geben Sie für `--replication-source-identifier` den Amazon-Ressourcennamen (ARN) der Quell-DB-Instance an.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   aws rds create-db-cluster \
     --db-cluster-identifier mymultiazdbcluster \
     --replication-source-identifier arn:aws:rds:us-
   east-2:123456789012:db:mydbinstance
     --engine postgres \
     --db-cluster-instance-class db.m5d.large \
     --storage-type io1 \
     --iops 1000 \
     --db-subnet-group-name defaultvpc \
     --backup-retention-period 1
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   aws rds create-db-cluster ^
     --db-cluster-identifier mymultiazdbcluster ^
     --replication-source-identifier arn:aws:rds:us-
   east-2:123456789012:db:mydbinstance
     --engine postgres ^
     --db-cluster-instance-class db.m5d.large ^
     --storage-type io1 ^
     --iops 1000 ^
     --db-subnet-group-name defaultvpc ^
     --backup-retention-period 1
   ```

2. Halten Sie alle Transaktionen in die Quell-DB-Instance an und warten Sie anschließend, bis alle Updates für das Lesereplikat abgeschlossen wurden.

   Datenbank-Updates werden im Lesereplikat durchgeführt, nachdem sie in der primären DB-Instance vorgenommen wurden. Diese Replikationsverzögerung kann erheblich variieren. Verwenden Sie die Metrik `Replica Lag`, um zu bestimmen, wann alle Aktualisierungen am Lesereplikat vorgenommen wurden. Weitere Informationen zur Replikationsverzögerung finden Sie unter Überwachen der Lesereplikation.

3. Wenn Sie bereit sind, stufen Sie das Lesereplikat zu einem eigenständigen Multi-AZ-DB-Cluster hoch.

   Verwenden Sie den AWS CLI -Befehl `promote-read-replica-db-cluster`, um ein Lesereplikat eines Multi-AZ-DB-Clusters hochzustufen. Geben Sie für `--db-cluster-identifier` die ID des Lesereplikats des Multi-AZ-DB-Clusters an.

   ```
   aws rds promote-read-replica-db-cluster --db-cluster-identifier mymultiazdbcluster
   ```

4. Warten Sie, bis der Status des hochgestuften Multi-AZ-DB-Clusters `Available` lautet.
5. Weisen Sie Ihre Anwendungen an, den hochgestuften Multi-AZ-DB-Cluster zu verwenden.

Löschen Sie optional die Single-AZ-Bereitstellung oder die Multi-AZ-Bereitstellung der DB-Instance, wenn sie nicht mehr benötigt wird. Detaillierte Anweisungen finden Sie unter Löschen einer DB-Instance.

RDS API

Um eine Single-AZ-Bereitstellung oder eine Multi-AZ-DB-Instance-Bereitstellung mithilfe einer Read Replica auf einen Multi-AZ-DB-Cluster zu migrieren, führen Sie die folgenden Schritte mit dem aus. RDS API

1. Erstellen Sie das Lesereplikat des Multi-AZ-DB-Clusters.

   Verwenden Sie die Operation `CreateDBCluster` mit dem erforderlichen Parameter `DBClusterIdentifier`, um ein Lesereplikat des Multi-AZ-DB-Clusters zu erstellen. Geben Sie für `ReplicationSourceIdentifier` den Amazon-Ressourcennamen (ARN) der Quell-DB-Instance an.
2. Halten Sie alle Transaktionen in die Quell-DB-Instance an und warten Sie anschließend, bis alle Updates für das Lesereplikat abgeschlossen wurden.

   Datenbank-Updates werden im Lesereplikat durchgeführt, nachdem sie in der primären DB-Instance vorgenommen wurden. Diese Replikationsverzögerung kann erheblich variieren. Verwenden Sie die Metrik `Replica Lag`, um zu bestimmen, wann alle Aktualisierungen am Lesereplikat vorgenommen wurden. Weitere Informationen zur Replikationsverzögerung finden Sie unter Überwachen der Lesereplikation.
3. Wenn Sie bereit sind, stufen Sie das Lesereplikat zu einem eigenständigen Multi-AZ-DB-Cluster hoch.

Verwenden Sie die Operation `PromoteReadReplicaDBCluster` mit dem erforderlichen Parameter `DBClusterIdentifier`, um ein Lesereplikat des Multi-AZ-DB-Clusters hochzustufen. Geben Sie die ID des Lesereplikats des Multi-AZ-DB-Clusters an.

4. Warten Sie, bis der Status des hochgestuften Multi-AZ-DB-Clusters `Available` lautet.
5. Weisen Sie Ihre Anwendungen an, den hochgestuften Multi-AZ-DB-Cluster zu verwenden.

Löschen Sie optional die Single-AZ-Bereitstellung oder die Multi-AZ-Bereitstellung der DB-Instance, wenn sie nicht mehr benötigt wird. Detaillierte Anweisungen finden Sie unter Löschen einer DB-Instance.

Einschränkungen beim Erstellen eines Lesereplikats eines Multi-AZ-DB-Clusters

Die folgenden Einschränkungen gelten für die Erstellung eines Lesereplikats eines Multi-AZ-DB-Clusters aus einer Single-AZ-Bereitstellung oder einer Multi-AZ-Bereitstellung einer DB-Instance.

- Sie können keine Multi-AZ-DB-Cluster-Lesereplik in einer anderen Datenbank als der AWS-Konto , die AWS-Konto die Quell-DB-Instance besitzt, erstellen.
- Sie können keine Multi-AZ-DB-Cluster-Lesereplik in einer anderen AWS-Region als der Quell-DB-Instance erstellen.
- Sie können ein Lesereplikat eines Multi-AZ-DB-Clusters nicht auf einen bestimmten Zeitpunkt wiederherstellen.
- Die Speicherverschlüsselung muss dieselben Einstellungen für die Quell-DB-Instance und den Multi-AZ-DB-Cluster haben.
- Wenn die Quell-DB-Instance verschlüsselt ist, muss die Multi-AZ-DB-Cluster-Lesereplik mit demselben Schlüssel verschlüsselt werden. KMS
- Wenn die Quell-DB-Instance General Purpose SSD (GP3) -Speicher verwendet und weniger als 400 GiB an zugewiesenem Speicher hat, können Sie die IOPS für den Multi-AZ-DB-Cluster bereitgestellte Read Replica nicht ändern.
- Wenn Sie ein Nebenversions-Upgrade für die Quell-DB-Instance durchführen möchten, müssen Sie das Nebenversions-Upgrade zuerst auf dem Lesereplikat des Multi-AZ-DB-Clusters vornehmen.
- Wenn Sie ein kleineres Versions-Upgrade auf einer Read Replica RDS für Postgre SQL Multi-AZ-DB-Cluster durchführen, wechselt die Reader-DB-Instance nach dem Upgrade nicht zur Writer-DB-Instance. Daher kann es in Ihrem DB-Cluster zu Ausfallzeiten kommen, während Amazon die RDS Writer-Instance aktualisiert.

- Sie können kein Hauptversions-Upgrade auf einer Multi-AZ-DB-Cluster-Read Replica durchführen.
- Sie können ein Hauptversions-Upgrade auf der Quell-DB-Instance des Lesereplikats eines Multi-AZ-DB-Clusters durchführen, die Replikation auf das Lesereplikat wird jedoch angehalten und kann nicht neu gestartet werden.
- Das Lesereplikat des Multi-AZ-DB-Clusters unterstützt keine kaskadierenden Lesereplikate.
- RDSFür Postgre SQL können Read Replicas für Multi-AZ-DB-Cluster kein Failover durchführen.

Erstellen eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster

Sie können ein DB-Instance-Lesereplikat aus einem Multi-AZ-DB-Cluster erstellen, um für leseintensive Datenbank-Workloads über die Rechenkapazität oder die I/O-Kapazität des Clusters hinaus zu skalieren. Sie können diesen übermäßigen Datenverkehr an Lesevorgängen einem oder mehreren DB-Instance-Lesereplikaten zuweisen. Sie können auch Lesereplikate verwenden, um von einem Multi-AZ-DB-Cluster zu einer DB-Instance zu migrieren.

Wenn Sie ein Lesereplikat erstellen möchten, geben Sie einen Multi-AZ-DB-Cluster als Replikationsquelle an. Eine der Reader-Instances des Multi-AZ-DB-Clusters ist immer die Quelle der Replikation, nicht die Writer-Instance. Diese Bedingung stellt sicher, dass das Replikat immer mit dem Quell-Cluster synchronisiert ist, auch im Falle eines Failovers.

Themen

- Vergleichen von Reader-DB-Instances und DB-Instance-Lesereplikaten
- Überlegungen
- Erstellen eines DB-Instance-Lesereplikats
- Hochstufen des DB-Instance-Lesereplikats
- Einschränkungen für die Erstellung eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster

Vergleichen von Reader-DB-Instances und DB-Instance-Lesereplikaten

Ein DB-Instance-Lesereplikat eines Multi-AZ-DB-Clusters unterscheidet sich in den folgenden Punkten von den Reader-DB-Instances des Multi-AZ-DB-Clusters:

- Die Reader-DB-Instances fungieren als automatische Failover-Ziele, DB-Instance-Lesereplikate hingegen nicht.

- Reader-DB-Instances müssen eine Änderung gegenüber der Writer-DB-Instance bestätigen, bevor ein Commit für die Änderung ausgeführt werden kann. Bei DB-Instance-Lesereplikaten werden Updates jedoch asynchron in das Lesereplikat des Lesereplikats kopiert, ohne dass eine Bestätigung erforderlich ist.
- Reader-DB-Instances verwenden immer die gleiche Instance-Klasse, den gleichen Speichertyp und die gleiche Engine-Version wie die Writer-DB-Instance des Multi-AZ-DB-Clusters. DB-Instance-Lesereplikate müssen jedoch nicht unbedingt dieselben Konfigurationen wie der Quell-Cluster verwenden.
- Sie können ein DB-Instance-Lesereplikat zu einer eigenständigen DB-Instance hochstufen. Eine Reader-DB-Instance eines Multi-AZ-DB-Clusters können Sie nicht zu einer eigenständigen Instance hochstufen.
- Der Reader-Endpunkt leitet nur Anfragen an die Reader-DB-Instances des Multi-AZ-DB-Clusters weiter. Er leitet niemals Anfragen an ein DB-Instance-Lesereplikat weiter.

Weitere Informationen über Reader- und Writer-DB-Instances finden Sie unter the section called “Multi-AZ-DB-Cluster-Architektur”.

Überlegungen

Beachten Sie Folgendes, bevor Sie DB-Instance-Lesereplikat aus einem Multi-AZ-DB-Cluster erstellen:

- Wenn Sie das DB-Instance-Lesereplikat erstellen, muss es dieselbe Hauptversion wie seine Quelle und dieselbe oder eine höhere Nebenversion haben. Nachdem Sie das Lesereplikat erstellt haben, können Sie es optional auf eine höhere Nebenversion als den Quell-Cluster aktualisieren.
- Wenn Sie das DB-Instance-Lesereplikat erstellen, muss der zugewiesene Speicher dem zugewiesenen Speicher des Multi-AZ-DB-Quell-Clusters entsprechen. Sie können den zugewiesenen Speicher ändern, nachdem das Lesereplikat erstellt wurde.
- RDSFür My SQL muss der `gtid-mode` Parameter `ON` für den Multi-AZ-Quell-DB-Cluster auf gesetzt werden. Weitere Informationen finden Sie unter the section called “DB-Cluster-Parametergruppen”.
- Eine aktive, langlaufende Transaktion kann den Prozess der Erstellung des Lesereplikats verlangsamen. Wir empfehlen Ihnen zu warten, bis langlaufende Transaktionen abgeschlossen sind, bevor ein Lesereplikat erstellt wird.
- Wenn Sie den Multi-AZ-Quell-DB-Cluster für ein DB-Instance-Lesereplikat löschen, werden alle Lesereplikate, auf die geschrieben wird, zu eigenständigen DB-Instances hochgestuft.

Erstellen eines DB-Instance-Lesereplikats

Sie können mit, oder eine DB-Instance-Read-Replica aus einem Multi-AZ-DB-Cluster erstellen. AWS Management Console AWS CLI RDS API

> **Note**
>
> Wir empfehlen dringend, dass Sie alle Read Replicas in derselben auf Amazon basierenden Virtual Private Cloud (VPC) VPC des Multi-AZ-Quell-DB-Clusters erstellen.
> Wenn Sie eine Read Replica in einem anderen Multi-AZ-DB-Cluster als VPC dem Quell-Cluster erstellen, können sich die Bereiche Classless Inter-Domain Routing (CIDR) zwischen dem Replikat und dem System überschneiden. RDS CIDRDurch eine Überlappung wird das Replikat instabil, was sich negativ auf Anwendungen auswirken kann, die eine Verbindung mit dem Replikat herstellen. Wenn beim Erstellen des Lesereplikats eine Fehlermeldung angezeigt wird, wählen Sie eine andere Ziel-DB-Subnetzgruppe aus. Weitere Informationen finden Sie unter the section called “Arbeiten mit einer DB-Instance in einer VPC”.

Konsole

Führen Sie die folgenden Schritte über die AWS Management Console aus, um ein DB-Instance-Lesereplikat aus einem Multi-AZ-DB-Cluster zu erstellen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Multi-AZ-DB-Cluster aus, den Sie als Quelle für ein Lesereplikat verwenden möchten.
4. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.
5. Vergewissern Sie sich, dass unter Replikatquelle der richtige Multi-AZ-DB-Cluster ausgewählt ist.
6. Geben Sie unter DB-Kennung einen Namen für das Lesereplikat ein.
7. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die DB-Instance an. Weitere Informationen zu einer Einstellung finden Sie unter the section called “Verfügbare Einstellungen”.

> **Note**
>
> Der zugewiesene Speicher für das DB-Instance-Lesereplikat muss dem zugewiesenen Speicher des Multi-AZ-DB-Quell-Clusters entsprechen.

8. Wählen Sie Read Replica erstellen aus.

AWS CLI

Verwenden Sie den Befehl, um eine Read Replica einer DB-Instance aus einem Multi-AZ-DB-Cluster zu erstellen. AWS CLI `create-db-instance-read-replica` Geben Sie für `--source-db-cluster-identifier` die ID des Multi-AZ-DB-Clusters an.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier myreadreplica \
  --source-db-cluster-identifier mymultiazdbcluster
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance-read-replica ^
  --db-instance-identifier myreadreplica ^
  --source-db-cluster-identifier mymultiazdbcluster
```

RDS API

Verwenden Sie die Operation `CreateDBInstanceReadReplica`, um ein DB-Instance-Lesereplikat aus einem Multi-AZ-DB-Cluster zu erstellen.

Hochstufen des DB-Instance-Lesereplikats

Wenn Sie das DB-Instance-Lesereplikat nicht mehr benötigen, können Sie es zu einer eigenständigen DB-Instance hochstufen. Wenn Sie ein Lesereplikat hochstufen, wird die DB-Instance neu gestartet, bevor sie wieder verfügbar ist. Detaillierte Anweisungen finden Sie unter the section called “Hochstufen eines Lesereplikats”.

Wenn Sie das Lesereplikat verwenden, um eine Multi-AZ-Bereitstellung eines DB-Clusters zu einer Single-AZ- oder einer Multi-AZ-Bereitstellung einer DB-Instance zu migrieren, sollten Sie

alle Transaktionen beenden, die auf den DB-Quell-Cluster geschrieben werden. Warten Sie anschließend, bis alle Updates für das Lesereplikat abgeschlossen wurden. Datenbank-Updates werden für das Lesereplikat durchgeführt, nachdem sie in einer der Reader-DB-Instances des Multi-AZ-DB-Clusters vorgenommen wurden. Diese Replikationsverzögerung kann erheblich variieren. Verwenden Sie die Metrik `ReplicaLag`, um zu bestimmen, wann alle Aktualisierungen am Lesereplikat vorgenommen wurden. Weitere Informationen zur Replikationsverzögerung finden Sie unter the section called “Überwachen der Lesereplikation”.

Nachdem Sie das Lesereplikat hochgestuft haben, warten Sie, bis der Status der hochgestuften DB-Instance `Available` lautet, bevor Sie Ihre Anwendungen an die hochgestufte DB-Instance weiterleiten. Löschen Sie optional die Multi-AZ-Bereitstellung des DB-Clusters, wenn Sie sie nicht mehr benötigen. Detaillierte Anweisungen finden Sie unter the section called “Löschen eines Multi-AZ-DB-Clusters”.

Einschränkungen für die Erstellung eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster

Die folgenden Einschränkungen gelten für die Erstellung eines Lesereplikats eines DB-Instance-Lesereplikats aus einer Multi-AZ-Bereitstellung eines DB-Clusters.

- Sie können keine Read Replica einer DB-Instance in einem anderen System als dem erstellen AWS-Konto , dem der AWS-Konto Multi-AZ-Quell-DB-Cluster gehört.
- Sie können keine Read Replica einer DB-Instance in einem anderen Multi-AZ-DB-Cluster als AWS-Region dem Quell-DB-Cluster erstellen.
- Sie können ein DB-Instance-Lesereplikat nicht auf einen bestimmten Zeitpunkt wiederherstellen.
- Die Speicherverschlüsselung muss dieselben Einstellungen für den Multi-AZ-DB-Quell-Cluster und das DB-Instance-Lesereplikat haben.
- Wenn der Multi-AZ-Quell-DB-Cluster verschlüsselt ist, muss die Read Replica der DB-Instance mit demselben Schlüssel verschlüsselt werden. KMS
- Wenn Sie ein Nebenversions-Upgrade für den Multi-AZ-DB-Quell-Cluster durchführen möchten, müssen Sie das Nebenversions-Upgrade zuerst auf dem DB-Instance-Lesereplikat vornehmen.
- Das DB-Instance-Lesereplikat unterstützt keine kaskadierenden Lesereplikate.
- RDSFür Postgre SQL muss auf dem Multi-AZ-Quell-DB-Cluster die SQL Postgre-Version 13.11, 14.8 oder 15.2.R2 oder höher ausgeführt werden, um eine DB-Instance-Read-Replica zu erstellen.
- Sie können ein Hauptversions-Upgrade auf dem Multi-AZ-Quell-DB-Instance des Lesereplikats einer DB-Instance durchführen, die Replikation auf das Lesereplikat wird jedoch angehalten und kann nicht neu gestartet werden.

Einrichtung der externen Replikation von Multi-AZ-DB-Clustern für Amazon RDS

Sie können die Replikation zwischen einem Multi-AZ-DB-Cluster und einer Datenbank außerhalb von Amazon RDS einrichten.

Die externe Replikation ermöglicht Multi-AZ-DB-Clustern die Replikation von Daten zwischen einer RDS DB-Instance und einer externen Datenbank, entweder vor Ort oder in einer anderen Cloud-Umgebung. Dies ist vorteilhaft für die Notfallwiederherstellung, die Datenmigration und die Aufrechterhaltung der Konsistenz zwischen Systemen an verschiedenen Standorten. In diesem Abschnitt werden die Voraussetzungen für die Einrichtung der Replikation, die Konfiguration des Prozesses und wichtige Überlegungen wie Replikationslatenz, Bandbreite und Kompatibilität mit verschiedenen Datenbank-Engines behandelt.

RDSfür mein SQL

Um die externe Replikation für einen RDS for My SQL Multi-AZ-DB-Cluster einzurichten, müssen Sie binäre Protokolldateien auf den DB-Instances innerhalb des Clusters so lange aufbewahren, dass die Änderungen auf das Replikat angewendet werden, bevor Amazon die Binärprotokolldatei RDS löscht. Konfigurieren Sie dazu die Aufbewahrung von Binärprotokollen, indem Sie die `mysql.rds_set_configuration` gespeicherte Prozedur aufrufen und den Parameter angeben. `binlog retention hours` Weitere Informationen finden Sie unter the section called “binlog retention hours”.

Der Standardwert für `binlog retention hours` ist`NULL`, was bedeutet, dass Binärprotokolle nicht aufbewahrt werden (0 Stunden). Wenn Sie die externe Replikation für einen Multi-AZ-DB-Cluster einrichten möchten, müssen Sie den Parameter auf einen anderen Wert als `NULL` setzen.

Sie können die Aufbewahrung von Binärprotokollen nur von der Writer-DB-Instance des Multi-AZ-DB-Clusters aus konfigurieren, und die Einstellung wird asynchron an alle Reader-DB-Instances weitergegeben.

Darüber hinaus empfehlen wir dringend, die GTID basierte Replikation auf Ihrem externen Replikat zu aktivieren. Wenn dann eine der DB-Instances ausfällt, können Sie die Replikation von einer anderen fehlerfreien DB-Instance innerhalb des Clusters aus fortsetzen. Weitere Informationen finden Sie in der SQL Dokumentation Meine Dokumentation unter Replikation mit globalen Transaktions-Identifikatoren.

RDSfür Postgre SQL

Um die externe Replikation für einen SQL Multi-AZ-DB-Cluster RDS für Postgre einzurichten, müssen Sie die logische Replikation aktivieren. Detaillierte Anweisungen finden Sie unter the section called “SQLLogische Postgre-Replikation mit Multi-AZ-DB-Clustern”.

Löschen eines Multi-AZ-DB-Clusters für Amazon RDS

Sie können einen DB-Multi-AZ-DB-Cluster mit dem AWS Management Console AWS CLI, dem oder dem löschen. RDS API

Die zum Löschen eines Multi-AZ-DB-Clusters benötigte Zeit kann in Abhängigkeit von den folgenden Faktoren variieren:

- Der Aufbewahrungszeitraum für Backups (d. h. wie viele Backups gelöscht werden sollen).
- Wie viele Daten werden gelöscht.
- Ob ein letzter Snapshot erstellt wurde.

Der Löschschutz muss auf dem Multi-AZ-DB-Cluster deaktiviert werden, bevor Sie ihn löschen können. Weitere Informationen finden Sie unter the section called “Voraussetzungen für das Löschen einer DB-Instance”. Sie können den Löschschutz deaktivieren, indem Sie den Multi-AZ-DB-Cluster ändern. Weitere Informationen finden Sie unter the section called “Ändern eines Multi-AZ-DB-Clusters”.

Konsole

Löschen eines Multi-AZ-DB-Clusters

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann den Multi-AZ-DB-Cluster aus, den Sie löschen möchten.
3. Klicken Sie bei Actions auf Delete.
4. Wählen Sie Letzten Snapshot erstellen? um einen endgültigen DB-Snapshot für den Multi-AZ-DB-Cluster zu erstellen.

   Wenn Sie einen endgültigen Snapshot erstellen, geben Sie einen Namen für den endgültigen Snapshot-Namen ein.
5. Wählen Sie Automatische Backups beibehalten, um automatisierte Backups beizubehalten.
6. Geben Sie **`delete me`** in das Feld ein.
7. Wählen Sie Löschen.

AWS CLI

Um einen Multi-AZ-DB-Cluster mithilfe von zu löschen AWS CLI, rufen Sie den delete-db-clusterBefehl mit den folgenden Optionen auf:

- `--db-cluster-identifier`
- `--final-db-snapshot-identifier` oder `--skip-final-snapshot`

Example Mit einem letzten Snapshot

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Example Mit keinem letzten Snapshot

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --skip-final-snapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --skip-final-snapshot
```

RDS API

Um einen Multi-AZ-DB-Cluster mithilfe von Amazon zu löschen RDSAPI, rufen Sie den eleteDBClusterD-Vorgang mit den folgenden Parametern auf:

- `DBClusterIdentifier`
- `FinalDBSnapshotIdentifier` oder `SkipFinalSnapshot`

Einschränkungen von Multi-AZ-DB-Clustern für Amazon RDS

Ein Multi-AZ-DB-Cluster verfügt über eine Writer-DB-Instance und zwei Reader-DB-Instances in drei separaten Availability Zones. Multi-AZ-DB-Cluster bieten hohe Verfügbarkeit, erhöhte Kapazität für Lese-Workloads und eine geringere Latenz im Vergleich zu Multi-AZ-Bereitstellungen. Weitere Informationen zu Multi-AZ-DB-Clustern finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS.

Die folgenden Einschränkungen gelten für Multi-AZ-DB-Cluster.

- Multi-AZ-DB-Cluster unterstützen die folgenden Funktionen nicht:
  - IPv6 Verbindungen (Dual-Stack-Modus)
  - Regionsübergreifende automatisierte Backups
  - Kerberos-Authentifizierung
  - Den Port ändern. Alternativ können Sie einen Multi-AZ-DB-Cluster zu einem bestimmten Zeitpunkt wiederherstellen und einen anderen Port angeben.
  - Optionsgruppen
  - Point-in-time-recovery (PITR) für gelöschte Cluster
  - Automatische Speicherskalierung durch Einstellung des maximal zugewiesenen Speichers. Alternativ können Sie den Speicher manuell skalieren.
  - Den Multi-AZ-DB-Cluster stoppen und starten
  - Kopieren eines Snapshot eines Multi-AZ-DB-Clusters
  - Verschlüsselung eines unverschlüsselten Multi-AZ-DB-Clusters
- Multi-AZ-DB-Cluster von RDS for MySQL unterstützen nur die folgenden gespeicherten Systemprozeduren:
  - `mysql.rds_rotate_general_log`
  - `mysql.rds_rotate_slow_log`
  - `mysql.rds_show_configuration`
  - `mysql.rds_set_external_master_with_auto_position`
  - `mysql.rds_set_configuration`
- Multi-AZ-DB-Cluster von RDS für PostgreSQL unterstützen die folgenden Erweiterungen nicht: und. `aws_s3` `pg_transport`
- Multi-AZ-DB-Cluster von RDS for PostgreSQL unterstützen nicht die Verwendung eines benutzerdefinierten DNS-Servers für ausgehenden Netzwerkzugriff.

Erweiterter Support für Amazon RDS mit Amazon RDS

Mit RDS Extended Support können Sie gegen Aufpreis eine Datenbank auf einer Hauptversion der Engine auch nach Ablauf des Standard-Supports von RDS weiter betreiben.

Sie können eine Datenbank nur bei RDS Extended Support registrieren, indem Sie RDS Extended Support aktivieren, wenn Sie eine DB-Instance zum ersten Mal erstellen oder wiederherstellen. Sie können Ihren RDS Extended Support-Registrierungsstatus für bestehende DB-Instances nur aktualisieren, wenn Sie sie wiederherstellen.

Wenn Sie RDS Extended Support während der Erstellung oder Wiederherstellung einer DB-Instance aktiviert haben, registriert Amazon RDS Amazon die DB-Instance nach dem Ende des Standard-Supports für RDS Amazon automatisch bei RDS Extended Support. Die automatische Registrierung bei RDS Extended Support ändert nichts an der Datenbank-Engine und beeinträchtigt nicht die Verfügbarkeit oder Leistung Ihrer DB-Instance.

RDS Extended Support bietet die folgenden Updates und technischen Support:

- Sicherheitsupdates für kritische und wichtige Sicherheitsaktualisierungen CVEs für Ihre DB-Instance oder Ihren DB-Cluster, einschließlich der Datenbank-Engine
- Bugfixes und Patches für kritische Probleme
- Die Möglichkeit, Supportanfragen zu eröffnen und Hilfe bei der Fehlerbehebung im Rahmen des standardmäßigen Amazon RDS Service Level Agreements zu erhalten

Dieses kostenpflichtige Angebot gibt Ihnen mehr Zeit für ein Upgrade auf eine unterstützte Hauptversion der Engine. Das Enddatum des Standard-Supports für RDS für MySQL, Version 5.7, ist beispielsweise der 29. Februar 2024. Sie sind jedoch nicht bereit, vor diesem Datum manuell auf RDS for MySQL Version 8.0 zu aktualisieren. In diesem Fall registriert Amazon RDS Ihre Datenbanken am 29. Februar 2024 automatisch bei RDS Extended Support, und Sie können RDS for MySQL Version 5.7 weiterhin ausführen. Ab dem 1. März 2024 berechnet Ihnen Amazon RDS automatisch den RDS Extended Support.

RDS Extended Support ist bis zu 3 Jahre nach dem RDS-Standard-Supports für eine Hauptversion der Engine verfügbar . Wenn Sie nach dieser Zeit Ihre Haupt-Engine-Version nicht auf eine unterstützte Version aktualisiert haben, aktualisiert Amazon RDS automatisch Ihre Haupt-Engine-Version. Wir empfehlen, so schnell wie möglich auf eine unterstützte Haupt-Engine-Version zu aktualisieren.

Weitere Informationen zum Ende des Standard-Supports für RDS und zum Ende des erweiterten Supports für RDS finden Sie unter Unterstützte MySQL-Hauptversionen in Amazon RDS und im Veröffentlichungskalender für Amazon RDS for PostgreSQL.

Themen

- Überblick über Amazon RDS Extended Support
- Gebühren für Amazon RDS Extended Support
- Versionen mit Amazon RDS Extended Support
- Amazon RDS und Kundenverantwortung mit Amazon RDS Extended Support
- Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support
- Anzeige der Registrierung Ihrer DB-Instances oder Multi-AZ-DB-Cluster, Aurora-DB-Cluster in Amazon RDS Extended Support
- Support-Daten für Engine-Versionen in Amazon RDS Extended Support anzeigen
- Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support

Überblick über Amazon RDS Extended Support

Wenn Sie nach dem Ende des Standard-Supports für RDS den RDS Extended Support während der Erstellung oder Wiederherstellung Ihrer DB-Instances nicht deaktiviert haben, registriert Amazon RDS diese automatisch bei RDS Extended Support. Amazon RDS aktualisiert Ihre DB-Instance automatisch auf die letzte Nebenversion, die vor dem Ende des Standard-Supports für RDS veröffentlicht wurde, sofern Sie diese Version noch nicht ausführen. Amazon RDS aktualisiert Ihre Nebenversion erst nach Ablauf des Standard-Supportdatums für RDS für Ihre Haupt-Engine-Version.

Sie können neue Datenbanken mit wichtigen Engine-Versionen erstellen, die das Ende des Standard-Supports für RDS erreicht haben. RDS registriert diese neuen Datenbanken automatisch beim RDS Extended Support und berechnet Ihnen für dieses Angebot Gebühren.

Wenn Sie vor dem Ende des Standard-Supports für RDS ein Upgrade auf eine Engine durchführen, für die der RDS noch verfügbar ist, wird Amazon RDS Ihre Engine nicht für RDS Extended Support registrieren.

Wenn Sie versuchen, einen Snapshot einer Datenbank wiederherzustellen, die mit einer Engine kompatibel ist, deren Standardsupport für RDS abgelaufen ist, aber nicht für RDS Extended Support registriert ist, versucht Amazon RDS Amazon , den Snapshot so zu aktualisieren, dass er mit der

neuesten Engine-Version kompatibel ist, die noch unter RDS steht. Wenn die Wiederherstellung fehlschlägt, registriert Amazon RDS Ihre Engine automatisch bei RDS Extended Support mit einer Version, die mit dem Snapshot kompatibel ist.

Sie können die Registrierung für RDS Extended Support jederzeit beenden. Um die Registrierung zu beenden, aktualisieren Sie jede registrierte Engine auf eine neuere Engine-Version, die immer noch unter der Standardunterstützung von RDS steht. Das Ende der Registrierung für RDS Extended Support wird an dem Tag wirksam, an dem Sie ein Upgrade auf eine neuere Engine-Version abschließen, für die noch der RDS gilt.

Weitere Informationen zum Ende des Standard-Supports für RDS und zum Ende des erweiterten Supports für RDS finden Sie unter Unterstützte MySQL-Hauptversionen in Amazon RDS und im Veröffentlichungskalender für Amazon RDS for PostgreSQL.

Gebühren für Amazon RDS Extended Support

Ab dem Tag nach dem RDS Ende des Standard-Supportdatums fallen Gebühren für alle Engines an, die für den RDS erweiterten Support registriert sind. Das Datum für das RDS Ende des Standard-Supports finden Sie Hauptversionen im Veröffentlichungskalender für Amazon RDS for Postgre SQL. RDSFür Standby-Instances in Multi-AZ-Bereitstellungen fallen Gebühren für erweiterten Support an.

Die zusätzliche Gebühr für den RDS erweiterten Support endet automatisch, wenn Sie eine der folgenden Maßnahmen ergreifen:

- Führen Sie ein Upgrade auf eine Engine-Version durch, für die der Standardsupport gilt.
- Löschen Sie die Datenbank, auf der eine Hauptversion nach dem Ende des Standard-Supports von RDS ausgeführt wird.

Die Gebühren werden wieder aufgenommen, wenn Ihre Ziel-Engine-Version in RDS future in den erweiterten Support aufgenommen wird.

RDSFür Postgre SQL 11 beispielsweise am 1. März 2024 in den erweiterten Support aufgenommen, die Gebühren beginnen jedoch erst am 1. April 2024. Sie führen am 30. April 2024 ein Upgrade Ihrer RDSfür Postgre SQL 11-Datenbank auf RDSfür Postgre SQL 12 durch. Für Postgre SQL 11 werden Ihnen nur 30 Tage Extended Support in Rechnung gestellt. RDS Sie laufen weiterhin RDSfür Postgre SQL 12 auf dieser DB-Instance, nachdem der Standardsupport RDS am 28. Februar 2025 abgelaufen ist. Für Ihre Datenbank fallen ab dem 1. März 2025 wieder Gebühren für den RDS erweiterten Support an.

Weitere Informationen finden Sie unter Amazon RDS for My SQL pricing und Amazon RDS for SQL Postgre-pricing Amazon .

Vermeidung von Gebühren für Amazon RDS Extended Support

Sie können vermeiden, dass Ihnen der RDS erweiterte Support in Rechnung gestellt wird, indem Sie RDS daran hindern, nach Ablauf des Aurora-Supports eine DB-Instance oder einen Multi-AZ-DB-Cluster zu erstellen oder wiederherzustellen. Verwenden Sie dazu das AWS CLI oder das RDSAPI.

Geben Sie in der AWS CLIopen-source-rds-extended-support-disabled die `--engine-lifecycle-support` Option an. Geben Sie im RDS API `open-source-rds-extended-support-disabled` für den `LifeCycleSupport` Parameter an. Weitere Informationen finden Sie unter Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters, oder Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, .

Versionen mit Amazon RDS Extended Support

RDS Extended Support ist nur für Hauptversionen verfügbar. Er ist nicht für Nebenversionen verfügbar.

RDS Extended Support ist für RDS für MySQL und für RDS für PostgreSQL verfügbar. Weitere Informationen finden Sie unter Hauptversionen und im Veröffentlichungskalender für Amazon RDS for PostgreSQL in den Versionshinweisen zu Amazon RDS for PostgreSQL.

Mithilfe der AWS CLI oder der RDS-API können Sie sich auch Informationen zu den Supportdaten für Engine-Versionen anzeigen lassen. Weitere Informationen finden Sie unter Support-Daten für Engine-Versionen in Amazon RDS Extended Support anzeigen.

Versionsbenennung für Amazon RDS Extended Support

Amazon RDS wird neue Nebenversionen mit Fixes und CVE-Patches für Engines auf RDS Extended Support veröffentlichen. Weitere Informationen finden Sie unter Amazon RDS-Versionen mit erweitertem Support für RDS für MySQL und Amazon RDS Extended Support-Updates für RDS for PostgreSQL in den Versionshinweisen zu Amazon RDS for PostgreSQL.

Die Namen dieser Nebenversionen werden beispielsweise in der Form major.minor-RDS.YYYYMMDD.patch.YYYYMMDD(für RDS für MySQL) oder 5.7.44-RDS.20240208.R2.20240210 11.22-RDS.20240208.R2.20240210 (für RDS für PostgreSQL) angegeben.

Major

Für MySQL ist die Hauptversionsnummer sowohl die Ganzzahl als auch der erste Bruchteil der Versionsnummer, zum Beispiel 8.0. Ein Upgrade der Hauptversion erhöht den Hauptteil der Versionsnummer. Ein Upgrade von 5.7.44 auf 8.0.33 ist beispielsweise ein Upgrade der Hauptversion, wobei 5.7 und 8.0 die Hauptversionsnummern sind.

Für PostgreSQL ist die Hauptversionsnummer die Ganzzahl, zum Beispiel 11.

Neben-RDS.yyyymmdd

Für MySQL ist die Nebenversionsnummer der dritte Teil der Versionsnummer, zum Beispiel das 44-RDS.20240208 in5.7.44-RDS.20240208.

Für PostgeSQL ist die Nebenversionsnummer der zweite Teil der Versionsnummer, zum Beispiel das In22-RDS.20240208. 11.22-RDS.20240208

Das Datum ist das Datum, an dem Amazon RDS die Nebenversion von Amazon RDS erstellt hat.

Patch

Die Patch-Version folgt auf das Datum, an dem Amazon RDS die Amazon RDS-Nebenversion erstellt hat, z. B. R2 in 5.7.44-RDS.20240208.R2 oder11.22-RDS.20240208.R2.

Eine Amazon RDS-Patch-Version enthält wichtige Bugfixes, die einer Amazon RDS-Nebenversion nach ihrer Veröffentlichung hinzugefügt wurden.

YYYYMMDD

Das Datum ist, an dem Amazon RDS die Patch-Version erstellt hat, z. B. 20240210 in oder. 5.7.44-RDS.20240208.R2.20240210 11.22-RDS.20240208.R2.20240210

Eine veraltete Version von Amazon RDS ist ein Sicherheitspatch, der wichtige Sicherheitskorrekturen enthält, die nach der Veröffentlichung zu einer Nebenversion hinzugefügt wurden. Er enthält keine Korrekturen, die das Verhalten einer Engine ändern könnten.

Amazon RDS und Kundenverantwortung mit Amazon RDS Extended Support

Im Folgenden werden die Verantwortlichkeiten von Amazon RDS Amazon und Ihre Aufgaben im Zusammenhang mit RDS Extended Support beschrieben.

Themen

- Verantwortlichkeiten RDS
- Ihre Aufgaben

Verantwortlichkeiten RDS

Nach Ablauf des Standard-Supports von RDS stellt Amazon RDS Patches, Bugfixes und Upgrades für Engines bereit, die für den RDS erweiterten Support registriert sind. Dies gilt für bis zu 3 Jahre oder bis Sie die Engines nicht mehr verwenden, je nachdem, was zuerst eintritt.

Die Patches werden für Kritisch und Hoch CVEs gemäß den CVSS Schweregraden der National Vulnerability Database (NVD) bereitgestellt. Weitere Informationen finden Sie unter Kennzahlen zu Schwachstellen.

Ihre Aufgaben

Sie sind verantwortlich für die Installation der Patches, Bugfixes und Upgrades, die für DB-Instances oder Multi-AZ-DB-Cluster, Aurora-DB-Cluster werden, die für RDS Extended Support registriert sind. Amazon RDS behält sich das Recht vor, solche Patches, Bugfixes und Upgrades jederzeit zu ändern, zu ersetzen oder zurückzuziehen. Wenn ein Patch erforderlich ist, um Sicherheits- oder kritische Stabilitätsprobleme zu beheben, behält sich Amazon RDS das Recht vor, Ihre DB-Instances oder Multi-AZ-DB-Cluster, mit dem Patch zu aktualisieren oder zu verlangen, dass Sie den Patch installieren.

Sie sind auch dafür verantwortlich, Ihre Engine vor RDS Ablauf des Extended Support-Datums auf eine neuere Engine-Version aufzurüsten. Das RDS Ende des Extended Support-Datums liegt in der Regel 3 Jahre nach dem RDSStandard-Supportdatum . Das Datum, an dem der erweiterte Support für Ihre Datenbank-Haupt-Engine-Version RDS endet, finden Sie Hauptversionen im Veröffentlichungskalender für Amazon RDS for Postgre SQL.

Wenn Sie Ihre Engine nicht aufrüsten, versucht Amazon RDS Amazon nach RDS Ablauf des Extended Support-Datums, Ihre Engine auf eine neuere Engine-Version aufzurüsten, die vom RDS unterstützt wird. Wenn das Upgrade fehlschlägt, behält sich Amazon RDS Amazon das Recht vor, die DB-Instance oder den Multi-AZ-DB-Cluster, den Aurora-DB-Cluster , auf dem die Engine läuft, nach dem RDS des Standard-Supports zu löschen. Vorher bewahrt Amazon RDSAmazon jedoch Ihre Daten aus dieser Engine auf.

Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support

Wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, erstellen, wählen Sie in der Konsole die Option Extended Support aktivieren RDS oder verwenden Sie die Option Extended Support im AWS CLI oder den Parameter in. RDS API Wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, einen bei Amazon RDS Extended Support registrieren, ist diese für die gesamte Lebensdauer der DB-Instance oder des Multi-AZ-DB-Clusters, des oder des globalen Clusters dauerhaft im RDS Extended Support registriert.

Wenn Sie die Konsole verwenden, müssen Sie Enable RDS Extended Support auswählen. Die Einstellung ist standardmäßig nicht ausgewählt.

Wenn Sie das AWS CLI oder das verwenden RDS API und die Einstellung RDS Extended Support nicht angeben, aktiviert Amazon RDS standardmäßig RDS Extended Support. Wenn Sie mithilfe von AWS CloudFormationoder anderen Services automatisieren, sorgt dieses Standardverhalten dafür, dass Ihre Datenbank auch nach Ablauf des Standard-Supports von RDS verfügbar bleibt.

Sie können die Registrierung für RDS Extended Support verhindern, indem Sie mit AWS CLIoder die RDSAPIeine DB-Instance oder einen Multi-AZ-DB-Cluster, einen Aurora-DB-Cluster erstellen.

Themen

- RDSErweitertes Support-Verhalten
- Überlegungen zum RDS erweiterten Support
- Erstellen Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, mit RDS erweitertem Support.

RDSErweitertes Support-Verhalten

Die folgende Tabelle fasst zusammen, was passiert, wenn eine Hauptversion der Engine das Ende der RDS erreicht.

RDSStatus des erweiterten Support*	Behavior
Enabled	Amazon RDS berechnet Ihnen den RDS erweiterten Support.

RDSStatus des erweiterten Support*	Behavior
Disabled	Amazon RDS aktualisiert Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster, Ihren auf eine unterstützte Engine-Version. Dieses Upgrade findet am oder kurz nach dem Ende des Standard-Supports von RDS statt.

* In der RDS Konsole wird der Status RDS Extended Support als Ja oder Nein angezeigt. Im Feld AWS CLI oder RDS API wird der Status RDS Erweiterter Support als `open-source-rds-extended-support` oder angezeigt`open-source-rds-extended-support-disabled`.

Überlegungen zum RDS erweiterten Support

Bevor Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, erstellen, sollten Sie die folgenden Punkte berücksichtigen:

- Nach Ablauf des RDS des Standard-Supports können Sie die Erstellung einer neuen DB-Instance oder eines neuen Multi-AZ-DB-Clusters, verhindern und Gebühren für RDS erweiterten Support vermeiden. Verwenden Sie dazu das AWS CLI oder das RDSAPI. Geben Sie in der AWS CLI`open-source-rds-extended-support-disabled` die `--engine-lifecycle-support` Option an. Geben Sie im RDS API `open-source-rds-extended-support-disabled` für den `LifeCycleSupport` Parameter an. Wenn Sie angeben `open-source-rds-extended-support-disabled` und das Datum für das Ende des Standard-Supports für RDS abgelaufen ist, schlägt die Erstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, immer fehl.
- RDSExtended Support wird auf Clusterebene festgelegt. Mitglieder eines Clusters haben immer dieselbe Einstellung für RDS Extended Support in der RDS Konsole AWS CLI, `--engine-lifecycle-support` `EngineLifecycleSupport` im und im RDSAPI.

Weitere Informationen finden Sie unter MySQL-Versionen und Veröffentlichungskalender für Amazon RDS for Postgre SQL.

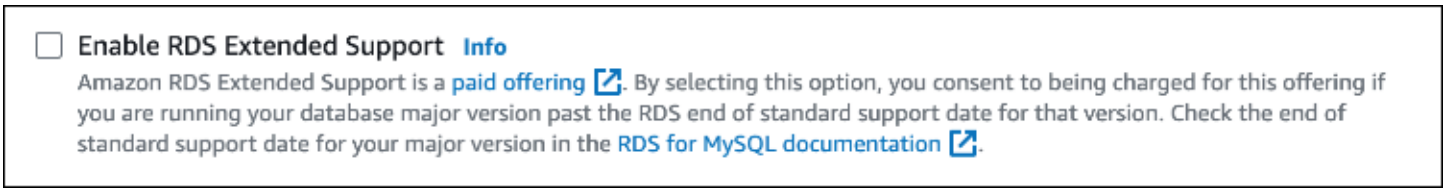
Erstellen Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, mit RDS erweitertem Support.

Sie können eine DB-Instance oder einen Multi-AZ-DB-Cluster, mit einer RDS Extended Support-Version erstellen, indem Sie die AWS Management Console AWS CLI, oder die RDS API verwenden.

Konsole

Wenn Sie , eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen, wählen Sie im Abschnitt Engine-Optionen die Option Enable RDS Extended Support aus. Diese Einstellung ist standardmäßig nicht ausgewählt.

Die folgende Abbildung zeigt die Einstellung „RDSExtended Support aktivieren“:



AWS CLI

Wenn Sie den AWS CLI Befehl (Multi-AZ-DB-Cluster) ausführen, wählen Sie RDS Extended Support aus, indem Sie `open-source-rds-extended-support` für die `--engine-lifecycle-support` Option angeben. Standardmäßig ist diese Option auf `open-source-rds-extended-support` eingestellt.

Um die Erstellung einer neuen DB-Instance oder eines Multi-AZ-DB-Clusters, nach Ablauf des RDS zu verhindern, geben Sie `open-source-rds-extended-support-disabled` für die `--engine-lifecycle-support` Option an. Auf diese Weise vermeiden Sie alle damit verbundenen Gebühren für RDS erweiterten Support.

RDS API

Wenn Sie den RDS API Amazon-Betrieb C reateDBInstance oder C reateDBCluster (Multi-AZ DB-Cluster) verwenden, wählen Sie RDS Extended Support aus, indem Sie den `EngineLifecycleSupport` Parameter auf `open-source-rds-extended-support` setzen. Dieser Parameter ist standardmäßig auf `open-source-rds-extended-support` festgelegt.

Um die Erstellung einer neuen DB-Instance oder eines Multi-AZ-DB-Clusters, nach Ablauf des RDS zu verhindern, geben Sie `open-source-rds-extended-support-disabled` für den

`EngineLifecycleSupport` Parameter an. Auf diese Weise vermeiden Sie alle damit verbundenen Gebühren für RDS erweiterten Support.

Weitere Informationen finden Sie unter den folgenden Themen:

- Um eine DB-Instance zu erstellen, befolgen Sie die Anweisungen für Ihre spezifische DB-Engine in Erstellen einer Amazon RDS-DB-Instance.
- Befolgen Sie die Anweisungen für Ihre DB-Engine unter Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen, um einen Multi-AZ-DB-Cluster zu erstellen.

Anzeige der Registrierung Ihrer DB-Instances oder Multi-AZ-DB-Cluster, Aurora-DB-Cluster in Amazon RDS Extended Support

Sie können die Registrierung Ihrer DB-Instances oder Multi-AZ-DB-Cluster, Aurora-DB-Cluster in RDS Extended Support mithilfe der AWS Management Console, der oder der AWS CLI RDS-API anzeigen.

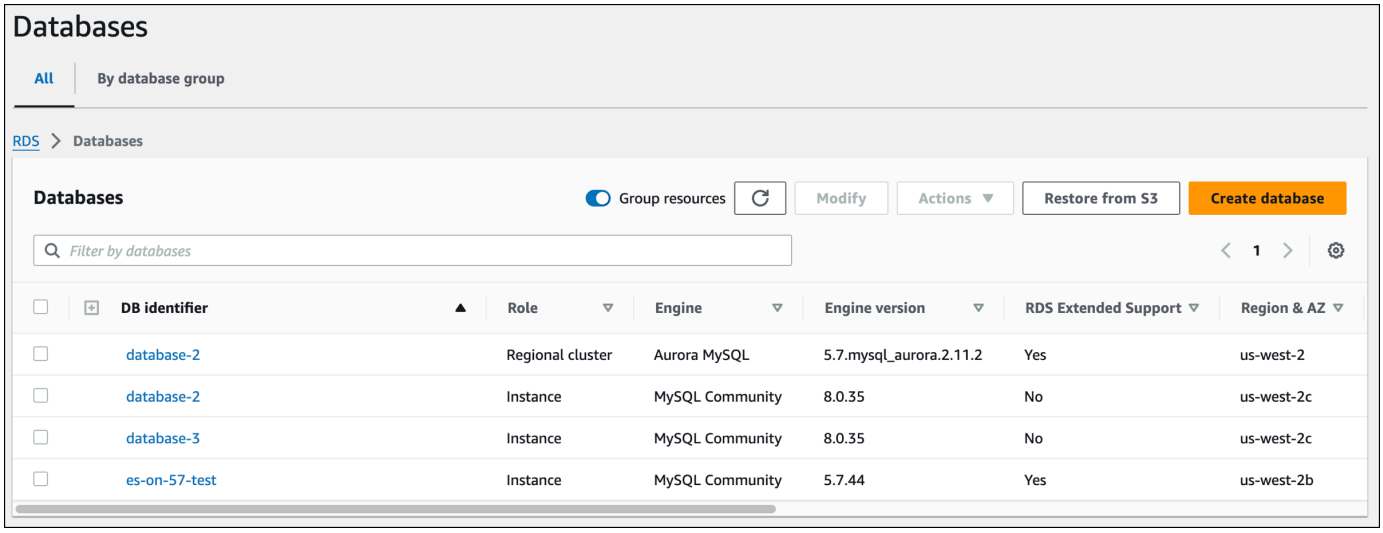
> **Note**
>
> Die Spalte RDS Extended Support in der Konsole, die `-engine-lifecycle-support` Option - in der AWS CLI und der `EngineLifecycleSupport` Parameter in der RDS-API geben nur die Registrierung für RDS Extended Support an. Die Gebühren für RDS Extended Support beginnen erst, wenn Ihre DB-Engine-Version das Ende des RDS-Standardsupports erreicht hat. Weitere Informationen finden Sie unter Hauptversionen und im Veröffentlichungskalender für Amazon RDS for PostgreSQL in den Versionshinweisen zu Amazon RDS for PostgreSQL.
> Sie haben beispielsweise eine RDS for MySQL 5.7-Datenbank, die bei RDS Extended Support registriert ist. Am 1. März 2024 begann Amazon RDS, Ihnen den erweiterten RDS-Support für diese Datenbank in Rechnung zu stellen. Am 31. Juli 2024 haben Sie diese Datenbank auf RDS für MySQL 8.0 aktualisiert. Der RDS Extended Support-Status für diese Datenbank bleibt aktiviert. Die Gebühren für RDS Extended Support für diese Datenbank wurden jedoch eingestellt, da RDS for MySQL 8.0 das Ende der Standardunterstützung für RDS noch nicht erreicht hatte. Amazon RDS berechnet Ihnen den erweiterten RDS-Support für diese Datenbank erst am 1. August 2026, wenn der RDS-Standardsupport für RDS for MySQL 8.0 endet.

Konsole

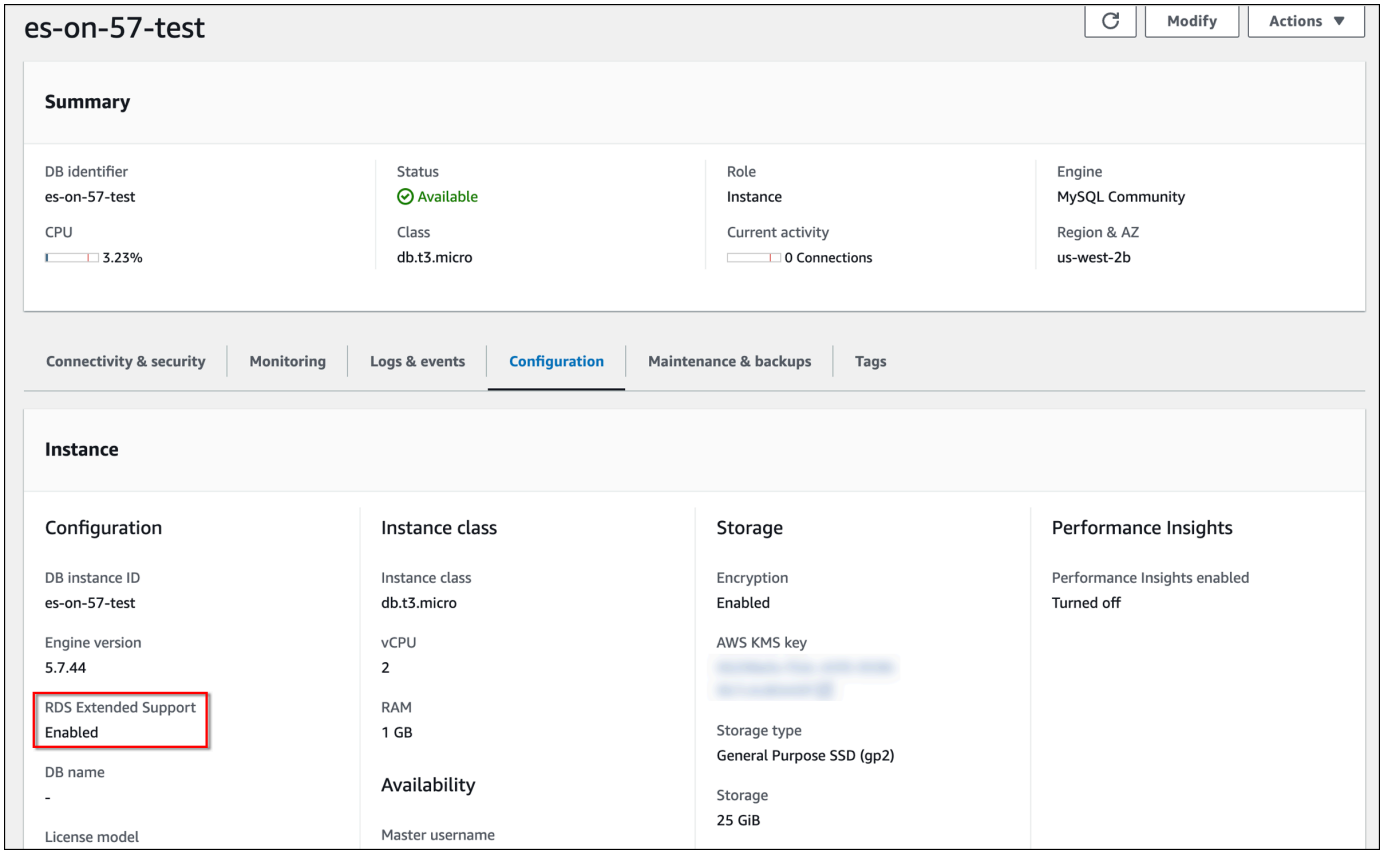
So zeigen Sie die Registrierung Ihrer DB-Instances oder Multi-AZ-DB-Cluster, Aurora-DB-Cluster in RDS Extended Support an

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus. Der Wert unter RDS Extended Support gibt an, ob eine DB-Instance oder ein Multi-AZ-DB-Cluster, für RDS Extended Support registriert ist. Wenn kein Wert angezeigt wird, ist RDS Extended Support für Ihre Datenbank nicht verfügbar.

> **Tip**
>
> Wenn die Spalte RDS Extended Support nicht angezeigt wird, wählen Sie das Symbol Einstellungen und aktivieren Sie dann RDS Extended Support.



3. Sie können die Registrierung auch auf der Registerkarte Konfiguration für jede Datenbank einsehen. Wählen Sie unter DB-ID eine Datenbank aus. Suchen Sie auf der Registerkarte Konfiguration unter Erweiterter Support nach, ob die Datenbank registriert ist oder nicht.



AWS CLI

Um die Registrierung Ihrer Datenbanken bei RDS Extended Support mithilfe des Befehls oder oder (Multi-AZ DB Clusters) anzuzeigen AWS CLI, führen Sie den Befehl (Multi-AZ DB Clusters) aus.

Wenn RDS Extended Support für eine Datenbank verfügbar ist, enthält die Antwort den ParameterEngineLifecycleSupport. Der Wert `open-source-rds-extended-support` gibt an, dass eine DB-Instance oder ein Multi-AZ-DB-Cluster, bei RDS Extended Support registriert ist. Der Wert `open-source-rds-extended-support-disabled` gibt an, dass die Registrierung der DB-Instance oder des Multi-AZ-DB-Clusters, des Aurora-DB-Clusters in RDS Extended Support deaktiviert wurde.

Beispiel

Der folgende Befehl gibt Informationen für alle Ihre DB-Instances zurück:

```
aws rds describe-db-instances
```

Die folgende Antwort zeigt, dass eine PostgreSQL-Engine, die auf der DB-Instance ausgeführt wird, bei RDS Extended Support registriert `database-1` ist:

```
{
    "DBInstanceIdentifier": "database-1",
    "DBInstanceClass": "db.t3.large",
    "Engine": "postgres",
    ...
    "EngineLifecycleSupport": "open-source-rds-extended-support"
}
```

RDS-API

https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_DescribeDBClusters.html

Wenn RDS Extended Support für eine Datenbank verfügbar ist, enthält die Antwort den Parameter`EngineLifecycleSupport`. Der Wert `open-source-rds-extended-support` gibt an, dass eine DB-Instance oder ein Multi-AZ-DB-Cluster, bei RDS Extended Support registriert ist. Der Wert `open-source-rds-extended-support-disabled` gibt an, dass die Registrierung der DB-Instance oder des Multi-AZ-DB-Clusters, des Aurora-DB-Clusters in RDS Extended Support deaktiviert wurde.

Support-Daten für Engine-Versionen in Amazon RDS Extended Support anzeigen

Sie können Informationen zu Support-Daten für Engine-Versionen für Ihre DB-Instances oder Multi-AZ-DB-Cluster, in Amazon RDS Extended Support anzeigen, indem Sie die AWS CLI oder die RDS-API verwenden. Diese Informationen können Ihnen bei der Planung von Upgrades helfen.

AWS CLI Befehle und RDS-API-Operationen geben Start- und Enddaten für RDS und RDS Extended Support zurück. Diese Daten finden Sie auch in den Tabellen mit den wichtigsten Engine-Versionen. Weitere Informationen finden Sie unter Unterstützte MySQL-Hauptversionen in Amazon RDS und im Veröffentlichungskalender für Amazon RDS for PostgreSQL in den Versionshinweisen zu Amazon RDS for PostgreSQL.

AWS CLI

Führen Sie den Befehl describe-db-major-engine-versions aus, um die Start- und Enddaten für RDS Standard Support und RDS Extended Support für Ihre wichtigsten Engine-Versionen mit dem AWS CLI anzuzeigen.

Dieser Befehl gibt die folgenden relevanten Parameter für Open-Source-Engines (MariaDB, MySQL und PostgreSQL) zurück. Für kommerzielle Engines (Db2, SQL Server und Oracle) werden diese Parameter nicht zurückgegeben.

- `SupportedEngineLifecycles`— Dieser Parameter ist eine Reihe von Objekten, die `LifecycleSupportName``LifecycleSupportStartDate`, und enthalten`LifecycleSupportEndDate`.
- `LifecycleSupportName`— Dieser Parameter gibt die Art der Unterstützung an, in der sich die Engine-Version befindet: RDS Standard Support (`open-source-rds-standard-support`) oder RDS Extended Support (`open-source-rds-extended-support`). Für MariaDB gibt dieser Parameter nur die RDS-Standardunterstützung () `open-source-rds-standard-support` zurück.
- `LifecycleSupportStartDate`— Dieser Parameter listet das Startdatum für entweder den RDS oder den RDS Extended Support für die Hauptversion der Engine auf, abhängig vom Wert von`LifecycleSupportName`.
- `LifecycleSupportEndDate`— Dieser Parameter listet das Enddatum für entweder den RDS oder den RDS Extended Support für die Hauptversion der Engine auf, abhängig vom Wert von`LifecycleSupportName`.

Beispiel

Das Antwortbeispiel zeigt die folgenden Informationen:

- Die Start- und Enddaten für die unterstützten Engine-Lebenszyklen `open-source-rds-standard-support` und `open-source-rds-extended-support` für MySQL 5.7. RDS Extended Support ist für MySQL 5.7 verfügbar.
- Die Start- und Enddaten für den unterstützten Engine-Lebenszyklus `open-source-rds-standard-support` für MariaDB 10.6. RDS Extended Support ist für MariaDB 10.6 nicht verfügbar.
- Keine Informationen zu den unterstützten Engine-Lebenszyklen für SQL Server Enterprise Edition 13, da es sich bei SQL Server um eine kommerzielle Engine handelt.

```
{
    "DBMajorEngineVersions": [
        {
            "Engine": "mysql",
            "MajorEngineVersion": "5.7",
            "SupportedEngineLifecycles": [
                {
                    "LifecycleSupportName": "open-source-rds-standard-support",
                    "LifecycleSupportStartDate": "2016-02-22T00:00:00+00:00",
                    "LifecycleSupportEndDate": "2024-02-29T23:59:59.999000+00:00"
                },
                {
                    "LifecycleSupportName": "open-source-rds-extended-support",
                    "LifecycleSupportStartDate": "2024-03-01T00:00:00+00:00",
                    "LifecycleSupportEndDate": "2027-02-28T23:59:59.999000+00:00"
                }
            ]
        },
        {
            "Engine": "mariadb",
            "MajorEngineVersion": "10.6",
            "SupportedEngineLifecycles": [
                {
                    "LifecycleSupportName": "open-source-rds-standard-support",
                    "LifecycleSupportStartDate": "2022-02-03T00:00:00+00:00",
                    "LifecycleSupportEndDate": "2026-07-31T23:59:59.999000+00:00"
                }
            ]
        },
        {
            "Engine": "sqlserver-ee",
            "MajorEngineVersion": "13.00"
        },
        ...
    ]
}
```

RDS-API

Verwenden Sie den DBMajor EngineVersions Vorgang Describe, um die Start- und Enddaten für RDS und RDS Extended Support für Ihre wichtigsten Engine-Versionen mithilfe der RDS-API anzuzeigen.

Wenn RDS Extended Support für eine Engine-Version verfügbar ist, enthält die Antwort den Parameter `SupportedEngineLifeCycles` als Array mit zwei Objekten. Ein Objekt enthält die Start- und Enddaten für die Standardunterstützung von RDS . Das zweite Objekt enthält das Start- und Enddatum für RDS Extended Support.

Wenn RDS Extended Support für eine Open-Source-Engine-Version (MariaDB, MySQL und PostgreSQL) nicht verfügbar ist, enthält die Antwort nur den Parameter `SupportedEngineLifeCycles` als Array mit einem einzigen Objekt. Dieses Objekt enthält das Start- und Enddatum für die RDS-Standardunterstützung.

Wenn die Engine-Version für eine kommerzielle Engine (Db2, SQL Server und Oracle) bestimmt ist, enthält die Antwort den Parameter `SupportedEngineLifeCycles` nicht.

Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support

Wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, wiederherstellen, wählen Sie in der Konsole die Option Extended Support aktivieren RDS oder verwenden Sie die Option Extended Support im AWS CLI oder den Parameter in. RDS API Wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, für RDS Extended Support registrieren, wird diese für die gesamte Lebensdauer der DB-Instance oder des Multi-AZ-DB-Clusters, des oder des globalen Clusters dauerhaft im RDS Extended Support registriert.

Die Standardeinstellung für die RDS Extended Support-Einstellung hängt davon ab, ob Sie die Konsole AWS CLI, die oder die RDS API zum Wiederherstellen Ihrer Datenbank verwenden. Wenn Sie die Konsole verwenden, RDSExtended Support aktivieren nicht auswählen und die Haupt-Engine-Version, die Sie wiederherstellen, das RDS des Standard-Supports überschritten hat, aktualisiert Amazon Ihre DB-Instance RDS automatisch auf eine neuere Engine-Version. Wenn Sie das AWS CLI oder das verwenden RDS API und die Einstellung für den RDS erweiterten Support nicht angeben, aktiviert Amazon RDS standardmäßig den RDS erweiterten Support. Wenn Sie mithilfe von AWS CloudFormationoder anderen Services automatisieren, sorgt dieses Standardverhalten dafür, dass Ihre Datenbank auch nach Ablauf des Standard-Supports von RDS verfügbar bleibt. Sie können den RDS erweiterten Support deaktivieren, indem Sie den AWS CLI oder den verwenden RDSAPI.

Themen

- RDSErweitertes Support-Verhalten

- Überlegungen zum RDS erweiterten Support
- Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit RDS erweitertem Support

RDSErweitertes Support-Verhalten

In der folgenden Tabelle wird zusammengefasst, was passiert, wenn eine Haupt-Engine-Version einer DB-Instance oder eines Multi-AZ-DB-Clusters, , den Sie wiederherstellen, das RDS der Standardunterstützung erreicht hat.

RDSStatus des erweiterten Support*	Behavior
Enabled	Amazon RDS berechnet Ihnen Gebühren für den RDS erweiterten Support.
Deaktiviert **	Nach Abschluss der Wiederherstellung aktualisiert Amazon RDS automatisch Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster, Ihren auf eine neuere Engine-Version (in einem future Wartungsfenster).

* In der RDS Konsole wird der Status RDS Extended Support als Ja oder Nein angezeigt. Im Feld AWS CLI oder RDS API wird der Status RDS Erweiterter Support als `open-source-rds-extended-support` oder angezeigt`open-source-rds-extended-support-disabled`.

** Diese Option ist nur verfügbar, wenn eine DB-Instance oder ein Multi-AZ-DB-Cluster wiederhergestellt wird, auf dem Postgre SQL 12 und höher oder My SQL 8 und höher ausgeführt wird.

Überlegungen zum RDS erweiterten Support

Bevor Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster, wiederherstellen, sollten Sie die folgenden Punkte berücksichtigen:

- Wenn Sie nach Ablauf des RDS des Standard-Supports eine DB-Instance oder einen Multi-AZ-DB-Cluster, einen von Amazon S3 wiederherstellen möchten, können Sie dies nur tun, indem Sie die AWS CLI oder die RDS API verwenden. Verwenden Sie die `--engine-lifecycle-support`

Option im AWS CLI Befehl restore-db-cluster-from-s3 oder den `EngineLifecycleSupport` Parameter in der Operation R estoreDBCluster fromS3. RDS API

- Wenn Sie verhindern möchten, dass RDS Ihre Datenbanken auf RDS Extended Support-Versionen zurücksetzt, geben Sie `open-source-rds-extended-support-disabled` im AWS CLI oder im an RDSAPI. Auf diese Weise vermeiden Sie alle damit verbundenen Gebühren für den RDS erweiterten Support.

  Wenn Sie diese Einstellung angeben, aktualisiert RDSAmazon Ihre wiederhergestellte Datenbank automatisch auf eine neuere, unterstützte Hauptversion. Wenn das Upgrade die Prüfungen vor dem Upgrade nicht bestanden hat, führt RDSAmazon ein sicheres Rollback zur Version der RDS Extended Support Engine durch. Diese Datenbank bleibt im RDS erweiterten Support-Modus, und Amazon RDS Amazon berechnet Ihnen den RDS erweiterten Support in Rechnung, bis Sie Ihre Datenbank manuell aktualisieren.

  Wenn Sie beispielsweise einen My SQL 5.7-Snapshot wiederherstellen, ohne den RDS erweiterten Support zu nutzen, versucht AmazonRDS, Ihre Datenbank automatisch auf My SQL 8.0 zu aktualisieren. Wenn dieses Upgrade aufgrund eines Problems fehlschlägt, das Sie lösen müssen, RDS führt Amazon ein Rollback der Datenbank auf My SQL 5.7 durch. Amazon RDS behält den RDS erweiterten Support für die Datenbank bei, bis Sie das Problem beheben können. Beispielsweise kann ein Upgrade fehlschlagen, weil nicht genügend Speicherplatz zur Verfügung steht. Nachdem Sie das Problem behoben haben, müssen Sie das Upgrade einleiten. Nach dem ersten Versuch, Ihre Datenbank zu aktualisieren, versucht Amazon RDS nicht, sie erneut zu aktualisieren.
- RDSExtended Support wird auf Clusterebene festgelegt. Mitglieder eines Clusters haben immer dieselbe Einstellung für RDS Extended Support in der RDS Konsole AWS CLI, `--engine-lifecycle-support` `EngineLifecycleSupport` im und im RDSAPI.

Weitere Informationen finden Sie unter MySQL-Versionen und Veröffentlichungskalender für Amazon RDS for Postgre SQL.

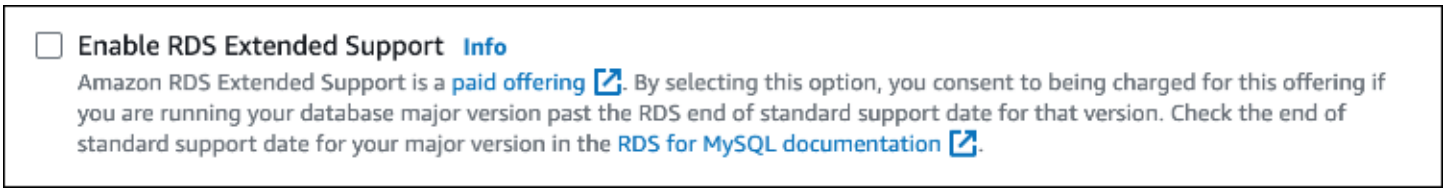
Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit RDS erweitertem Support

Sie können eine DB-Instance oder einen Multi-AZ-DB-Cluster, mit einer RDS Extended Support-Version wiederherstellen, indem Sie die AWS Management Console AWS CLI, oder die RDS API verwenden.

Konsole

Wenn Sie , eine DB-Instance oder einen Multi-AZ-DB-Cluster wiederherstellen, wählen Sie im Abschnitt Engine-Optionen die Option RDSErweiterte Support aktivieren aus. Wenn Sie diese Einstellung nicht auswählen und die Haupt-Engine-Version, die Sie wiederherstellen, das RDS Ihren Ihren Multi-AZ-DB-Cluster automatisch auf eine Version mit RDS.

Die folgende Abbildung zeigt die Einstellung „RDSExtended Support aktivieren“:



AWS CLI

Wenn Sie den AWS CLI Befehl oder restore-db-cluster-from-snapshot ausführen, wählen Sie RDS Extended Support aus, indem Sie für die Option angeben. `open-source-rds-extended-support` `--engine-lifecycle-support`

Wenn Sie Gebühren im Zusammenhang mit RDS Extended Support vermeiden möchten, setzen Sie die `--engine-lifecycle-support` Option auf`open-source-rds-extended-support-disabled`. Standardmäßig ist diese Option auf eingestellt`open-source-rds-extended-support`.

Sie können diesen Wert auch mit den folgenden AWS CLI Befehlen angeben:

- restore-db-cluster-from-s3
- restore-db-cluster-to-point-in-time
- restore-db-instance-from-3
- restore-db-instance-to-point-in-time

RDS API

Wenn Sie die RDS API Operation F romDBSnapshot oder R estoreDBCluster FromSnapshot Amazon verwenden, wählen Sie RDS Extended Support, indem Sie den `EngineLifecycleSupport` Parameter auf setzen`open-source-rds-extended-support`.

Wenn Sie Gebühren im Zusammenhang mit RDS Extended Support vermeiden möchten, setzen Sie den `EngineLifecycleSupport` Parameter auf`open-source-rds-extended-support-`

`disabled`. Dieser Parameter ist standardmäßig auf `open-source-rds-extended-support` festgelegt.

Sie können diesen Wert auch mithilfe der folgenden RDS API Operationen angeben:

- R estoreDBCluster aus S3
- RestoreDBClusterToPointInTime
- R estoreDBInstance aus S3
- RestoreDBInstanceToPointInTime

Weitere Informationen zur Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters finden Sie in Wiederherstellung auf einer DB-Instance den Anweisungen für Ihre DB-Engine unter.

Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates

Bei einer Blau/Grün-Bereitstellung wird eine Produktionsdatenbankumgebung in eine separate, synchronisierte Staging-Umgebung kopiert. Mithilfe von Amazon RDS Blue/Green Deployments können Sie Änderungen an der Datenbank in der Staging-Umgebung vornehmen, ohne die Produktionsumgebung zu beeinträchtigen. Sie können beispielsweise die Haupt- oder Nebenversion der DB-Engine aktualisieren oder Datenbankparameter in der Staging-Umgebung ändern. Wenn Sie bereit sind, können Sie die Staging-Umgebung zur neuen Produktionsdatenbankumgebung heraufstufen, wobei die Ausfallzeiten in der Regel unter einer Minute liegen.

> **Note**
>
> Derzeit werden Blue/Green-Bereitstellungen nur für RDS für MariaDB, RDS für MySQL und RDS für PostgreSQL unterstützt. Informationen zur Verfügbarkeit von Amazon Aurora finden Sie unter Verwenden von Amazon Aurora Blue/Green Deployments für Datenbank-Updates im Amazon Aurora Aurora-Benutzerhandbuch.

Themen

- Überblick über Amazon RDS Amazon
- Eine blaue/grüne Bereitstellung in Amazon RDS erstellen
- Eine blaue/grüne Bereitstellung in Amazon RDS anzeigen
- Umschalten einer blauen/grünen Bereitstellung in Amazon RDS
- Löschen einer blauen/grünen Bereitstellung in Amazon RDS

Überblick über Amazon RDS Amazon

Mithilfe von Amazon RDS Blue/Green Deployments können Sie Datenbankänderungen vornehmen und testen, bevor Sie sie in einer Produktionsumgebung implementieren. Mit einer Blau/Grün-Bereitstellung wird eine Staging-Umgebung erstellt, die die Produktionsumgebung kopiert. In einer Blau/Grün-Umgebung ist die blaue Umgebung die aktuelle Produktionsumgebung. Die grüne Umgebung ist die Staging-Umgebung und bleibt mit der aktuellen Produktionsumgebung synchron.

Sie können Änderungen an den RDS-DB-Instances in der grünen Umgebung vornehmen, ohne die Produktions-Workloads zu beeinträchtigen. Sie können beispielsweise die Haupt- oder Nebenversion der DB-Engine aktualisieren, die zugrundeliegende Dateisystemkonfiguration aktualisieren oder Datenbankparameter in der Staging-Umgebung ändern. Sie können Änderungen in der grünen Umgebung gründlich testen. Wenn Sie bereit sind, können Sie zwischen den Umgebungen wechseln, um die grüne Umgebung zur neuen Produktionsumgebung zu machen. Die Umstellung dauert in der Regel weniger als eine Minute, ohne dass Daten verloren gehen und Anwendungsänderungen erforderlich sind.

Da die grüne Umgebung eine Kopie der Topologie der Produktionsumgebung ist, umfasst die grüne Umgebung die von der DB-Instance verwendeten Funktionen. Zu diesen Funktionen gehören die Lesereplikate, die Speicherkonfiguration, DB-Snapshots, automatische Backups, Performance Insights und die verbesserte Überwachung. Wenn es sich bei der blauen DB-Instance um eine Multi-AZ-Bereitstellung handelt, entspricht die grüne DB-Instance ebenfalls einer Multi-AZ-Bereitstellung.

> **Note**
>
> Derzeit werden blue/green Bereitstellungen nur für RDS für MariaDB, RDS für MySQL und RDS für PostgreSQL unterstützt. Informationen zur Verfügbarkeit von Amazon Aurora finden Sie unter Überblick über Amazon Aurora Blue/Green-Bereitstellungen im Amazon Aurora Aurora-Benutzerhandbuch.
>
> Unter bestimmten Bedingungen verwendet RDS for PostgreSQL logische Replikation anstelle von physischer Replikation, um die grüne Umgebung mit der blauen Umgebung synchron zu halten. Weitere Informationen finden Sie unter the section called “Postgre-Replikationsmethoden SQL”.

Themen

- Verfügbarkeit von Regionen und Versionen
- Vorteile der Verwendung von Amazon Blue/Green RDS-Bereitstellungen

- Arbeitsablauf einer Bereitstellung blue/green
- Autorisieren des Zugriffs auf Amazon RDS Amazon Blue/Green-Bereitstellungsvorgänge
- Einschränkungen und Überlegungen für Amazon RDS Amazon
- Bewährte Methoden für Amazon RDS Amazon in Blau/Grün
- SQLPostgre-Replikationsmethoden für blaue/grüne Bereitstellungen

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen finden Sie unter the section called “Blau/Grün-Bereitstellungen”.

Vorteile der Verwendung von Amazon Blue/Green RDS-Bereitstellungen

Durch die Verwendung von Amazon RDS Blue/Green Deployments können Sie über Sicherheitspatches auf dem Laufenden bleiben, die Datenbankleistung verbessern und neuere Datenbankfunktionen mit kurzen, vorhersehbaren Ausfallzeiten einführen. Blue/green Bereitstellungen reduzieren die Risiken und Ausfallzeiten bei Datenbank-Updates, wie z. B. bei Upgrades größerer oder kleinerer Engine-Versionen.

Blau/Grün-Bereitstellungen bieten die folgenden Vorteile:

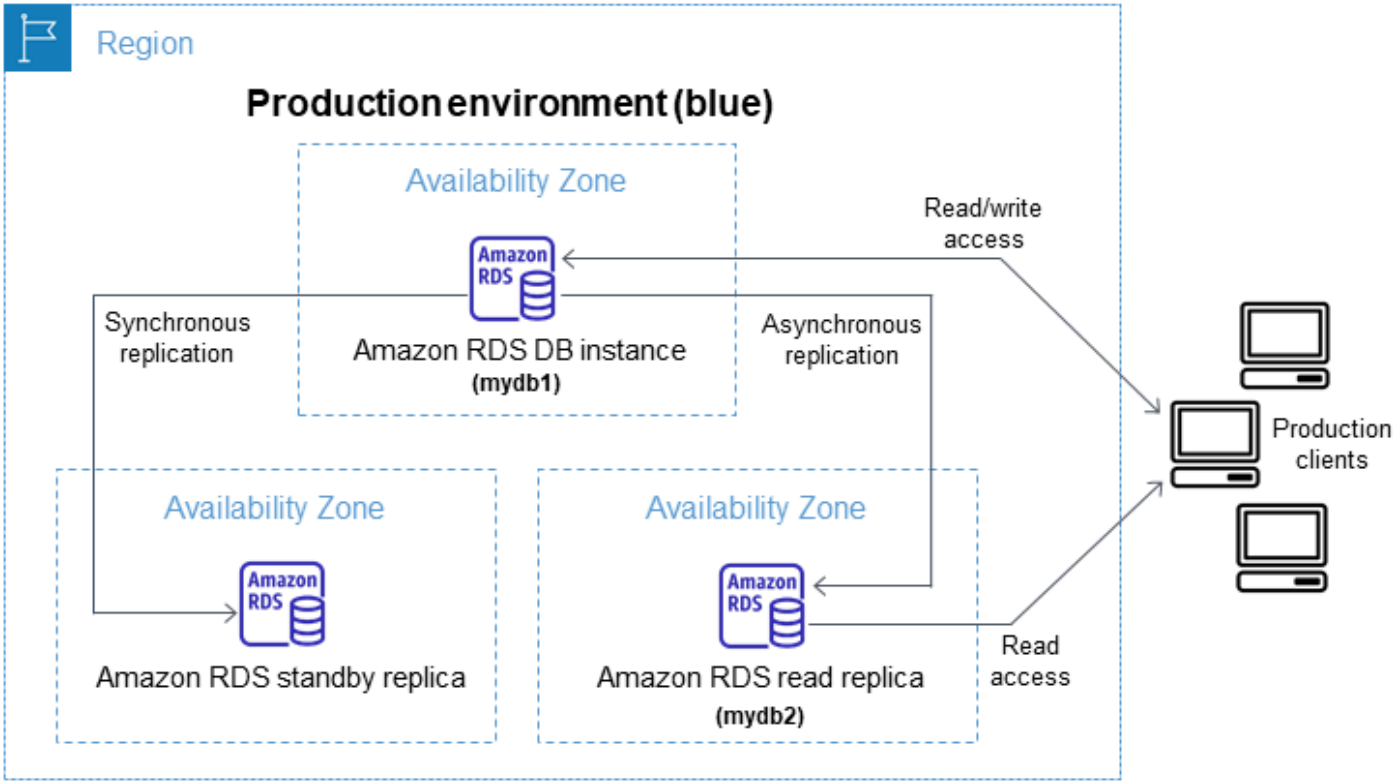
- Sie können ganz einfach eine produktionsreife Staging-Umgebung erstellen.
- Sie können Datenbankänderungen von der Produktionsumgebung in die Staging-Umgebung automatisch replizieren.
- Sie können Datenbankänderungen in einer sicheren Staging-Umgebung testen, ohne die Produktionsumgebung zu beeinträchtigen.
- Sie bleiben mit Datenbank-Patches und Systemaktualisierungen auf dem Laufenden.
- Sie können neue Datenbankfunktionen implementieren und testen.
- Sie können Ihre Staging-Umgebung auf die neue Produktionsumgebung umstellen, ohne Änderungen an Ihrer Anwendung vorzunehmen.
- Die Umstellung erfolgt durch den Einsatz des integrierten Integritätsschutzes völlig sicher.
- Es treten keine Datenverluste während der Umstellung auf.
- Die Umstellung erfolgt schnell, in der Regel in weniger als einer Minute, abhängig von Ihrer Workload.

Arbeitsablauf einer Bereitstellung blue/green

Führen Sie die folgenden wichtigen Schritte aus, wenn Sie eine blue/green Bereitstellung für Datenbankupdates verwenden.

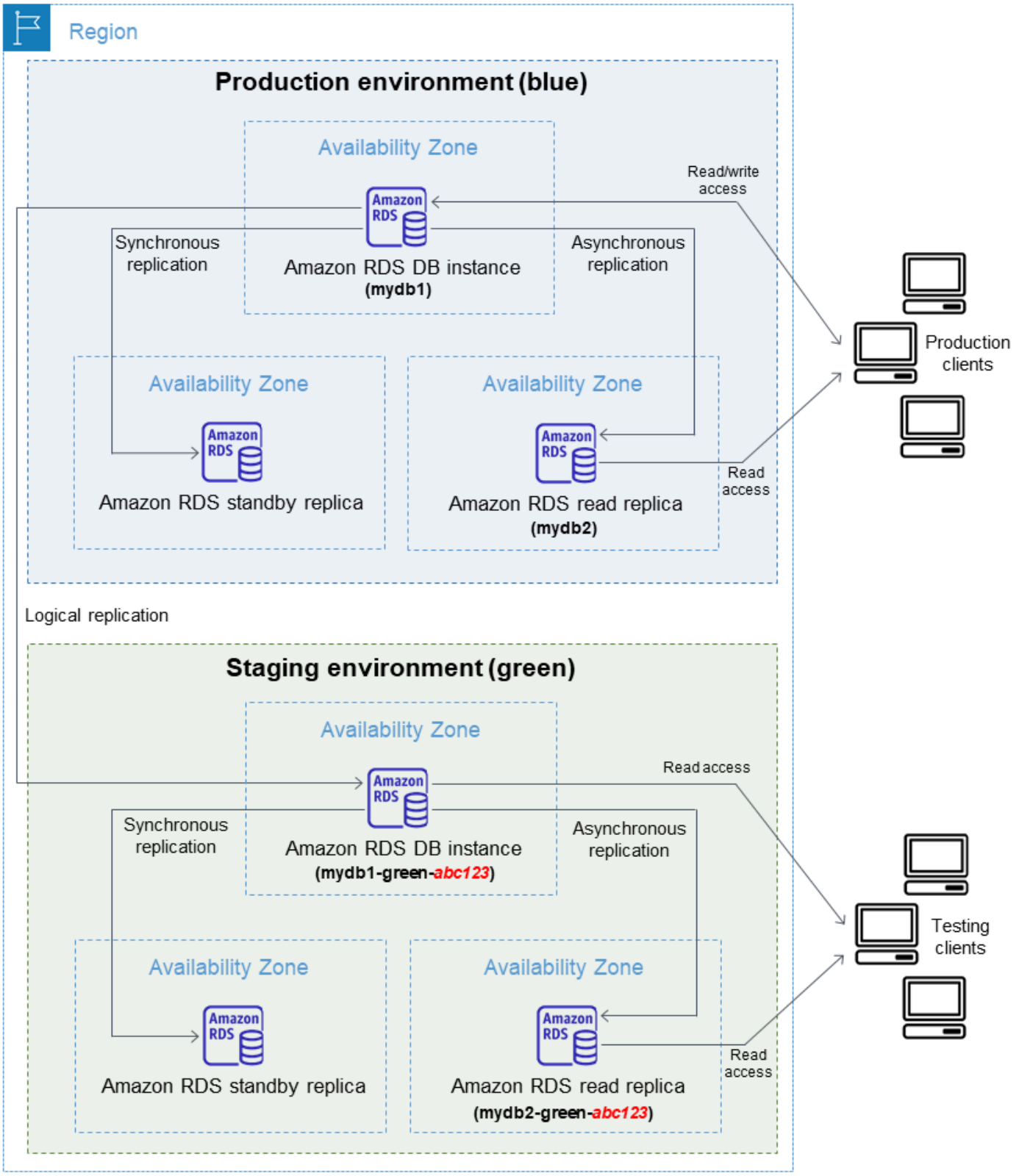
1. Identifizieren Sie eine Produktionsumgebung, die aktualisiert werden muss.

   Die Produktionsumgebung in dieser Abbildung verfügt beispielsweise über eine Multi-AZ-Bereitstellung der DB-Instance (mydb1) und ein Lesereplikat (mydb2).



2. Erstellen Sie die blue/green Bereitstellung. Detaillierte Anweisungen finden Sie unter Eine blaue/grüne Bereitstellung in Amazon RDS erstellen .

   Die folgende Abbildung zeigt ein Beispiel für eine blue/green Bereitstellung der Produktionsumgebung aus Schritt 1. Bei der Erstellung der blue/green Bereitstellung kopiert RDS die komplette Topologie und Konfiguration der primären DB-Instance, um eine grüne Umgebung zu schaffen. Den kopierten DB-Instance-Namen wird `-green-random-characters` angehängt. Die Staging-Umgebung im Image enthält eine Multi-AZ-DB-Instance-Bereitstellung (mydb1-green-*abc123*) und eine Read Replica (mydb2-green-). *abc123*



Wenn Sie die blue/green Bereitstellung erstellen, können Sie Ihre DB-Engine-Version aktualisieren und eine andere DB-Parametergruppe für die DB-Instances in der grünen Umgebung angeben.

RDS konfiguriert auch die Replikation von der primären DB-Instance in der blauen Umgebung zur primären DB-Instance in der grünen Umgebung.

Nachdem Sie die blue/green Bereitstellung erstellt haben, ist die DB-Instance in der grünen Umgebung standardmäßig schreibgeschützt.

3. Nehmen Sie bei Bedarf weitere Änderungen an der Staging-Umgebung vor. Sie können beispielsweise die DB-Instance-Klasse ändern, die von einer oder mehreren DB-Instances in der grünen Umgebung verwendet wird.

   Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

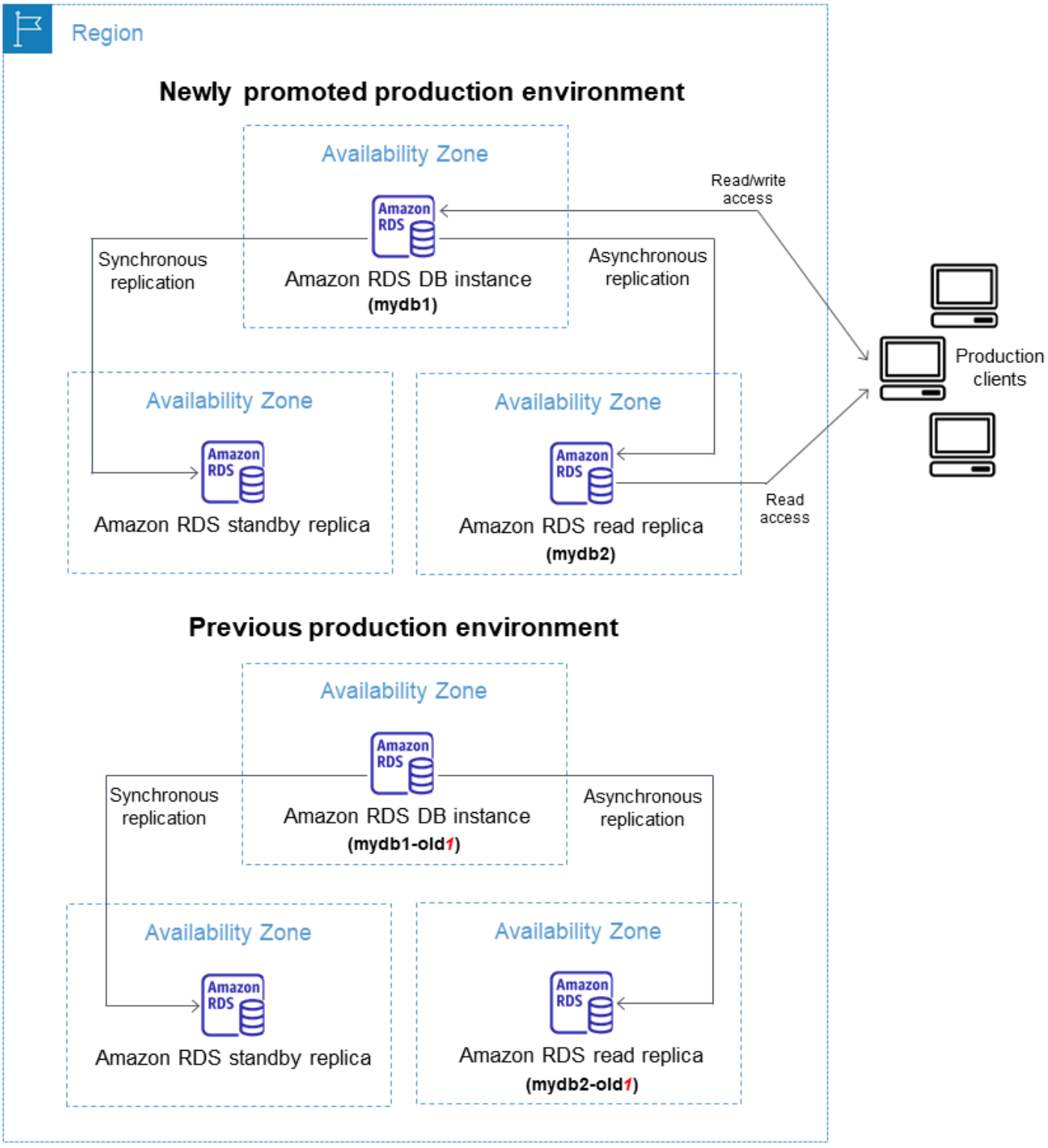
4. Testen Sie Ihre Staging-Umgebung.

   Während des Testens empfehlen wir, dass Sie Ihre Datenbanken in der grünen Umgebung schreibgeschützt lassen. Aktivieren Sie Schreibvorgänge in der grünen Umgebung mit Vorsicht, da sie zu Replikationskonflikten führen können. Sie können auch zu ungewollten Daten in den Produktionsdatenbanken nach der Umstellung führen. Um Schreibvorgänge für RDS for MySQL zu aktivieren, setzen Sie den `read_only` Parameter auf und warten Sie`1`, bis die Parametergruppe synchronisiert ist. Da `read_only` es sich um einen dynamischen Parameter handelt, ist ein Neustart nicht erforderlich. Wechseln Sie nach der Synchronisierung `read_only` von `1` zu`0`. Für RDS for PostgreSQL-Bereitstellungen, die logische Replikation verwenden, setzen Sie den `default_transaction_read_only` Parameter auf `off` Sitzungsebene auf. Für Benutzer, die physische Replikation verwenden, können Sie in der grünen Umgebung keine Schreibvorgänge aktivieren.

5. Wenn Sie bereit sind, wechseln Sie zur Umstellung der Staging-Umgebung auf die neue Produktionsumgebung. Detaillierte Anweisungen finden Sie unter Umschalten einer blauen/grünen Bereitstellung in Amazon RDS.

   Die Umstellung führt zu Ausfallzeiten. Die Ausfallzeit beträgt normalerweise weniger als eine Minute, kann aber je nach Workload länger sein.

   Die folgende Abbildung zeigt die DB-Instances nach der Umstellung.



Nach der Umstellung werden die DB-Instances, die sich in der grünen Umgebung befanden, zu den neuen Produktions-DB-Instances. Die Namen und Endpunkte in der aktuellen Produktionsumgebung werden der neu umgestellten Produktionsumgebung zugewiesen,

sodass keine Änderungen an Ihrer Anwendung erforderlich sind. Infolgedessen fließt Ihr Produktionsdatenverkehr jetzt in die neue Produktionsumgebung. Die DB-Instances in der vorherigen blauen Umgebung werden umbenannt, indem `-old`*n* an den aktuellen Namen angehängt wird, wobei *n* eine Zahl ist. Angenommen, der Name der DB-Instance in der blauen Umgebung lautet `mydb1`. Nach dem Switchover lautet der Name der DB-Instance. `mydb1-old1`

In dem Beispiel in der Abbildung werden bei der Umstellung die folgenden Änderungen vorgenommen:

- Die Multi-AZ-Bereitstellung der DB-Instance in der grünen Umgebung mit dem Namen `mydb1-green-abc123` wird zur Produktions-Multi-AZ-Bereitstellung der DB-Instance und trägt nun den Namen `mydb1`.
- Das Lesereplikat der grünen Umgebung namens `mydb2-green-abc123` wird zum Produktionslesereplikat `mydb2`.
- Die Multi-AZ-Bereitstellung der DB-Instance in der blauen Umgebung namens `mydb1` wird in `mydb1-old1` umbenannt.
- Das Lesereplikat der blauen Umgebung mit dem Namen `mydb2` wird in `mydb2-old1` umbenannt.

6. Wenn Sie eine blue/green Bereitstellung nicht mehr benötigen, können Sie sie löschen. Detaillierte Anweisungen finden Sie unter Löschen einer blauen/grünen Bereitstellung in Amazon RDS.

   Nach der Umstellung wird die vorherige Produktionsumgebung nicht gelöscht, sodass Sie sie bei Bedarf für Regressionstests verwenden können.

Autorisieren des Zugriffs auf Amazon RDS Amazon Blue/Green-Bereitstellungsvorgänge

Benutzer müssen über die erforderlichen Berechtigungen verfügen, um Operationen im Zusammenhang mit Blau/Grün-Bereitstellungen ausführen zu können. Sie können IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Sie können diese Richtlinien dann den IAM-Berechtigungssätzen oder -Rollen zuordnen, die diese Berechtigungen benötigen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.

Der Benutzer, der eine Blau/Grün-Bereitstellung erstellt, muss über Berechtigungen zum Ausführen folgender RDS-Operationen verfügen:

- `rds:CreateBlueGreenDeployment`
- `rds:AddTagsToResource`
- `rds:CreateDBInstanceReadReplica`

Der Benutzer, der eine Blau/Grün-Bereitstellung umstellt, muss über Berechtigungen zum Ausführen folgender RDS-Operationen verfügen:

- `rds:SwitchoverBlueGreenDeployment`
- `rds:ModifyDBInstance`
- `rds:PromoteReadReplica`

Der Benutzer, der eine Blau/Grün-Bereitstellung löscht, muss über Berechtigungen zum Ausführen folgender RDS-Operation verfügen:

- `rds:DeleteBlueGreenDeployment`
- `rds:DeleteDBInstance`

Amazon RDS stellt in Ihrem Namen Ressourcen in der Staging-Umgebung bereit und ändert sie. Zu diesen Ressourcen gehören DB-Instances, die eine intern definierte Namenskonvention verwenden. Daher dürfen angehängte IAM-Richtlinien keine unvollständigen Muster für Ressourcennamen enthalten, wie `my-db-prefix-*` z. Nur Platzhalter (*) werden unterstützt. Im Allgemeinen empfehlen wir, Ressourcen-Tags und andere unterstützte Attribute anstelle von Platzhaltern zu verwenden, um den Zugriff auf diese Ressourcen zu steuern. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon RDS.

Einschränkungen und Überlegungen für Amazon RDS Amazon

Blaue/grüne Bereitstellungen in Amazon RDS erfordern eine sorgfältige Abwägung von Faktoren wie Replikationssteckplätzen, Ressourcenmanagement, Instance-Größe und potenziellen Auswirkungen auf die Datenbankleistung. Die folgenden Abschnitte enthalten Anleitungen zur Optimierung Ihrer Bereitstellungsstrategie, um minimale Ausfallzeiten, reibungslose Übergänge und eine effektive Verwaltung Ihrer Datenbankumgebung sicherzustellen.

Themen

- Einschränkungen für blue/green Bereitstellungen
- Überlegungen zu blue/green Bereitstellungen

Einschränkungen für blue/green Bereitstellungen

Die folgenden Einschränkungen gelten für blue/green Bereitstellungen.

Themen

- Allgemeine Einschränkungen für Blau/Grün-Bereitstellungen
- Einschränkungen von RDS für MySQL für blue/green Bereitstellungen
- Einschränkungen von RDS für PostgreSQL für blue/green Bereitstellungen mit physischer Replikation
- Einschränkungen von RDS for PostgreSQL für Bereitstellungen mit logischer Replikation blue/green

Allgemeine Einschränkungen für Blau/Grün-Bereitstellungen

Die folgenden allgemeinen Einschränkungen gelten für blue/green Bereitstellungen:

- Blaue/grüne Bereitstellungen unterstützen nicht die Verwaltung von Masterbenutzerkennwörtern mit. AWS Secrets Manager
- Wenn Dedicated Log Volume (DLV) in der blauen Datenbank aktiviert ist, muss es auf allen DB-Instances, einschließlich Read Replicas, aktiviert sein.
- Während der Umstellung sind für die blauen und grünen Umgebungen keine Null-ETL-Integrationen mit Amazon Redshift möglich. Sie müssen zuerst die Integration löschen und umstellen. Anschließend erstellen Sie die Integration neu.
- Der Event Scheduler (`event_scheduler`Parameter) muss in der grünen Umgebung deaktiviert sein, wenn Sie eine Bereitstellung erstellen. blue/green Dadurch wird verhindert, dass Ereignisse in der grünen Umgebung generiert werden und zu Inkonsistenzen führen.
- Sie können eine unverschlüsselte DB-Instance nicht in eine verschlüsselte DB-Instance ändern.
- Sie können einen blauen nicht in eine höhere Engine-Version als den entsprechenden grünen ändern.
- Die Ressourcen in der blauen und der grünen Umgebung müssen sich in demselben AWS-Konto befinden.
- Blau/Grün-Bereitstellungen werden für die folgenden Funktionen nicht unterstützt:
  - Amazon-RDS-Proxy
  - Kaskadierende Lesereplikate

- Regionsübergreifende Lesereplikate
- AWS CloudFormation
- Multi-AZ-DB-Cluster-Bereitstellungen

  Blau/Grün-Bereitstellungen werden für Multi-AZ-Bereitstellungen von DB-Instances unterstützt. Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Einschränkungen von RDS für MySQL für blue/green Bereitstellungen

Die folgenden Einschränkungen gelten für blue/green Bereitstellungen von RDS for MySQL:

- Der blaue kann kein externes Binlog-Replikat sein.
- Wenn die Quelldatenbank einer benutzerdefinierten Optionsgruppe zugeordnet ist, können Sie bei der Erstellung der Bereitstellung kein Hauptversions-Upgrade angeben. blue/green

  In diesem Fall können Sie eine blue/green Bereitstellung erstellen, ohne ein Hauptversions-Upgrade anzugeben. Anschließend können Sie die Datenbank in der grünen Umgebung aktualisieren. Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.
- Blaue/grüne Bereitstellungen unterstützen den AWS JDBC-Treiber für MySQL nicht. Weitere Informationen finden Sie unter Bekannte Einschränkungen von. GitHub

Einschränkungen von RDS für PostgreSQL für blue/green Bereitstellungen mit physischer Replikation

Die folgenden Einschränkungen gelten für RDS for blue/green PostgreSQL-Bereitstellungen, die physische Replikation verwenden. Eine Erläuterung, wann blue/green Bereitstellungen physische Replikation statt logischer Replikation verwenden, finden Sie unter. the section called “Postgre-Replikationsmethoden SQL”

- Nachdem die grüne Umgebung erstellt wurde, können Sie kein manuelles Upgrade der Hauptversion durchführen.
- Blaue/grüne Bereitstellungen, die physische Replikation verwenden, unterstützen keine Schemaänderungen in der grünen Umgebung, da diese ausschließlich schreibgeschützt ist.
- Bei der blauen DB-Instance kann es sich nicht um eine logische Quelle (Herausgeber) oder ein Replikat (Abonnent) handeln.

Einschränkungen von RDS for PostgreSQL für Bereitstellungen mit logischer Replikation blue/green

Die folgenden Einschränkungen gelten für Blue/Green-Bereitstellungen, die logische Replikation verwenden. Eine Erläuterung, wann blue/green Bereitstellungen logische Replikation statt physischer Replikation verwenden, finden Sie unter. the section called “Postgre-Replikationsmethoden SQL”

- Nicht protokollierte Tabellen werden nicht in die grüne Umgebung repliziert, .
- Der blaue kann keine logische Quelle (Herausgeber) oder Replikat (Abonnent) sein.
- Wenn die blaue DB-Instance als fremder Server einer FDW-Erweiterung (Foreign Data Wrapper) konfiguriert ist, müssen Sie den Endpunktnamen des Instance- anstelle von IP-Adressen verwenden. Dadurch kann die Konfiguration auch nach der Umstellung funktionsfähig bleiben.
- In einer blue/green Bereitstellung benötigt jede Datenbank einen logischen Replikationssteckplatz. Mit zunehmender Anzahl von Datenbanken nimmt der Ressourcenaufwand zu, was möglicherweise zu Verzögerungen bei der Replikation führen kann, insbesondere wenn die nicht ausreichend skaliert ist. Die Auswirkungen hängen von Faktoren wie der Datenbankauslastung und der Anzahl der Verbindungen ab. Um dies zu minimieren, sollten Sie erwägen, Ihre DB-Instance-Klasse zu skalieren oder die Anzahl der Datenbanken auf der zu reduzieren.
- Der Prozess zum Anwenden der logischen Replikation in der grünen Umgebung erfolgt über einen einzigen Thread. Wenn die blaue Umgebung ein hohes Volumen an Schreibverkehr generiert, kann die grüne Umgebung möglicherweise nicht Schritt halten. Dies kann zu Verzögerungen oder Fehlern bei der Replikation führen, insbesondere bei Workloads, die einen kontinuierlich hohen Schreibdurchsatz erzeugen. Stellen Sie sicher, dass Sie Ihre Workloads gründlich testen. Für Szenarien, die größere Versionsupgrades und die Verarbeitung umfangreicher Schreib-Workloads erfordern, sollten Sie alternative Ansätze wie die Verwendung von AWS Database Migration Service (AWS DMS) logischer Replikation in Betracht ziehen.
- Das Erstellen neuer Partitionen in partitionierten Tabellen wird bei Blue/Green-Bereitstellungen für RDS for PostgreSQL nicht unterstützt. Das Erstellen neuer Partitionen beinhaltet z. B. DDL-Operationen (Data Definition Language)`CREATE TABLE`, die nicht von der blauen Umgebung in die grüne Umgebung repliziert werden. Bestehende partitionierte Tabellen und ihre Daten werden jedoch in die grüne Umgebung repliziert.
- Die folgenden Einschränkungen gelten für PostgreSQL-Erweiterungen:
  - Die `pg_partman` Erweiterung muss in der blauen Umgebung deaktiviert werden, wenn Sie eine blue/green Bereitstellung erstellen. Die Erweiterung führt DDL-Operationen wie etwa `CREATE TABLE` durch, die die logische Replikation von der blauen in die grüne Umgebung unterbrechen.
  - Die `pg_cron` Erweiterung muss nach der Erstellung der blue/green Bereitstellung in allen grünen Datenbanken deaktiviert bleiben. Die Erweiterung verfügt über Hintergrund-Worker,

die als Superuser ausgeführt werden und die Schreibschutzeinstellung der grünen Umgebung umgehen, was zu Replikationskonflikten führen kann.

- Die `pgactive` Erweiterungen `pglogical` und müssen in der blauen Umgebung deaktiviert sein, wenn Sie eine blue/green Bereitstellung erstellen. Nachdem Sie die grüne Umgebung zur neuen Produktionsumgebung umgestellt haben, können Sie die Erweiterungen wieder aktivieren. Dazu kann die blaue Datenbank kein logischer Subscriber einer externen Instance sein.
- Wenn Sie die `pgAudit` Erweiterung verwenden, muss sie in den gemeinsam genutzten Bibliotheken (`shared_preload_libraries`) der benutzerdefinierten DB-Parametergruppen sowohl für die blaue als auch für die grüne DB-Instance verbleiben. Weitere Informationen finden Sie unter the section called “Einrichtung der pgAudit Erweiterung”.

Spezifische Einschränkungen der logischen Replikation für Bereitstellungen blue/green

PostgreSQL hat bestimmte Einschränkungen in Bezug auf die logische Replikation, die sich in Einschränkungen bei der Erstellung von blue/green Bereitstellungen für niederschlagen.

In der folgenden Tabelle werden die Einschränkungen der logischen Replikation beschrieben, die für Blau/Grün-Bereitstellungen für RDS für PostgreSQL gelten. Weitere Informationen finden Sie in der Dokumentation zur logischen Replikation in PostgreSQL.

Einschränkung	Erklärung
Data Definition Language (DDL)-Anweisungen wie `CREATE TABLE` und `CREATE SCHEMA` werden nicht von der blauen in die grüne Umgebung repliziert.	Wenn Amazon RDS eine DDL-Änderung in der blauen Umgebung erkennt, gehen die grünen Datenbanken in den Status Replikation herabgestuft über. Sie müssen die blue/green Bereitstellung und alle grünen Datenbanken löschen und sie dann neu erstellen.
DCL-Anweisungen (Data Control Language), wie	Wenn Amazon RDS PostgreSQL einen Versuch erkennt, eine DCL-Anweisung in der blauen Umgebung auszuführen, wird eine Warnmeldung angezeigt. Es ist keine Konfiguration oder API verfügbar, um dieses

Einschränkung	Erklärung
z. B. `GRANT` und`REVOKE`, werden nicht von der blauen Umgebung in die grüne Umgebung repliziert.	Verhalten zu ändern, da dies eine Einschränkung des blue/green Bereitstellungsprozesses darstellt.
`NEXTVAL`-Operationen an Sequenzobjekten werden nicht zwischen der blauen und der grünen Umgebung synchronisiert.	Während der Umstellung erhöht Amazon RDS die Sequenzwerte in der grünen Umgebung so, dass sie denen in der blauen Umgebung entsprechen. Wenn Sie Tausende von Sequenzen haben, kann dies die Umstellung verzögern.

Einschränkung	Erklärung
Große Objekte in der blauen Umgebung werden nicht in die grüne Umgebung repliziert. Dazu gehören sowohl vorhandene große Objekte als auch alle während des blue/green Bereitstellungsprozesses neu erstellten oder geänderten großen Objekte.	Wenn Amazon RDS die Erstellung oder Änderung großer Objekte in der blauen Umgebung erkennt, die in der `pg_largeobject` -Systemtabelle gespeichert sind, gehen die grünen Datenbanken in den Status Replikation herabgestuft über. Sie müssen die blue/green Bereitstellung und alle grünen Datenbanken löschen und sie dann neu erstellen.
Materialisierte Ansichten werden in der grünen Umgebung nicht automatisch aktualisiert.	Durch das Aktualisieren materialisierter Ansichten in der blauen Umgebung werden sie in der grünen Umgebung nicht aktualisiert. Nach dem Switchover können Sie sie manuell mit dem Befehl REFRESH MATERIALIZED VIEW aktualisieren oder eine Aktualisierung planen.
UPDATE- und DELETE-Operationen sind für Tabellen, die keinen Primärschlüssel haben, nicht zulässig.	Bevor Sie ein blue/green Deployment erstellen, stellen Sie sicher, dass alle Tabellen über einen Primärschlüssel oder eine Verwendung verfügen. `REPLICA IDENTITY FULL` Verwenden Sie diese Option jedoch nur, `REPLICA IDENTITY FULL` wenn kein primärer oder eindeutiger Schlüssel vorhanden ist, da dies die Replikationsleistung beeinträchtigt. Weitere Informationen finden Sie in der PostgreSQL-Dokumentation.

Überlegungen zu blue/green Bereitstellungen

Amazon RDS verfolgt Ressourcen in blue/green Bereitstellungen mit dem `DbiResourceId` jeder Ressource. Diese Ressourcen-ID ist eine AWS-Region eindeutige, unveränderliche Kennung für die Ressource.

Die Ressourcen-ID ist von der Instance-ID des getrennt. Jede einzelne ist in der Datenbankkonfiguration in der RDS-Konsole aufgeführt.

Der Name (Instanz-ID) einer Ressource ändert sich, wenn Sie zu einer blue/green Bereitstellung wechseln, aber jede Ressource behält dieselbe Ressourcen-ID. Eine DB-Instance-ID in der blauen Umgebung könnte beispielsweise `mydb` lauten. Nach der Umstellung könnte diese DB-Instance in `mydb-old1` umbenannt sein. Die Ressourcen-ID der DB-Instance ändert sich während der Umstellung jedoch nicht. Wenn Sie also die grünen Ressourcen auf die neuen Produktionsressourcen umstellen, stimmen ihre Ressourcen IDs nicht mit den blauen Ressourcen überein IDs , die zuvor in Produktion waren.

Nachdem Sie eine blue/green Bereitstellung umgestellt haben, sollten Sie erwägen, die Ressource IDs auf die Ressourcen der neu umgestellten Produktionsressourcen für integrierte Funktionen und Dienste zu aktualisieren, die Sie mit den Produktionsressourcen verwendet haben. Berücksichtigen Sie insbesondere die folgenden Aktualisierungen:

- Wenn Sie die Filterung mithilfe der RDS-API und der RDS-Ressource durchführen IDs, passen Sie die für die Filterung IDs verwendete Ressource nach dem Switchover an.
- Wenn Sie Ressourcen CloudTrail für die Überwachung verwenden, passen Sie die Benutzer von so an, CloudTrail dass sie die neue Ressource IDs nach dem Switchover verfolgen. Weitere Informationen finden Sie unter überwachen Amazon RDS API ruft an AWS CloudTrail.
- Wenn Sie die Performance Insights Insights-API verwenden, passen Sie die Ressource IDs in API-Aufrufen nach dem Switchover an. Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.

  Sie können eine Datenbank mit demselben Namen nach der Umstellung überwachen, diese enthält jedoch nicht die Daten, die vor der Umstellung vorhanden waren.
- Wenn Sie Ressourcen IDs in IAM-Richtlinien verwenden, stellen Sie sicher, dass Sie bei Bedarf die Ressource IDs der neu übertragenen Ressourcen hinzufügen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.
- Wenn Ihrer IAM-Rollen zugeordnet sind, stellen Sie sicher, dass Sie diese nach dem Switchover erneut zuordnen. Angehängte Rollen werden nicht automatisch in die grüne Umgebung kopiert.

- Wenn Sie sich mithilfe der IAM-Datenbankauthentifizierung bei Ihrer DB-Instance authentifizieren, stellen Sie sicher, dass in der für den Datenbankzugriff verwendeten IAM-Richtlinie sowohl die blauen als auch die grünen Datenbanken unter dem Element `Resource` der Richtlinie aufgeführt sind. Dies ist erforderlich, um nach der Umstellung eine Verbindung mit der grünen Datenbank herzustellen. Weitere Informationen finden Sie unter the section called “Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff”.
- Wenn Sie AWS Backup automatische Backups von Ressourcen in einer Blau/Grün-Umgebung verwalten, passen Sie die IDs verwendete Ressource AWS Backup nach dem Switchover an. Weitere Informationen finden Sie unter Wird AWS Backup zur Verwaltung automatisierter Backups für Amazon verwendet RDS.
- Wenn Sie einen manuellen oder automatisierten DB-Snapshot für eine DB-Instance wiederherstellen möchten, die Teil einer blue/green Bereitstellung war, stellen Sie sicher, dass Sie den richtigen DB-Snapshot wiederherstellen, indem Sie den Zeitpunkt überprüfen, zu dem der Snapshot erstellt wurde. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.
- Wenn Sie ein früheres automatisches Backup in einer blauen Umgebung beschreiben oder es zu einem bestimmten Zeitpunkt wiederherstellen möchten, verwenden Sie die Ressourcen-ID für die Operation.

  Da sich der Name der DB-Instance während der Umstellung ändert, können Sie ihren vorherigen Namen nicht für `DescribeDBInstanceAutomatedBackups`- oder `RestoreDBInstanceToPointInTime`-Operationen verwenden.

  Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.
- Wenn Sie einer DB-Instance in der grünen Umgebung einer Blau/Grün-Bereitstellung ein Lesereplikat hinzufügen, ersetzt das neue Lesereplikat bei der Umstellung kein Lesereplikat in der blauen Umgebung. Das neue Lesereplikat wird jedoch nach der Umstellung in der neuen Produktionsumgebung beibehalten.
- Nach dem Umschalten AWS Database Migration Service (AWS DMS) können die Replikationsaufgaben nicht fortgesetzt werden, da der Checkpoint aus der blauen Umgebung in der grünen Umgebung ungültig ist. Sie müssen die DMS-Aufgabe mit einem neuen Checkpoint neu erstellen, um die Replikation fortzusetzen.
- Wenn Sie eine DB-Instance in der grünen Umgebung einer blue/green Bereitstellung löschen, können Sie keine neue DB-Instance erstellen, um sie in der Bereitstellung zu ersetzen. blue/green

Wenn Sie eine neue DB-Instance mit demselben Namen und Amazon-Ressourcennamen (ARN) wie die gelöschte DB-Instance erstellen, hat sie eine andere `DbiResourceId`. Sie ist folglich nicht Teil der grünen Umgebung.

Das folgende Verhalten ergibt sich, wenn Sie eine DB-Instance in der grünen Umgebung löschen:

- Wenn die DB-Instance in der blauen Umgebung mit dem gleichen Namen vorhanden ist, wird sie nicht auf die DB-Instance in der grünen Umgebung umgestellt. Diese DB-Instance wird nicht umbenannt, indem dem DB-Instance-Namen `-old`*n* hinzugefügt wird.
- Jede Anwendung, die auf die DB-Instance in der blauen Umgebung verweist, verwendet nach der Umstellung weiterhin dieselbe DB-Instance.

Das gleiche Verhalten gilt für DB-Instances und Lesereplikate.

Bewährte Methoden für Amazon RDS Amazon in Blau/Grün

Im Folgenden finden Sie bewährte Methoden für blaue/grüne Bereitstellungen.

Themen

- Allgemeine bewährte Methoden für blaue/grüne Bereitstellungen
- RDS für MySQL — Best Practices für blaue/grüne Bereitstellungen
- Bewährte Methoden für RDS für PostgreSQL für Blue/Green-Bereitstellungen

Allgemeine bewährte Methoden für blaue/grüne Bereitstellungen

Beachten Sie bei der Erstellung einer Blau/Grün-Bereitstellung die folgenden allgemeinen bewährten Methoden.

- Testen Sie die DB-Instances in der grünen Umgebung gründlich, bevor Sie umstellen.
- Halten Sie Ihre Datenbanken in der grünen Umgebung schreibgeschützt. Wir empfehlen, Schreibvorgänge in der grünen Umgebung mit Vorsicht zu aktivieren, da sie zu Replikationskonflikten führen können. Sie können auch zu ungewollten Daten in den Produktionsdatenbanken nach der Umstellung führen.
- Wenn Sie eine blaue/grüne Bereitstellung verwenden, um Schemaänderungen zu implementieren, nehmen Sie nur replikationskompatible Änderungen vor.

Sie können beispielsweise neue Spalten am Ende einer Tabelle hinzufügen, ohne die Replikation von der blauen zur grünen Bereitstellung zu unterbrechen. Schemaänderungen, wie das Umbenennen von Spalten oder Tabellen, führen jedoch dazu, dass die Replikation zur Grün-Bereitstellung unterbrochen wird.

Weitere Informationen zu replikationskompatiblen Änderungen finden Sie in der MySQL-Dokumentation unter Replikation mit unterschiedlichen Tabellendefinitionen auf Quelle und Replikat sowie unter Einschränkungen in der Dokumentation zur logischen Replikation in PostgreSQL.

> **Note**
>
> Diese Einschränkung gilt nicht für RDS for PostgreSQL Blue/Green-Bereitstellungen, die physische Replikation verwenden. Weitere Informationen finden Sie unter the section called “Einschränkungen von RDS für PostgreSQL für blue/green Bereitstellungen mit physischer Replikation”.

- Nachdem Sie die Blau/Grün-Bereitstellung erstellt haben, führen Sie gegebenenfalls Lazy Loading durch. Stellen Sie sicher, dass das Laden der Daten abgeschlossen ist, bevor Sie umstellen. Weitere Informationen finden Sie unter Verzögertes Laden und Speicherinitialisierung für blaue/grüne Bereitstellungen.
- Wenn Sie auf eine Blau/Grün-Bereitstellung umstellen, befolgen Sie die bewährten Methoden für die Umstellung. Weitere Informationen finden Sie unter the section called “Bewährte Methoden für die Umstellung”.

RDS für MySQL — Best Practices für blaue/grüne Bereitstellungen

Beachten Sie die folgenden bewährten Methoden, wenn Sie eine blaue/grüne Bereitstellung aus einem erstellen.

- Vermeiden Sie die Verwendung von nicht-transaktionalen Speicher-Engines wie MyISAM, die nicht für die Replikation optimiert sind.
- Optimieren Sie Read Replicas und die grüne Umgebung für die Replikation von Binärprotokollen. Falls von Ihrer DB-Engine unterstützt, aktivieren Sie GTID-, parallel und absturzsichere Replikation, um Datenkonsistenz und Beständigkeit zu gewährleisten, bevor Sie Ihre Blue/Green-Bereitstellung erstellen. Weitere Informationen finden Sie unter Verwenden der GTID basierten Replikation.

- Wenn es in der grünen Umgebung zu Verzögerungen bei der Replikation kommt, sollten Sie Folgendes berücksichtigen:
  - Setzen Sie den `innodb_flush_log_at_trx_commit` Parameter vorübergehend auf 2in der grünen DB-Parametergruppe. Wenn die Replikation aufgeholt hat, kehren Sie zum Standardwert 1 vor dem Switchover zurück. Wenn der temporäre Parameterwert unerwartet heruntergefahren wird oder es zu einem Absturz kommt, erstellen Sie die grüne Umgebung neu, um unentdeckte Datenbeschädigungen zu vermeiden.
  - Um die Schreiblatenz zu reduzieren und den Replikationsdurchsatz zu verbessern, ändern Sie vorübergehend grüne Multi-AZ-DB-Instances in Single-AZ-DB-Instances. Aktivieren Sie Multi-AZ unmittelbar vor dem Switchover erneut.

Bewährte Methoden für RDS für PostgreSQL für Blue/Green-Bereitstellungen

Beachten Sie die folgenden bewährten Methoden, wenn Sie eine blaue/grüne Bereitstellung aus einer RDS for PostgreSQL-DB-Instance erstellen.

Themen

- Allgemeine Best Practices für RDS für PostgreSQL für Blue/Green-Bereitstellungen
- Bewährte Methoden für RDS für PostgreSQL für Blue/Green-Bereitstellungen mit physischer Replikation
- Bewährte Methoden für RDS für PostgreSQL für Blue/Green-Bereitstellungen mit logischer Replikation

Allgemeine Best Practices für RDS für PostgreSQL für Blue/Green-Bereitstellungen

Beachten Sie die folgenden allgemeinen bewährten Methoden, wenn Sie eine blaue/grüne Bereitstellung aus einer RDS for PostgreSQL-DB-Instance erstellen.

- Aktualisieren Sie alle Ihre PostgreSQL-Erweiterungen auf die neueste Version, bevor Sie eine Blau/Grün-Bereitstellung erstellen. Weitere Informationen finden Sie unter the section called “Aktualisierung von Postgre-Erweiterungen SQL”.
- Transaktionen mit langer Laufzeit können zu erheblichen Verzögerungen bei Replikaten führen. Um die Verzögerung bei Replikaten zu verringern, sollten Sie Folgendes in Betracht ziehen:
  - Reduzieren Sie lang andauernde Transaktionen, die verzögert werden können, bis die grüne Umgebung die blaue Umgebung eingeholt hat.

- Reduzieren Sie Massenoperationen in der blauen Umgebung, bis die grüne Umgebung die blaue Umgebung eingeholt hat.
- Initiieren Sie an stark frequentierten Tischen einen manuellen Vakuumfrostvorgang, bevor Sie die blaue/grüne Bereitstellung erstellen.
- Deaktivieren Sie für PostgreSQL Version 12 und höher den `index_cleanup` Parameter für große oder ausgelastete Tabellen, um die normale Wartungsrate bei blauen Datenbanken zu erhöhen. Weitere Informationen finden Sie unter the section called “Möglichst schnelles Bereinigen einer Tabelle”.

  > **Note**
  >
  > Wenn die Indexbereinigung während des Staubsaugens regelmäßig übersprungen wird, kann dies zu einer Aufblähung des Index führen, wodurch die Scanleistung beeinträchtigt werden kann. Es hat sich bewährt, diesen Ansatz nur zu verwenden, wenn Sie eine blaue/grüne Bereitstellung verwenden. Sobald die Bereitstellung abgeschlossen ist, empfehlen wir, die regelmäßige Indexwartung und -bereinigung wieder aufzunehmen.

- Eine langsame Replikation kann dazu führen, dass Sender und Empfänger häufig neu gestartet werden, was die Synchronisation verzögert. Um sicherzustellen, dass sie aktiv bleiben, deaktivieren Sie Timeouts, indem Sie den `wal_sender_timeout` Parameter auf `0` in der blauen Umgebung und den `wal_receiver_timeout` Parameter auf `0` in der grünen Umgebung setzen.
- Um zu verhindern, dass Write-Ahead-Log-Segmente (WAL) aus der blauen Umgebung entfernt werden, setzen Sie den `wal_keep_segments` Parameter für PostgreSQL Version 13 und niedriger auf 15625. Für Version 14 und höher setzen Sie den `wal_keep_size` Parameter auf 1 TiB, wenn genügend freier Speicherplatz vorhanden ist.

Bewährte Methoden für RDS für PostgreSQL für Blue/Green-Bereitstellungen mit physischer Replikation

Bei der physischen Replikation erstellt Amazon RDS eine Read Replica der Quell-DB-Instance. Informationen zu verwandten Parametern, Überwachung, Optimierung und Fehlerbehebung finden Sie unterthe section called “Arbeiten mit Read Replicas in RDS for PostgreSQL”.

Eine Erläuterung, wann bei Bereitstellungen in Blau/Grün physische Replikation statt logischer Replikation verwendet wird, finden Sie unter. the section called “Postgre-Replikationsmethoden SQL”

Bewährte Methoden für RDS für PostgreSQL für Blue/Green-Bereitstellungen mit logischer Replikation

Beachten Sie beim Erstellen von blue/green deployment that uses logical replication. For an explanation of when blue/green Bereitstellungen die folgenden bewährten Methoden, bei denen logische Replikation statt physischer Replikation verwendet wird, siehe. the section called “Postgre-Replikationsmethoden SQL”

- Wenn Ihre Datenbank über ausreichend freien Speicher verfügt, erhöhen Sie den Wert des `logical_decoding_work_mem` DB-Parameters in der blauen Umgebung. Dadurch muss auf der Festplatte weniger dekodiert werden und stattdessen wird Arbeitsspeicher beansprucht. Weitere Informationen finden Sie in der PostgreSQL-Dokumentation.
  - Mithilfe der Metrik können Sie den Transaktionsüberlauf überwachen, der auf die `ReplicationSlotDiskUsage` CloudWatch Festplatte geschrieben wird. Diese Metrik bietet Einblicke in die Festplattennutzung von Replikationssteckplätzen und hilft so zu erkennen, wann Transaktionsdaten die Speicherkapazität überschreiten und auf der Festplatte gespeichert werden. Mit der `FreeableMemory` CloudWatch Metrik können Sie den freien Speicher überwachen. Weitere Informationen finden Sie unter the section called “Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS”.
  - In RDS für PostgreSQL Version 14 und höher können Sie die Größe logischer Überlaufdateien mithilfe der `pg_stat_replication_slots` Systemansicht überwachen.
- Wenn Sie die `aws_s3` Erweiterung verwenden, gewähren Sie der grünen DB-Instance über eine IAM-Rolle Zugriff auf Amazon S3, nachdem die grüne Umgebung erstellt wurde. Dadurch können die Import- und Exportbefehle auch nach der Umstellung weiter funktionieren. Detaillierte Anweisungen finden Sie unter the section called “Einrichten des Zugriffs auf einen Amazon S3-Bucket”.
- Überprüfen Sie die Leistung Ihrer UPDATE- und DELETE-Anweisungen und prüfen Sie, ob die Erstellung eines Indexes für die in der WHERE-Klausel verwendete Spalte diese Abfragen optimieren kann. Dies kann die Leistung verbessern, wenn die Operationen in der grünen Umgebung wiederholt werden.
- Wenn Sie Trigger verwenden, stellen Sie sicher, dass sie das Erstellen, Aktualisieren und Löschen von, und `pg_catalog.pg_replication_slots` Objekten`pg_catalog.pg_publication`, deren Namen mit 'rds' beginnen`pg_catalog.pg_subscription`, nicht beeinträchtigen.
- Wenn Sie eine höhere Engine-Version für die grüne Umgebung angeben, führen Sie den `ANALYZE` Vorgang für alle Datenbanken aus, um die `pg_statistic` Tabelle zu aktualisieren. Optimizer-Statistiken werden während eines größeren Versionsupgrades nicht übertragen. Sie müssen

daher alle Statistiken neu generieren, um Leistungsprobleme zu vermeiden. Weitere bewährte Methoden bei größeren Versionsupgrades finden Sie unter. the section called “Durchführen eines Hauptversions-Upgrades”

- Vermeiden Sie es, Trigger so zu konfigurieren, als `ENABLE REPLICA` `ENABLE ALWAYS` ob der Trigger auf der Quelle verwendet wird, um Daten zu manipulieren. Andernfalls leitet das Replikationssystem die Änderungen weiter und führt den Trigger aus, was zu Duplizierungen führt.

SQLPostgre-Replikationsmethoden für blaue/grüne Bereitstellungen

Amazon RDS for Postgre verwendet SQL hauptsächlich physische Replikation für die blue/green deployments. However, if you request a major version upgrade when you create the blue/green Bereitstellung, und Ihre Quell-DB-Instance führt eine der in der folgenden Tabelle aufgeführten SQL Postgre-Versionen aus. Amazon RDS verwendet stattdessen die logische Replikation.

In der folgenden Tabelle wird beschrieben, wann Amazon physische und wann logische Replikation für SQL Postgre-Blue/Green-Bereitstellungen RDS verwendet.

Quellversion der Postgre-DB-Instance SQL	Upgrade-Aktion bei blauer/grüner Bereitstellung	Methode der Replikation
• 16.1 und alle höheren Haupt- und Nebenversionen • 15.4 und höhere 15-Versionen • 14.9 und höhere 14-Versionen • 13.12 und höhere 13-Versionen • 12.16 und höhere 12-Versionen • 11.21 und höhere 11-Versionen	Aktualisierung der Hauptversion (grüne Instanz auf einer höheren Major-Engine-Version als blaue)	Logische Replikation
Alle unterstützten Versionen	Upgrade einer kleineren Version oder kein Upgrade	Physische Replikation

Quellversion der Postgre-DB-Instance SQL	Upgrade-Aktion bei blauer/grüner Bereitstellung	Methode der Replikation
	(grüne Instanz auf derselben Haupt-Engine-Version wie blaue)	

> **Note**
>
> Größere Versions-Upgrades werden nicht für Blue/Green-Bereitstellungen mit Quellcode RDS für SQL Postgre-Versionen 15.3 und niedriger, 14.8 und niedriger, 13.11 und niedriger, 12.15 und niedriger oder 11.20 und niedriger unterstützt.

Informationen zu den Einschränkungen von Blue/Green-Bereitstellungen, die physische und logische Replikation verwenden, finden Sie in den folgenden Abschnitten:

- the section called “Einschränkungen von RDS für PostgreSQL für blue/green Bereitstellungen mit physischer Replikation”
- the section called “Einschränkungen von RDS for PostgreSQL für Bereitstellungen mit logischer Replikation blue/green ”

Eine blaue/grüne Bereitstellung in Amazon RDS erstellen

Wenn Sie eine Blau/Grün-Bereitstellung erstellen, geben Sie die DB-Instance an, die in die Bereitstellung kopiert werden soll. Die von Ihnen ausgewählte DB-Instance ist die Produktions-DB-Instance und wird in der blauen Umgebung zur primären DB-Instance. Diese DB-Instance wird in die grüne Umgebung kopiert und RDS konfiguriert die Replikation von der DB-Instance in der blauen Umgebung zur DB-Instance in der grünen Umgebung.

RDS kopiert die Topologie und die Funktionen der blauen Umgebung in einen Staging-Bereich. Wenn die blaue DB-Instance Repliken gelesen hat, werden sie als Repliken der grünen Instance kopiert. Der zugewiesene Speicher aller grünen Replikate entspricht der grünen primären Instance, während andere Speicherparameter von den blauen Replikaten übernommen werden.

Wenn es sich bei der blauen DB-Instance um eine Multi-AZ-Bereitstellung handelt, wird die grüne DB-Instance als Multi-AZ-Bereitstellung erstellt.

Themen

- Vorbereiten einer Blau-Grün-Bereitstellung
- Angeben von Änderungen bei der Erstellung einer Blau/Grün-Bereitstellung
- Verzögertes Laden und Speicherinitialisierung für blaue/grüne Bereitstellungen
- Erstellen einer Blau/Grün-Bereitstellung
- Einstellungen für die Erstellung von blauen/grünen Bereitstellungen

Vorbereiten einer Blau-Grün-Bereitstellung

Abhängig von der Engine, auf der Ihre ausgeführt wird, müssen Sie bestimmte Schritte ausführen, bevor Sie eine blaue/grüne Bereitstellung erstellen.

Themen

- Vorbereiten einer RDS for MySQL- oder RDS for MariaDB-DB-Instance für eine Blue/Green-Bereitstellung
- Vorbereiten einer RDS for PostgreSQL-DB-Instance für eine blaue/grüne Bereitstellung mit physischer Replikation
- Vorbereiten einer RDS for PostgreSQL-DB-Instance für eine blaue/grüne Bereitstellung mit logischer Replikation

Vorbereiten einer RDS for MySQL- oder RDS for MariaDB-DB-Instance für eine Blue/Green-Bereitstellung

Bevor Sie eine blaue/grüne Bereitstellung für eine RDS for MySQL- oder RDS for MariaDB-DB-Instance erstellen, müssen Sie automatische Backups aktivieren. Detaillierte Anweisungen finden Sie unter the section called “Aktivieren von automatisierten Backups”.

Vorbereiten einer RDS for PostgreSQL-DB-Instance für eine blaue/grüne Bereitstellung mit physischer Replikation

Bevor Sie eine blaue/grüne RDS-Bereitstellung für PostgreSQL erstellen, die physische Replikation verwendet, stellen Sie sicher, dass Sie Folgendes tun. Eine Liste der Versionen, bei denen physische

Replikation und logische Replikation verwendet wird, finden Sie unter. the section called “Postgre-Replikationsmethoden SQL”

- Aktivieren Sie automatische Backups auf der DB-Instance. Detaillierte Anweisungen finden Sie unter the section called “Aktivieren von automatisierten Backups”.
- Stellen Sie sicher, dass die DB-Instance nicht die Quelle oder das Ziel der externen Replikation ist. Weitere Informationen finden Sie unter the section called “Allgemeine Einschränkungen”.

Vorbereiten einer RDS for PostgreSQL-DB-Instance für eine blaue/grüne Bereitstellung mit logischer Replikation

Bevor Sie eine blaue/grüne RDS-Bereitstellung für PostgreSQL erstellen, die logische Replikation verwendet, stellen Sie sicher, dass Sie Folgendes tun. Eine Liste der Versionen, die logische Replikation statt physischer Replikation verwenden, finden Sie unter. the section called “Postgre-Replikationsmethoden SQL”

- Ordnen Sie die Instance einer benutzerdefinierten DB-Parametergruppe zu, bei der die logische Replikation (`rds.logical_replication`) aktiviert ist. Für die Replikation von der blauen zur grünen Umgebung ist die logische Replikation erforderlich. Detaillierte Anweisungen finden Sie unter the section called “Ändern von Parametern in einer DB-Parametergruppe”.

  Da für blaue/grüne Bereitstellungen mindestens ein Hintergrund-Worker pro Datenbank erforderlich ist, stellen Sie sicher, dass Sie die folgenden Konfigurationseinstellungen an Ihre Arbeitslast anpassen. Anweisungen zum Optimieren der einzelnen Einstellungen finden Sie unter Konfigurationseinstellungen in der PostgreSQL-Dokumentation.

  - `max_replication_slots`
  - `max_wal_senders`
  - `max_logical_replication_workers`
  - `max_worker_processes`

  Nachdem Sie die logische Replikation aktiviert und alle Konfigurationsoptionen festgelegt haben, stellen Sie sicher, dass Sie die DB-Instance neu starten, damit die Änderungen wirksam werden. Blau/Grün-Bereitstellungen erfordern, dass die WDBriter-Instance mit der DB-Parametergruppe synchronisiert ist, andernfalls schlägt die Erstellung fehl. Weitere Informationen finden Sie unter the section called “Neustarten einer DB-Instance”.

- Stellen Sie sicher, dass die DB-Instance nicht die Quelle oder das Ziel der externen Replikation ist. Weitere Informationen finden Sie unter the section called “Allgemeine Einschränkungen”.
- Stellen Sie sicher, dass alle Tabellen in der DB-Instance einen Primärschlüssel haben. Die logische PostgreSQL-Replikation lässt keine `UPDATE`- oder `DELETE`-Operationen mit Tabellen zu, die keinen Primärschlüssel haben.
- RDS for PostgreSQL verwendet die native logische Replikation von PostgreSQL und speichert Write-Ahead-Logs (WAL) -Segmente auf der blauen Instance, bis sie in der grünen Umgebung wiedergegeben werden. Bevor Sie eine blaue/grüne Bereitstellung erstellen, stellen Sie sicher, dass die blaue Instance über ausreichende Kapazität verfügt, indem Sie die folgenden Metriken überprüfen:
  - `FreeStorageSpace`
  - `TransactionLogsGeneration`
  - `TransactionLogsDiskUsage`
  - `OldestReplicationSlotLag`

  Um den zusätzlichen Speicherbedarf für die blaue Instance abzuschätzen, sollten Sie die `TransactionLogsGeneration` CloudWatch Metrik in Zeiten hoher Arbeitslast überwachen. Wenn Ihr Workload beispielsweise innerhalb von 24 Stunden 100 GB WAL-Daten generiert, stellen Sie sicher, dass Sie über mindestens 100 GB zusätzlichen Speicherplatz verfügen, um WAL-Segmente für einen Tag aufnehmen zu können. Weitere Informationen finden Sie unter Überwachen von Metriken in einer DB-Instance.

Angeben von Änderungen bei der Erstellung einer Blau/Grün-Bereitstellung

Sie können die folgenden Änderungen an der DB-Instance in der grünen Umgebung vornehmen, wenn Sie die Blau/Grün-Bereitstellung erstellen:

Sie können nach der Bereitstellung weitere Änderungen an der DB-Instance in der grünen Umgebung vornehmen. Sie können beispielsweise eine höhere Engine-Version oder eine andere Parametergruppe angeben.

Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Themen

- Angeben einer höheren Engine-Version

- Geben Sie eine andere DB-Parametergruppe an
- Ändern Sie die Speicher- und Leistungseinstellungen
- Aktivieren von RDS-optimierten Schreibvorgängen
- Aktualisieren der Speicherkonfiguration

Angeben einer höheren Engine-Version

Sie können eine höhere Engine-Version angeben, wenn Sie ein DB-Engine-Upgrade testen möchten. Bei der Umstellung wird die Datenbank auf die von Ihnen angegebene Haupt- oder Unterversion der DB-Engine aktualisiert.

Geben Sie eine andere DB-Parametergruppe an

Sie können testen, wie sich Parameteränderungen auf die DB-Instances in der grünen Umgebung auswirken, oder im Falle eines Upgrades eine Parametergruppe für eine neue Hauptversion der DB-Engine angeben.

Wenn Sie eine andere DB-Parametergruppe festlegen, wird die angegebene DB-Parametergruppe allen DB-Instances in der grünen Umgebung zugeordnet. Wenn Sie eine andere Parametergruppe festlegen, wird jede DB-Instance in der grünen Umgebung der Parametergruppe der entsprechenden blauen DB-Instance zugeordnet.

Ändern Sie die Speicher- und Leistungseinstellungen

Passen Sie die Speicher- und Leistungseinstellungen in einer umweltfreundlichen Umgebung an, um die Ressourcenzuweisung zu optimieren. Zu diesen Einstellungen gehören zugewiesener Speicher, bereitgestellte IOPS, Speichertyp und Speicherdurchsatz (für GP3-Speicher).

Sie können den Speichertyp der grünen DB-Instance auf gp2, gp3, io1 oder io2 ändern. Für GP3-Speicher können Sie auch den Speicherdurchsatz anpassen, um die Datenübertragungsleistung für Workloads mit hohem Bedarf zu verbessern oder um die Kosten für weniger intensive Anwendungen zu senken. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.

Sie können sich auch dafür entscheiden, den zugewiesenen Speicherplatz in einer umweltfreundlichen Umgebung zu erhöhen oder zu verringern. Eine Speicherreduzierung erfolgt jedoch nur, wenn der zugewiesene Zielspeicher mindestens 20% über der aktuellen Speichernutzung liegt. Wenn Sie den zugewiesenen Speicherplatz verringern, initiiert Amazon RDS ein Upgrade der

Speicherkonfiguration. Weitere Informationen finden Sie unter the section called “Aktualisieren der Speicherkonfiguration”.

Wenn die blaue DB-Instance Magnetspeicher verwendet, müssen Sie die grüne DB-Instance in den Speichertyp General Purpose oder Provisioned IOPS ändern, um den zugewiesenen Speicher zu erhöhen oder zu verringern.

Aktivieren von RDS-optimierten Schreibvorgängen

Sie können eine blaue/grüne Bereitstellung verwenden, um ein Upgrade auf eine DB-Instance-Klasse durchzuführen, die RDS-optimierte Schreibvorgänge unterstützt. Sie können RDS-optimierte Schreibvorgänge nur für eine Datenbank aktivieren, die mit einer unterstützten DB-Instance-Klasse erstellt wurde. Daher erstellt diese Option eine grüne Datenbank, die eine unterstützte DB-Instance-Klasse verwendet, so dass Sie RDS-optimierte Schreibvorgänge für die grüne DB-Instance aktivieren können.

Wenn Sie von einer DB-Instance-Klasse, die RDS-optimierte Schreibvorgänge nicht unterstützt, auf eine aktualisieren, die dies tut, müssen Sie auch die Speicherkonfiguration der grünen DB-Instance aktualisieren. Weitere Informationen finden Sie unter the section called “Aktualisieren der Speicherkonfiguration”.

Sie können nur die DB-Instance-Klasse der primären grünen DB-Instance aktualisieren. Standardmäßig übernehmen Lesereplikate in der grünen Umgebung die DB-Instance-Einstellungen von der blauen Umgebung. Nach erfolgreichem Erstellen der grünen Umgebung müssen Sie die DB-Instance-Klasse der Lesereplikate in der grünen Umgebung manuell ändern.

Je nach Engine-Version und Instance-Klasse der blauen DB-Instance werden einige Instance-Klassen-Upgrades nicht unterstützt. Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”.

Aktualisieren der Speicherkonfiguration

Wenn Ihre blaue Datenbank nicht die neueste Speicherkonfiguration aufweist, kann RDS die grüne DB-Instance von der älteren Speicherkonfiguration (32-Bit-Dateisystem) zu der bevorzugten Konfiguration migrieren. Sie können Blau/Grün-Bereitstellungen von RDS verwenden, um die Skalierungsbeschränkungen in Bezug auf Speicher und Dateigröße für ältere 32-Bit-Dateisysteme zu überwinden. Darüber hinaus ändert diese Einstellung die Speicherkonfiguration so, dass sie mit RDS-optimierten Schreibvorgängen kompatibel ist, wenn die angegebene DB-Instance-Klasse optimierte Schreibvorgänge unterstützt.

> **Note**
>
> Die Aktualisierung der Speicherkonfiguration ist ein I/O-intensiver Vorgang und führt zu längeren Erstellungszeiten für das blue/green deployments. The storage upgrade process is faster if the blue DB instance uses Provisioned IOPS SSD (io1 or io2 Block Express) storage, and if you provisoned the green environment with an instance size of 4xlarge or larger. Storage upgrades involving General Purpose SSD (gp2) storage can deplete your I/O Guthaben, was wiederum zu längeren Upgrade-Zeiten führt. Weitere Informationen finden Sie unter the section called “DB-Instance-Speicher”.
> Während des Speicher-Upgrades ist die grüne DB-Instance vorübergehend nicht verfügbar, während die blaue DB-Instance weiterhin verfügbar ist. Die Replikation wird während dieser Zeit unterbrochen. Überwachen Sie den Speicher auf der blauen Instance und ziehen Sie eine Skalierung in Betracht, wenn der Speicherplatz 90% erreicht, da die grüne Instance nach dem Upgrade automatisch um 10% skaliert wird.

Diese Option ist nur verfügbar, wenn Ihre blaue Datenbank nicht die neueste Speicherkonfiguration aufweist oder wenn Sie die DB-Instance-Klasse in derselben Anfrage ändern. Sie können die Speicherkonfiguration nur aktualisieren, wenn Sie zunächst eine blaue/grüne Bereitstellung erstellen.

Verzögertes Laden und Speicherinitialisierung für blaue/grüne Bereitstellungen

Wenn Sie eine Blaue/Grün-Bereitstellung vornehmen, erstellt Amazon RDS die primäre DB-Instance in der grünen Umgebung durch Wiederherstellung aus einem DB-Snapshot. Nach der Erstellung laden die grüne DB-Instance und ihre Read Replicas weiterhin Daten im Hintergrund. Dabei wird ein Prozess ausgeführt, der als Lazy Loading bezeichnet wird.

Beim verzögerten Laden werden Datenblöcke nur geladen, wenn Anwendungen sie anfordern. Wenn Sie versuchen, auf Daten zuzugreifen, die noch nicht geladen wurden, ruft Amazon EBS sie sofort von Amazon S3 ab, während die verbleibenden Daten weiterhin im Hintergrund geladen werden. Weitere Informationen finden Sie unter Amazon-EBS-Schnappschüsse.

Um die Leistung des vollen Volumes zu beschleunigen, bietet Amazon RDS eine Speicherinitialisierung, bei der alle Blöcke im Green-Environment-Volume gelesen werden. Amazon EBS lädt proaktiv Blöcke von Amazon S3 herunter und bietet so von der ersten Verwendung an maximale Volumenleistung. Die Speicherinitialisierung erfolgt vollständig im Hintergrund, sodass

keine Auswirkungen auf die Verfügbarkeit Ihrer DB-Instance oder laufende Aktivitäten wie Patches oder Upgrades auftreten.

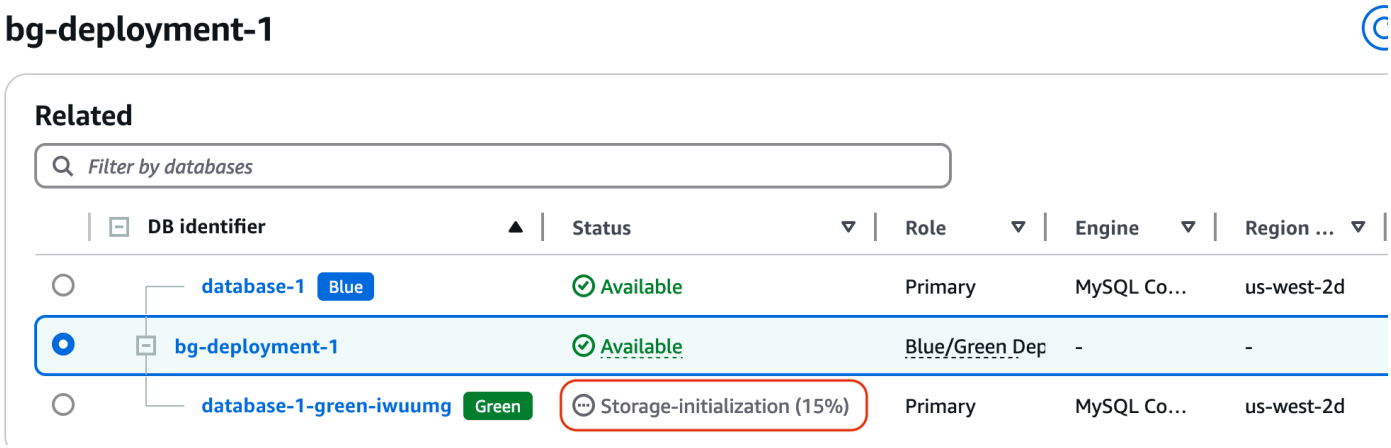
Die Speicherinitialisierung ist nur für Instances in Blau/Grün-Bereitstellungen mit den Datenträgertypen`gp2`,`gp3`, `io1` und verfügbar. `io2` Sie unterstützt alle Instance-Klassen mit Ausnahme der Familien t3 und t4. Wenn Sie eine grüne DB-Instance in einer Single-AZ-Bereitstellung in eine Multi-AZ-DB-Instance-Bereitstellung ändern, umfasst die Speicherinitialisierung den sekundären Knoten in der Multi-AZ-Konfiguration.

Während der Speicherinitialisierung bleibt die Instance vollständig verfügbar und für Datenbankoperationen nutzbar, obwohl der Speicher möglicherweise erst nach Abschluss der Initialisierung die volle Leistung erreicht. Während der Speicherinitialisierung ändert sich der Gesamtstatus der Instance in Speicherinitialisierung, und die Fortschrittsanzeige gibt die Mindestinitialisierungsstufe für alle Volumes der DB-Instance an.

Verwenden Sie die Konsole oder die Amazon RDS-API AWS CLI, um die Speicherinitialisierung zu überwachen.

Console

In der AWS Management Console sehen Sie den Fortschritt der Speicherinitialisierung mit dem DB-Instance-Status.



AWS CLI

Mit dem AWS CLI können Sie die Speicherinitialisierung mit dem Befehl überwachen. [describe-db-instances](#) Das `PercentProgress` Feld in der Antwort zeigt, wie viel Prozent der Daten von Amazon S3 abgerufen wurden.

```
aws rds describe-db-instances --db-instance-identifier my-db-instance
```

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "my-db-instance",
            "DBInstanceClass": "db.m5.2xlarge",
            "Engine": "postgres",
            "DBInstanceStatus": "storage-initialization",
            ...
            "PercentProgress": "34"
        }
    ]
}
```

Amazon RDS API

Mit der Amazon RDS-API rufen Sie den Status der Speicherinitialisierung ab, indem Sie die DBInstances Aktion Describe aufrufen.

Die Fortschrittsanzeige wird aktualisiert, wenn die Hintergrundinitialisierung voranschreitet, sodass Sie die Speicherbereitschaft verfolgen können, bevor die vollständige Speicherinitialisierung abgeschlossen ist. Die Speicherinitialisierung ermöglicht eine optimierte Leistung, sobald Ihre grüne DB-Instance voll funktionsfähig ist.

Erstellen einer Blau/Grün-Bereitstellung

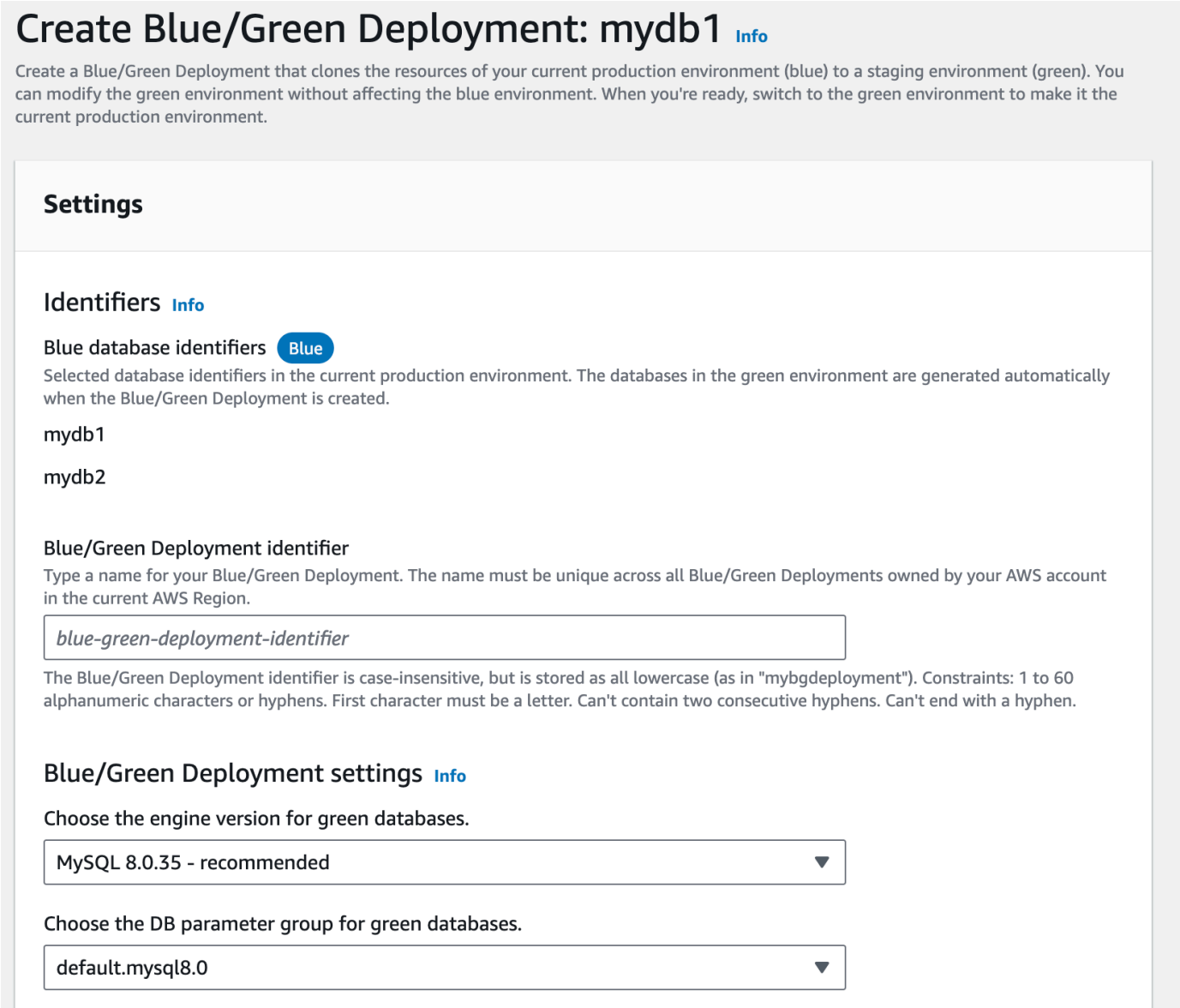
Sie können eine blaue/grüne Bereitstellung mithilfe der AWS Management Console, der oder der AWS CLI RDS-API erstellen.

Konsole

So erstellen Sie eine Blau/Grün-Bereitstellung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance zum Kopieren in eine grüne Umgebung aus.
3. Wählen Sie Aktionen, Blau/Grün-Bereitstellung erstellen.

   Die Seite „Blaue/grüne Bereitstellung erstellen“ wird angezeigt.



4. Überprüfen Sie die blauen Datenbankkennungen. Stellen Sie sicher, dass sie mit den DB-Instances übereinstimmen, die Sie in der blauen Umgebung erwarten. Wenn dies nicht der Fall ist, wählen Sie Cancel (Abbrechen) aus.
5. Geben Sie für den blauen/grünen Bereitstellungsnamen einen Namen für Ihre blaue/grüne Bereitstellung ein.
6. Geben Sie in den verbleibenden Abschnitten die Einstellungen für die grüne Umgebung an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter the section called “Verfügbare Einstellungen”.

   Sie können nach der Bereitstellung weitere Änderungen an den Datenbanken in der grünen Umgebung vornehmen.

7. Wählen Sie Erstellen aus.

AWS CLI

Verwenden Sie den create-blue-green-deploymentBefehl, um mit dem AWS CLI eine blaue/grüne Bereitstellung zu erstellen. Informationen zu allen verfügbaren Optionen finden Sie unterthe section called “Verfügbare Einstellungen”.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name my-blue-green-deployment \
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 \
    --target-engine-version 8.0.31 \
    --target-db-parameter-group-name mydbparametergroup
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-blue-green-deployment ^
    --blue-green-deployment-name my-blue-green-deployment ^
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 ^
    --target-engine-version 8.0.31 ^
    --target-db-parameter-group-name mydbparametergroup
```

RDS-API

Verwenden Sie den `CreateBlueGreenDeployment`Vorgang, um mithilfe der Amazon RDS-API eine blaue/grüne Bereitstellung zu erstellen. Informationen zu den jeweiligen Optionen finden Sie unter the section called “Verfügbare Einstellungen”.

Einstellungen für die Erstellung von blauen/grünen Bereitstellungen

In der folgenden Tabelle werden die Einstellungen erläutert, die Sie wählen können, wenn Sie eine blaue/grüne Bereitstellung erstellen. Weitere Informationen zu den AWS CLI Optionen finden Sie unter create-blue-green-deployment. Weitere Informationen zu den RDS-API-Parametern finden Sie unter CreateBlueGreenDeployment.

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Allocated storage	Die Menge an Speicherplatz, die Ihrer grünen DB-Instance zugewiesen werden soll (in Gibibytes). Sie können wählen, ob Sie den zugewiesenen Speicher erhöhen oder verringern möchten.
	
	Wenn Ihre blaue DB-Instance Magnetspeicher (`standard`) verwendet, müssen Sie die grüne DB-Instance in den Speichertyp General Purpose oder Provisioned IOPS ändern, um den zugewiese nen Speicher in der grünen Umgebung zu ändern.
	
	Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Sp eicher.
		CLI-Option:
		
		`--target-allocated-storage`
		
		API-Parameter:
		
		`TargetAllocatedStorage`

Blaue/grüne Bereitstellungs-ID	Ein Name für die blaue/grüne Bereitstellung.	CLI-Option:
		
		`--blue-green-deployment-nam e`
		
		API-Parameter:
		
		`BlueGreenDeploymentName`

Blauer Datenbank bezeichner	Die ID des , den Sie in die grüne Umgebung kopieren möchten. Wenn Sie die CLI oder API verwenden, geben Sie den Amazon Resource Name (ARN) des an.	CLI-Option:
		
		`--source`
		
		API-Parameter:
		
		`Source`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

für grüne Datenbanken	Eine Parametergruppe, die den Datenbanken in der grünen Umgebung zugeordnet werden soll.	CLI-Option:
		
		`--target-db-parameter-group-name`
		
		`--target-db-cluster-parameter-group-name`
		
		API-Parameter:
		
		`TargetDBParameterGroupName`
		
		`TargetDBClusterParameterGroupName`

Aktiviert optimierte Schreibvorgänge für grüne Datenbanken	Aktivieren Sie RDS-optimierte Schreibvorgänge auf der grünen primären DB-Instance. Weitere Informationen finden Sie unter the section called “Aktivieren von RDS-optimierten Schreibvorgängen”. Wenn Sie von einer DB-Instance-Klasse, die RDS-optimierte Schreibvorgänge nicht unterstützt, zu einer DB-Instance-Klasse wechseln, die RDS-optimierte Schreibvorgänge unterstützt, müssen Sie auch die Speicherkonfiguration aktualisieren. Weitere Informationen finden Sie unter the section called “Aktualisieren der Speicherkonfiguration”.	Für die CLI und API wird die Angabe einer Ziel-DB-Instance-Klasse, die RDS-optimierte Schreibvorgänge unterstützt, diese automatisch auf der grünen primären DB-Instance aktiviert.

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Engine-Version für grüne Datenbanken	Führen Sie ein Upgrade des in der grünen Umgebung auf die angegebene DB-Engine-Version durch.
	
	Falls nicht angegeben, wird jede Datenbank, in der grünen Umgebung, mit derselben Engine-Version wie der entsprechende in der blauen Umgebung erstellt.
	
	Wenn Sie sich für eine for PostgreSQL-DB-Instance entscheiden, die logische Replikation verwendet, überprüfen und bestätigen Sie die Einschränkungen der logischen Replikation. Weitere Informationen finden Sie unter the section called “Spezifische Einschränkungen der logischen Replikation für Bereitstellungen blue/green ”.
		CLI-Option:
		
		`--target-engine-version`
		
		RDS-API-Parameter:
		
		`TargetEngineVersion`

Grüne DB-Instance-Klasse	Zum Beispiel die Rechen- und Speicherkapazität jeder DB-Instance in der grünen Umgebung`db.m5d.xlarge` .
	
	Diese Option ist nur sichtbar, wenn Sie RDS-optimierte Schreibvorgänge für die grüne Datenbank aktivieren.
		CLI-Option:
		
		`--target-db-instance-class`
		
		RDS-API-Parameter:
		
		`TargetDBInstanceClass`

Konsolene instellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Bereitgestellte IOPS	Die Anzahl der bereitgestellten Eingabe-/Ausgabeoperationen pro Sekunde (IOPS), die ursprünglich der grünen Datenbank zugewiesen werden sollen.
	
	Dieser Wert gilt nur für die grüne primäre DB-Instance, nicht für grüne Replikate.
		CLI-Option:
		
		`--target-iops`
		
		RDS-API-Parameter:
		
		`TargetIops`

Upgrade der Speicherk onfiguration	Wählen Sie aus, ob Sie die Konfiguration Ihres Speicherd ateisystems aktualisieren möchten. Wenn Sie diese Einstellung aktivieren, migriert RDS die grüne Datenbank vom alten Speicherd ateisystem zur bevorzugten Konfiguration.
	
	Diese Option ist nur verfügbar, wenn Ihre blaue Datenbank nicht die neueste Speicherkonfiguration aufweist oder wenn Sie RDS-optim ierte Schreibvorgänge in derselben Anfrage aktivieren. Sie können die Speicherkonfiguration nur aktualisi eren, wenn Sie zunächst eine blaue/grüne Bereitstellung erstellen.
	
	Weitere Informationen finden Sie unter the section called “Upgrade des Speicherdateisystems”.
		CLI-Option:
		
		`--upgrade-target-storage-co nfig`
		
		RDS-API-Parameter:
		
		`UpgradeTargetStorageConfig`

Konsoleneinstellung	Beschreibung der Einstellung	CLI-Option und RDS-API-Parameter

Storage throughput (Speicherdurchsatz)	Der Wert für den Speicherdurchsatz für die grüne Datenbank. Diese Einstellung ist nur sichtbar, wenn Sie General Purpose SSD (gp3) als Speichertyp wählen.
	
	Dieser Wert gilt nur für die grüne primäre DB-Instance, nicht für grüne Replikate.
	
	Weitere Informationen finden Sie unter GP3-Speicher (empfohlen).
		CLI-Option:
		
		`--target-storage-throughput`
		
		RDS-API-Parameter:
		
		`TargetStorageThroughput`

Speichertyp	Der Speichertyp für die grüne Datenbank. Die folgenden Speichertypen werden unterstützt:
	
	• Allzweck-SSD (GP2)
	• Allzweck-SSD (gp3)
	• Bereitgestellte IOPS (io1)
	• Bereitgestellte IOPS-SSD (io2)
	
	Dieser Wert gilt nur für die grüne primäre DB-Instance, nicht für grüne Replikate.
	
	Weitere Informationen finden Sie unter Amazon RDS-Speichertypen.
		CLI-Option:
		
		`--target-storage-type`
		
		RDS-API-Parameter:
		
		`TargetStorageType`

Eine blaue/grüne Bereitstellung in Amazon RDS anzeigen

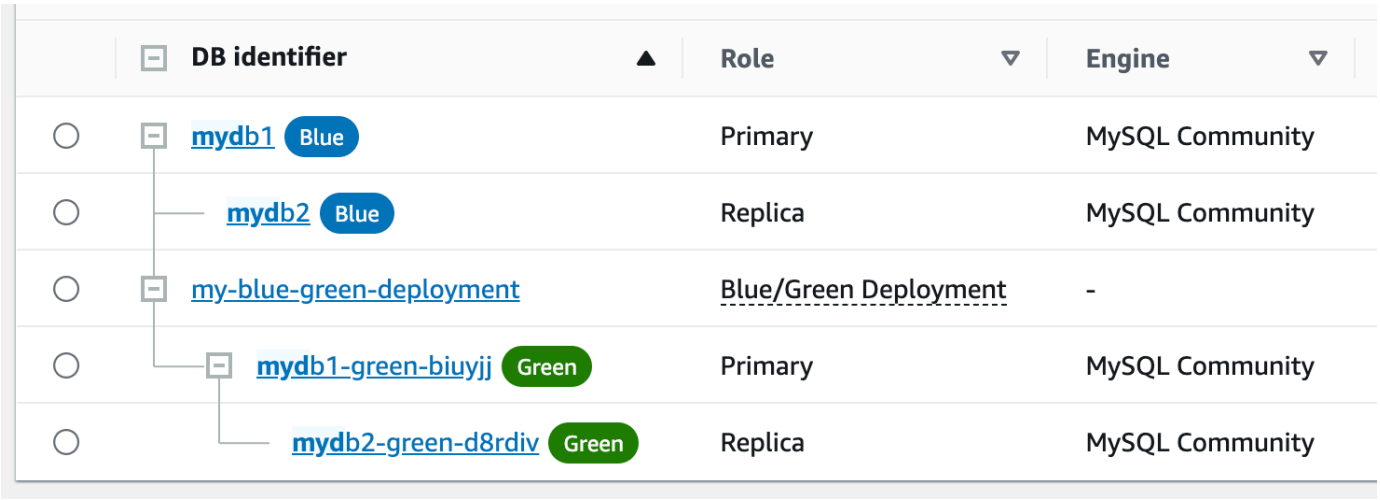
Sie können die Details zu einer blauen/grünen Bereitstellung mithilfe der AWS Management Console, der oder der AWS CLI RDS-API anzeigen.

Außerdem können Sie Ereignisse anzeigen und abonnieren, um Informationen über eine Blau/Grün-Bereitstellung zu erhalten. Weitere Informationen finden Sie unter Blau/Grün-Bereitstellungsereignisse.

Konsole

So zeigen Sie Details über eine Blau/Grün-Bereitstellung an

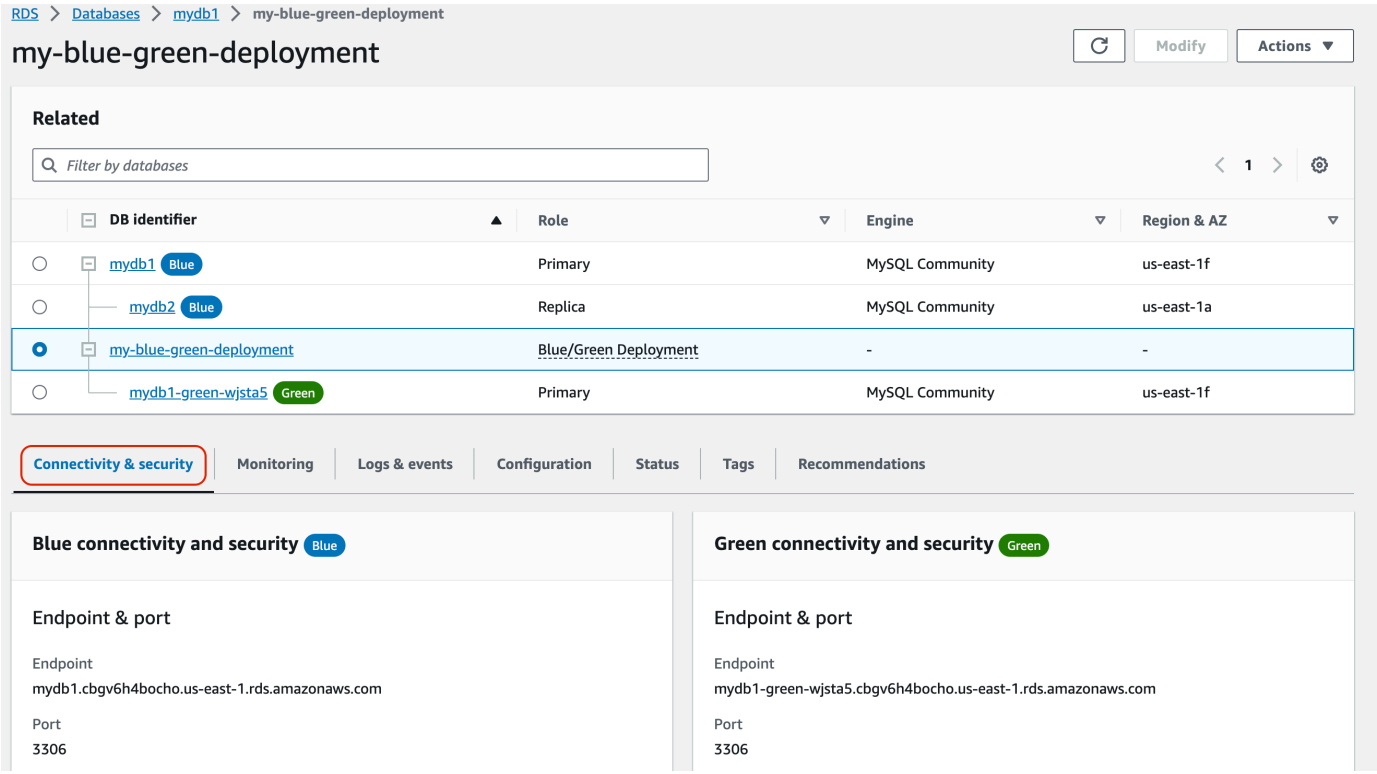
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus und suchen Sie dann in der Liste nach der Blau/Grün-Bereitstellung.



   Der Wert Role (Rolle) für die Blau/Grün-Bereitstellung ist Blue/Green Deployment (Blau/Grün-Bereitstellung).
3. Wählen Sie den Namen der Blau/Grün-Bereitstellung aus, die Sie anzeigen möchten, um die Details anzeigen zu lassen.

   Jede Registerkarte verfügt über einen Abschnitt für die blaue und einen Abschnitt für die grüne Bereitstellung. Beispielsweise kann die DB-Engine-Version auf der Registerkarte Konfiguration in der blauen Umgebung und in der grünen Umgebung unterschiedlich sein, wenn Sie die DB-Engine-Version in der grünen Umgebung aktualisieren.

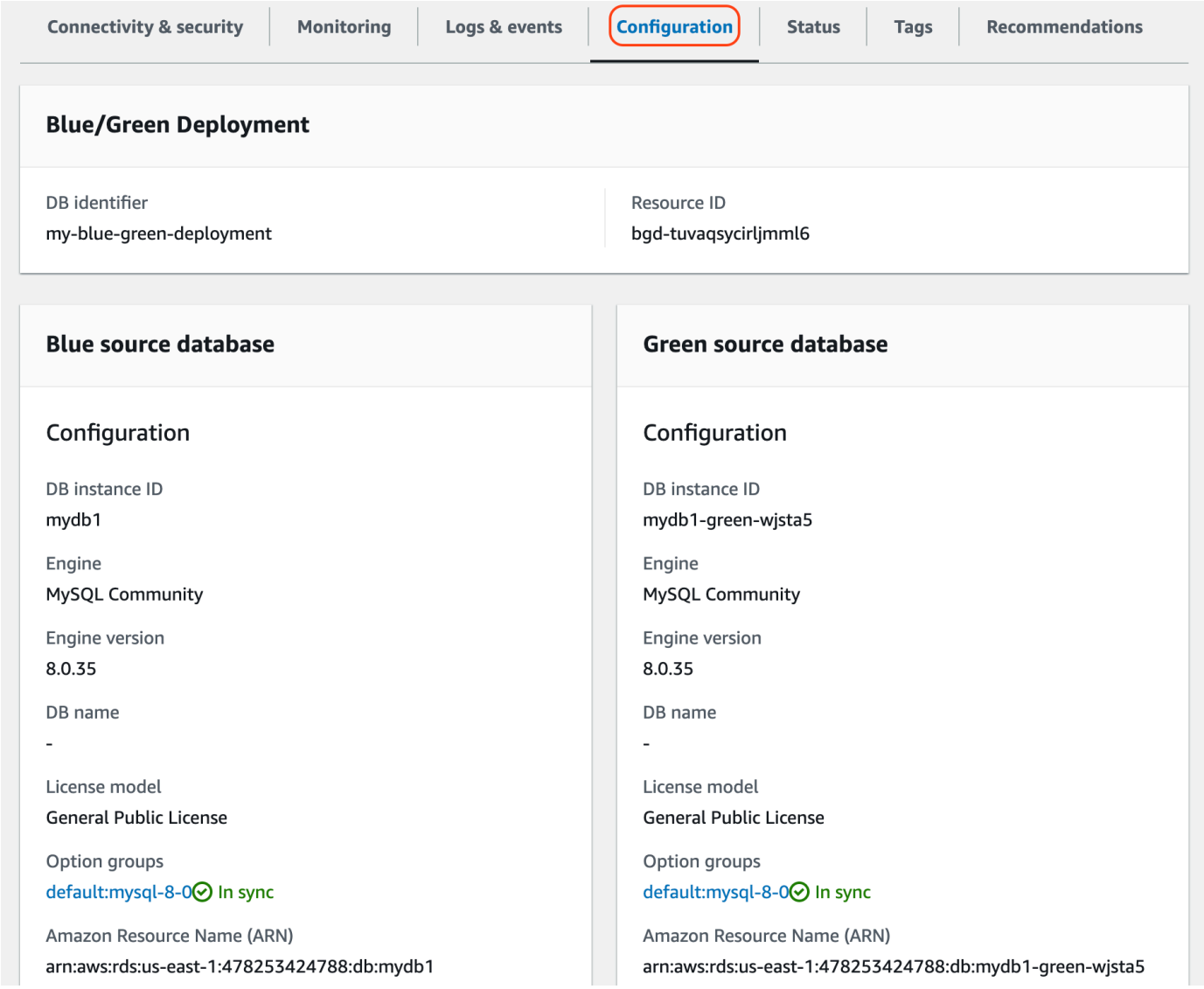
   Die folgende Abbildung zeigt ein Beispiel für die Registerkarte Konnektivität und Sicherheit:



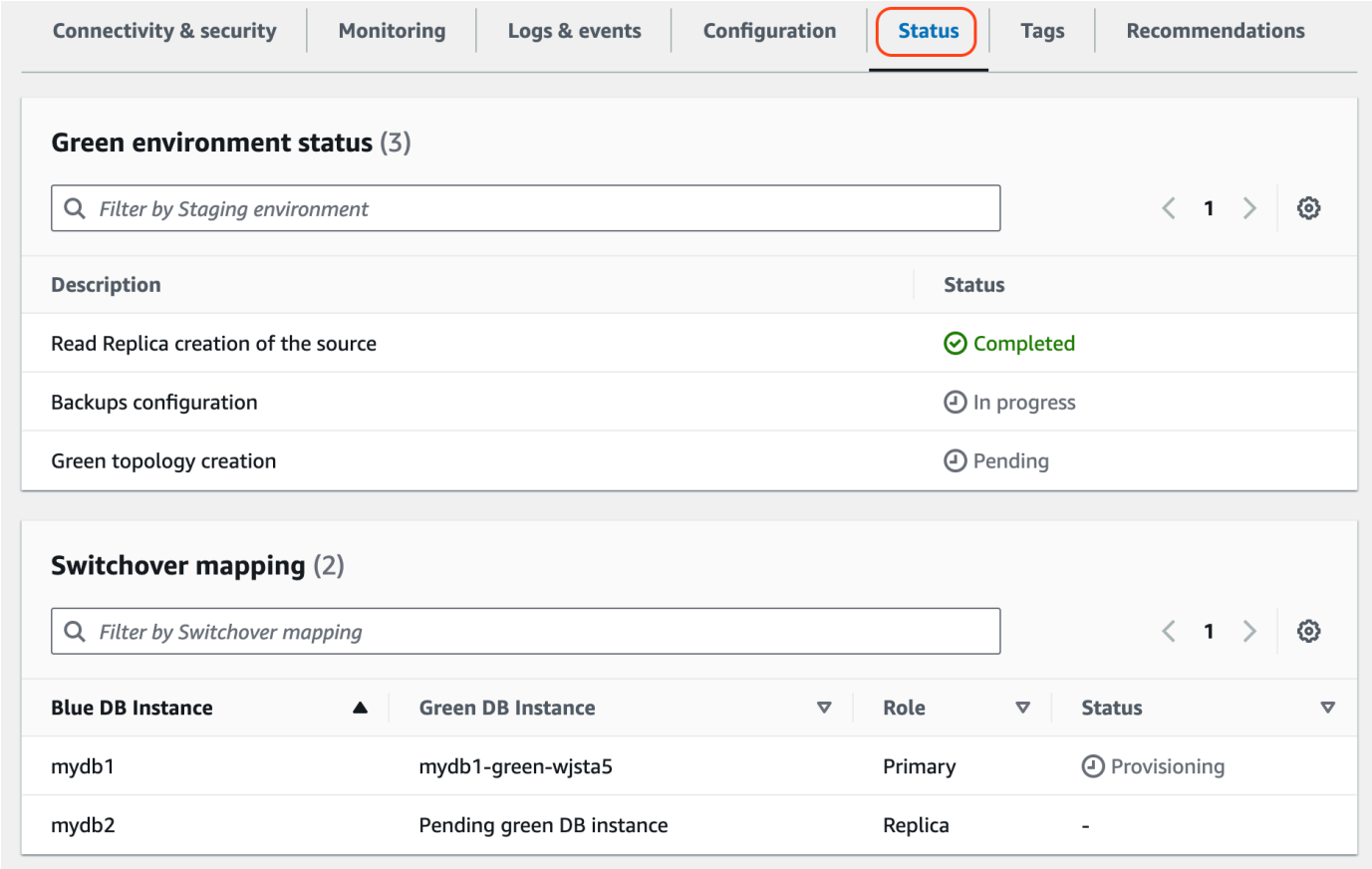
Die Registerkarte Konnektivität und Sicherheit enthält auch einen Abschnitt namens Replikation, in dem der aktuelle Status der Replikation und die Replikatverzögerung zwischen der blauen und der grünen Umgebung angezeigt werden. Wenn der Replikationsstatus `Replicating` lautet, wird die Blau/Grün-Bereitstellung erfolgreich repliziert.

Bei blau/grünen Bereitstellungen von RDS for PostgreSQL, die logische Replikation verwenden, kann sich der Replikationsstatus auf einen Wert ändern, `Replication degraded` wenn Sie in der blauen Umgebung nicht unterstützte DDL- oder große Objektänderungen vornehmen. Weitere Informationen finden Sie unter the section called “Spezifische Einschränkungen der logischen Replikation für Bereitstellungen blue/green ”.

Die folgende Abbildung zeigt ein Beispiel für die Registerkarte „Konfiguration“:



Die folgende Abbildung zeigt ein Beispiel für die Registerkarte Status:



AWS CLI

Verwenden Sie den describe-blue-green-deploymentsBefehl, um die Details zu einer blauen/grünen Bereitstellung mithilfe von anzuzeigen. AWS CLI

Example Anzeigen der Details einer Blau/Grün-Bereitstellung durch Filtern nach ihrem Namen

Wenn Sie den describe-blue-green-deploymentsBefehl verwenden, können Sie nach dem `--blue-green-deployment-name` filtern.

Das folgende Beispiel zeigt die Details für eine Blau/Grün-Bereitstellung mit dem Namen *`my-blue-green-deployment`*.

```
aws rds describe-blue-green-deployments \
  --filters Name=blue-green-deployment-name,Values=my-blue-green-deployment
```

Example Anzeigen der Details einer Blau/Grün-Bereitstellung durch Angabe der entsprechenden ID

Wenn Sie den describe-blue-green-deploymentsBefehl verwenden, können Sie die `--blue-green-deployment-identifier` Option angeben.

Das folgende Beispiel zeigt die Details für eine Blau/Grün-Bereitstellung mit der ID *bgd-1234567890abcdef*.

```
aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-1234567890abcdef
```

RDS-API

Wenn Sie die Details einer Blau/Grün-Bereitstellung mithilfe der Amazon-RDS-API anzeigen möchten, verwenden Sie den `DescribeBlueGreenDeployments`-Vorgang und geben Sie `BlueGreenDeploymentIdentifier` an.

Umschalten einer blauen/grünen Bereitstellung in Amazon RDS

Bei einem Switchover wird die grüne Umgebung zur neuen Produktionsumgebung. Wenn die grüne DB-Instance Repliken gelesen hat, werden diese ebenfalls umgestellt. Vor der Umstellung wird der Produktionsdatenverkehr an die DB-Instance und die Lesereplikate in der blauen Umgebung weitergeleitet. Nach der Umstellung wird der Produktionsdatenverkehr an die DB-Instance und die Lesereplikate in der grünen Umgebung weitergeleitet.

Das Umschalten auf eine blaue/grüne Bereitstellung ist nicht dasselbe wie das Heraufstufen des grünen innerhalb der blauen/grünen Bereitstellung. Wenn Sie den grünen manuell heraufstufen, indem Sie im Menü Aktionen die Option Heraufstufen wählen, wird die Replikation zwischen der blauen und der grünen Umgebung unterbrochen und die blaue/grüne Bereitstellung wechselt in den Status Ungültige Konfiguration.

Themen

- Umstellungs-Timeout
- Integrationsschutz der Umstellung
- Umstellungsaktionen
- Bewährte Methoden für die Umstellung
- Überprüfung der CloudWatch Metriken vor dem Switchover
- Überwachung der Replikatverzögerung vor dem Switchover

- Umstellen einer Blau/Grün-Bereitstellung
- Nach der Umstellung

Umstellungs-Timeout

Sie können einen Umstellungs-Timeout zwischen 30 Sekunden und 3 600 Sekunden (eine Stunde) festlegen. Wenn die Umstellung länger als angegeben dauert, werden alle Änderungen rückgängig gemacht und es werden keine Änderungen an einer der Umgebungen vorgenommen. Der Standardwert für den Timeout beträgt 300 Sekunden (fünf Minuten).

Integrationsschutz der Umstellung

Wenn Sie eine Umstellung starten, führt Amazon RDS einige grundlegende Prüfungen durch, um zu testen, ob die blaue und die grüne Umgebung für die Umstellung bereit sind. Diese Prüfungen werden als Integrationsschutz der Umstellung. Dieser Integrationsschutz verhindert eine Umstellung, wenn die Umgebungen dafür nicht bereit sind. Mit diesem Schutz werden längere Ausfallzeiten als erwartet vermieden und Datenverluste zwischen der blauen und der grünen Umgebung verhindern, die sich ergeben könnten, wenn die Umstellung gestartet wird.

Amazon RDS führt die folgenden Integrationsschutzprüfungen in der grünen Umgebung durch:

- Zustand der Replikation – Es wird geprüft, ob der Replikationsstatus der grünen primären DB-Instance fehlerfrei ist. Die grüne primäre DB-Instance ist ein Replikat der blauen primären DB-Instance.
- Replikationsverzögerung – Es wird geprüft, ob die Replikatverzögerung der grünen primären DB-Instance innerhalb der für die Umstellung zulässigen Grenzwerte liegt. Die zulässigen Grenzwerte basieren auf dem angegebenen Timeout-Zeitraum. Die Replikatverzögerung gibt an, wie weit die grüne primäre DB-Instance hinter ihrer blauen primären DB-Instance zurückbleibt. Weitere Informationen finden Sie unter the section called “Überwachung der Replikatverzögerung vor dem Switchover”.
- Aktive Schreibvorgänge – Es wird sichergestellt, dass es auf der grünen primären DB-Instance keine aktiven Schreibvorgänge gibt.

Amazon RDS führt die folgenden Integrationsschutzprüfungen in der blauen Umgebung durch:

- Externe Replikation — Stellt für RDS for PostgreSQL sicher, dass es sich bei der blauen Umgebung nicht um eine selbstverwaltete logische Quelle (Herausgeber) oder Replikat (Abonnent)

handelt. Ist dies der Fall, empfehlen wir, die selbstverwalteten Replikations-Slots und Abonnements für alle Datenbanken in der blauen Umgebung zu löschen, mit dem Switchover fortzufahren und sie dann neu zu erstellen, um die Replikation fortzusetzen. Prüft für RDS für MySQL und RDS für MariaDB, ob es sich bei der Blue-Datenbank nicht um ein externes Binlog-Replikat handelt. Ist dies der Fall, stellen Sie sicher, dass sie nicht aktiv repliziert wird.

- Lang andauernde aktive Schreibvorgänge – Es wird sichergestellt, dass auf der blauen primären DB-Instance keine lang andauernden aktiven Schreibvorgänge vorhanden sind, da diese die Replikatverzögerung erhöhen können.
- Lang andauernde DDL-Anweisungen – Es wird sichergestellt, dass auf der blauen primären DB-Instance keine lang andauernden DDL-Anweisungen vorhanden sind, da diese die Replikatverzögerung erhöhen können.
- Nicht unterstützte PostgreSQL-Änderungen — Bei blau/grünen Bereitstellungen von RDS for PostgreSQL, die logische Replikation verwenden, wird sichergestellt, dass in der blauen Umgebung keine DDL-Änderungen und keine Hinzufügungen oder Änderungen großer Objekte vorgenommen wurden. Weitere Informationen finden Sie unter the section called “Spezifische Einschränkungen der logischen Replikation für Bereitstellungen blue/green ”.

  Wenn Amazon RDS nicht unterstützte PostgreSQL-Änderungen erkennt, ändert es den Replikationsstatus auf `Replication degraded` und benachrichtigt Sie, dass Switchover für die blue/green deployment. To proceed with switchover, we recommend that you delete and recreate the blue/green Bereitstellung und alle grünen Datenbanken nicht verfügbar ist. Wählen Sie dazu Aktionen, Mit grünen Datenbanken löschen aus.

Umstellungsaktionen

Wenn Sie auf eine Blau/Grün-Bereitstellung umstellen, führt RDS die folgenden Aktionen aus:

1. Es führt Integritätsschutzprüfungen durch, um zu überprüfen, ob die blaue und die grüne Umgebung bereit für die Umstellung sind.
2. Es stoppt neue Schreibvorgänge auf der primären DB-Instance in beiden Umgebungen.
3. Es trennt Verbindungen mit den DB-Instances in beiden Umgebungen und erlaubt keine neuen Verbindungen.
4. Es wartet, bis die Replikation in der grünen Umgebung aufgeholt hat, sodass die grüne Umgebung mit der blauen Umgebung synchron ist.
5. Es benennt die DB-Instances in beiden Umgebungen um.

RDS benennt die DB-Instances in der grünen Umgebung um, sodass sie den jeweiligen DB-Instances in der blauen Umgebung entsprechen. Angenommen, der Name einer DB-Instance in der blauen Umgebung lautet `mydb`. Nehmen wir außerdem an, dass der Name der entsprechenden DB-Instance in der grünen Umgebung `mydb-green-abc123` lautet. Während der Umstellung wird der Name der DB-Instance in der grünen Umgebung in `mydb` geändert.

RDS benennt die DB-Instances in der blauen Umgebung um, indem `-old`*n* an den aktuellen Namen angehängt wird, wobei *n* eine Zahl ist. Angenommen, der Name einer DB-Instance in der blauen Umgebung lautet `mydb`. Nach der Umstellung könnte der Name der DB-Instance `mydb-old1` lauten.

RDS benennt auch die Endpunkte in der grünen Umgebung um, sodass sie mit den entsprechenden Endpunkten in der blauen Umgebung übereinstimmen und keine Anwendungsänderungen erforderlich sind.

6. Erlaubt Verbindungen mit Datenbanken in beiden Umgebungen.
7. Erlaubt neue Schreibvorgänge auf der primären DB-Instance in der neuen Produktionsumgebung.

   Nach dem Switchover erlaubt der nur Lesevorgänge, bis Sie den `read_only` Parameter (für RDS für MySQL) oder den `default_transaction_read_only` Parameter (für RDS für PostgreSQL) auf setzen `0` und die DB-Instance neu starten.

Sie können den Status eines Switchovers mit Amazon EventBridge überwachen. Weitere Informationen finden Sie unter the section called “Blau/Grün-Bereitstellungsereignisse”.

Wenn Sie Tags in der blauen Umgebung konfiguriert haben, werden diese Tags beim Switchover in die neue Produktionsumgebung kopiert. Weitere Informationen zu Tags erhalten Sie unter Verschlagwortung von Amazon-Ressourcen RDS.

Wenn die Umstellung startet und aus irgendeinem Grund vorzeitig beendet wird, werden alle Änderungen rückgängig gemacht und es werden keine Änderungen an einer der Umgebungen vorgenommen.

Bewährte Methoden für die Umstellung

Wir empfehlen Ihnen dringend, sich an bewährte Methoden zu halten und vor der Umstellung die folgenden Aufgaben auszuführen:

- Testen Sie die Ressourcen in der grünen Umgebung gründlich. Stellen Sie sicher, dass sie ordnungsgemäß und effizient funktionieren.
- Überwachen Sie relevante CloudWatch Amazon-Metriken. Weitere Informationen finden Sie unter the section called “Überprüfung der CloudWatch Metriken vor dem Switchover”.
- Ermitteln Sie den besten Zeitpunkt für die Umstellung.

  Während der Umstellung werden Schreibvorgänge in beiden Umgebungen von den Datenbanken abgeschnitten. Ermitteln Sie einen Zeitpunkt, an dem der Datenverkehr in Ihrer Produktionsumgebung am niedrigsten ist. Transaktionen mit langer Laufzeit, z. B. aktive DDLs, können Ihre Umstellungszeit verlängern, was zu längeren Ausfallzeiten Ihrer Produktionsworkloads führt.

  Wenn Ihre DB-Instances, eine große Anzahl von Verbindungen aufweisen, sollten Sie erwägen, diese manuell auf das für Ihre Anwendung erforderliche Minimum zu reduzieren, bevor Sie die blue/green deployment. One way to achieve this is to create a script that monitors the status of the blue/green Bereitstellung umstellen und mit der Bereinigung der Verbindungen beginnen, wenn festgestellt wird, dass sich der Status zu geändert hat. `SWITCHOVER_IN_PROGRESS`
- Stellen Sie sicher, dass die DB-Instances in beiden Umgebungen den Status `Available` aufweisen.
- Stellen Sie sicher, dass die primäre DB-Instance in der grünen Umgebung fehlerfrei ist und repliziert wird.
- Stellen Sie sicher, dass Ihre Netzwerk- und Client-Konfigurationen den DNS-Cache Time-To-Live (TTL) nicht über fünf Sekunden hinaus erhöhen. Dies ist die Standardeinstellung für RDS-DNS-Zonen.
  Andernfalls senden Anwendungen nach der Umstellung weiterhin Schreibdatenverkehr an die blaue Umgebung.
- Stellen Sie sicher, dass das Laden der Daten abgeschlossen ist, bevor Sie umstellen. Weitere Informationen finden Sie unter the section called “Verzögertes Laden und Speicherinitialisierung”.
- Gehen Sie für , RDS für PostgreSQL, blaue/grüne Bereitstellungen, die logische Replikation verwenden, wie folgt vor:
  - Überprüfen Sie die Einschränkungen der logischen Replikation und ergreifen Sie vor dem Switchover alle erforderlichen Maßnahmen. Weitere Informationen finden Sie unter the section called “Spezifische Einschränkungen der logischen Replikation für Bereitstellungen blue/green ”.
  - Führen Sie die `ANALYZE`-Operation aus, um die Tabelle `pg_statistics` zu aktualisieren. Dadurch wird das Risiko von Leistungsproblemen nach dem Switchover reduziert.

> **Note**
>
> Während einer Umstellung können Sie keine der DB-Instances in der Umstellung ändern.

Überprüfung der CloudWatch Metriken vor dem Switchover

- `DatabaseConnections` – Verwenden Sie diese Metrik, um das Aktivitätsniveau in der Blau/Grün-Bereitstellung abzuschätzen, und stellen Sie sicher, dass der Wert für Ihre Bereitstellung auf einem akzeptablen Niveau liegt, bevor Sie umsteigen. Wenn Performance Insights aktiviert ist, ist `DBLoad` eine genauere Metrik.

Weitere Informationen finden Sie unter the section called “CloudWatch Metriken für RDS”.

Überwachung der Replikatverzögerung vor dem Switchover

Bevor Sie zu einer blauen/grünen Bereitstellung wechseln, sollten Sie sicherstellen, dass die Replikatverzögerung nahezu Null ist, um Ausfallzeiten zu reduzieren.

RDS für MySQL und RDS für MariaDB

Überprüfen Sie bei blauen/grünen MySQL - und MariaDB-Bereitstellungen die CloudWatch Metrik in der grünen Umgebung, um die aktuelle Replikatverzögerung zu ermitteln. Weitere Informationen finden Sie unter the section called “Diagnose und Lösung bei Verzögerungen zwischen Read Replicas (Lesereplikaten)”.

Anweisungen zur Identifizierung der aktuellen Replikatverzögerung finden the section called “Überwachen und Optimieren des Replikationsprozesses” Sie unter Blau/Grün-Bereitstellungen von PostgreSQL, die physische Replikationsreplikation verwenden.

Bei blauen/grünen PostgreSQL-Bereitstellungen, die logische Replikation verwenden, überprüfen Sie die `OldestReplicationSlotLag` CloudWatch Metrik in der blauen Umgebung, um die aktuelle Replikatverzögerung zu ermitteln. Weitere Informationen finden Sie unter the section called “Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS”.

Darüber hinaus können Sie die folgende SQL-Abfrage in der blauen Umgebung ausführen:

```
SELECT slot_name,
       confirmed_flush_lsn as flushed,
       pg_current_wal_lsn(),
       (pg_current_wal_lsn() - confirmed_flush_lsn) AS lsn_distance
FROM pg_catalog.pg_replication_slots
WHERE slot_type = 'logical';

slot_name        |    flushed    | pg_current_wal_lsn | lsn_distance
-----------------+---------------+--------------------+------------
logical_replica1 | 47D97/CF32980 | 47D97/CF3BAC8      | 37192
```

Die `confirmed_flush_lsn` steht für die letzte Protokollsequenznummer (LSN), die an das Replikat gesendet wurde. Die `pg_current_wal_lsn` stellt dar, wo sich die Datenbank jetzt befindet. Ein `lsn_distance` Wert von 0 bedeutet, dass das Replikat abgeholt wurde.

Eine Erläuterung, wann bei blauen/grünen Bereitstellungen physische Replikation und wann logische Replikation verwendet wird, finden Sie unter. the section called “Postgre-Replikationsmethoden SQL”

Umstellen einer Blau/Grün-Bereitstellung

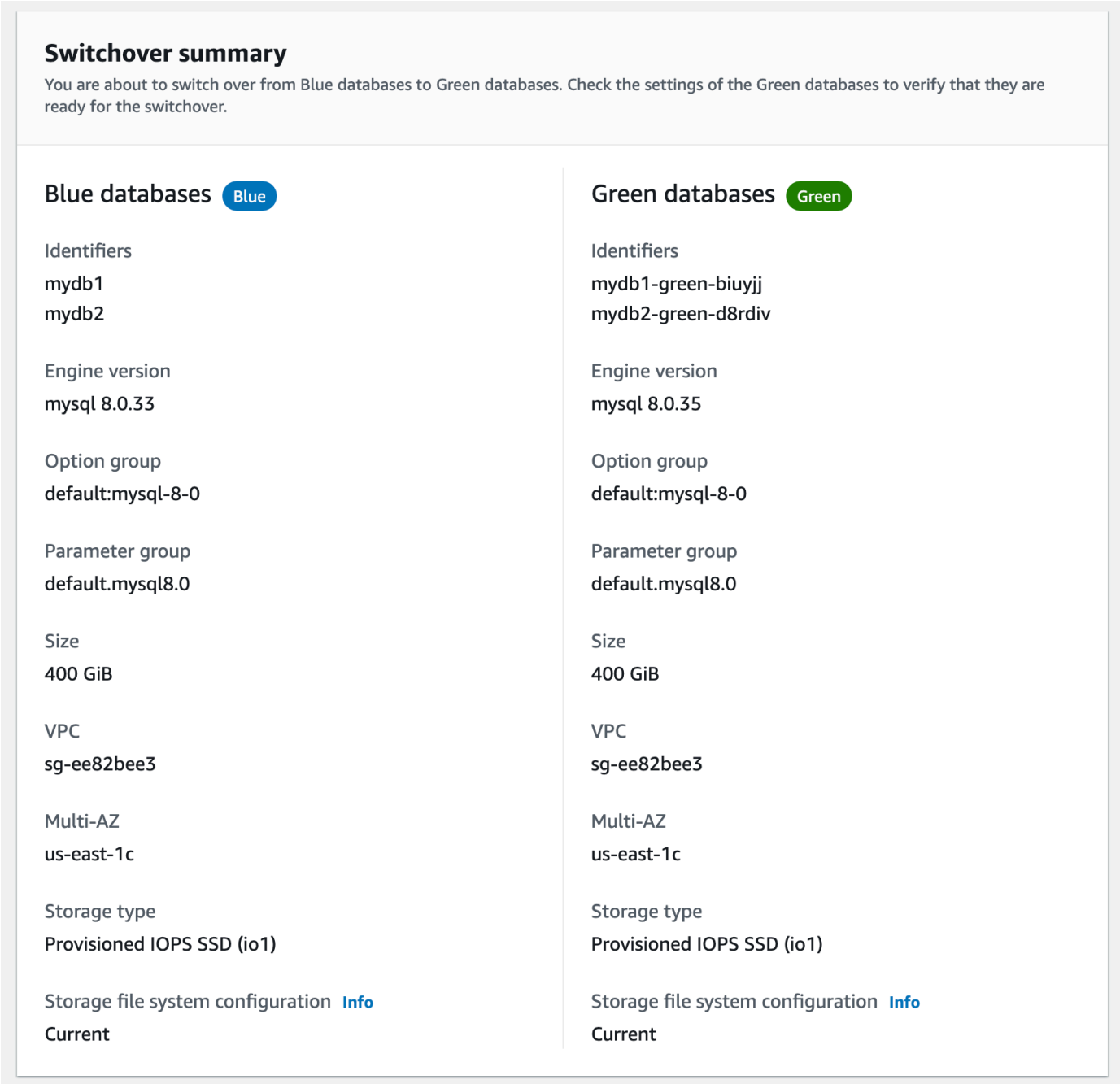
Sie können mit der, der oder der AWS Management Console RDS-API zu einer AWS CLI blauen/grünen Bereitstellung wechseln.

Konsole

So stellen Sie eine Blau/Grün-Bereitstellung um

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die Blau/Grün-Bereitstellung aus, die Sie umstellen möchten.
3. Wählen Sie unter Actions (Aktionen) die Option Switch over (Umstellen) aus.

   Die Seite Switch Over (Umstellen) wird angezeigt.



4. Sehen Sie sich auf der Seite Switchover (Umstellen) die Umstellungszusammenfassung an. Stellen Sie sicher, dass die Ressourcen in beiden Umgebungen Ihren Erwartungen entsprechen. Wenn dies nicht der Fall ist, wählen Sie Cancel (Abbrechen) aus.
5. Geben Sie unter Timeout-Einstellungen das Zeitlimit für die Umstellung ein.
6. Wenn auf Ihrer Instance RDS für PostgreSQL ausgeführt wird, überprüfen und bestätigen Sie die vor der Umstellung zu berücksichtigenden Empfehlungen. Weitere Informationen finden Sie unter the section called “Spezifische Einschränkungen der logischen Replikation für Bereitstellungen blue/green ”.

7. Wählen Sie Switch over (Umstellen) aus.

AWS CLI

Verwenden Sie den switchover-blue-green-deploymentBefehl mit den folgenden Optionen AWS CLI, um mit dem zu einer blauen/grünen Bereitstellung zu wechseln:

- `--blue-green-deployment-identifier`— Geben Sie die Ressourcen-ID der blauen/grünen Bereitstellung an.
- `--switchover-timeout` – Geben Sie das Zeitlimit für die Umstellung in Sekunden an. Der Standardwert ist 300.

Example Umstellen einer Blau/Grün-Bereitstellung

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --switchover-timeout 600
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds switchover-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --switchover-timeout 600
```

RDS-API

Verwenden Sie die `SwitchoverBlueGreenDeployment`-Operation mit den folgenden Parametern, um mithilfe der Amazon-RDS-API eine Blau/Grün-Bereitstellung umzustellen:

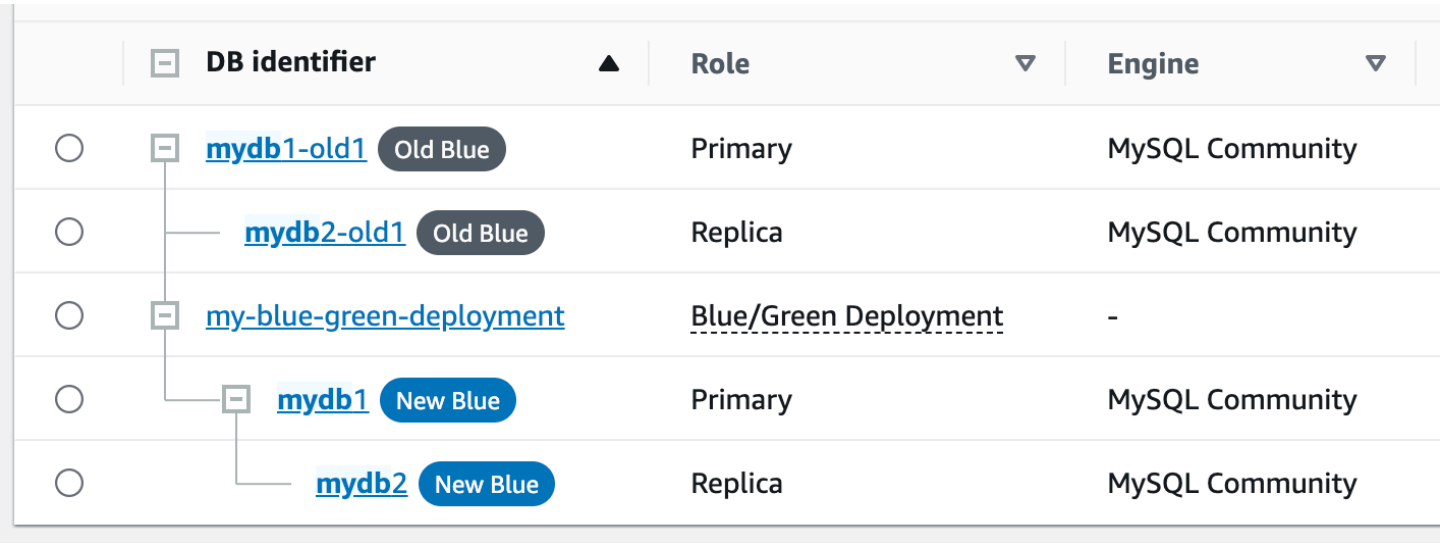
- `BlueGreenDeploymentIdentifier`— Geben Sie die Ressourcen-ID der blauen/grünen Bereitstellung an.
- `SwitchoverTimeout` – Geben Sie das Zeitlimit für die Umstellung in Sekunden an. Der Standardwert ist 300.

Nach der Umstellung

Nach einer Umstellung werden die DB-Instances in der vorherigen blauen Umgebung beibehalten. Für diese Ressourcen fallen die Standardkosten an. Die Replikation zwischen der blauen und der grünen Umgebung werden gestoppt.

RDS benennt die DB-Instances in der blauen Umgebung um, indem `-old`*n* an den aktuellen Namen angehängt wird, wobei *n* eine Zahl ist. Die DB-Instances in der alten blauen Umgebung sind schreibgeschützt, bis Sie den `read_only` Parameter (für RDS für MySQL) oder den `default_transaction_read_only` Parameter (für RDS für PostgreSQL) auf setzen. `0` RDS benennt die DB-Instances als in der grünen Umgebung. `-new`*n*

Wenn Sie die blaue/grüne Bereitstellungsressource löschen, behält RDS die `-new`*n* Ressourcen `-old`*n* und bei.



Aktualisierung des übergeordneten Knotens für Verbraucher

RDS bietet vollständig verwaltete Read Replicas. Es bietet jedoch auch die Möglichkeit, selbstverwaltete Replikate einzurichten, die auch als externe Replikate bezeichnet werden. Mit externen Replikaten können Sie Ressourcen von Drittanbietern als Replikationsziele verwenden.

Wenn Sie eine blaue/grüne Bereitstellung von RDS für MariaDB oder RDS für MySQL umgestellt haben und der blaue vor dem Switchover externe Replikate oder Binärprotokollverbraucher hatte, müssen Sie deren übergeordneten Knoten nach dem Switchover aktualisieren, um die Replikationskontinuität aufrechtzuerhalten.

Um den übergeordneten Knoten zu aktualisieren

1. Nach dem Switchover gibt die , die sich zuvor in der grünen Umgebung befand, ein Ereignis aus, das den Namen der Master-Log-Datei und die Master-Log-Position enthält. Um das Ereignis zu finden, navigieren Sie zur RDS-Konsole und wählen Sie im linken Navigationsbereich Ereignisse aus.
2. Filtern Sie vor dem Switchover nach Ereignissen, bei denen die Quelle der Name der alten Green ist.
3. Suchen Sie das Ereignis, das die binären Log-Koordinaten enthält. Die Ereignismeldung ähnelt:`Binary log coordinates in green environment after switchover: file mysql-bin-changelog.000003 and position 40134574`.
4. Stellen Sie sicher, dass der Verbraucher oder das Replikat alle Binärprotokolle aus der alten blauen Umgebung übernommen hat. Verwenden Sie dann die bereitgestellten Binärprotokollkoordinaten, um die Replikation auf den Verbrauchern fortzusetzen. Wenn Sie beispielsweise ein MySQL-Replikat auf ausführen EC2, können Sie die folgenden Befehle verwenden:

   MySQL 8.0.22 und niedrigere Haupt- und Nebenversionen

   ```
   CHANGE MASTER TO MASTER_HOST='{new-writer-endpoint}', MASTER_LOG_FILE='mysql-bin-changelog.000003', MASTER_LOG_POS=40134574;
   ```

   MySQL 8.0.23 und höhere Haupt- und Nebenversionen

   ```
   CHANGE REPLICATION SOURCE TO SOURCE_HOST='{new-writer-endpoint}',
    SOURCE_LOG_FILE='mysql-bin-changelog.000003', SOURCE_LOG_POS=40134574;
   ```

Wenn es sich bei dem Verbraucher um eine andere RDS for MySQL- oder RDS for MariaDB-DB-Instance handelt, führen Sie die folgenden gespeicherten Prozeduren der Reihe nach aus:

1. the section called “mysql.rds_stop_replication”
2. mysql.rds_reset_external_master (für Version 8.0 und niedriger) oder mysql_rds_reset_external_source (für Version 8.4 und höher)
3. mysql.rds_set_external_master (für Version 8.0 und niedriger) oder mysql_rds_set_external_source (für Version 8.4 und höher)
4. the section called “mysql.rds_start_replication”

Löschen einer blauen/grünen Bereitstellung in Amazon RDS

Sie können eine Blau/Grün-Bereitstellung vor oder nach der Umstellung löschen.

Wenn Sie eine Blau/Grün-Bereitstellung löschen, bevor Sie sie umstellen, löscht Amazon RDS optional die/den DB-Instances in der Grün-Umgebung:

- Wenn Sie die DB-Instances in der grünen Umgebung (`--delete-target`) löschen möchten, stellen Sie sicher, dass ihr Löschschutz nicht aktiviert ist.
- Wenn Sie die/den DB-Instances in der Grün-Umgebung (`--no-delete-target`) nicht löschen, werden/wird die Instances beibehalten, sie sind/ist jedoch nicht länger Teil einer Blau-/Grün-Bereitstellung. Für RDS for MySQL wird die Replikation zwischen den Umgebungen fortgesetzt. Für RDS for PostgreSQL wird die grüne Umgebung zu einer eigenständigen Umgebung heraufgestuft, sodass die Replikation gestoppt wird.

Die Option zum Löschen der grünen Datenbanken ist in der Konsole nach der Umstellung nicht verfügbar. Wenn Sie blaue/grüne Bereitstellungen mit dem löschen, können Sie die `--delete-target` Option nicht angeben AWS CLI, wenn der Bereitstellungsstatus lautet. `SWITCHOVER_COMPLETED`

> **Important**
>
> Das Löschen einer Blau/Grün-Bereitstellung wirkt sich nicht auf die blaue Umgebung aus.

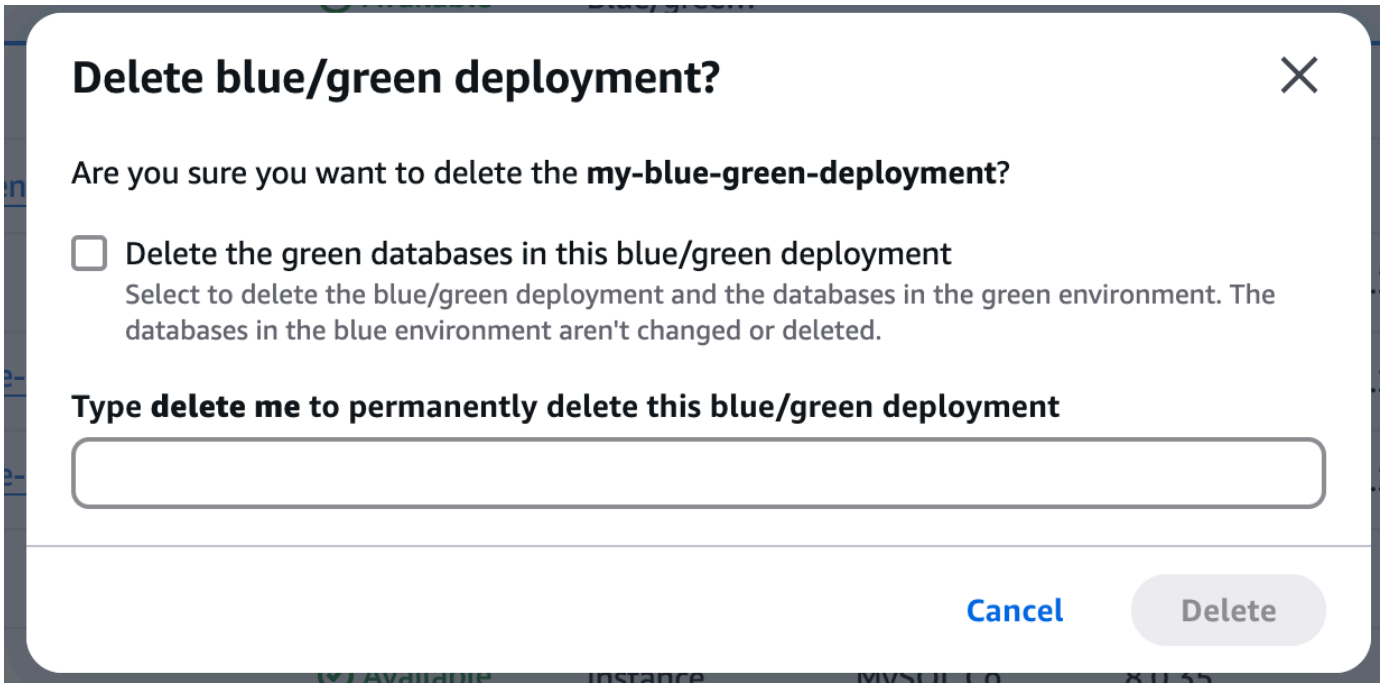
Sie können eine blaue/grüne Bereitstellung mithilfe der AWS Management Console, der oder der AWS CLI RDS-API löschen.

Konsole

So löschen Sie eine Blau/Grün-Bereitstellung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die Blau/Grün-Bereitstellung aus, die Sie löschen möchten.
3. Klicken Sie bei Actions auf Delete.

Die Bereitstellung „Blau/Grün löschen“? ein Fenster wird angezeigt.



Um die grünen Datenbanken zu löschen, wählen Sie Die grünen Datenbanken in dieser blauen/grünen Bereitstellung löschen aus.

4. Geben Sie **`delete me`** in das Feld ein.
5. Wählen Sie Löschen aus.

AWS CLI

Um eine blaue/grüne Bereitstellung mithilfe von zu löschen AWS CLI, verwenden Sie den delete-blue-green-deploymentBefehl mit den folgenden Optionen:

- `--blue-green-deployment-identifier`— Die Ressourcen-ID der blauen/grünen Bereitstellung, die gelöscht werden soll.
- `--delete-target` – Gibt an, dass die DB-Instances in der grünen Umgebung gelöscht werden . Sie können diese Option nicht angeben, wenn die Blau/Grün-Bereitstellung den Status `SWITCHOVER_COMPLETED` hat.
- `--no-delete-target` – Gibt an, dass die DB-Instances in der grünen Umgebung beibehalten werden .

Example Löschen einer Blau/Grün-Bereitstellung und der DB-Instances in der grünen Umgebung

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --delete-target
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --delete-target
```

Example Löschen einer Blau/Grün-Bereitstellung unter Beibehaltung der DB-Instances in der grünen Umgebung

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --no-delete-target
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --no-delete-target
```

RDS-API

Verwenden Sie die `DeleteBlueGreenDeployment`-Operation mit den folgenden Parametern, um mithilfe der Amazon-RDS-API eine Blau/Grün-Bereitstellung zu löschen:

- `BlueGreenDeploymentIdentifier`— Die Ressourcen-ID der blauen/grünen Bereitstellung, die gelöscht werden soll.
- `DeleteTarget` – Geben Sie `TRUE` an, um die DB-Instances in der grünen Umgebung zu löschen, oder `FALSE`, um sie beizubehalten. Darf nicht `TRUE` sein, wenn die Blau/Grün-Bereitstellung den Status `SWITCHOVER_COMPLETED` hat.

Daten sichern, wiederherstellen und exportieren

In diesem Abschnitt wird gezeigt, wie Sie Daten aus einer RDS Amazon-DB-Instance oder einem Multi-AZ-DB-Cluster sichern, wiederherstellen und exportieren.

Informationen zur Sicherung Ihrer Amazon RDS DB-Instance oder Ihres Multi-AZ-DB-Clusters finden Sie in den folgenden Themen.

- Einführung in Backups
- Verwaltung automatisierter Backups
- Verwaltung manueller Backups

Informationen zur Wiederherstellung Ihrer Amazon RDS DB-Instance oder Ihres Multi-AZ-DB-Clusters finden Sie unterWiederherstellung auf einer DB-Instance.

Informationen zum Kopieren, Teilen oder Exportieren von DB-Snapshots finden Sie in den folgenden Themen.

- Kopieren eines DB-Snapshots für Amazon RDS
- Einen DB-Snapshot für Amazon teilen RDS
- Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS

Informationen zum Anzeigen automatisierter Backups, die von verwaltet werden AWS Backup, finden Sie unterWird AWS Backup zur Verwaltung automatisierter Backups für Amazon verwendet RDS.

Einführung in Backups

Amazon RDS erstellt und speichert automatische Backups Ihrer DB-Instance oder Ihres Multi-AZ-DB-Clusters während des Backup-Fensters Ihrer DB-Instance. RDSerstellt einen Speichervolume-Snapshot Ihrer DB-Instance und sichert dabei die gesamte DB-Instance und nicht nur einzelne Datenbanken. RDSspeichert die automatisierten Backups Ihrer DB-Instance gemäß dem von Ihnen angegebenen Aufbewahrungszeitraum für Backups. Bei Bedarf können Sie Ihre DB-Instance zu einem beliebigen Zeitpunkt während der Aufbewahrungsdauer des Backups wiederherstellen.

Automatische Backups erfolgen nach diesen Regeln:

- Ihre DB-Instance muss den Status `available` aufweisen, damit ein automatisiertes Backup durchgeführt werden kann. Automatisierte Backups erfolgen nicht, während sich Ihre DB-Instance in einem anderen Status als `available` befindet (beispielsweise `storage_full`).
- Automatisierte Backups erfolgen nicht, während eine DB-Snapshot-Kopie für dieselbe Datenbank in derselben AWS-Region ausgeführt wird.

Sie können Ihre DB-Instance auch sichern, indem Sie manuell einen DB-Snapshot erstellen. Weitere Informationen zum manuellen Erstellen eines DB-Snapshots finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Der erste Snapshot einer DB-Instance enthält die Daten der vollständigen Datenbank. Bei den nachfolgenden Snapshots derselben Datenbank handelt es sich um inkrementelle Snapshots, d. h. es werden nur die Daten gespeichert, die sich seit der letzten Snapshot-Speicherung geändert haben.

Sie können automatische und manuelle DB-Snapshots kopieren und manuelle DB-Snapshots freigeben. Weitere Informationen zum Kopieren eines DB-Snapshots finden Sie unter Kopieren eines DB-Snapshots für Amazon RDS. Weitere Informationen zum Freigeben eines DB-Snapshots finden Sie unter Einen DB-Snapshot für Amazon teilen RDS.

Sicherungsspeicher

Ihr RDS Amazon-Backup-Speicher für jede Region AWS-Region besteht aus den automatisierten Backups und manuellen DB-Snapshots für diese Region. Der gesamte Speicherplatz für das Backup entspricht der Summe des Speichers für alle Backups in dieser Region. Durch das Verschieben eines DB-Snapshots in eine andere Region wird der Sicherungsspeicher in dieser Region vergrößert. Backups werden in Amazon S3 gespeichert.

Weitere Informationen zu den Kosten für Backup-Speicher finden Sie unter RDSAmazon-Preise.

Falls Sie automatisierte Backups beim Löschen einer DB-Instance beibehalten, bleiben die Backups während des gesamten Aufbewahrungszeitraums gespeichert. Falls Sie Retain automated backups (Automatische Backups aufbewahren) beim Löschen einer DB-Instance nicht wählen, werden die automatischen Backups mit der DB-Instance gelöscht. Nach dem Löschen können die automatischen Backups nicht wiederhergestellt werden. Wenn Sie Amazon einen endgültigen DB-Snapshot RDS erstellen lassen möchten, bevor Ihre DB-Instance gelöscht wird, können Sie diesen zur Wiederherstellung Ihrer DB-Instance verwenden. Optional können Sie auch einen zuvor erstellten manuellen Snapshot verwenden. Manuelle Snapshots werden nicht gelöscht. Pro Region sind bis zu 100 manuelle Snapshots zulässig.

Verwaltung automatisierter Backups

In diesem Abschnitt wird gezeigt, wie Sie automatisierte Backups für DB-Instances und Multi-AZ-DB-Cluster verwalten.

Themen

- Backup-Fenster
- Aufbewahrungszeitraum für Backups
- Aktivieren von automatisierten Backups
- Aufbewahren automatisierter Backups
- Löschen aufbewahrter automatisierter Backups
- Automatisierte Backups mit nicht unterstützten My SQL Storage Engines
- Automatisierte Backups mit nicht unterstützten MariaDB-Speicher-Engines
- Automatisierte Backups auf ein anderes replizieren AWS-Region

Backup-Fenster

Automatisierte Backups werden täglich während des bevorzugten Zeitfensters für das Backup durchgeführt. Wenn das Backup mehr Zeit als im Zeitfenster angegeben benötigt, wird es nach Ablauf des Zeitfensters zu Ende geführt. Das Zeitfenster für das Backup darf sich nicht mit dem wöchentlichen Wartungsfenster für die DB-Instance oder den Multi-AZ-DB-Cluster überschneiden.

Während des Zeitfensters für das automatisierte Backup können I/O-Speichervorgänge kurzzeitig ausgesetzt werden, bis der Sicherungsprozess gestartet wird (normalerweise nur wenige Sekunden). Bei Multi-AZ-Bereitstellungen können für einen Zeitraum von wenigen Minuten längere Latenzzeiten während eines Backups auftreten. Für MariaDB, MySQL, Oracle und Postgre SQL wird die I/O-Aktivität auf Ihrem Primärserver während des Backups für Multi-AZ-Bereitstellungen nicht unterbrochen, da das Backup aus dem Standby übernommen wird. Bei SQL Servern wird die I/O-Aktivität während des Backups sowohl für Single-AZ- als auch für Multi-AZ-Bereitstellungen kurzzeitig unterbrochen, da das Backup von der Primärseite übernommen wird. Bei Db2 wird die I/O-Aktivität auch während des Backups kurzzeitig unterbrochen, obwohl das Backup aus dem Standby-Modus stammt.

Automatisierte Backups können gelegentlich übersprungen werden, wenn die DB-Instance oder der Cluster zu dem Zeitpunkt, zu dem ein Backup gestartet werden soll, einen hohen Workload aufweist.

Wenn eine Sicherung übersprungen wird, können Sie trotzdem eine point-in-time-recovery (PITR) ausführen, und im nächsten Backup-Fenster wird trotzdem versucht, eine Sicherung durchzuführen. Weitere Informationen zu finden Sie PITR unterWiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Wenn Sie bei der Erstellung der DB-Instance oder des Multi-AZ-DB-Clusters kein bevorzugtes Backup-Fenster angeben, RDS weist Amazon ein standardmäßiges 30-minütiges Backup-Fenster zu. Dieses Fenster wird nach dem Zufallsprinzip aus einem Zeitblock von jeweils 8 Stunden ausgewählt. AWS-Region In der folgenden Tabelle sind die Zeitblöcke für jeden AWS-Region Block aufgeführt, denen die Standard-Backup-Fenster zugewiesen werden.

Name der Region	Region	Zeitblock
USA Ost (Nord-Virginia)	us-east-1	03:00 — 11:00 UTC
USA Ost (Ohio)	us-east-2	03:00 — 11:00 UTC
USA West (Nordkalifornien)	us-west-1	06:00 — 14:00 UTC
USA West (Oregon)	us-west-2	06:00 — 14:00 UTC
Afrika (Kapstadt)	af-south-1	03:00 — 11:00 UTC
Asien-Pazifik (Hongkong)	ap-east-1	06:00 — 14:00 UTC
Asien-Pazifik (Hyderabad)	ap-south-2	06:30 — 14:30 UTC
Asien-Pazifik (Jakarta)	ap-southeast-3	08:00 — 16:00 UTC
Asien-Pazifik (Malaysia)	ap-southeast-5	09:00 — 17:00 Uhr UTC
Asien-Pazifik (Melbourne)	ap-southeast-4	11:00 — 19:00 UTC

Name der Region	Region	Zeitblock
Asien-Pazifik (Mumbai)	ap-south-1	16:30 — 00:30 UTC
Asien-Pazifik (Osaka)	ap-northeast-3	00:00 — 08:00 UTC
Asien-Pazifik (Seoul)	ap-northeast-2	13:00 — 21:00 UTC
Asien-Pazifik (Singapur)	ap-southeast-1	14:00 — 22:00 UTC
Asien-Pazifik (Sydney)	ap-southeast-2	12:00 — 20:00 Uhr UTC
Asien-Pazifik (Tokio)	ap-northeast-1	13:00 — 21:00 UTC
Kanada (Zentral)	ca-central-1	03:00 — 11:00 UTC
Kanada West (Calgary)	ca-west-1	18:00 — 02:00 UTC
China (Beijing)	cn-north-1	06:00 — 14:00 UTC
China (Ningxia)	cn-northwest-1	06:00 — 14:00 UTC
Europa (Frankfurt)	eu-central-1	20:00 — 04:00 UTC
Europa (Irland)	eu-west-1	22:00 — 06:00 UTC
Europa (London)	eu-west-2	22:00 — 06:00 UTC
Europa (Mailand)	eu-south-1	02:00 — 10:00 UTC
Europa (Paris)	eu-west-3	07:29 — 14:29 UTC
Europa (Spanien)	eu-south-2	02:00 — 10:00 UTC
Europa (Stockholm)	eu-north-1	23:00 — 07:00 UTC
Europa (Zürich)	eu-central-2	02:00 — 10:00 UTC

Name der Region	Region	Zeitblock
Israel (Tel Aviv)	il-central-1	03:00 — 11:00 UTC
Naher Osten (Bahrain)	me-south-1	06:00 — 14:00 UTC
Naher Osten () UAE	me-central-1	05:00 — 13:00 UTC
Südamerika (São Paulo)	sa-east-1	23:00 — 07:00 UTC
AWS GovCloud (US-Ost)	us-gov-east-1	17:00 — 01:00 UTC
AWS GovCloud (US-West)	us-gov-west-1	06:00 — 14:00 Uhr UTC

Aufbewahrungszeitraum für Backups

Sie können den Aufbewahrungszeitraum für Backups einstellen, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen. Wenn Sie eine DB-Instance mit Amazon RDS API oder dem erstellen AWS CLI und die Aufbewahrungsdauer für Backups nicht festlegen, beträgt die Standard-Aufbewahrungsdauer für Backups einen Tag. Wenn Sie eine DB-Instance mithilfe der Konsole erstellen, beträgt die Standard-Aufbewahrungsfrist für Backups sieben Tage.

Sie können den Aufbewahrungszeitraum für Backups ändern, nachdem Sie eine DB-Instance oder einen Cluster erstellt haben. Der Aufbewahrungszeitraum für Backups einer DB-Instance kann auf einen Wert zwischen 0 und 35 Tagen festgelegt werden. Wenn Sie den Wert des Aufbewahrungszeitraums für Backups auf 0 setzen, deaktivieren Sie automatische Backups. Für einen Multi-AZ-DB-Cluster können Sie die Aufbewahrungsdauer für Backups auf 1 bis 35 Tage festlegen. Das Limit für manuelle Snapshots (100 pro Region) gilt für automatisierte Backups nicht.

> **Important**
>
> Ein Ausfall tritt auf, wenn Sie die Aufbewahrungsdauer für Backups einer DB-Instance von 0 auf einen Wert ungleich Null oder von einem Wert ungleich Null auf 0 ändern.

RDSbeinhaltet nicht die Zeit, die in dem `stopped` Zustand verbracht wurde, in dem die Aufbewahrungsdauer für Backups berechnet wurde. Automatisierte Backups werden nicht erstellt, während eine DB-Instance oder ein Cluster gestoppt ist. Backups können länger als die Aufbewahrungsfrist für Backups beibehalten werden, wenn eine DB-Instance gestoppt wurde.

Aktivieren von automatisierten Backups

Wenn das automatisierte Backup für Ihre DB-Instance nicht aktiviert ist, können Sie es jederzeit aktivieren. Sie aktivieren die automatisierte Backup, indem Sie den Aufbewahrungszeitraum für Backups auf einen Wert größer als null festlegen. Wenn automatisierte Backups aktiviert sind, wird Ihre DB-Instance offline gestellt und es wird sofort ein Backup erstellt.

> **Note**
>
> Wenn Sie Ihre Backups in verwalten AWS Backup, können Sie keine automatisierten Backups aktivieren. Weitere Informationen finden Sie unter Wird AWS Backup zur Verwaltung automatisierter Backups für Amazon verwendet RDS.

Konsole

So aktivieren Sie automatisierte Backups direkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die DB-Instance oder den Multi-AZ-DB-Cluster, die/den Sie ändern möchten.
3. Wählen Sie Ändern aus.
4. Wählen Sie für die Aufbewahrungsdauer von Backup einen positiven Wert ungleich Null, z. B. drei Tage.
5. Klicken Sie auf Continue.
6. Wählen Sie Apply immediately (Sofort anwenden) aus.
7. Wählen Sie DB-Instance ändern oder Cluster ändern aus, um Ihre Änderungen zu speichern und automatisierte Backups zu aktivieren.

AWS CLI

Verwenden Sie den modify-db-clusterBefehl AWS CLI modify-db-instanceoder, um automatische Backups zu aktivieren.

Verwenden Sie die folgenden Parameter:

- `--db-instance-identifier` (oder `--db-cluster-identifier` für einen Multi-AZ-DB-Cluster)
- `--backup-retention-period`
- `--apply-immediately` oder `--no-apply-immediately`

In diesem Beispiel aktivieren wir automatische Backups, indem wir den Aufbewahrungszeitraum für Backups auf drei Tage festlegen. Die Änderungen werden sofort übernommen.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my_db_instance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my_db_instance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

RDS-API

Um automatische Backups zu aktivieren, verwenden Sie den DBCluster RDS-API-Vorgang Modify DBInstance oder Modify mit den folgenden erforderlichen Parametern:

- `DBInstanceIdentifier` oder `DBClusterIdentifier`
- `BackupRetentionPeriod`

Anzeigen automatisierter Sicherungen

Um Ihre beibehaltenen automatisierten Backups anzuzeigen, wählen Sie zuerst im Navigationsbereich Automatisierte Backups aus und dann Beibehaltung aus. Um einzelne Snapshots anzuzeigen, die mit einem beibehaltenen automatisierten Backup verknüpft sind, wählen Sie im Navigationsbereich Snapshot aus. Alternativ können Sie einzelne Snapshots beschreiben, die einem aufbewahrten automatischen Backup zugeordnet sind. Von da können Sie eine DB-Instance direkt aus einem dieser Snapshots wiederherstellen.

Automatisierte Snapshot-Namen folgen dem Muster `rds:<database-name>-yyyy-mm-dd-hh-mm` und `yyyy-mm-dd-hh-mm` stellen das Datum und die Uhrzeit der Erstellung des Snapshots dar.

Verwenden Sie einen der folgenden Befehle AWS CLI, um die automatisierten Backups für Ihre vorhandenen DB-Instances mithilfe von zu beschreiben:

```
aws rds describe-db-instance-automated-backups --db-instance-
identifier DBInstanceIdentifier
```

or

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Zum Beschreiben Ihrer automatischen Backups mit der RDS-API rufen Sie die `DescribeDBInstanceAutomatedBackups`-Aktion mit einem der folgenden Parameter auf:

- `DBInstanceIdentifier`
- `DbiResourceId`

Aufbewahren automatisierter Backups

> **Note**
>
> Sie können nur automatisierte Backups von DB-Instances, nicht von Multi-AZ-DB-Clustern aufbewahren.

Beim Löschen einer DB-Instance können Sie festlegen, dass automatisierte Backups beibehalten werden. Die automatisierten Backups bleiben für den Aufbewahrungszeitraum erhalten, der zu dem Zeitpunkt für die DB-Instance festgelegt war, als Sie sie gelöscht haben.

Aufbewahrte automatische Backups enthalten System-Snapshots und Transaktionsprotokolle aus einer DB-Instance. Sie enthalten auch Ihre DB-Instance-Eigenschaften wie zugeteilten Speicher und DB-Instance-Klasse, die sie zu einer aktiven Instance wiederherstellen sollen.

Bei Aufbewahrung automatisierter Backups und manueller Snapshots fallen Gebühren an, bis sie gelöscht werden. Weitere Informationen finden Sie unter Aufbewahrungskosten.

Sie können automatisierte Backups für RDS Instances speichern, auf denen die Db2-, MariaDB-, My-, Postgre- SQLSQL, Oracle- und Microsoft Server-Engines ausgeführt werden. SQL

Mithilfe von, und können Sie gespeicherte automatische Backups wiederherstellen oder entfernen. AWS Management Console RDS API AWS CLI

Themen

- retention period (Aufbewahrungszeitraum)
- Anzeigen von beibehaltenen Backups
- Wiederherstellung
- Aufbewahrungskosten
- Einschränkungen

retention period (Aufbewahrungszeitraum)

Die System-Snapshots und Transaktionsprotokolle in einem aufbewahrten automatischen Backup laufen auf dieselbe Art wie für die Quell-DB-Instance ab. Da keine neuen Snapshots vorhanden sind und keine Protokolle für diese Instance erstellt wurden, laufen die automatischen Backups letztendlich vollständig ab. Sie sind effektiv so lange aktiv, wie es ihr letzter System-Snapshot gewesen wäre, je nach den Einstellungen für den Aufbewahrungszeitraum in der Quell-Instance zum Zeitpunkt des Löschens. Aufbewahrte automatische Backups werden vom System entfernt, nachdem ihr letzter System-Snapshot abläuft.

Sie können einen aufbewahrten automatisierten Backup auf dieselbe Art entfernen, wie Sie eine DB-Instance löschen. Sie können gespeicherte automatische Backups mithilfe der Konsole oder des RDS API Vorgangs entfernen`DeleteDBInstanceAutomatedBackup`.

Finale Snapshots sind unabhängig von aufbewahrten automatischen Backups. Wir empfehlen dringend das Erstellen eines finalen Snapshots, selbst wenn Sie automatisierte Backups aufbewahren, da aufbewahrte automatisierte Backups irgendwann ablaufen. Der finale Snapshot läuft nicht ab.

Anzeigen von beibehaltenen Backups

Um Ihre beibehaltenen automatisierten Backups anzuzeigen, wählen Sie zuerst im Navigationsbereich Automatisierte Backups aus und dann Beibehaltung aus. Um einzelne Snapshots anzuzeigen, die mit einem beibehaltenen automatisierten Backup verknüpft sind, wählen Sie im Navigationsbereich Snapshot aus. Alternativ können Sie einzelne Snapshots beschreiben, die einem aufbewahrten automatischen Backup zugeordnet sind. Von da können Sie eine DB-Instance direkt aus einem dieser Snapshots wiederherstellen.

Verwenden Sie den folgenden Befehl AWS CLI, um Ihre gespeicherten automatisierten Backups mithilfe von zu beschreiben:

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Um Ihre gespeicherten automatisierten Backups mithilfe von zu beschreiben RDSAPI, rufen Sie die `DescribeDBInstanceAutomatedBackups`Aktion mit dem `DbiResourceId` Parameter auf.

Wiederherstellung

Informationen zum Wiederherstellen von DB-Instances aus automatisierten Backups finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Aufbewahrungskosten

Die Kosten eines aufbewahrten automatischen Backups setzen sich aus den Kosten des Gesamtspeichers der System-Snapshots zusammen, die diesem zugeordnet sind. Für Transaktionsprotokolle oder Instance-Metadaten wird keine zusätzliche Gebühr erhoben. Alle anderen Preisregeln für Backups gelten für wiederherstellbare Instances.

Beispiel: Angenommen, Ihr gesamter zugeordneter Speicher zum Ausführen von Instances beträgt 100 GB. Angenommen, Sie haben einem aufbewahrten automatischen Backup zudem 50 GB manuelle Snapshots sowie 75 GB System-Snapshots zugewiesen. In diesem Fall werden Ihnen nur die zusätzlichen 25 GB Sicherungsspeicher in Rechnung gestellt: (50 GB + 75 GB) – 100 GB = 25 GB.

Einschränkungen

Die folgenden Einschränkungen gelten für aufbewahrte automatische Backups:

- Die maximale Anzahl von gespeicherten automatisierten Backups in einer AWS Region ist 40. Sie ist nicht im DB-Instances-Kontingent enthalten. Sie können 40 laufende DB-Instances und weitere 40 aufbewahrte automatisierte Backups gleichzeitig haben.
- Aufbewahrte automatische Backups enthalten keine Informationen über Parameter oder Optionsgruppen.
- Sie können eine gelöschte Instance zu jedem Zeitpunkt wiederherstellen, der zum Zeitpunkt des Löschens innerhalb der Aufbewahrungsdauer liegt.
- Sie können ein aufbewahrtes automatisiertes Backup nicht ändern. Das liegt daran, dass sie aus Systemsicherungen, Transaktionsprotokollen und den DB-Instance-Eigenschaften besteht, die zum Zeitpunkt des Löschens der Quell-Instance existierten.

Löschen aufbewahrter automatisierter Backups

Sie können aufbewahrte automatisierte Backups löschen, wenn sie nicht mehr benötigt werden.

Konsole

So löschen Sie eine aufbewahrte automatisierte Backup:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
3. Wählen Sie auf der Registerkarte Retained (Beibehalten) die beibehaltene automatisierte Backup, die Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Geben Sie auf der Bestätigungsseite **`delete me`** und wählen Sie Löschen aus.

AWS CLI

Sie können ein gespeichertes automatisiertes Backup löschen, indem Sie den AWS CLI Befehl delete-db-instance-automated-backup mit der folgenden Option verwenden:

- `--dbi-resource-id` – Die Ressourcenkennung für die Quell-DB-Instance.

Sie können die Ressourcen-ID für die Quell-DB-Instance einer gespeicherten automatisierten Sicherung ermitteln, indem Sie den AWS CLI Befehl describe-db-instance-automated-backups ausführen.

Example

Im folgenden Beispiel wird das aufbewahrte automatisierte Backup mit der Quell-DB-Instance-Ressourcenkennung `db-123ABCEXAMPLE` gelöscht.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id db-123ABCEXAMPLE
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id db-123ABCEXAMPLE
```

RDS API

Sie können ein gespeichertes automatisiertes Backup löschen, indem Sie die RDS API Amazon-Operation D eleteDBInstance AutomatedBackup mit dem folgenden Parameter verwenden:

- `DbiResourceId` – Die Ressourcenkennung für die Quell-DB-Instance.

  Sie können die Ressourcen-ID für die Quell-DB-Instance eines gespeicherten automatisierten Backups mithilfe der RDS API Amazon-Operation D ermitteln escribeDBInstance AutomatedBackups.

Deaktivieren von automatisierten Backups

Es kann in bestimmten Situationen vorteilhaft sein, das automatisierte Backup vorübergehend zu deaktivieren, z. B. wenn große Datenmengen geladen werden.

> **Important**
>
> Wir raten dringend davon ab, automatische Backups zu deaktivieren, da dadurch die Wiederherstellung deaktiviert point-in-time wird. Durch die Deaktivierung der automatisierten

Backups für eine DB-Instance oder einen Multi-AZ-DB-Cluster werden alle vorhandenen automatisierten Backups für die Datenbank gelöscht. Wenn Sie automatisierte Backups deaktivieren und erneut aktivieren, können Wiederherstellungen nur ab dem Zeitpunkt der Reaktivierung des automatischen Backups durchgeführt werden.

Konsole

So deaktivieren Sie das automatisierte Backup direkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die DB-Instance oder den Multi-AZ-DB-Cluster, die/den Sie ändern möchten.
3. Wählen Sie Ändern aus.
4. Wählen Sie unter Aufbewahrungszeitraum für Backups den Wert 0 days (0 Tage).
5. Klicken Sie auf Continue.
6. Wählen Sie Apply immediately (Sofort anwenden) aus.
7. Wählen Sie DB-Instance ändern oder Cluster ändern aus, um Ihre Änderungen zu speichern und automatisierte Backups zu deaktivieren.

AWS CLI

Um automatische Backups sofort zu deaktivieren, verwenden Sie den modify-db-clusterBefehl modify-db-instanceoder und setzen Sie den Aufbewahrungszeitraum für Backups auf 0 mit`--apply-immediately`.

Example

Im folgenden Beispiel werden automatisierte Backups für einen Multi-AZ-DB-Cluster direkt deaktiviert.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --backup-retention-period 0 \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --backup-retention-period 0 ^
    --apply-immediately
```

Sie können herausfinden, ob die Änderungen wirksam sind, indem Sie `describe-db-instances` für die DB-Instance (oder `describe-db-clusters` für einen Multi-AZ-DB-Cluster) aufrufen, bis als Aufbewahrungszeitraum für Backups der Wert 0 und als Status von `mydbcluster` „available" angezeigt wird.

```
aws rds describe-db-clusters --db-cluster-identifier mydcluster
```

RDS API

Um automatische Backups sofort zu deaktivieren, rufen Sie die odifyDBCluster Operation M odifyDBInstance oder M mit den folgenden Parametern auf:

- `DBInstanceIdentifier = mydbinstance` (oder `DBClusterIdentifier = mydbcluster`)
- `BackupRetentionPeriod = 0`

Example

```
https://rds.amazonaws.com/
    ?Action=ModifyDBInstance
    &DBInstanceIdentifier=mydbinstance
    &BackupRetentionPeriod=0
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2009-10-14T17%3A48%3A21.746Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Automatisierte Backups mit nicht unterstützten My SQL Storage Engines

Für die My SQL DB-Engine werden automatisierte Backups nur für die InnoDB-Speicher-Engine unterstützt. Die Verwendung dieser Funktionen mit anderen My SQL Storage-Engines, einschließlich

MyISAM, kann bei der Wiederherstellung aus Backups zu unzuverlässigem Verhalten führen. Insbesondere weil Speicher-Engines wie My ISAM keine zuverlässige Wiederherstellung nach einem Absturz unterstützen, können Ihre Tabellen im Falle eines Absturzes beschädigt werden. Aus diesem Grund empfehlen wir die Verwendung der InnoDB-Speicher-Engine.

- Um bestehende Meine ISAM Tabellen in InnoDB-Tabellen zu konvertieren, können Sie zum Beispiel den `ALTER TABLE` folgenden Befehl verwenden: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Wenn Sie My verwenden, können Sie versuchenISAM, Tabellen, die nach einem Absturz beschädigt wurden, mithilfe des `REPAIR` Befehls manuell zu reparieren. Weitere Informationen finden Sie in der REPAIRTABLEErklärung in der SQL Dokumentation zu My. Wie in der SQL Dokumentation „Meine“ erwähnt, besteht jedoch eine gute Chance, dass Sie nicht alle Ihre Daten wiederherstellen können.
- Wenn Sie vor der Wiederherstellung einen Snapshot Ihrer Meine ISAM Tabellen erstellen möchten, gehen Sie wie folgt vor:

  1. Stoppen Sie alle Aktivitäten in Ihren Meine ISAM Tabellen (d. h. schließen Sie alle Sitzungen).

     Sie können alle Sitzungen beenden, indem Sie den Befehl mysql.rds_kill für jeden Prozess ausführen, der von dem Befehl `SHOW FULL PROCESSLIST` ausgegeben wird.

  2. Sperren und leeren Sie jeden Ihrer Meine ISAM Tische. Mit dem folgenden Befehl werden z. B. die beiden Tabellen `myisam_table1` und `myisam_table2` gesperrt und ausgelagert:

     ```
     mysql> FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;
     ```

  3. Erstellen Sie einen Snapshot Ihrer DB-Instance oder Ihres Multi-AZ-DB-Clusters. Wenn der Snapshot abgeschlossen ist, lösen Sie die Sperren und setzen Sie die Aktivität in Meine ISAM Tabellen fort. Sie können die Sperren für Ihre Tabellen mit dem folgenden Befehl aufheben:

     ```
     mysql> UNLOCK TABLES;
     ```

  Durch diese Schritte wird My ISAM gezwungen, die im Arbeitsspeicher gespeicherten Daten auf die Festplatte zu übertragen, sodass bei der Wiederherstellung aus einem DB-Snapshot ein sauberer Start gewährleistet ist. Weitere Informationen zum Erstellen eines DB-Snapshots finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Automatisierte Backups mit nicht unterstützten MariaDB-Speicher-Engines

Für die MariaDB-Engine werden automatische Backups nur mit der InnoDB-Speicher-Engine unterstützt. Die Verwendung dieser Funktionen mit anderen MariaDB-Speicher-Engines einschließlich Aria kann zu unzuverlässigem Verhalten bei der Wiederherstellung anhand von Backups führen. Obwohl Aria eine absturzsichere Alternative zu My istISAM, können Ihre Tabellen im Falle eines Absturzes dennoch beschädigt werden. Aus diesem Grund empfehlen wir die Verwendung der InnoDB-Speicher-Engine.

- Wenn Sie vorhandene Aria-Tabellen in InnoDB-Tabellen konvertieren möchten, können Sie den Befehl `ALTER TABLE` verwenden. Beispiel: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Wenn Sie Aria verwenden, können Sie versuchen, die bei einem Absturz beschädigten Tabellen mit dem Befehl `REPAIR TABLE` manuell zu reparieren. Weitere Informationen finden Sie unter http://mariadb.com/kb/en/mariadb/repair-table/.
- Gehen Sie wie folgt vor, wenn Sie vor der Wiederherstellung einen Snapshot Ihrer Aria-Tabellen erstellen möchten:
  1. Stoppen Sie alle Aktivitäten für Ihre Aria-Tabellen (d. h. beenden Sie alle Sitzungen).
  2. Sperren Sie alle Aria-Tabellen und lagern Sie sie aus.
  3. Erstellen Sie einen Snapshot Ihrer DB-Instance oder Ihres Multi-AZ-DB-Clusters. Wenn der Snapshot erstellt wurde, heben Sie die Sperren für die Aria-Tabellen auf und setzen Sie die Aktivitäten wieder fort. Mit diesen Schritten wird eine Auslagerung der Daten im Arbeitsspeicher auf den Datenträger von Aria erzwungen, sodass bei der Wiederherstellung von einem DB-Snapshot ein sauberer Start sichergestellt ist.

Automatisierte Backups auf ein anderes replizieren AWS-Region

Für zusätzliche Notfallwiederherstellungsfunktionen können Sie Ihre Amazon RDS-Datenbank-Instance so konfigurieren, dass Snapshots und Transaktionsprotokolle an ein Ziel AWS-Region Ihrer Wahl repliziert werden. Wenn die Sicherungsreplikation für eine DB-Instance konfiguriert ist, initiiert RDS eine regionsübergreifende Kopie aller Snapshots und Transaktionsprotokolle, sobald sie auf der DB-Instance bereit sind.

Für die Datenübertragung fallen Gebühren für die DB-Snapshot-Kopie an. Nachdem der DB-Snapshot kopiert wurde, gelten für den Speicher in der Zielregion Standardgebühren. Weitere Einzelheiten finden Sie unter RDS-Preise.

Ein Beispiel für die Verwendung der Backup-Replikation finden Sie im AWS Online-Techtalk Managed Disaster Recovery with Amazon RDS for Oracle Cross-Region Automated Backups.

> **Note**
>
> Die automatische Backup-Replikation wird für Multi-AZ-DB-Cluster nicht unterstützt. Sie wird jedoch für Multi-AZ-DB-Instance-Bereitstellungen unterstützt. Weitere Informationen zu Einschränkungen für automatisierte Backups finden Sie unter. Kontingente und Beschränkungen für Amazon RDS

Informationen zur Konfiguration und Verwaltung automatisierter Backups für Amazon RDS finden Sie in den folgenden Themen.

Themen

- Aktivierung regionsübergreifender automatisierter Backups für Amazon RDS
- Informationen zu replizierten Backups für Amazon finden RDS
- Wiederherstellung zu einem bestimmten Zeitpunkt aus einem replizierten Backup für Amazon RDS
- Stoppen der automatisierten Backup-Replikation für Amazon RDS
- Löschen replizierter Backups für Amazon RDS

Unterstützung für Multi-AZ-Bereitstellungen

Die regionsübergreifende automatische Backreplikation wird für Multi-AZ-DB-Instance-Bereitstellungen für die folgenden Engines unterstützt:

- RDS für Db2
- RDS for MariaDB
- RDS for MySQL
- RDS für Oracle
- RDS for PostgreSQL
- RDS für SQL Server

Die regionsübergreifende automatische Backup-Replikation wird für Multi-AZ-DB-Cluster nicht unterstützt.

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Versions- und Regionsverfügbarkeit mit regionsübergreifenden automatisierten Backups finden Sie unter Unterstützte Regionen und DB-Engines für regionsübergreifende automatisierte Backups in Amazon RDS.

Quell- und Zielunterstützung AWS-Region

Die Backup-Replikation wird zwischen den folgenden unterstützt AWS-Regionen.

Quellregion	Verfügbare Zielregionen
Afrika (Kapstadt)	Europa (Frankfurt), Europa (Irland), Europa (London)
Asien-Pazifik (Hongkong)	Asien-Pazifik (Singapur), Asien-Pazifik (Tokio)
Asien-Pazifik (Hyderabad)	Asien-Pazifik (Mumbai)
Asien-Pazifik (Melbourne)	Asien-Pazifik (Sydney)
Asien-Pazifik (Malaysia)	Asien-Pazifik (Singapur)
Asien-Pazifik (Mumbai)	Asien-Pazifik (Hyderabad), Asien-Pazifik (Singapur) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon)

Quellregion	Verfügbare Zielregionen
Asia Pacific (Osaka)	Asien-Pazifik (Tokio)
Asia Pacific (Seoul)	Asien-Pazifik (Singapur), Asien-Pazifik (Tokio) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon)
Asien-Pazifik (Singapur)	Asien-Pazifik (Hongkong), Asien-Pazifik (Malaysia), Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Sydney), Asien-Pazifik (Tokio) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon)
Asien-Pazifik (Sydney)	Asien-Pazifik (Melbourne), Asien-Pazifik (Singapur) USA Ost (Nord-Virginia), USA West (Nordkalifornien), USA West (Oregon)
Asien-Pazifik (Tokio)	Asien-Pazifik (Hongkong), Asien-Pazifik (Osaka), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon)
Kanada (Zentral)	Kanada West (Calgary) Europa (Irland) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Nordkalifornien), USA West (Oregon)
Kanada West (Calgary)	Kanada (Zentral)
China (Peking)	China (Ningxia)
China (Ningxia)	China (Peking)

Quellregion	Verfügbare Zielregionen
Europa (Frankfurt)	Afrika (Kapstadt) Europa (Irland), Europa (London), Europa (Paris), Europa (Stockholm), Europa (Zürich) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon)
Europa (Irland)	Afrika (Kapstadt) Kanada (Zentral) Europa (Frankfurt), Europa (London), Europa (Paris), Europa (Stockholm), Europa (Zürich) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Nordkalifornien), USA West (Oregon)
Europa (London)	Afrika (Kapstadt) Europa (Frankfurt), Europa (Irland), Europa (Paris), Europa (Stockholm) USA Ost (Nord-Virginia)
Europe (Paris)	Europa (Frankfurt), Europa (Irland), Europa (London), Europa (Stockholm) USA Ost (Nord-Virginia)
Europe (Stockholm)	Europa (Frankfurt), Europa (Irland), Europa (London), Europa (Paris) USA Ost (Nord-Virginia)
Europa (Zürich)	Europa (Frankfurt), Europa (Irland)
Südamerika (São Paulo)	USA Ost (Nord-Virginia), USA Ost (Ohio)
AWS GovCloud (US-Ost)	AWS GovCloud (US-West)

Quellregion	Verfügbare Zielregionen
AWS GovCloud (US-West)	AWS GovCloud (US-Ost)
USA Ost (Nord-Virginia)	Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik Canada (Central) Europa (Frankfurt), Europa (Irland), Europa (London), Europa (Paris), Europa (Stockholm) Südamerika (São Paulo) USA Ost (Ohio), USA West (Nordkalifornien), USA West (Oregon)
US East (Ohio)	Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Tokio) Canada (Central) Europa (Frankfurt), Europa (Irland) Südamerika (São Paulo) USA Ost (Nord-Virginia), USA West (Nordkalifornien), USA West (Oregon)
USA West (Nordkalifornien)	Asien-Pazifik (Sydney) Canada (Central) Europa (Irland) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Oregon)

Quellregion	Verfügbare Zielregionen
USA West (Oregon)	Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik Canada (Central) Europa (Frankfurt), Europa (Irland) USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Nordkali fornien)

Sie können den `describe-source-regions` AWS CLI Befehl auch verwenden, um herauszufinden, welche sich gegenseitig replizieren AWS-Regionen können. Weitere Informationen finden Sie unter Informationen zu replizierten Backups für Amazon finden RDS.

Einschränkungen

Im Folgenden sind Einschränkungen für regionsübergreifende automatische Backups für Amazon RDS aufgeführt.

- Die automatische Backup-Replikation wird für Multi-AZ-DB-Cluster nicht unterstützt.
- Standardmäßig können Sie bis zu 20 regionsübergreifende automatisierte Backups pro Person haben. AWS-Konto

Aktivierung regionsübergreifender automatisierter Backups für Amazon RDS

Sie können die Backup-Replikation auf neuen oder vorhandenen DB-Instances mithilfe der RDS Amazon-Konsole aktivieren. Sie können auch den `start-db-instance-automated-backups-replication` AWS CLI Befehl oder die `StartDBInstanceAutomatedBackupsReplication` RDS API Operation verwenden. Sie können AWS-Region für jedes Ziel bis zu 20 Backups replizieren. AWS-Konto

> **Note**
>
> Um automatisierte Backups replizieren zu können, sollten Sie diese unbedingt aktivieren. Weitere Informationen finden Sie unter Aktivieren von automatisierten Backups.

Konsole

Sie können die Sicherungs-Replikation für eine neue oder vorhandene DB-Instance aktivieren:

- Bei einer neuen DB-Instance aktivieren Sie diese, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Einstellungen für DB-Instances.
- Verwenden Sie für eine vorhandene DB-Instance das folgende Verfahren.

Aktivieren Sie die Sicherungs-Replikation für eine vorhandene DB-Instance wie folgt:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
3. Wählen Sie auf der Registerkarte Aktuelle Region die DB-Instance aus, für die Sie die Sicherungs-Replikation aktivieren möchten.
4. Wählen Sie für Aktionen die Option Regionsübergreifende Replikation verwalten aus.
5. Wählen Sie unter Backup replication (Backup-Replikation) die Option Enable replication to another AWS-Region (Replikation für eine andere Region aktivieren) aus.
6. Wählen Sie die Zielregion aus.
7. Wählen Sie den Aufbewahrungszeitraum für replizierte Backups aus.
8. Wenn Sie die Verschlüsselung auf der Quell-DB-Instance aktiviert haben, wählen Sie die Option AWS KMS keyfür die Verschlüsselung der Backups oder geben Sie einen Schlüssel ARN ein.
9. Wählen Sie Save (Speichern) aus.

In der Quellregion werden replizierte Backups auf der Registerkarte Aktuelle Region der Seite Automated backups (Automatisierte Backups) aufgeführt. In der Zielregion werden replizierte Backups auf der Registerkarte Replizierte Backups der Seite Automated backups (Automatisierte Backups) aufgeführt.

AWS CLI

Aktivieren Sie die Backup-Replikation mithilfe des `start-db-instance-automated-backups-replication` AWS CLI Befehls.

Im folgenden CLI Beispiel werden automatische Backups von einer DB-Instance in der Region USA West (Oregon) in die Region USA Ost (Nord-Virginia) repliziert. Außerdem werden die replizierten Backups mithilfe einer AWS KMS key in der Zielregion verschlüsselt.

Aktivieren Sie die Sicherungs-Replikation wie folgt:

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds start-db-instance-automated-backups-replication \
  --region us-east-1 \
  --source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" \
  --kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" \
  --backup-retention-period 7
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds start-db-instance-automated-backups-replication ^
  --region us-east-1 ^
  --source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" ^
  --kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" ^
  --backup-retention-period 7
  ```

  Die `--source-region` Option ist erforderlich, wenn Sie Backups zwischen den Regionen AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) verschlüsseln. Geben Sie für `--source-region` die AWS-Region der Quell-DB-Instance an.

  Wenn `--source-region` nicht angegeben ist, geben Sie unbedingt einen `--pre-signed-url`-Wert an. Eine vorsignierte Anfrage enthält URL eine mit URL Signature Version 4 signierte Anfrage für den `start-db-instance-automated-backups-replication` Befehl, der in der Quelle aufgerufen wird. AWS-RegionWeitere Informationen zu dieser *`pre-signed-url`* Option finden Sie unter start-db-instance-automated-backups-replication in der Befehlsreferenz.AWS CLI

RDS API

Aktivieren Sie die Backup-Replikation, indem Sie den `StartDBInstanceAutomatedBackupsReplication`RDSAPIVorgang mit den folgenden Parametern verwenden:

- `Region`(wenn Sie den API Vorgang nicht von der Zielregion aus aufrufen)
- `SourceDBInstanceArn`

- `BackupRetentionPeriod`
- `KmsKeyId` (optional)
- `PreSignedUrl` (wird benötigt, wenn Sie es verwenden `KmsKeyId`)

> **Note**
>
> Wenn Sie die Backups verschlüsseln, müssen Sie auch eine URL vorsignierte Datei hinzufügen. Weitere Informationen zu vorsignierten URLs Anfragen finden Sie unter Authentifizieren von Anfragen: Verwenden von Abfrageparametern (AWS Signature Version 4) in der Amazon Simple Storage Service API Reference und Signature Version 4-Signaturprozess in der AWS Allgemeinen Referenz.

Informationen zu replizierten Backups für Amazon finden RDS

Sie können die folgenden CLI Befehle verwenden, um Informationen zu replizierten Backups zu finden:

- `describe-source-regions`
- `describe-db-instances`
- `describe-db-instance-automated-backups`

Im folgenden `describe-source-regions` Beispiel wird die Quelle aufgeführt, AWS-Regionen aus der automatische Backups in die Zielregion USA West (Oregon) repliziert werden können.

Zeigen Sie Informationen über Quellregionen wie folgt an:

- Führen Sie den folgenden Befehl aus.

```
aws rds describe-source-regions --region us-west-2
```

Die Ausgabe zeigt, dass Backups von US East (N. Virginia), aber nicht von USA Ost (Ohio) oder USA West (Nordkalifornien), in repliziert werden können USA West (Oregon).

```
{
    "SourceRegions": [
```

```
        ...
        {
            "RegionName": "us-east-1",
            "Endpoint": "https://rds.us-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-east-2",
            "Endpoint": "https://rds.us-east-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
            "RegionName": "us-west-1",
            "Endpoint": "https://rds.us-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        }
    ]
}
```

Das folgende `describe-db-instances`-Beispiel zeigt die automatisierten Backups für eine DB-Instance.

Zeigen Sie die replizierten Backups für eine DB-Instance wie folgt an:

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds describe-db-instances \
  --db-instance-identifier mydatabase
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds describe-db-instances ^
  --db-instance-identifier mydatabase
  ```

Die Ausgabe enthält die replizierten Backups.

```
{
```

```
    "DBInstances": [
        {
            "StorageEncrypted": false,
            "Endpoint": {
                "HostedZoneId": "Z1PVIF0B656C1W",
                "Port": 1521,
            ...

            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
        }
    ]
}
```

Das folgende `describe-db-instance-automated-backups`-Beispiel zeigt die automatisierten Backups für eine DB-Instance.

Zeigen Sie automatisierte Backups für eine DB-Instance wie folgt an:

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-identifier mydatabase
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-identifier mydatabase
  ```

Die Ausgabe zeigt die Quell-DB-Instance und automatisierte Backups in USA West (Oregon), auf die Backups in US East (N. Virginia) repliziert werden.

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
```

```
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:09:53Z",
            }
            ...
        }
    ]
}
```

Im folgenden `describe-db-instance-automated-backups`-Beispiel wird die Option `--db-instance-automated-backups-arn` verwendet, um die replizierten Backups in der Zielregion anzuzeigen.

Zeigen Sie replizierte Backups wie folgt an:

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

Die Ausgabe zeigt die Quell-DB-Instance in USA West (Oregon) mit replizierten Backups in US East (N. Virginia).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:01:23Z"
            },
            "AllocatedStorage": 50,
            "BackupRetentionPeriod": 7,
            "Status": "replicating",
            "Port": 1521,
            ...
        }
    ]
}
```

Wiederherstellung zu einem bestimmten Zeitpunkt aus einem replizierten Backup für Amazon RDS

Sie können eine DB-Instance mithilfe der RDS Amazon-Konsole aus einem replizierten Backup zu einem bestimmten Zeitpunkt wiederherstellen. Sie können auch den `restore-db-instance-to-point-in-time` AWS CLI Befehl oder die `RestoreDBInstanceToPointInTime` RDS API Operation verwenden.

Allgemeine Informationen zu point-in-time recovery (PITR) finden Sie unterWiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

> **Note**
>
> Beachten Sie die folgenden Einschränkungen der DB-Engine, wenn automatische Backups repliziert werden: AWS-Regionen
>
> - RDSBei „SQLServer“ werden Optionsgruppen nicht kopiert.
> - RDSBei Oracle werden die folgenden Optionen nicht kopiert:`NATIVE_NETWORK_ENCRYPTION`, `OEMOEM_AGENT`, und`SSL`.

Wenn Sie Ihrer DB-Instance eine benutzerdefinierte Optionsgruppe zugeordnet haben, können Sie diese Optionsgruppe in der Zielregion neu erstellen. Stellen Sie dann die DB-Instance in der Zielregion wieder her und verknüpfen Sie die benutzerdefinierte Optionsgruppe mit ihr. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.

Konsole

Stellen Sie eine DB-Instance von einem replizierten Backup zu einem bestimmten Zeitpunkt wie folgt wieder her:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der Auswahl der Region die Zielregion (in die Backups repliziert werden) aus.
3. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
4. Wählen Sie auf der Registerkarte Replizierte Backups die DB-Instance aus, die Sie wiederherstellen möchten.
5. Wählen Sie unter Aktionen die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.
6. Wählen Sie Späteste Wiederherstellungszeit, um auf den spätesten möglichen Zeitpunkt wiederherzustellen oder wählen Sie Benutzerdefiniert, um eine Zeit auszuwählen.

   Geben Sie bei Auswahl von Custom (Benutzerdefiniert) das Datum und die Uhrzeit ein, zu dem/der Sie die Instance wiederherstellen möchten.

   > **Note**
   >
   > Die Zeiten werden in Ihrer lokalen Zeitzone angezeigt, was durch einen Offset von der koordinierten Weltzeit (UTC) gekennzeichnet ist. Zum Beispiel ist UTC -5 Eastern Standard Time/Central Daylight Time.

7. Geben Sie für DB-Instance-Kennung den Namen der wiederhergestellten DB-Ziel-Instance ein.
8. (Optional) Wählen Sie bei Bedarf andere Optionen aus, z. B. das Aktivieren von Autoscaling.
9. Wählen Sie Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

AWS CLI

Verwenden Sie den `restore-db-instance-to-point-in-time` AWS CLI Befehl, um eine neue DB-Instance zu erstellen.

Stellen Sie eine DB-Instance von einer replizierten Sicherung zu einem bestimmten Zeitpunkt wie folgt wieder her:

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" \
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2020-10-14T23:45:00.000Z
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2020-10-14T23:45:00.000Z
```

RDS API

Um eine DB-Instance zu einem bestimmten Zeitpunkt wiederherzustellen, rufen Sie den `RestoreDBInstanceToPointInTime`RDSAPIAmazon-Vorgang mit den folgenden Parametern auf:

- `SourceDBInstanceAutomatedBackupsArn`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Stoppen der automatisierten Backup-Replikation für Amazon RDS

Sie können die Backup-Replikation für DB-Instances mithilfe der RDS Amazon-Konsole beenden. Sie können auch den `stop-db-instance-automated-backups-replication` AWS CLI Befehl oder die `StopDBInstanceAutomatedBackupsReplication` RDS API Operation verwenden.

Replizierte Backups werden beibehalten, abhängig vom Aufbewahrungszeitraum für Backups, der bei ihrer Erstellung festgelegt wurde.

Konsole

Stoppen Sie die Sicherungs-Replikation auf der Seite Automated backups (Automatisierte Backups) in der Quellregion.

Um die Backup-Replikation auf eine zu beenden AWS-Region

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie die Quellregion aus der Regionsauswahl aus.
3. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
4. Wählen Sie auf der Registerkarte Aktuelle Region die DB-Instance aus, für die Sie die Sicherungs-Replikation stoppen möchten.
5. Wählen Sie für Aktionen die Option Regionsübergreifende Replikation verwalten aus.
6. Deaktivieren Sie unter Backup replication (Backup-Replikation) das Kontrollkästchen Enable replication to another AWS-Region (Replizierung in eine andere Region aktivieren).
7. Wählen Sie Save (Speichern) aus.

Replizierte Backups werden auf der Registerkarte Beibehaltung auf der Seite Automated backups (Automatisierte Backups) in der Zielregion aufgeführt.

AWS CLI

Stoppen Sie die Backup-Replikation mithilfe des `stop-db-instance-automated-backups-replication` AWS CLI Befehls.

Im folgenden CLI Beispiel wird verhindert, dass automatische Backups einer DB-Instance in der Region USA West (Oregon) repliziert werden.

Stoppen der Sicherungs-Replikation

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds stop-db-instance-automated-backups-replication \
  --region us-east-1 \
  --source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds stop-db-instance-automated-backups-replication ^
  --region us-east-1 ^
  --source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
  ```

RDS API

Beenden Sie die Backup-Replikation, indem Sie den StopDBInstanceAutomatedBackupsReplicationRDSAPIVorgang mit den folgenden Parametern ausführen:

- `Region`
- `SourceDBInstanceArn`

Löschen replizierter Backups für Amazon RDS

Sie können replizierte Backups für DB-Instances mithilfe der RDS Amazon-Konsole löschen. Sie können auch den `delete-db-instance-automated-backups` AWS CLI Befehl oder die `DeleteDBInstanceAutomatedBackup` RDS API Operation verwenden.

Konsole

Löschen Sie replizierte Backups in der Zielregion auf der Seite Automated backups (Automatisierte Backups).

Löschen Sie replizierte Backups wie folgt:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie die Zielregion aus der Regionsauswahl aus.
3. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
4. Wählen Sie auf der Registerkarte Replizierte Backups die DB-Instance aus, für die Sie die replizierten Backups löschen möchten.
5. Klicken Sie bei Actions auf Delete.
6. Geben Sie auf der Bestätigungsseite **`delete me`** ein, und wählen Sie die Option Delete (Löschen) aus.

AWS CLI

Löschen Sie replizierte Backups mithilfe des `delete-db-instance-automated-backup` AWS CLI Befehls.

Sie können den `describe-db-instances`CLIBefehl verwenden, um die Amazon-Ressourcennamen (ARNs) der replizierten Backups zu finden. Weitere Informationen finden Sie unter Informationen zu replizierten Backups für Amazon finden RDS.

Löschen Sie replizierte Backups wie folgt:

- Führen Sie einen der folgenden Befehle aus.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds delete-db-instance-automated-backup \
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds delete-db-instance-automated-backup ^
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

RDS API

Löschen Sie replizierte Backups, indem Sie den `DeleteDBInstanceAutomatedBackup`RDSAPIVorgang mit dem `DBInstanceAutomatedBackupsArn` Parameter verwenden.

Verwaltung manueller Backups

In diesem Abschnitt wird gezeigt, wie manuelle Backups für DB-Instances und DB-Cluster verwaltet werden.

Informationen zum Erstellen eines DB-Snapshots für ein Single-AZ DB-Instance finden Sie unterErstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Hinweise zum Erstellen eines DB-Snapshots für einen Multi-AZ-DB-Cluster finden Sie unterErstellen eines Multi-AZ-DB-Cluster-Snapshots für Amazon RDS.

Hinweise zum Löschen eines DB-Snapshots finden Sie unterLöschen eines DB-Snapshots für Amazon RDS.

Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS

Amazon RDS erstellt einen Speichervolumen-Snapshot Ihrer DB-Instance und sichert die gesamte DB-Instance und nicht nur einzelne Datenbanken. Das Erstellen dieses DB-Snapshots in einer Single-AZ-DB-Instance bewirkt eine kurzzeitige I/O-Unterbrechung, die – je nach Größe und Klasse Ihrer DB-Instance – einige Sekunden bis Minuten dauern kann. Für MariaDB, MySQL, Oracle und Postgre SQL wird die I/O-Aktivität auf Ihrem Primärserver während des Backups für Multi-AZ-Bereitstellungen nicht unterbrochen, da das Backup aus dem Standby-Modus stammt. Bei SQL Servern wird die I/O-Aktivität während des Backups für Multi-AZ-Bereitstellungen kurzzeitig unterbrochen.

Beim Erstellen eines DB-Snapshots wählen Sie die DB-Instance aus, die gesichert werden soll, und benennen den DB-Snapshot, damit Sie später mit diesem eine Wiederherstellung ausführen können. Die Zeit, die für die Erstellung eines Snapshots benötigt wird, hängt von der Größe Ihrer Datenbanken ab. Da der Snapshot das gesamte Speichervolume umfasst, wirkt sich die Größe von Dateien, wie z. B. temporären Dateien, auch auf die Zeitdauer aus, die für die Erstellung des Snapshots benötigt wird.

> **Note**
>
> Ihre DB-Instance muss den Status `available` aufweisen, um einen DB-Snapshot zu erstellen.
> Bei SQL Postgre-DB-Instances werden Daten in Tabellen ohne Protokollierung möglicherweise nicht aus Snapshots wiederhergestellt. Weitere Informationen finden Sie unter Bewährte Methoden für die Arbeit mit PostgreSQL.

Im Gegensatz zu automatisierten Backups unterliegen manuelle Snapshots nicht dem Aufbewahrungszeitraum für Backups. Snapshots laufen nicht ab.

Für sehr langfristige Backups von MariaDB-SQL, My- und SQL Postgre-Daten empfehlen wir, Snapshot-Daten nach Amazon S3 zu exportieren. Wenn die Hauptversion der DB-Engine nicht mehr unterstützt wird, können Sie diese Version nicht über einen Snapshot wiederherstellen. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS.

Sie können einen DB-Snapshot mit dem AWS Management Console, dem oder dem AWS CLI erstellen. RDS API

Konsole

So erstellen Sie einen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   Die Liste manuellen Snapshots wird angezeigt.
3. Wählen Sie Take Snapshot (Snapshot erstellen) aus.

   Das Fenster Take DB Snapshot (DB-Snapshot erstellen) wird angezeigt.
4. Wählen Sie die DB-Instance aus, für die Sie einen Snapshot erstellen möchten.
5. Geben Sie den Snapshot-Namen ein.
6. Wählen Sie Take Snapshot (Snapshot erstellen) aus.

Die Liste der manuellen Snapshots wird angezeigt, wobei der Status des neuen DB-Snapshots als `Creating` angezeigt wird. Nachdem sein Status`Available`, können Sie die Erstellungszeit sehen.

AWS CLI

Wenn Sie mit dem einen DB-Snapshot erstellen AWS CLI, müssen Sie angeben, welche DB-Instance Sie sichern möchten, und Ihrem DB-Snapshot einen Namen geben, damit Sie ihn später wiederherstellen können. Sie können dies tun, indem Sie den AWS CLI `create-db-snapshot`Befehl mit den folgenden Parametern verwenden:

- `--db-instance-identifier`
- `--db-snapshot-identifier`

In diesem Beispiel erstellen Sie einen DB-Snapshot mit dem Namen *`mydbsnapshot`* für eine DB-Instance namens *`mydbinstance`*.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier mydbinstance \
    --db-snapshot-identifier mydbsnapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier mydbinstance ^
    --db-snapshot-identifier mydbsnapshot
```

RDS API

Wenn Sie mit Amazon einen DB-Snapshot erstellen RDSAPI, müssen Sie angeben, welche DB-Instance Sie sichern möchten, und Ihrem DB-Snapshot einen Namen geben, damit Sie ihn später wiederherstellen können. Sie können dies tun, indem Sie den RDS API `CreateDBSnapshot`Amazon-Befehl mit den folgenden Parametern verwenden:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Erstellen eines Multi-AZ-DB-Cluster-Snapshots für Amazon RDS

Stellen Sie beim Erstellen eines Multi-AZ-DB-Cluster-Snapshots sicher, welchen Multi-AZ-DB-Cluster Sie sichern möchten, und geben Sie dann Ihrem DB-Cluster-Snapshot einen Namen, damit Sie ihn später wiederherstellen können. Sie können auch einen Multi-AZ-DB-Cluster-Snapshot teilen. Detaillierte Anweisungen finden Sie unter the section called “Freigeben eines DB Snapshots”.

Sie können einen Multi-AZ-DB-Cluster-Snapshot mit dem AWS Management Console AWS CLI, dem oder dem erstellen. RDS API

Für sehr langfristige Sicherungen empfehlen wir den Export von Snapshot-Daten in Amazon S3. Wenn die Hauptversion der DB-Engine nicht mehr unterstützt wird, können Sie diese Version nicht über einen Snapshot wiederherstellen. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS.

Konsole

So erstellen Sie einen DB-Cluster-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie in der Liste den Multi-AZ-DB-Cluster aus, für den Sie einen Snapshot erstellen möchten.
4. Wählen Sie für Aktionen die Option Take snapshot (Snapshot aufnehmen).

   Das Fenster Take DB Snapshot (DB-Snapshot erstellen) wird angezeigt.
5. Geben Sie den Namen des Snapshots in das Feld Snapshot name (Snapshot-Name) ein.
6. Wählen Sie Take Snapshot (Snapshot erstellen) aus.

Die Snapshots-Seite wird angezeigt, wobei der Status des neuen Multi-AZ-DB-Cluster-Snapshots als`Creating` angezeigt wird. Nachdem sein Status`Available`, können Sie die Erstellungszeit sehen.

AWS CLI

Sie können einen Multi-AZ-DB-Cluster-Snapshot erstellen, indem Sie den AWS CLI create-db-cluster-snapshotBefehl mit den folgenden Optionen verwenden:

- `--db-cluster-identifier`
- `--db-cluster-snapshot-identifier`

In diesem Beispiel erstellen Sie einen Multi-AZ-DB-Cluster-Snapshot mit dem Namen *`mein Multi-AZDB-Cluster-Snapshot`* für einen DB-Cluster namens *`mein Multiazdb-Cluster`*.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mymultiazdbcluster \
    --db-cluster-snapshot-identifier mymultiazdbclustersnapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-cluster-snapshot ^
    --db-cluster-identifier mymultiazdbcluster ^
    --db-cluster snapshot-identifier mymultiazdbclustersnapshot
```

RDS API

Sie können einen Multi-AZ-DB-Cluster-Snapshot erstellen, indem Sie den Amazon RDS API CreateDBCluster Snapshot-Vorgang mit den folgenden Parametern verwenden:

- `DBClusterIdentifier`
- `DBClusterSnapshotIdentifier`

Löschen eines Multi-AZ-DB-Cluster-Snapshots

Sie können von Amazon verwaltete Multi-AZ-DB-Snapshots löschenRDS, wenn Sie sie nicht mehr benötigen. Detaillierte Anweisungen finden Sie unter the section called “Löschen eines DB-Snapshots”.

Löschen eines DB-Snapshots für Amazon RDS

Sie können von Amazon verwaltete DB-Snapshots löschenRDS, wenn Sie sie nicht mehr benötigen.

> **Note**
>
> Verwenden Sie die AWS Backup Konsole, um Backups zu löschen AWS Backup, die von verwaltet werden. Informationen zu AWS Backup finden Sie im AWS Backup -Developer-Handbuch.

Löschen eines DB-Snapshots

Sie können einen manuellen, gemeinsam genutzten oder öffentlichen DB-Snapshot mit dem AWS Management Console AWS CLI, dem oder dem löschen RDSAPI.

Um einen geteilten oder öffentlichen Snapshot zu löschen, müssen Sie sich bei dem AWS Konto anmelden, dem der Snapshot gehört.

Wenn Sie vorhandene automatisierte DB-Snapshots löschen möchten, ohne die DB-Instance zu löschen, ändern Sie den Aufbewahrungszeitraum von Backups für die DB-Instance in "0". Die automatisierten Snapshots werden gelöscht, nachdem die Änderung angewandt wurde. Sie können die Änderung sofort übernehmen, wenn Sie nicht bis zum nächsten Wartungszeitraum warten möchten. Nachdem die Änderung vorgenommen wurde, können Sie automatische Backups wieder aktivieren, indem Sie für den Aufbewahrungszeitraum von Backups einen Wert größer als "0" festlegen. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Bei Aufbewahrung automatisierter Backups und manueller Snapshots fallen Gebühren an, bis sie gelöscht werden. Weitere Informationen finden Sie unter Aufbewahrungskosten.

Wenn Sie eine DB-Instance gelöscht haben, können Sie ihren automatisierten DB-Snapshots löschen, indem Sie die automatisierten Backups für die DB-Instance entfernen. Weitere Informationen zu automatischen Backups finden Sie unter Einführung in Backups.

Konsole

So löschen Sie einen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich die Option Snapshots.

   Die Liste manuellen Snapshots wird angezeigt.
3. Wählen Sie den DB-Snapshot aus, den Sie löschen möchten.
4. Wählen Sie unter Actions (Aktionen) die Option Delete Snapshot (Snapshot löschen) aus.
5. Wählen Sie auf der Bestätigungsseite die Option Delete (Löschen) aus.

AWS CLI

Sie können einen DB-Snapshot löschen, indem Sie den AWS CLI Befehl verwenden delete-db-snapshot.

Zum Löschen eines DB-Snapshots werden die folgenden Optionen verwendet.

- `--db-snapshot-identifier` – Die Kennung des DB-Snapshots.

Example

Der folgende Code löscht den DB-Snapshot `mydbsnapshot`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot
```

RDS API

Sie können einen DB-Snapshot mithilfe der RDS API Amazon-Operation D löscheneleteDBSnapshot.

Zum Löschen eines DB-Snapshots werden die folgenden Parameter verwendet.

- `DBSnapshotIdentifier` – Die Kennung des DB-Snapshots.

Wiederherstellung auf einer DB-Instance

In diesem Abschnitt wird gezeigt, wie eine Wiederherstellung auf einer DB-Instance durchgeführt wird. Auf dieser Seite wird gezeigt, wie aus einem DB-Snapshot eine Amazon RDS-DB-Instance wiederhergestellt wird.

Amazon RDS erstellt einen Snapshot für das Speichervolume der DB-Instance, damit die gesamte DB-Instance gesichert wird und nicht nur einzelne Datenbanken. Sie können eine neue DB-Instance erstellen, indem Sie einen DB-Snapshot wiederherstellen. Wenn Sie die DB-Instance wiederherstellen, können Sie den Namen des DB-Cluster-Snapshots angeben, aus dem die Wiederherstellung gestartet werden soll, und anschließend einen Namen für die neue DB-Instance angeben, die bei dieser Wiederherstellung erstellt wird. Sie können keine Wiederherstellung aus einem DB-Snapshot auf eine bestehende DB-Instance vornehmen. Bei der Wiederherstellung wird eine neue DB-Instance erstellt.

Sie können die wiederhergestellte DB-Instance nutzen, sobald ihr Status `available` ist. Die DB-Instance wird weiterhin Daten im Hintergrund laden. Dies wird als Lazy Loading bezeichnet.

Wenn Sie auf noch nicht geladene Daten zugreifen, lädt die DB-Instance sofort die angeforderten Daten von Amazon S3 herunter und fährt dann im Hintergrund mit dem Laden der restlichen Daten fort. Weitere Informationen finden Sie unter Amazon-EBS-Schnappschüsse.

Um die Auswirkungen des Lazy Loading auf Tabellen zu verringern, auf die Sie einen schnellen Zugriff benötigen, können Sie Vorgänge ausführen, die vollständige Tabellenscans beinhalten, z. B. `SELECT *`. Dadurch kann Amazon RDS alle gesicherten Tabellendaten von S3 herunterladen.

Sie können eine DB-Instance wiederherstellen und einen anderen Speichertyp als den im Quell-DB-Snapshot verwendeten angeben. In diesem Fall ist der Wiederherstellungsvorgang langsamer, da zusätzliche Arbeit für die Verlagerung der Daten in einen neuen Speichertyp erforderlich ist. Wenn die Wiederherstellung von oder auf einem magnetischen Speicher erfolgt, ist der Migrationsprozess am langsamsten. Das liegt daran, dass magnetische Speicher nicht über die IOPS-Fähigkeit der bereitgestellten IOPS- oder Allzweck (SSD)-Speicher verfügen.

Sie können AWS CloudFormation es verwenden, um eine DB-Instance aus einem DB-Instance-Snapshot wiederherzustellen. Weitere Informationen finden Sie unter AWS: :RDS:: DBInstance im AWS CloudFormation Benutzerhandbuch.

> **Note**
>
> Sie können eine DB-Instance nicht aus einem DB-Snapshot wiederherstellen, der freigegeben und verschlüsselt ist. Stattdessen können Sie eine Kopie des DB-Snapshots erstellen und die DB-Instance aus der Kopie wiederherstellen. Weitere Informationen finden Sie unter Kopieren eines DB-Snapshots für Amazon RDS.

Informationen zum Wiederherstellen einer DB-Instance mit einer RDS Extended Support-Version finden Sie unterWiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support.

Wiederherstellung aus einem Snapshot

Sie können eine DB-Instance mithilfe der AWS Management Console, der oder der RDS-API aus einem DB-Snapshot wiederherstellen. AWS CLI

> **Note**
>
> Sie können den Speicherplatz nicht reduzieren, wenn Sie eine DB-Instance wiederherstellen. Wenn Sie den zugewiesenen Speicherplatz erhöhen, muss er um mindestens 10 Prozent erhöht werden. Wenn Sie versuchen, den Wert um weniger als 10 Prozent zu erhöhen, erhalten Sie einen Fehler. Sie können den zugewiesenen Speicher bei der Wiederherstellung von DB-Instances von RDS für SQL Server nicht erhöhen.

Konsole

So stellen Sie eine DB-Instance aus einem DB-Snapshot wieder her

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot für die Wiederherstellung aus.
4. Wählen Sie in Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen) aus.
5. Geben Sie auf der Seite Restore Snapshot (Snapshot wiederherstellen) unter DB instance identifier (DB-Instance-Kennung) den Namen der wiederhergestellten Instance ein.

6. Geben Sie andere Einstellungen an, z. B. die Größe des zugewiesenen Speichers.

   Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

7. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

AWS CLI

Um eine DB-Instance aus einem DB-Snapshot wiederherzustellen, verwenden Sie den AWS CLI Befehl restore-db-instance-from-db-snapshot.

In diesem Beispiel führen Sie eine Wiederherstellung aus einem vorher erstellten DB-Snapshot mit dem Namen durc `mydbsnapshot`. Sie stellen auf eine neue DB-Instance namens wieder he `mynewdbinstance`. In diesem Beispiel wird auch die Größe des zugewiesenen Speichers festgelegt.

Sie können andere Einstellungen festlegen. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-snapshot-identifier mydbsnapshot \
    --allocated-storage 100
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-snapshot-identifier mydbsnapshot ^
    --allocated-storage 100
```

Dieser Befehl gibt etwa die folgende Ausgabe zurück:

```
DBINSTANCE  mynewdbinstance  db.t3.small  MySQL     50       sa             creating
 3  n  8.0.28  general-public-license
```

RDS-API

Um eine DB-Instance aus einem DB-Snapshot wiederherzustellen, rufen Sie die Amazon RDS-API-Funktion Restore DBInstance From DBSnapshot mit den folgenden Parametern auf:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Überlegungen

Überlegungen zur Wiederherstellung einer DB-Instance aus einem DB-Snapshot finden Sie in den folgenden Themen.

Themen

- Überlegungen zu Parametergruppen
- Überlegungen zu Sicherheitsgruppen
- Überlegungen zu Optionsgruppen
- Überlegungen zu Ressourcen-Markierungen
- Überlegungen zu Db2
- Überlegungen zu Microsoft SQL Server
- Überlegungen zu MySQL
- Überlegungen zu Oracle Database

Überlegungen zu Parametergruppen

Wir empfehlen Ihnen, die DB-Parametergruppe für alle DB-Snapshots aufzubewahren, die Sie erstellen, damit Sie Ihrer wiederhergestellten DB-Instance die korrekte Parametergruppe zuordnen können.

Die Standard-DB-Parametergruppe ist der wiederhergestellten Instance zugeordnet, es sei denn, Sie wählen eine andere Instance aus. In der Standard-DB-Parametergruppe sind keine benutzerdefinierten Parametereinstellungen verfügbar.

Sie können die Parametergruppe angeben, wenn Sie die DB-Instance wiederherstellen.

Weitere Informationen zu DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Überlegungen zu Sicherheitsgruppen

Wenn Sie eine DB-Instance wiederherstellen, werden die Standard-VPC (Virtual Private Cloud), DB-Subnetzgruppe und VPC-Sicherheitsgruppe mit der wiederhergestellten Instance verknüpft, wenn Sie keine anderen angeben.

- Wenn Sie die Amazon-RDS-Konsole verwenden, können Sie eine benutzerdefinierte VPC-Sicherheitsgruppe angeben, die der Instance zugeordnet werden soll, oder eine neue VPC-Sicherheitsgruppe erstellen.
- Wenn Sie die verwenden AWS CLI, können Sie eine benutzerdefinierte VPC-Sicherheitsgruppe angeben, die der Instance zugeordnet werden soll, indem Sie die `--vpc-security-group-ids` Option in den `restore-db-instance-from-db-snapshot` Befehl aufnehmen.
- Wenn Sie die Amazon RDS-API verwenden, können sie den `VpcSecurityGroupIds.VpcSecurityGroupId.N`-Parameter in der Aktion `RestoreDBInstanceFromDBSnapshot` einschließen.

Sobald die Wiederherstellung abgeschlossen ist und Ihre neue DB-Instance verfügbar ist, können Sie auch die VPC-Einstellungen ändern, indem Sie die DB-Instance bearbeiten. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Überlegungen zu Optionsgruppen

Wenn Sie eine DB-Instance wiederherstellen, ist die Standard-DB-Optionsgruppe meistens der wiederhergestellten DB-Instance zugeordnet.

Die Ausnahme ist, wenn die Quell-DB-Instance einer Optionsgruppe zugeordnet ist, die eine persistente oder permanente Option enthält. Wenn beispielsweise die Quell-DB-Instance Oracle Transparent Data Encryption (TDE) verwendet, muss die wiederhergestellte DB-Instance eine Optionsgruppe verwenden, die über die TDE-Option verfügt.

Wenn Sie eine DB-Instance in einer anderen VPC wiederherstellen, müssen Sie eine DB-Optionsgruppe zuweisen:

- Weisen Sie die Standard-Optionsgruppe für diese VPC-Gruppe der Instance zu.
- Weisen Sie eine weitere Optionsgruppe zu, die mit dieser VPC verknüpft ist.
- Erstellen einer neuen Optionsgruppe und Zuweisen von dieser zur DB-Instance Bei persistenten oder permanenten Optionen wie Oracle TDE müssen Sie eine neue Optionsgruppe erstellen, welche die persistente oder permanente Option enthält.

Weitere Informationen über DB-Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Überlegungen zu Ressourcen-Markierungen

Wenn Sie eine DB-Instance aus einem DB-Snapshot wiederherstellen, prüft RDS, ob Sie neue Tags angeben. Wenn ja, werden die neuen Tags zur wiederhergestellten DB-Instance hinzugefügt. Wenn es keine neuen Tags gibt, fügt RDS der wiederhergestellten DB-Instance die Tags aus der Quell-DB-Instance zum Zeitpunkt der Snapshot-Erstellung hinzu.

Weitere Informationen finden Sie unter Tags in DB-Snapshots kopieren.

Überlegungen zu Db2

Beim BYOL-Modell müssen Ihre Amazon RDS for Db2-DB-Instances mit einer benutzerdefinierten Parametergruppe verknüpft werden, die Ihre IBM Site ID und Ihr IBM Customer ID. Andernfalls schlagen Versuche fehl, eine DB-Instance aus einem Snapshot wiederherzustellen. Ihre Amazon RDS for Db2-DB-Instances müssen außerdem mit einer AWS License Manager selbstverwalteten Lizenz verknüpft sein. Weitere Informationen finden Sie unter Bringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit.

Beim AWS Marketplace Durchlaufmodell mit Db2-Lizenz benötigen Sie ein aktives AWS Marketplace Abonnement für die jeweilige IBM Db2 Edition, die Sie verwenden möchten. Wenn Sie noch keine haben, abonnieren Sie AWS Marketplace dafür Db2 IBM Db2 Ausgabe. Weitere Informationen finden Sie unter Db2-Lizenz über AWS Marketplace.

Überlegungen zu Microsoft SQL Server

Wenn Sie einen RDS for Microsoft SQL Server-DB-Snapshot in einer neuen Instance wiederherstellen, können Sie zu jeder Zeit die selbe Edition wie Ihr Snapshot wiederherstellen. In einigen Fällen können Sie auch die Edition Ihrer DB-Instance ändern. Die folgenden Einschränkungen beschreiben, wenn die Edition geändert wird:

- Der DB-Snapshot muss über genügend zugewiesenen Speicherplatz für die neue Edition verfügen.
- Es werden nur die folgenden Editionsänderungen unterstützt:
  - Von Standard Edition auf Enterprise Edition
  - Von Web Edition auf Standard Edition oder Enterprise Edition
  - Von Express Edition auf Web Edition, Standard Edition oder Enterprise Edition

Wenn Sie von einer Edition auf eine neuere wechseln möchten, die bei der Backup eines Snapshots nicht unterstützt wird, können Sie es mit der Funktion für natives Backup und Backup versuchen. SQL Server überprüft basierend auf den SQL Server-Funktionen, die Sie in Ihrer Datenbank aktiviert haben, ob eine Datenbank mit der neuen Edition kompatibel ist. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Überlegungen zu MySQL

Um Daten von einem RDS for MySQL-DB-Snapshot mit einer nicht unterstützten Engine-Version wiederherzustellen, müssen Sie Ihren DB-Snapshot möglicherweise mehrmals aktualisieren. Weitere Informationen zu Upgrade-Optionen finden Sie unterUpgrade-Optionen für DB-Snapshots mit nicht unterstützten Engine-Versionen für RDS for MySQL.

Weitere Informationen zum Upgrade der Engine-Version eines RDS for MySQL-DB-Snapshots finden Sie unterAktualisierung einer MySQL-DB-Snapshot-Engine-Version.

Überlegungen zu Oracle Database

Beachten Sie Folgendes, wenn Sie eine Oracle-Datenbank aus einem DB-Snapshot wiederherstellen:

- Bevor Sie einen DB-Snapshot wiederherstellen, können Sie ihn auf eine spätere Oracle-Datenbankversion aktualisieren. Weitere Informationen finden Sie unter Aktualisieren eines Oracle-DB-Snapshots.
- Wenn Sie einen Snapshot einer CDB-Instance wiederherstellen, die die Single-Tenant-Konfiguration verwendet, können Sie den PDB-Namen ändern. Sie können die PDB-Namen nicht ändern, wenn die CDB-Instance die Multi-Tenant-Konfiguration verwendet. Weitere Informationen finden Sie unter Sichern und Wiederherstellen einer CDB.
- Sie können den CDB-Namen nicht ändern. Dieser lautet immer `RDSCDB`. Dieser CDB-Name ist für alle CDB-Instances identisch.
- Sie können nicht direkt mit den Tenant-Datenbanken in einem DB-Snapshot interagieren. Wenn Sie einen Snapshot einer CDB-Instance wiederherstellen, die die Multi-Tenant-Konfiguration verwendet, stellen Sie alle ihre Tenant-Datenbanken wieder her. Sie können describe-db-snapshot-tenant-databases verwenden, um die Tenant-Datenbanken in einem DB-Snapshot zu überprüfen, bevor Sie ihn wiederherstellen.
- Wenn Sie Oracle verwenden GoldenGate, behalten Sie immer die Parametergruppe mit dem `compatible` Parameter bei. Wenn Sie eine DB-Instance aus einem DB-Snapshot

wiederherstellen, müssen Sie die Parametergruppe mit einem übereinstimmenden oder höheren Wert des Parameters `compatible` angeben.

- Sie können sich dafür entscheiden, Ihre Datenbank umzubenennen, wenn Sie einen DB-Snapshot wiederherstellen. Wenn die Gesamtgröße des Online-Redo-Logs mehr als 20 GB beträgt, setzt RDS Ihre Online-Redo-Log-Größe möglicherweise auf die Standardeinstellungen von 512 MB (4 x 128 MB) zurück. Durch die geringere Größe kann der Wiederherstellungsvorgang in angemessener Zeit abgeschlossen werden. Sie können die Online-Redo-Logs später erneut erstellen und die Größe ändern.
- Sie können Ihr Masterbenutzerpasswort in verwalten. AWS Secrets Manager Weitere Informationen finden Sie unter Überblick über die Verwaltung von Masterbenutzer-Passwörtern mit AWS Secrets Manager.

Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS

Sie können eine DB-Instance zu einem bestimmten Zeitpunkt wiederherstellen und dabei eine neue DB-Instance erstellen, ohne die Quell-DB-Instance zu ändern.

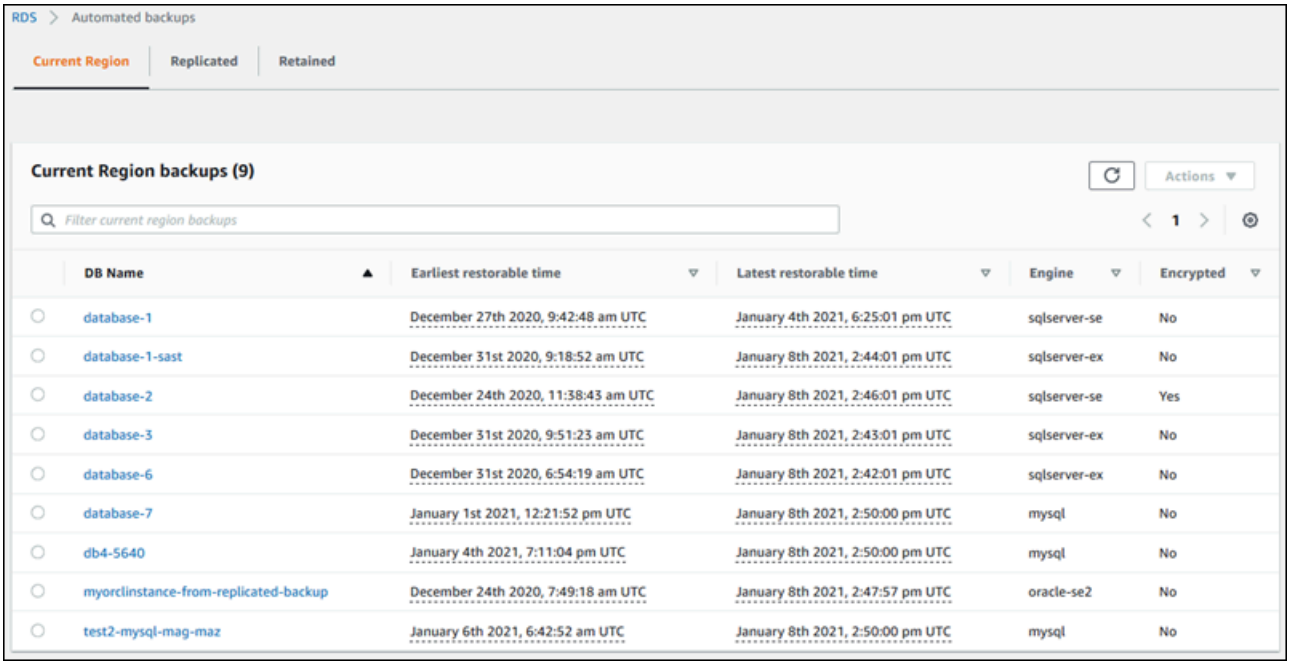
Wenn Sie eine DB-Instance zu einem bestimmten Zeitpunkt wiederherstellen, können Sie die Standard-VPC-Sicherheitsgruppe (Virtual Private Cloud) auswählen. Oder Sie können eine benutzerdefinierte VPC-Sicherheitsgruppe auf Ihre DB-Instance anwenden.

Wiederhergestellte DB-Instances werden automatisch dem Standard-DB-Parameter und Optionsgruppen zugeordnet. Sie können jedoch eine benutzerdefinierte Parametergruppe und eine Optionsgruppe anwenden, indem Sie diese während einer Wiederherstellung angeben.

Wenn die Quell-DB-Instance über Ressourcen-Tags verfügt, fügt RDS der wiederhergestellten DB-Instance die neuesten Tags hinzu.

RDS lädt Transaktionsprotokolle für DB-Instances alle fünf Minuten nach Amazon S3 hoch. Um den letzten wiederherstellbaren Zeitpunkt für eine DB-Instance zu sehen, verwenden Sie den AWS CLI describe-db-instancesBefehl und sehen Sie sich den Wert an, der im `LatestRestorableTime` Feld für die DB-Instance zurückgegeben wurde. Um die neueste Wiederherstellungszeit für jede DB-Instance in der Amazon RDS-Konsole anzuzeigen, wählen Sie Automatische Backups.

Sie können die Backup auf jeden beliebigen Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups vornehmen. Um den frühesten wiederherstellbaren Zeitpunkt für jede DB-Instance anzuzeigen, wählen Sie Automatische Backups in der Amazon RDS-Konsole aus.



> **Note**
>
> Es wird empfohlen, die gleiche oder eine ähnliche DB-Instance-Größe — und IOPS wiederherzustellen, wenn Speicher mit bereitgestellten IOPS verwendet wird — wie bei der Quell-DB-Instance verwendet wird. Möglicherweise wird ein Fehler angezeigt, wenn Sie beispielsweise eine DB-Instance-Größe mit einem inkompatiblen IOPS-Wert auswählen.

Informationen zum Wiederherstellen einer DB-Instance mit einer RDS Extended Support-Version finden Sie unterWiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support.

Bei einigen Amazon RDS-Datenbank-Engines müssen bei der Wiederherstellung ab einem bestimmten Zeitpunkt besondere Überlegungen angestellt werden:

- Wenn Sie die Passwortauthentifizierung mit einer Amazon RDS for Db2-DB-Instance verwenden, werden Benutzerverwaltungsaktionen`rdsadmin.add_user`, einschließlich, nicht in den Protokollen erfasst. Für diese Aktionen ist ein vollständiges Snapshot-Backup erforderlich.

  Beim BYOL-Modell müssen Ihre RDS für Db2-DB-Instances einer benutzerdefinierten Parametergruppe zugeordnet werden, die Ihre IBM Site ID und dein IBM Customer ID. Andernfalls schlagen Versuche fehl, eine DB-Instance auf einen bestimmten Zeitpunkt zurückzusetzen. Ihre Amazon RDS for Db2-DB-Instances müssen außerdem mit einer AWS License Manager

selbstverwalteten Lizenz verknüpft sein. Weitere Informationen finden Sie unter Bringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit.

Beim AWS Marketplace Durchlaufmodell mit Db2-Lizenz benötigen Sie ein aktives AWS Marketplace Abonnement für die jeweilige IBM Db2 Edition, die Sie verwenden möchten. Wenn Sie noch keine haben, abonnieren Sie AWS Marketplace dafür Db2 IBM Db2 Ausgabe. Weitere Informationen finden Sie unter Db2-Lizenz über AWS Marketplace.

- Wenn Sie eine RDS for Oracle-DB-Instance zu einem bestimmten Zeitpunkt wiederherstellen, können Sie eine andere DB-Engine, ein anderes Lizenzmodell und DBName (SID) für die wiederhergestellte DB-Instance angeben. Sie können auch angeben, dass RDS Masterbenutzerkennwörter unter speichern soll AWS Secrets Manager. Weitere Informationen finden Sie unter Überblick über die Verwaltung von Masterbenutzer-Passwörtern mit AWS Secrets Manager.
- Wenn Sie eine Microsoft SQL Server DB-Instance zu einem Zeitpunkt wiederherstellen, wird jede Datenbank innerhalb dieser Instance zu einem Zeitpunkt innerhalb von 1 Sekunde von jeder anderen Datenbank innerhalb der Instance wiederhergestellt. Transaktionen, die sich über mehrere Datenbanken innerhalb der Instance erstrecken, können inkonsistent wiederhergestellt werden.
- Für eine SQL Server-DB-Instance werden die Modi `OFFLINE`, `EMERGENCY` und `SINGLE_USER` nicht unterstützt. Wenn Sie eine Datenbank in einen dieser Modi schalten, wird die letzte wiederherstellbare Zeit für die gesamte Instance nicht mehr fortgesetzt.
- Einige Aktionen, wie z. B. das Ändern des Wiederherstellungsmodells einer SQL Server-Datenbank, können die Reihenfolge der Protokolle, die für die point-in-time Wiederherstellung verwendet werden, unterbrechen. In einigen Fällen kann Amazon RDS dieses Problem erkennen und die letzte wiederherstellbare Zeit wird an der Fortsetzung gehindert. In anderen Fällen, z. B. wenn eine SQL Server-Datenbank das Wiederherstellungsmodell `BULK_LOGGED` verwendet, wird die Unterbrechung der Protokollsequenz nicht erkannt. Es ist unter Umständen nicht möglich, eine SQL Server DB-Instance zu einem bestimmten Zeitpunkt wiederherzustellen, wenn die Protokollsequenz unterbrochen wurde. Aus diesen Gründen unterstützt Amazon RDS keine Änderung des Wiederherstellungsmodells von SQL Server-Datenbanken.

Sie können es auch verwenden AWS Backup , um Backups von Amazon RDS-DB-Instances zu verwalten. Wenn Ihre DB-Instance mit einem Backup-Plan verknüpft ist AWS Backup, wird dieser Backup-Plan für die point-in-time Wiederherstellung verwendet. Backups, die mit erstellt wurden, AWS Backup haben Namen, die auf enden`awsbackup:`*`AWS-Backup-job-number`*. Informationen zu AWS Backup finden Sie im AWS Backup Entwicklerhandbuch.

> **Note**
>
> Informationen in diesem Thema gelten für Amazon RDS. Weitere Informationen zum Wiederherstellen eines Amazon-Aurora-DB-Clusters finden Sie unter Wiederherstellen eines DB-Clusters zu einer bestimmten Zeit.

Sie können eine DB-Instance mithilfe der AWS Management Console, der oder der RDS-API auf einen bestimmten Zeitpunkt zurücksetzen. AWS CLI

> **Note**
>
> Sie können den Speicherplatz nicht reduzieren, wenn Sie eine DB-Instance wiederherstellen. Wenn Sie den zugewiesenen Speicherplatz erhöhen, muss er um mindestens 10 Prozent erhöht werden. Wenn Sie versuchen, den Wert um weniger als 10 Prozent zu erhöhen, erhalten Sie einen Fehler. Sie können den zugewiesenen Speicher bei der Wiederherstellung von DB-Instances von RDS für SQL Server nicht erhöhen.

Konsole

Wiederherstellen einer DB-Instance zu einer bestimmten Zeit

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.

   Die automatisierten Backups werden auf der Registerkarte Current Region (Aktuelle Region) angezeigt.
3. Wählen Sie die DB-Instance aus, die Sie wiederherstellen möchten.
4. Wählen Sie unter Aktionen die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

   Anschließend wird das Fenster Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) angezeigt.
5. Wählen Sie Späteste Wiederherstellungszeit, um auf den spätesten möglichen Zeitpunkt wiederherzustellen oder wählen Sie Benutzerdefiniert, um eine Zeit auszuwählen.

Geben Sie bei der Auswahl von Custom das Datum und die Uhrzeit ein, zu der Sie den Instance-Cluster wiederherstellen möchten.

> **Note**
>
> Zeiten werden in Ihrer lokalen Zeitzone angezeigt, die durch einen Offset von Coordinated Universal Time (UTC) angezeigt wird. Beispiel: UTC-5 ist Ost Standardzeit/ Zentral Sommerzeit.

6. Geben Sie für DB-Instance-Kennung den Namen der wiederhergestellten DB-Ziel-Instance ein. Der Name muss eindeutig sein.
7. Wählen Sie nach Bedarf andere Optionen wie die DB-Instance-Klasse und Speicher aus, sowie ob Sie Speicher-Autoscaling verwenden möchten.

   Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.
8. Wählen Sie Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

AWS CLI

Um eine DB-Instance zu einem bestimmten Zeitpunkt wiederherzustellen, verwenden Sie den AWS CLI Befehl restore-db-instance-to-, point-in-time um eine neue DB-Instance zu erstellen. In diesem Beispiel wird außerdem die Größe des zugewiesenen Speichers festgelegt und die automatische Speicherskalierung aktiviert.

Das Tagging von Ressourcen wird für diesen Vorgang unterstützt. Wenn Sie die Option `--tags` verwenden, werden die Tags der Quell-DB-Instance ignoriert und die bereitgestellten Tags verwendet. Andernfalls werden die neuesten Tags aus der Quell-Instance verwendet.

Sie können andere Einstellungen festlegen. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier mysourcedbinstance \
```

```
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2017-10-14T23:45:00.000Z \
    --allocated-storage 100 \
    --max-allocated-storage 1000
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier mysourcedbinstance ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2017-10-14T23:45:00.000Z ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000
```

RDS-API

Zum Wiederherstellen einer DB-Instance zu einer bestimmten Zeit rufen Sie den Amazon RDS-API `RestoreDBInstanceToPointInTime`-Betrieb mit den folgenden Parametern auf:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Wiederherstellen eines Multi-AZ-DB-Clusters zu einer bestimmten Zeit

Sie können einen Multi-AZ-DB-Cluster auf einen bestimmten Zeitpunkt wiederherstellen, wodurch ein neuer Multi-AZ-DB-Cluster erstellt wird.

RDS lädt Transaktionsprotokolle für Multi-AZ-DB-Cluster laufend nach Amazon S3 hoch. Sie können die Backup auf jeden beliebigen Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups vornehmen. Verwenden Sie den Befehl, um den frühesten wiederherstellbaren Zeitpunkt für einen Multi-AZ-DB-Cluster zu ermitteln. AWS CLI describe-db-clusters Sehen Sie sich den Wert an, der im `EarliestRestorableTime`-Feld für den DB-Cluster zurückgegeben wurde. Um die neueste wiederherstellbare Zeit für ein Multi-AZ-DB-Cluster anzuzeigen, sehen Sie sich den Wert an, der im `LatestRestorableTime` Feld für den DB-Cluster zurückgegeben wird.

Wenn Sie einen Multi-AZ-DB-Cluster auf einen bestimmten Zeitpunkt wiederherstellen, können Sie die Standard-VPC-Sicherheitsgruppe für Ihren Multi-AZ-DB-Cluster auswählen oder eine benutzerdefinierte VPC-Sicherheitsgruppe auf Ihren Multi-AZ-DB-Cluster anwenden.

Wiederhergestellte Multi-AZ-DB-Cluster werden automatisch dem Standard-DB-Cluster und Parametergruppen zugeordnet. Sie können jedoch eine benutzerdefinierte DB-Cluster-Parametergruppe anwenden, indem Sie sie bei einer Wiederherstellung angeben.

Wenn der Quell-DB-Cluster über Ressourcen-Tags verfügt, fügt RDS dem wiederhergestellten DB-Cluster die neuesten Tags hinzu.

> **Note**
>
> Es wird empfohlen, dieselbe oder eine ähnliche Multi-AZ-DB-Clustergröße wie den Quell-DB-Cluster wiederherzustellen. Es wird außerdem empfohlen, die Wiederherstellung mit demselben oder einem ähnlichen IOPS-Wert durchzuführen, wenn Sie bereitgestellten IOPS-Speicher verwenden. Möglicherweise wird ein Fehler angezeigt, wenn Sie beispielsweise eine DB-Cluster-Größe mit einem inkompatiblen IOPS-Wert auswählen.
> Wenn der Multi-AZ-Quell-DB-Cluster Allzweck-SSD-Speicher (GP3) verwendet und weniger als 400 GiB an zugewiesenem Speicher hat, können Sie die bereitgestellten IOPS für den wiederhergestellten DB-Cluster nicht ändern.

Informationen zur Wiederherstellung eines Multi-AZ-DB-Clusters mit einer RDS Extended Support-Version finden Sie unterWiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support.

Sie können einen Multi-AZ-DB-Cluster mithilfe der AWS Management Console, der oder der RDS-API auf AWS CLI einen bestimmten Zeitpunkt zurücksetzen.

Konsole

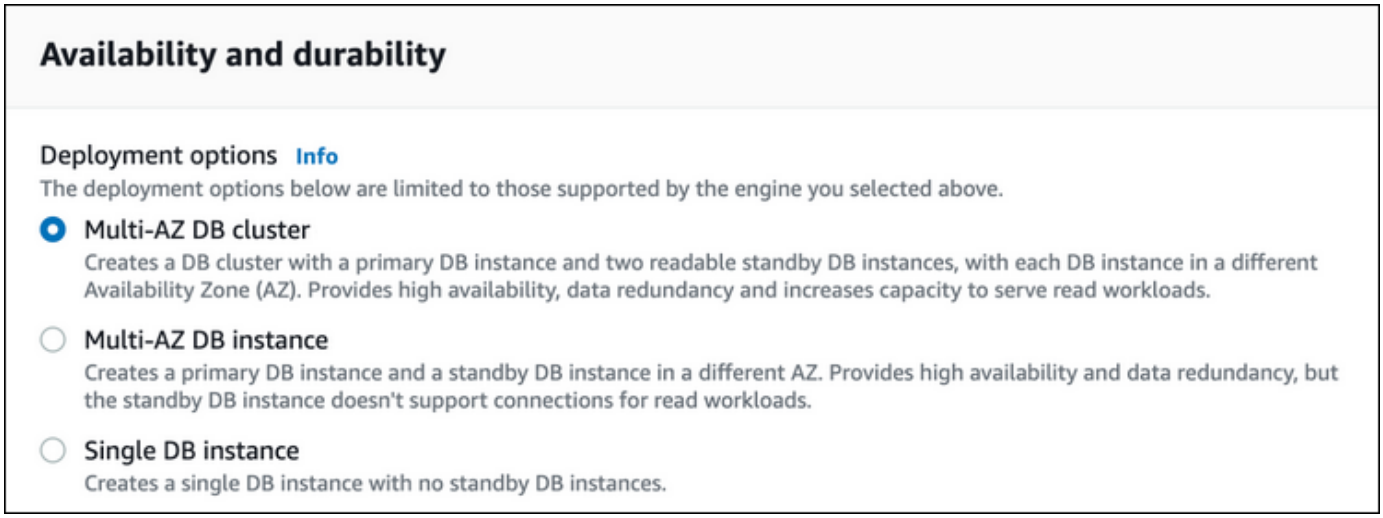
Wiederherstellen eines Multi-AZ-DB-Clusters zu einer bestimmten Zeit

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Multi-AZ-DB-Cluster aus, den Sie wiederherstellen möchten.
4. Wählen Sie unter Aktionen die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

   Anschließend wird das Fenster Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) angezeigt.
5. Wählen Sie Späteste Wiederherstellungszeit, um auf den spätesten möglichen Zeitpunkt wiederherzustellen oder wählen Sie Benutzerdefiniert, um eine Zeit auszuwählen.

   Wenn Sie Benutzerdefiniert wählen, geben Sie das Datum und die Uhrzeit ein, auf die Sie den Multi-AZ-DB-Cluster wiederherstellen möchten.

   > **Note**
   >
   > Zeiten werden in Ihrer lokalen Zeitzone angezeigt, die durch einen Offset von Coordinated Universal Time (UTC) angezeigt wird. Beispiel: UTC-5 ist Ost Standardzeit/ Zentral Sommerzeit.
6. Geben Sie bei DB cluster identifier (DB-Cluster-Kennung) den Namen für den wiederhergestellten Multi-AZ-DB-Cluster ein.
7. Wählen Sie unter Verfügbarkeit und Haltbarkeit die Option Multi-AZ-DB-Cluster aus.



8. Wählen Sie unter DB-Instance-Klasse eine DB-Instance-Klasse aus.

   Derzeit unterstützen Multi-AZ-DB-Cluster nur db.m6gd- und db.r6gd-DB-Instance-Klassen. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .
9. In den übrigen Abschnitten geben Sie die Einstellungen für Ihren DB-Cluster an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.
10. Wählen Sie Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

AWS CLI

Um einen Multi-AZ-DB-Cluster auf einen bestimmten Zeitpunkt zurückzusetzen, verwenden Sie den AWS CLI Befehl restore-db-cluster-to-, point-in-time um einen neuen Multi-AZ-DB-Cluster zu erstellen.

Derzeit unterstützen Multi-AZ-DB-Cluster nur db.m6gd- und db.r6gd-DB-Instance-Klassen. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Example

FürLinux, odermacOS: Unix

```
aws rds restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier mysourcemultiazdbcluster \
    --db-cluster-identifier mytargetmultiazdbcluster \
    --restore-to-time 2021-08-14T23:45:00.000Z \
    --db-cluster-instance-class db.r6gd.xlarge
```

Windows:

```
aws rds restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier mysourcemultiazdbcluster ^
    --db-cluster-identifier mytargetmultiazdbcluster ^
    --restore-to-time 2021-08-14T23:45:00.000Z ^
    --db-cluster-instance-class db.r6gd.xlarge
```

RDS-API

Um einen DB-Cluster zu einem bestimmten Zeitpunkt wiederherzustellen, rufen Sie den Amazon ClusterToPointInTime RDS-API-Vorgang RestoreDB mit den folgenden Parametern auf:

- `SourceDBClusterIdentifier`
- `DBClusterIdentifier`
- `RestoreToTime`

Wiederherstellen von einem Snapshot in einem Multi-AZ-DB-Cluster

Sie können einen Snapshot mithilfe der AWS Management Console, der AWS CLIoder der RDS-API in einem Multi-AZ-DB-Cluster wiederherstellen. Sie können jeden dieser Snapshots auf einem Multi-AZ-DB-Cluster wiederherstellen:

- Ein Snapshot einer Single-AZ-Bereitstellung
- Ein Snapshot einer Multi-AZ-DB-Cluster-Bereitstellung mit einer einzelnen DB-Instance
- Ein Snapshot eines Multi-AZ-DB-Clusters

Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

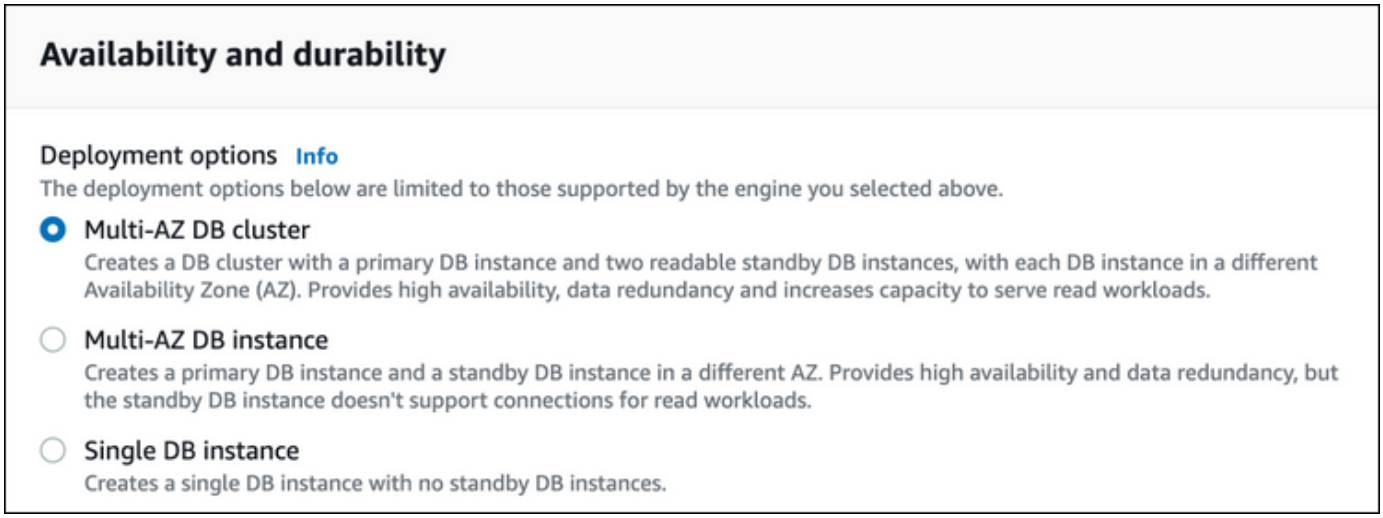
> **Tip**
>
> Sie können eine Single-AZ-Bereitstellung oder eine Multi-AZ-DB-Cluster-Bereitstellung zu einer Multi-AZ-DB-Cluster-Bereitstellung migrieren, indem Sie einen Snapshot wiederherstellen.

Informationen zum Wiederherstellen eines Multi-AZ-DB-Clusters mit einer RDS Extended Support-Version finden Sie unter Wiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support.

Konsole

So stellen Sie einen Snapshot auf einem Multi-AZ-DB-Cluster wieder

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den Snapshot für die Wiederherstellung aus.
4. Wählen Sie in Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen) aus.
5. Wählen Sie auf der Seite Snapshot wiederherstellen unter Verfügbarkeit und Haltbarkeit die Option Multi-AZ-DB-Cluster aus.



6. Geben Sie bei DB cluster identifier (DB-Cluster-Kennung) den Namen für den wiederhergestellten Multi-AZ-DB-Cluster ein.
7. In den übrigen Abschnitten geben Sie die Einstellungen für Ihren DB-Cluster an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern.
8. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

AWS CLI

Um einen Snapshot in einem Multi-AZ-DB-Cluster wiederherzustellen, verwenden Sie den AWS CLI Befehl restore-db-cluster-from-snapshot .

In dem folgenden Beispiel führen Sie eine Wiederherstellung aus einem vorher erstellten Snapshot mit dem Namen `mysnapshot` durch. Sie stellen in einem neuen Multi-AZ-DB-Cluster mit dem Namen `mynewmultiazdbcluster` wieder her. Sie geben auch die DB-Instance-Klasse an, die von den DB-Instances im Multi-AZ-DB-Cluster verwendet wird. Geben Sie entweder `mysql` oder `postgres` für die DB-Engine an.

Für die Option `--snapshot-identifier` können Sie entweder den Namen oder den Amazon-Ressourcennamen (ARN) verwenden, um einen DB-Cluster-Snapshot festzulegen. Sie können jedoch nur den ARN verwenden, um einen DB-Snapshot festzulegen.

Geben Sie für die Option `--db-cluster-instance-class` die DB-Instance-Klasse für den neuen Multi-AZ-DB-Cluster an. Multi-AZ-DB-Cluster unterstützen nur bestimmte DB-Instance-Klassen, z. B. db.m6gd und db.r6gd. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Sie können auch andere Optionen festlegen.

Example

Für Linux, macOSoder Unix:

```
aws rds restore-db-cluster-from-snapshot \
    --db-cluster-identifier mynewmultiazdbcluster \
    --snapshot-identifier mysnapshot \
    --engine mysql|postgres \
    --db-cluster-instance-class db.r6gd.xlarge
```

Windows:

```
aws rds restore-db-cluster-from-snapshot ^
    --db-cluster-identifier mynewmultiazdbcluster ^
    --snapshot-identifier mysnapshot ^
    --engine mysql|postgres ^
    --db-cluster-instance-class db.r6gd.xlarge
```

Nachdem Sie den DB-Cluster wiederhergestellt haben, können Sie den Multi-AZ-DB-Cluster der Sicherheitsgruppe hinzufügen, die mit dem DB-Cluster oder der DB-Instance verknüpft ist, mit dem bzw. der Sie den Snapshot erstellt haben, falls zutreffend. Durch Abschließen dieser Aktion werden dieselben Funktionen wie die des vorherigen DB-Clusters oder der DB-Instance bereitgestellt.

RDS-API

Um einen Snapshot in einem Multi-AZ-DB-Cluster wiederherzustellen, rufen Sie die RDS-API-Operation RestoreDBClusterFromSnapshot mit den folgenden Parametern auf:

- `DBClusterIdentifier`
- `SnapshotIdentifier`
- `Engine`

Sie können auch andere optionale Parameter angeben.

Nachdem Sie den DB-Cluster wiederhergestellt haben, können Sie den Multi-AZ-DB-Cluster der Sicherheitsgruppe hinzufügen, die mit dem DB-Cluster oder der DB-Instance verknüpft ist, mit dem bzw. der Sie den Snapshot erstellt haben, falls zutreffend. Durch Abschließen dieser Aktion werden dieselben Funktionen wie die des vorherigen DB-Clusters oder der DB-Instance bereitgestellt.

Wiederherstellen über einen Snapshot eines Multi-AZ-DB-Clusters in einer DB-Instance

Ein Snapshot eines Multi-AZ-DB-Clusters ist ein Speicher-Volume-Snapshot Ihres DB-Clusters, mit dem der gesamte DB-Cluster gesichert wird und nicht nur einzelne Datenbanken. Sie können einen Snapshot eines Multi-AZ-DB-Clusters in einer Single-AZ-Bereitstellung oder Multi-AZ-Bereitstellung der DB-Instance wiederherstellen. Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

> **Note**
>
> Sie können auch einen Snapshot eines Multi-AZ-DB-Clusters in einem neuen Multi-AZ-DB-Cluster wiederherstellen. Anweisungen finden Sie unter Wiederherstellen von einem Snapshot in einem Multi-AZ-DB-Cluster.

Informationen zur Wiederherstellung eines Multi-AZ-DB-Clusters mit einer RDS Extended Support-Version finden Sie unterWiederherstellung einer DB-Instance oder eines Multi-AZ-DB-Clusters, mit Amazon RDS Extended Support.

Verwenden Sie die AWS Management Console, oder die RDS-API AWS CLI, um einen Multi-AZ-DB-Cluster-Snapshot in einer Single-AZ-Bereitstellung oder einer Multi-AZ-DB-Instance-Bereitstellung wiederherzustellen.

Konsole

So stellen Sie einen Snapshot eines Multi-AZ-DB-Clusters in einer Single-AZ-Bereitstellung oder Multi-AZ-Bereitstellung der DB-Instance wieder her

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den Snapshot des Multi-AZ-DB-Clusters für die Wiederherstellung aus.
4. Wählen Sie in Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen) aus.
5. Wählen Sie auf der Seite Restore Snapshot (Snapshot wiederherstellen) unter Availability and durability (Verfügbarkeit und Beständigkeit) eine der folgenden Optionen aus:

- Single DB instance (Single-DB-Instance) – Stellt den Snapshot in einer DB-Instance ohne Standby-DB-Instance wieder her.
- Multi-AZ DB instance (Multi-AZ-DB-Instance) – Stellt den Snapshot in einer Multi-AZ-DB-Instance-Bereitstellung mit einer primären DB-Instance und einer Standby-DB-Instance wieder her.

6. Geben Sie für DB instance identifier (DB-Instance-Kennung) den Namen Ihrer wiederhergestellten DB-Instance ein.
7. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die DB-Instance an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.
8. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

AWS CLI

Um einen Multi-AZ-DB-Cluster-Snapshot in einer DB-Instance-Bereitstellung wiederherzustellen, verwenden Sie den AWS CLI Befehl restore-db-instance-from-db-snapshot.

In dem folgenden Beispiel führen Sie eine Wiederherstellung aus einem zuvor erstellten Snapshot eines Multi-AZ-DB-Clusters mit dem Namen `myclustersnapshot` durch. Die Wiederherstellung erfolgt in einer neuen Multi-AZ-DB-Instance-Bereitstellung mit einer primären DB-Instance mit dem Namen `mynewdbinstance`. Geben Sie für die Option `--db-cluster-snapshot-identifier` den Namen des Snapshots des Multi-AZ-DB-Clusters an.

Geben Sie für die Option `--db-instance-class` die DB-Instance-Klasse für die neue DB-Instance-Bereitstellung an. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Sie können auch andere Optionen festlegen.

Example

LinuxmacOSUnixFür, oder:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --engine mysql \
    --multi-az \
    --db-instance-class db.r6g.xlarge
```

Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-cluster-snapshot-identifier myclustersnapshot ^
    --engine mysql ^
    --multi-az ^
    --db-instance-class db.r6g.xlarge
```

Nachdem Sie die DB-instance wiederhergestellt haben, können Sie sie der Sicherheitsgruppe hinzufügen, die mit dem Multi-AZ-DB-Cluster verknüpft ist, mit dem Sie den Snapshot erstellt haben, falls zutreffend. Durch Abschließen dieser Aktion werden dieselben Funktionen wie die des vorherigen Multi-AZ-DB-Clusters bereitgestellt.

RDS-API

Um einen Multi-AZ-DB-Cluster-Snapshot in einer DB-Instance-Bereitstellung wiederherzustellen, rufen Sie den RDS-API-Vorgang RestoreDB InstanceFrom DBSnapshot mit den folgenden Parametern auf:

- `DBInstanceIdentifier`
- `DBClusterSnapshotIdentifier`
- `Engine`

Sie können auch andere optionale Parameter angeben.

Nachdem Sie die DB-instance wiederhergestellt haben, können Sie sie der Sicherheitsgruppe hinzufügen, die mit dem Multi-AZ-DB-Cluster verknüpft ist, mit dem Sie den Snapshot erstellt haben, falls zutreffend. Durch Abschließen dieser Aktion werden dieselben Funktionen wie die des vorherigen Multi-AZ-DB-Clusters bereitgestellt.

Tutorial: Eine RDS Amazon-DB-Instance aus einem DB-Snapshot wiederherstellen

Wenn Sie mit Amazon arbeiten, haben RDS Sie oft eine DB-Instance, mit der Sie gelegentlich arbeiten, die Sie aber nicht in Vollzeit benötigen. Nehmen wir beispielsweise an, Sie führen eine vierteljährliche Kundenumfrage durch, bei der eine EC2 Amazon-Instance zum Hosten einer Website für Kundenumfragen verwendet wird. Außerdem haben Sie eine DB-Instance, die zum Speichern der Umfrageergebnisse verwendet wird. Eine Möglichkeit, in einem solchen Szenario Geld zu sparen, besteht darin, nach Abschluss der Umfrage einen DB-Snapshot der DB-Instance zu verwenden. Anschließend löschen Sie die DB-Instance und stellen sie wieder her, wenn Sie die Umfrage erneut durchführen müssen.

Wenn Sie die DB-Instance wiederherstellen, geben Sie den Namen des DB-Snapshots für die Wiederherstellung an. Anschließend vergeben Sie einen Namen für die neue DB-Instance, die durch den Wiederherstellungsvorgang erstellt wird.

Genauere Informationen zum Wiederherstellen von DB-Instances aus Snapshots finden Sie unter Wiederherstellung auf einer DB-Instance.

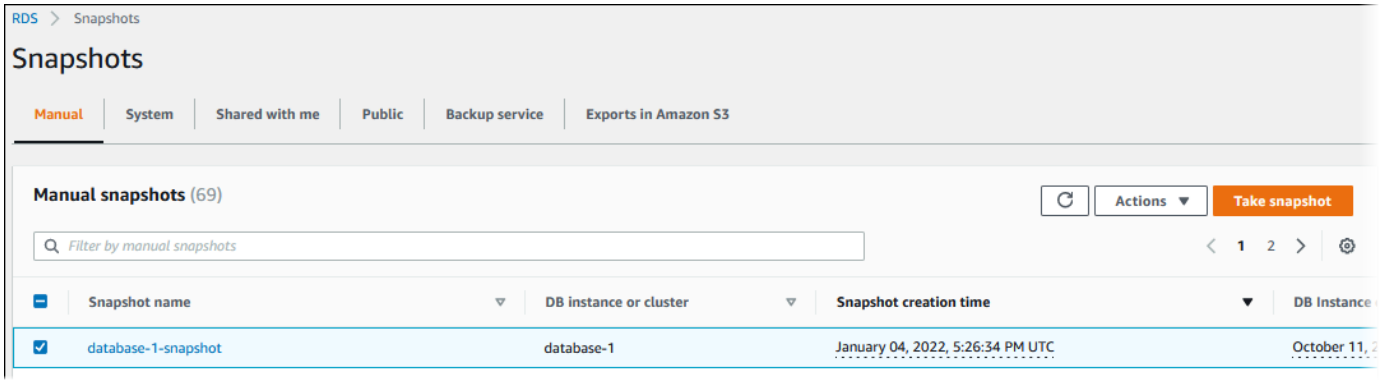
Für Informationen über AWS KMS Schlüsselverwaltung für AmazonRDS, sieheAWS KMS key Verwaltung.

Wiederherstellen einer DB-Instance aus einem DB-Snapshot

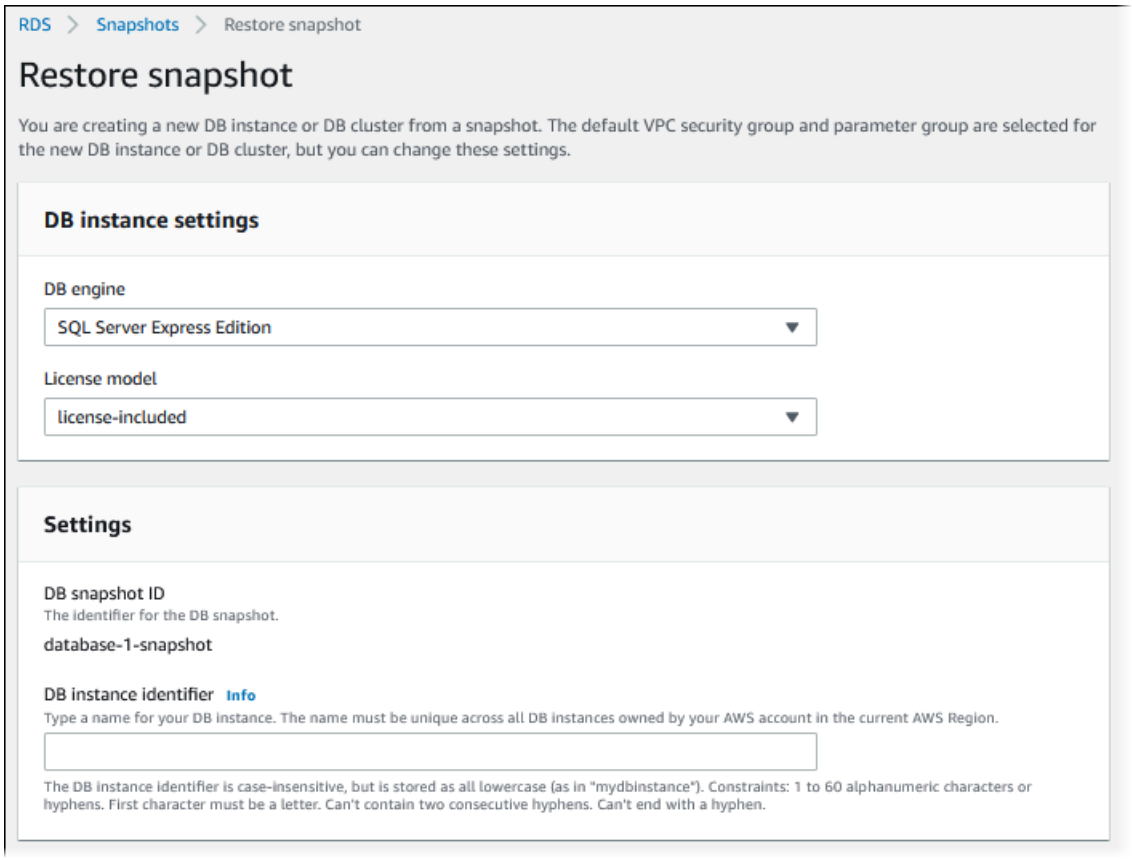
Gehen Sie wie folgt vor, um eine Wiederherstellung aus einem Snapshot im AWS Management Console.

So stellen Sie eine DB-Instance aus einem DB-Snapshot wieder her

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot für die Wiederherstellung aus.
4. Wählen Sie in Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen) aus.



Anschließend wird die Seite Restore snapshot (Snapshot wiederherstellen) angezeigt.



5. Verwenden Sie unter DB-Instance-Einstellungen die Standardeinstellungen für DB-Engine und Lizenzmodell (für Oracle oder Microsoft SQL Server).
6. Geben Sie unter Settings (Einstellungen) bei DB instance identifier (DB-Instance-Kennung) den eindeutigen Namen ein, den Sie für die wiederhergestellte DB-Instance verwenden möchten, zum Beispiel **`mynewdbinstance`**.

   Sie können den Namen dieser Instance während der Wiederherstellung aus einer DB-Instance verwenden, die nach der Erstellung des DB-Snapshots gelöscht wurde.

7. Wählen Sie unter Verfügbarkeit und Beständigkeit aus, ob eine Standby-Instance in einer anderen Availability Zone erstellt werden soll.

   Erstellen Sie für dieses Tutorial keine Standby-Instance.

8. Verwenden Sie unter Connectivity (Konnektivität) die Standardeinstellungen für folgende Optionen:

   - Virtuelle private Cloud (VPC)
   - DB subnet group (DB-Subnetzgruppe)
   - Öffentlicher Zugriff
   - VPCSicherheitsgruppe (Firewall)

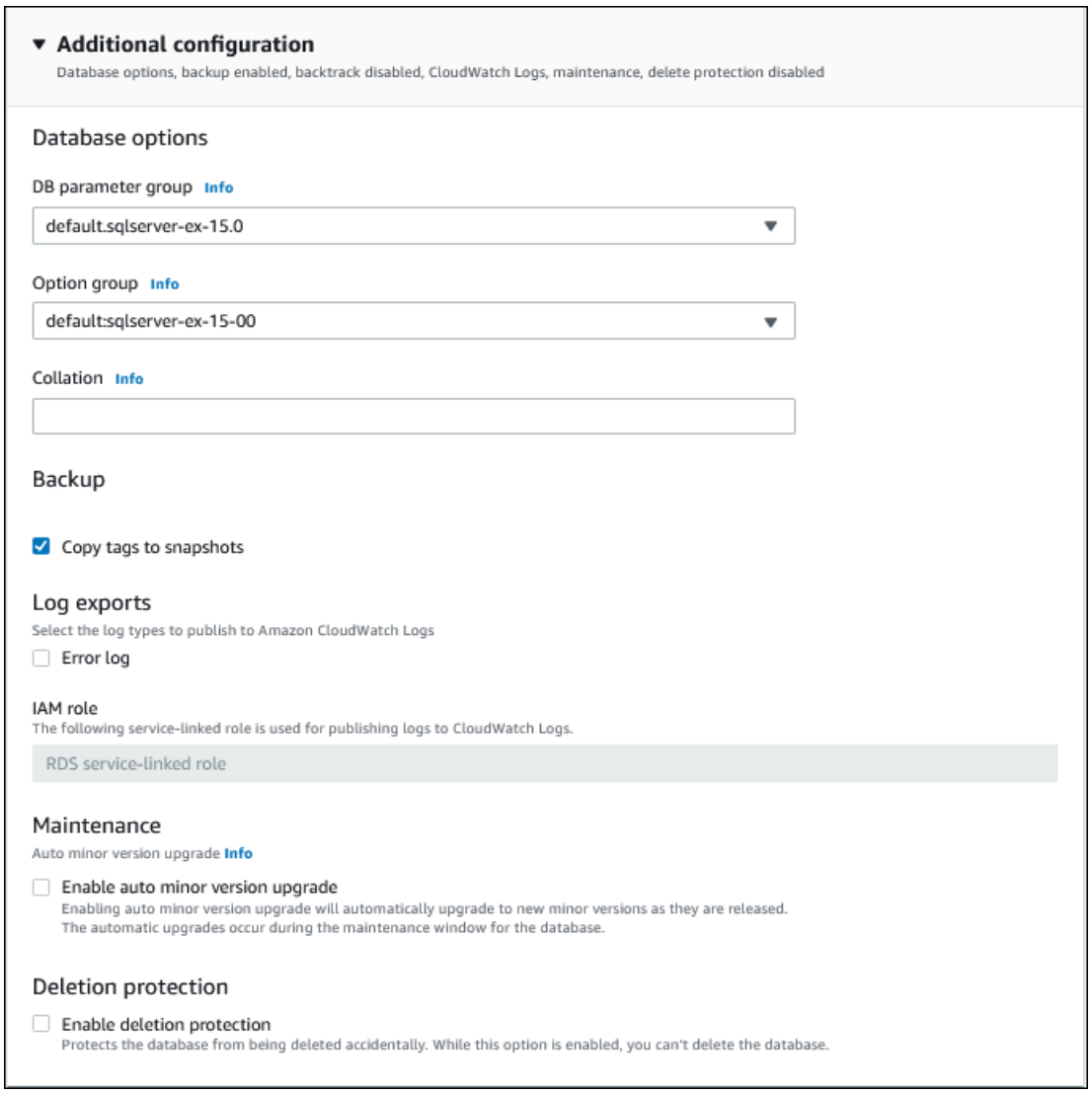
9. Wählen Sie die DB instance class (DB-Instance-Klasse) aus.

   Wählen Sie für dieses Tutorial Burstable classes (includes t classes) (Burst-fähige Klassen (einschließlich t-Klassen)) und dann db.t3.small aus.

10. Für Encryption (Verschlüsselung) verwenden Sie die Standardeinstellungen.

    Wenn die Quell-DB-Instance für den Snapshot verschlüsselt wurde, wird die wiederhergestellte DB-Instance ebenfalls verschlüsselt. Die Verschlüsselung kann nicht aufgehoben werden.

11. Erweitern Sie Additional configuration (Zusätzliche Konfiguration) unten auf der Seite.



12. Führen Sie unter Database options (Datenbankoptionen) folgende Schritte aus:

    a. Wählen Sie DB parameter group (DB-Parametergruppe) aus.

       Für dieses Tutorial verwenden Sie die Standard-DB-Parametergruppe.

    b. Wählen Sie die Option group (Optionsgruppe) aus.

       Für dieses Tutorial verwenden Sie die Standard-Optionsgruppe

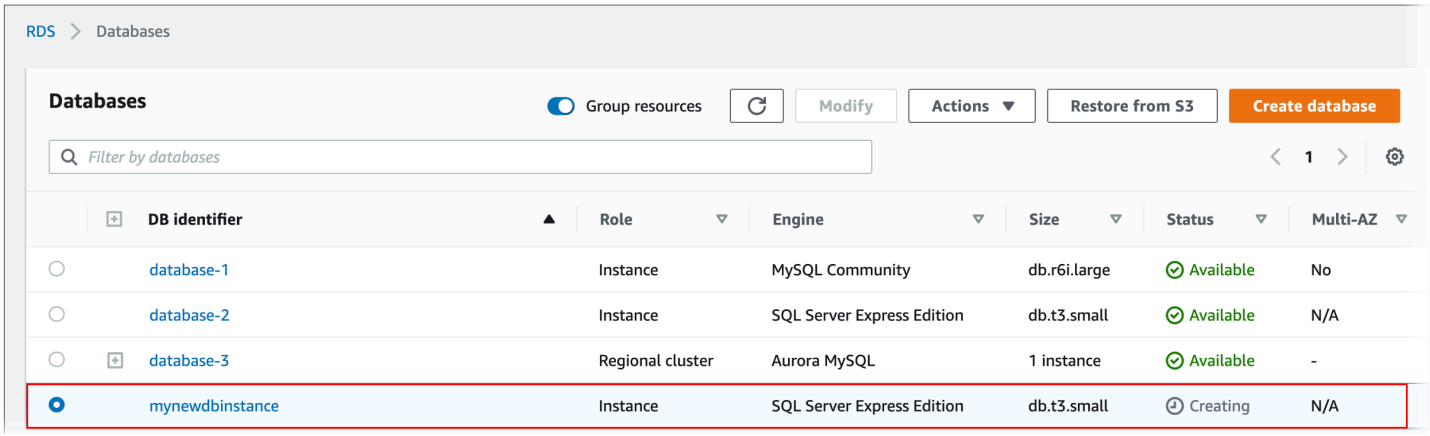
       > **Important**
       >
       > In einigen Fällen führen Sie möglicherweise eine Wiederherstellung aus einem DB-Snapshot einer DB-Instance durch, die eine persistente oder permanente Option

verwendet. In diesem Fall müssen Sie eine Optionsgruppe auswählen, die dieselbe Option verwendet.

c. Für Deletion protection (Löschschutz) setzen Sie ein Häkchen bei Enable deletion protection (Löschschutz aktivieren).

13. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

Auf der Seite Databases (Datenbanken) wird die wiederhergestellte DB-Instance mit dem Status `Creating` angezeigt.



Kopieren eines DB-Snapshots für Amazon RDS

Mit Amazon RDS können Sie automatisierte Backups oder manuelle DB-Snapshots kopieren. Nach dem Kopieren eines Snapshots ist die Kopie ein manueller Snapshot. Sie können mehrere Kopien eines automatisierten Backups oder eines manuellen Snapshots erstellen, aber jede Kopie muss über eine eindeutige Kennung verfügen.

Sie können einen Snapshot innerhalb desselben kopieren AWS-Region, Sie können einen Snapshot zwischen AWS-Regionen den anderen kopieren und Sie können gemeinsam genutzte Snapshots kopieren.

Kopieren eines DB-Snapshots

Für jeden AWS-Konto können Sie bis zu 20 DB-Snapshots gleichzeitig von einem AWS-Region zum anderen kopieren. Wenn Sie einen DB-Snapshot in einen anderen kopieren AWS-Region, erstellen Sie einen manuellen DB-Snapshot, der in diesem AWS-Region gespeichert wird. Beim Kopieren eines DB-Snapshots aus der Quelle AWS-Region fallen Amazon RDS-Datenübertragungsgebühren an.

Weitere Informationen zu den Kosten von Datenübertragungen finden Sie unter Amazon RDS – Preise.

Nachdem die DB-Snapshot-Kopie in der neuen Version erstellt wurde AWS-Region, verhält sich die DB-Snapshot-Kopie genauso wie alle anderen DB-Snapshots in dieser Version. AWS-Region

Sie können einen DB-Snapshot mit der AWS Management Console, der oder der AWS CLI RDS-API kopieren.

Konsole

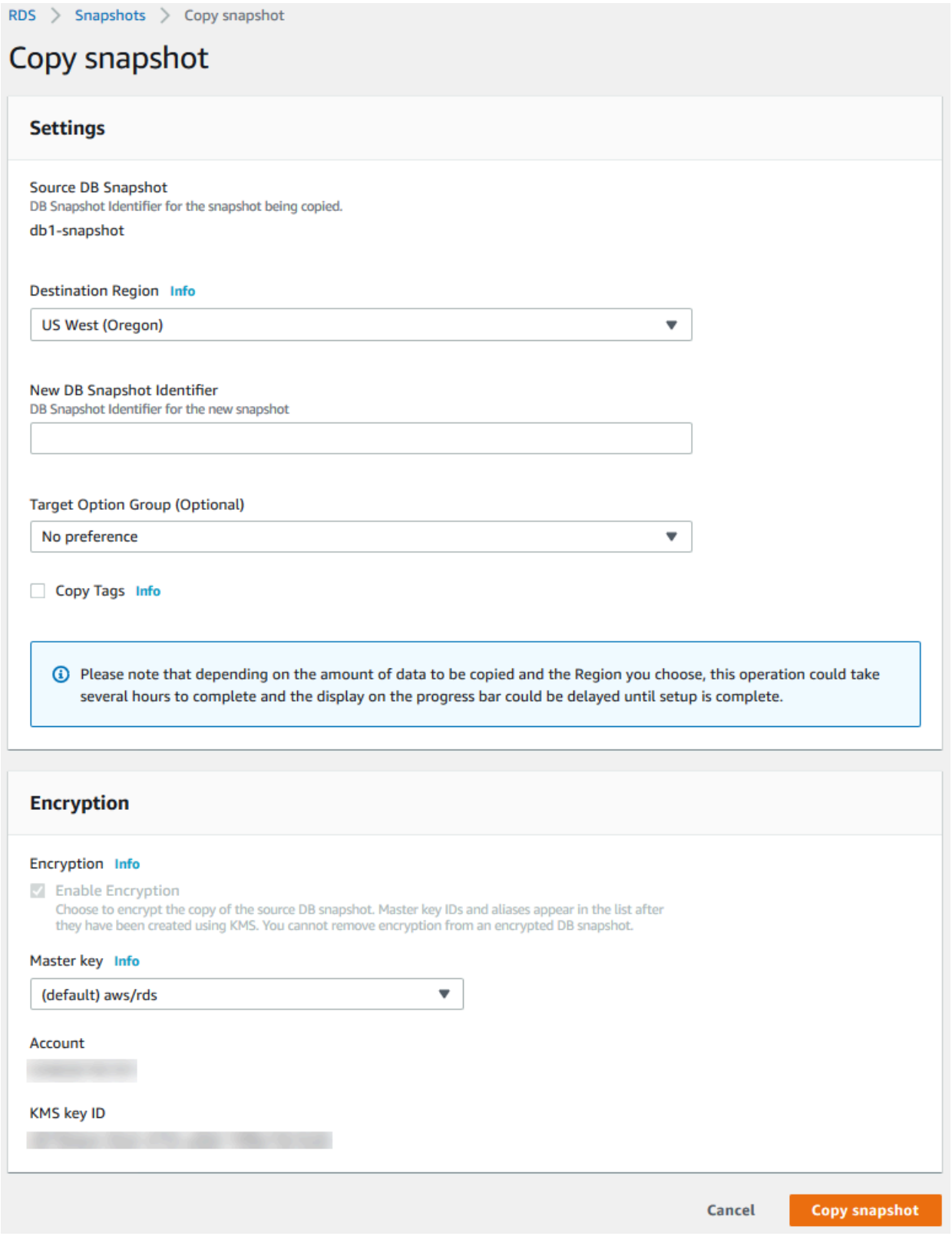
Mit dem folgenden Verfahren wird ein verschlüsselter oder unverschlüsselter DB-Snapshot in derselben Region AWS-Region oder in mehreren Regionen mithilfe von kopiert. AWS Management Console

So kopieren Sie einen DB-Snapshot

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

3. Wählen Sie den zu kopierenden DB-Snapshot.
4. Wählen Sie für Actions (Aktionen) die Option Copy Snapshot (Snapshot kopieren).

   Die Seite Snapshot kopieren wird angezeigt.



5. Wählen Sie unter Target option group (optional) (Zieloptionsgruppe (optional)) ggf. eine neue Optionsgruppe aus.

Geben Sie diese Option an, wenn Sie einen Snapshot von einem AWS-Region Snapshot in einen anderen kopieren und Ihre DB-Instance eine nicht standardmäßige Optionsgruppe verwendet.

Damit die Quell-DB-Instance für Oracle oder Microsoft SQL Server die transparente Datenverschlüsselung verwendet, müssen Sie diese Option beim regionsübergreifenden Kopieren angeben. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.

6. (Optional) Um den DB-Snapshot in eine andere zu kopieren AWS-Region, wählen Sie unter Zielregion die neue AWS-Region aus.

> **Note**
>
> Für das Ziel AWS-Region muss dieselbe Datenbank-Engine-Version verfügbar sein wie für die Quelle AWS-Region.

7. Geben Sie in New DB Snapshot Identifier (Neue DB-Snapshot-Kennung) den Namen der DB-Snapshot-Kopie ein.

   Sie können mehrere Kopien eines automatisierten Backups oder eines manuellen Snapshots erstellen, aber jede Kopie muss über eine eindeutige Kennung verfügen.

8. (Optional) Wählen Sie Copy Tags (Tags kopieren), um Tags und Werte aus dem Snapshot in die Kopie des Snapshots zu übernehmen.
9. (Optional) Für Verschlüsselung , gehen Sie folgendermaßen vor:

   a. Wählen Sie Enable Encryption (Verschlüsselung aktivieren), wenn der DB-Snapshot nicht verschlüsselt ist, die Kopie aber verschlüsselt werden soll.

   > **Note**
   >
   > Wenn der DB-Snapshot verschlüsselt ist, müssen Sie die Kopie verschlüsseln, damit das Kontrollkästchen bereits aktiviert ist.

   b. Für AWS KMS key geben Sie den KMS-Schlüsselbezeichner an, der zur Verschlüsselung der DB-Snapshot-Kopie verwendet werden soll.

10. Wählen Sie Copy Snapshot (Snapshot kopieren) aus.

AWS CLI

Sie können einen DB-Snapshot mit dem AWS CLI Befehl kopieren copy-db-snapshot. Wenn Sie den Snapshot in einen neuen kopieren AWS-Region, führen Sie den Befehl im neuen aus AWS-Region.

Zum Kopieren eines DB-Snapshots werden die folgenden Optionen verwendet. Nicht alle Optionen werden in allen Szenarien benötigt. Verwenden Sie die folgenden Beschreibungen und Beispiele, um die zu verwendenden Optionen zu ermitteln.

- `--source-db-snapshot-identifier` – Der Bezeichner des Quell-DB-Snapshots.
  - Wenn sich der Quell-Snapshot in derselben Datei AWS-Region wie die Kopie befindet, geben Sie eine gültige DB-Snapshot-ID an. Beispiel, `rds:mysql-instance1-snapshot-20130805`.
  - Wenn sich der Quell-Snapshot in derselben Datei AWS-Region wie die Kopie befindet und mit Ihrem geteilt wurde AWS-Konto, geben Sie einen gültigen DB-Snapshot-ARN an. Beispiel, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Wenn sich der Quell-Snapshot in einer anderen Version AWS-Region als der Kopie befindet, geben Sie einen gültigen DB-Snapshot-ARN an. Beispiel, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Wenn Sie aus einem freigegebenen manuellen DB-Snapshot kopieren, muss dieser Parameter der Amazon-Ressourcenname (ARN) des freigegebenen DB-Snapshots sein.
  - Wenn Sie einen verschlüsselten Snapshot kopieren, muss dieser Parameter das ARN-Format für die Quelle AWS-Region haben und mit dem `SourceDBSnapshotIdentifier` im `PreSignedUrl` Parameter übereinstimmen.
- `--target-db-snapshot-identifier` – Der Bezeichner für die neue Kopie des verschlüsselten DB-Snapshots.
- `--copy-option-group` – Kopieren Sie die Optionsgruppe aus einem Snapshot, der mit Ihrem AWS-Konto geteilt wurde.
- `--copy-tags` – Geben Sie diese Option an, damit Tags und Werte aus dem Snapshot in die Kopie des Snapshots übernommen werden.
- `--option-group-name` – Die Optionsgruppe, die der Kopie des Snapshots zugeordnet werden soll.

  Geben Sie diese Option an, wenn Sie einen Snapshot von einem AWS-Region Snapshot in einen anderen kopieren und Ihre DB-Instance eine nicht standardmäßige Optionsgruppe verwendet.

Damit die Quell-DB-Instance für Oracle oder Microsoft SQL Server die transparente Datenverschlüsselung verwendet, müssen Sie diese Option beim regionsübergreifenden Kopieren angeben. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.

- `--kms-key-id` – Der KMS-Schlüsselbezeichner für einen verschlüsselten DB-Snapshot. Der KMS-Schlüsselbezeichner ist der Amazon Resource Name (ARN), der Schlüsselbezeichner oder der Schlüsselalias für den KMS-Schlüssel.
  - Wenn Sie einen verschlüsselten DB-Snapshot von Ihrem kopieren AWS-Konto, können Sie einen Wert für diesen Parameter angeben, um die Kopie mit einem neuen KMS-Schlüssel zu verschlüsseln. Wenn Sie keinen Wert für diesen Parameter angeben, wird die Kopie des DB-Snapshots mit demselben KMS-Verschlüsselungsschlüssel wie der Quell-DB-Snapshot verschlüsselt.
  - Wenn Sie einen verschlüsselten DB-Snapshot, der gemeinsam genutzt wird AWS-Konto, von einem anderen kopieren, müssen Sie einen Wert für diesen Parameter angeben.
  - Falls Sie diesen Parameter beim Kopieren eines unverschlüsselten Snapshots angeben, wird die Kopie verschlüsselt.
  - Wenn Sie einen verschlüsselten Snapshot in einen anderen kopieren AWS-Region, müssen Sie einen KMS-Schlüssel für das Ziel angeben AWS-Region. KMS-Schlüssel sind spezifisch für AWS-Region das, in dem sie erstellt wurden, und Sie können keine Verschlüsselungsschlüssel von einem AWS-Region Schlüssel zum anderen verwenden AWS-Region.

Example Unverschlüsselte Quelle in derselben Region

Der folgende Code erstellt eine Kopie eines Snapshots mit dem neuen Namen`mydbsnapshotcopy`, der dem Quell-Snapshot entspricht AWS-Region . Beim Erstellen der Kopie werden die DB-Optionsgruppe und Tags des ursprünglichen Snapshots in die Snapshot-Kopie übernommen.

Für LinuxmacOS, oderUnix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --copy-option-group \
    --copy-tags
```

Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --copy-option-group ^
    --copy-tags
```

Example Unverschlüsselte Quelle in andere Region

Der folgende Code erstellt eine Kopie eines Snapshots mit dem neuen Namen`mydbsnapshotcopy`, AWS-Region in dem der Befehl ausgeführt wird.

Für LinuxmacOS, oderUnix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Example Verschlüsselte Quelle in andere Region

Im folgenden Codebeispiel wird ein verschlüsselter DB-Snapshot aus der USA West (Oregon)-Region in die US East (N. Virginia)-Region kopiert. Führen Sie den Befehl in der Zielregion (us-east-1) aus.

Für LinuxmacOS, oderUnix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --kms-key-id my-us-east-1-key \
    --option-group-name custom-option-group-name
```

Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-
instance1-snapshot-20161115 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --kms-key-id my-us-east-1-key ^
    --option-group-name custom-option-group-name
```

Der `--source-region` Parameter ist erforderlich, wenn Sie einen verschlüsselten Snapshot zwischen den Regionen AWS GovCloud (US-Ost) und AWS GovCloud (US-West) kopieren. Geben Sie für `--source-region` die AWS-Region der Quell-DB-Instance an.

Wenn `--source-region` nicht festgelegt ist, geben Sie einen `--pre-signed-url`-Wert an. Eine vorsignierte URL ist eine URL, die eine mit der Signaturversion 4 signierte Anforderung für den Befehl `copy-db-snapshot` enthält, die in der Quell- AWS-Region aufgerufen wird. Weitere Informationen zu dieser `pre-signed-url` Option finden Sie copy-db-snapshotin der AWS CLI Befehlsreferenz.

RDS-API

Sie können einen DB-Snapshot mithilfe der Amazon RDS-API-Operation Copy kopieren DBSnapshot. Wenn Sie den Snapshot in einen neuen kopieren AWS-Region, führen Sie die Aktion im neuen aus AWS-Region.

Zum Kopieren eines DB-Snapshots werden die folgenden Parameter verwendet. Nicht alle Parameter werden in allen Szenarien benötigt. Verwenden Sie die folgenden Beschreibungen und Beispiele, um die zu verwendenden Parameter zu ermitteln.

- `SourceDBSnapshotIdentifier` – Der Bezeichner des Quell-DB-Snapshots.
  - Wenn sich der Quell-Snapshot in derselben Datei AWS-Region wie die Kopie befindet, geben Sie eine gültige DB-Snapshot-ID an. Beispiel, `rds:mysql-instance1-snapshot-20130805`.
  - Wenn sich der Quell-Snapshot in derselben Datei AWS-Region wie die Kopie befindet und mit Ihrem geteilt wurde AWS-Konto, geben Sie einen gültigen DB-Snapshot-ARN an. Beispiel, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Wenn sich der Quell-Snapshot in einer anderen Version AWS-Region als der Kopie befindet, geben Sie einen gültigen DB-Snapshot-ARN an. Beispiel, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Wenn Sie aus einem freigegebenen manuellen DB-Snapshot kopieren, muss dieser Parameter der Amazon-Ressourcenname (ARN) des freigegebenen DB-Snapshots sein.

  - Wenn Sie einen verschlüsselten Snapshot kopieren, muss dieser Parameter das ARN-Format für die Quelle AWS-Region haben und mit dem `SourceDBSnapshotIdentifier` im `PreSignedUrl` Parameter übereinstimmen.
- `TargetDBSnapshotIdentifier` – Der Bezeichner für die neue Kopie des verschlüsselten DB-Snapshots.
- `CopyOptionGroup` – Legen Sie diesen Parameter auf `true` fest, um die Optionsgruppe aus einem geteilten Snapshots in die Kopie des Snapshots zu übernehmen. Der Standardwert ist `false`.
- `CopyTags` – Setzen Sie diesen Parameter auf `true`, damit Tags und Werte aus dem Snapshot in die Kopie des Snapshots übernommen werden. Der Standardwert ist `false`.
- `OptionGroupName` – Die Optionsgruppe, die der Kopie des Snapshots zugeordnet werden soll.

  Geben Sie diesen Parameter an, wenn Sie einen Snapshot von einem AWS-Region Snapshot in einen anderen kopieren und Ihre DB-Instance eine nicht standardmäßige Optionsgruppe verwendet.

  Damit die Quell-DB-Instance für Oracle oder Microsoft SQL Server die transparente Datenverschlüsselung verwendet, müssen Sie diesen Parameter beim regionsübergreifenden Kopieren angeben. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.
- `KmsKeyId` – Der KMS-Schlüsselbezeichner für einen verschlüsselten DB-Snapshot. Der KMS-Schlüsselbezeichner ist der Amazon Resource Name (ARN), der Schlüsselbezeichner oder der Schlüsselalias für den KMS-Schlüssel.
  - Wenn Sie einen verschlüsselten DB-Snapshot von Ihrem kopieren AWS-Konto, können Sie einen Wert für diesen Parameter angeben, um die Kopie mit einem neuen KMS-Schlüssel zu verschlüsseln. Wenn Sie keinen Wert für diesen Parameter angeben, wird die Kopie des DB-Snapshots mit demselben KMS-Verschlüsselungsschlüssel wie der Quell-DB-Snapshot verschlüsselt.
  - Wenn Sie einen verschlüsselten DB-Snapshot, der gemeinsam genutzt wird AWS-Konto, von einem anderen kopieren, müssen Sie einen Wert für diesen Parameter angeben.
  - Falls Sie diesen Parameter beim Kopieren eines unverschlüsselten Snapshots angeben, wird die Kopie verschlüsselt.
  - Wenn Sie einen verschlüsselten Snapshot in einen anderen kopieren AWS-Region, müssen Sie einen KMS-Schlüssel für das Ziel angeben AWS-Region. KMS-Schlüssel sind spezifisch für AWS-Region das, in dem sie erstellt wurden, und Sie können keine Verschlüsselungsschlüssel von einem AWS-Region Schlüssel zum anderen verwenden AWS-Region.

- `PreSignedUrl`— Die URL, die eine mit Signature Version 4 signierte Anfrage für den `CopyDBSnapshot` API-Vorgang in der Quelle enthält AWS-Region , die den zu kopierenden Quell-DB-Snapshot enthält.

  Geben Sie diesen Parameter an, wenn Sie mithilfe der Amazon RDS-API einen verschlüsselten DB-Snapshot AWS-Region von einem anderen kopieren. Sie können anstelle dieses Parameters die Option für die Quellregion angeben, wenn Sie einen verschlüsselten DB-Snapshot unter Verwendung der AWS-Region aus einer anderen AWS CLI kopieren.

  Die vorsignierte URL muss eine gültige Anforderung für die API-Operation `CopyDBSnapshot` sein, die in der Quell- AWS-Region ausgeführt werden kann, in der sich der zu kopierende verschlüsselte DB-Snapshot befindet. Die vorsignierte URL-Anforderung muss die folgenden Parameterwerte enthalten:

  - `DestinationRegion`— Das AWS-Region , in das der verschlüsselte DB-Snapshot kopiert wird. Dies AWS-Region ist derselbe, bei dem die `CopyDBSnapshot` Operation aufgerufen wird, die diese vorsignierte URL enthält.

    Nehmen wir beispielsweise an, Sie kopieren einen verschlüsselten DB-Snapshot aus der Region us-west-2 in die Region us-east-1. Sie rufen dann die Aktion `CopyDBSnapshot` in der Region us-east-1 auf und geben Sie eine vorsignierte URL an, die einen Aufruf an die Aktion `CopyDBSnapshot` in der Region us-west-2 enthält. In diesem Beispiel muss `DestinationRegion` in der vorsignierten URL auf die Region us-east-1 festgelegt werden.
  - `KmsKeyId` – Die KMS-Schlüsselkennung für den Schlüssel, der für die Verschlüsselung der Kopie des DB-Snapshots in der Ziel- AWS-Region verwendet werden soll. Dies ist derselbe Bezeichner sowohl für die `CopyDBSnapshot` Operation, die im Ziel aufgerufen wird, als auch für die Operation AWS-Region, die in der vorsignierten URL enthalten ist.
  - `SourceDBSnapshotIdentifier` – Der DB-Snapshot-Bezeichner des zu kopierenden verschlüsselten DB-Snapshots. Dieser Bezeichner muss im ARN-Format (Amazon-Ressourcenname) der Quell- AWS-Region angegeben werden. Wenn Sie beispielsweise einen verschlüsselten DB-Snapshot aus der Region us-west-2 kopieren, `SourceDBSnapshotIdentifier` sieht Ihr wie das folgende Beispiel aus:. `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115`

  Weitere Informationen zu mit Signature Version 4 signierte Anforderungen finden Sie nachstehend:

  - Authentifizieren von Anfragen: Verwenden von Abfrageparametern (AWS Signaturversion 4) in der Amazon Simple Storage Service API-Referenz
  - Signaturvorgang für Signaturversion 4 im Allgemeine AWS-Referenz

Example Unverschlüsselte Quelle in derselben Region

Der folgende Code erstellt eine Kopie eines Snapshots mit dem neuen Namen `mydbsnapshotcopy` im selben Format AWS-Region wie der Quell-Snapshot. Beim Erstellen der Kopie werden alle Tags des ursprünglichen Snapshots in die Snapshot-Kopie übernommen.

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &CopyTags=true
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=mysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example Unverschlüsselte Quelle in andere Region

Der folgende Code erstellt eine Kopie eines Snapshots mit dem neuen Namen `mydbsnapshotcopy` in der USA West (Nordkalifornien)-Region.

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-east-1%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example Verschlüsselte Quelle in andere Region

Der folgende Code erstellt eine Kopie eines Snapshots mit dem neuen Namen `mydbsnapshotcopy` in der US East (N. Virginia)-Region.

```
https://rds.us-east-1.amazonaws.com/
    ?Action=CopyDBSnapshot
    &KmsKeyId=my-us-east-1-key
    &OptionGroupName=custom-option-group-name
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCopyDBSnapshot
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBSnapshotIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%25253A123456789012%25253Asnapshot%25253Amysql-instance1-snapshot-20161115
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-west-2%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20161115
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Einschränkungen

Im folgenden werden einige Einschränkungen beim Kopieren von Snapshots aufgeführt:

- Sie können einen Snapshot nicht in die oder aus den China (Peking) oder der China (Ningxia)-Region kopieren.
- Sie können einen Snapshot zwischen AWS GovCloud (US-Ost) und AWS GovCloud (US-West) kopieren. Sie können jedoch keinen Snapshot zwischen diesen GovCloud (US-) Regionen und Regionen, die keine GovCloud (US-) Regionen sind, kopieren.
- Wenn Sie einen Quell-Snapshot löschen, bevor der Ziel-Snapshot verfügbar ist, kann das Kopieren des Snapshots fehlschlagen. Verifizieren Sie, dass der Ziel-Snapshot den Status `AVAILABLE` hat, bevor Sie einen Quell-Snapshot löschen.
- Pro Konto können bis zu 20 Snapshot-Kopieranforderungen an eine einzelne Zielregion aktiv sein.
- Wenn Sie mehrere Snapshot-Kopien für dieselbe Quell-DB-Instance anfordern, werden diese intern in die Warteschlange gestellt. Mit den später angeforderten Kopien wird erst begonnen, wenn die vorherigen Snapshot-Kopien fertiggestellt sind. Weitere Informationen finden Sie unter Warum ist die Erstellung meines EC2 AMI- oder EBS-Snapshots langsam? im AWS Knowledge Center.
- Je nach Umfang AWS-Regionen und Menge der zu kopierenden Daten kann eine regionsübergreifende Snapshot-Kopie Stunden in Anspruch nehmen. In einigen Fällen kann es eine große Anzahl von regionsübergreifenden Snapshot-Kopieranforderungen aus einer bestimmten -Region geben. In solchen Fällen stellt Amazon RDS neue regionsübergreifende Kopieranforderungen dieser -Quellregion gegebenenfalls in eine Warteschlange, bis aktive Kopiervorgänge abgeschlossen wurden. Zu Kopieranforderungen, die sich in der Warteschlange befinden, werden keine Fortschrittsinformationen angezeigt. Fortschrittsinformationen werden angezeigt, sobald der Kopiervorgang gestartet wird.
- Wenn eine Kopie noch aussteht, wenn Sie eine andere Kopie starten, wird die zweite Kopie nicht starten, bis nach Abschluss der ersten Kopie.
- Sie können keinen Snapshot eines Multi-AZ-DB-Clusters kopieren.
- Sie können nur Snapshots von DB-Instances, die io2-Volumes verwenden, dorthin kopieren, AWS-Regionen wo io2 Block Express-Volumes verfügbar sind. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.

Überlegungen

Hinweise zum Kopieren eines DB-Snapshots finden Sie in den folgenden Themen.

Themen

- Snapshot-Aufbewahrung
- Überlegungen zum Kopieren gemeinsam genutzter Snapshots

- Überlegungen zum Kopieren von Verschlüsselungssnapshots
- Überlegungen zum inkrementellen Kopieren von Snapshots
- Überlegungen zum regionsübergreifenden Kopieren von Snapshots
- Überlegungen zu Optionsgruppen
- Überlegungen zur Parametergruppe

Snapshot-Aufbewahrung

Amazon RDS löscht automatisierte Backups in verschiedenen Situationen:

- Am Ende des Aufbewahrungszeitraums.
- Wenn Sie automatisierte Backups für eine DB-Instance deaktivieren.
- Wenn Sie eine DB-Instance löschen.

Soll ein automatisches Backup länger aufbewahrt werden, erstellen Sie durch Kopieren einen manuellen Snapshot, der aufbewahrt wird, bis Sie ihn löschen. Amazon-RDS-Speicherkosten können für manuelle Snapshots anfallen, wenn sie Ihren Standardspeicherplatz überschreiten.

Weitere Information zu Sicherungsspeicherkosten finden Sie unter Amazon RDS – Preise.

Überlegungen zum Kopieren gemeinsam genutzter Snapshots

Sie können Snapshots kopieren, die von anderen AWS-Konten für Sie freigegeben wurden. In einigen Fällen können Sie einen verschlüsselten Snapshot, der von einem anderen AWS-Konto geteilt wurde, kopieren. In diesen Fällen müssen Sie Zugriff auf die Datei haben AWS KMS key , mit der der Snapshot verschlüsselt wurde.

> **Note**
>
> Amazon RDS-Speicherkosten fallen für geteilte Snapshots an, die Sie kopieren. Amazon RDS hängt möglicherweise den ARN der Quell-DB-Instance an den Snapshot an, den Sie kopiert haben.

Sie können einen gemeinsam genutzten DB-Snapshot kopieren, AWS-Regionen wenn der Snapshot unverschlüsselt ist. Ist der freigegebene DB-Snapshot jedoch verschlüsselt, können Sie ihn nur in dieselbe Region kopieren.

> **Note**
>
> Das Kopieren gemeinsam genutzter inkrementeller Snapshots in denselben AWS-Region wird unterstützt, wenn sie unverschlüsselt sind oder wenn sie mit demselben KMS-Schlüssel wie der ursprüngliche vollständige Snapshot verschlüsselt sind. Wenn Sie einen anderen KMS-Schlüssel verwenden, um nachfolgende Snapshots zu verschlüsseln, wenn sie kopiert werden, sind diese freigegebenen Snapshots vollständige Snapshots. Weitere Informationen finden Sie unter Überlegungen zum inkrementellen Kopieren von Snapshots.

Überlegungen zum Kopieren von Verschlüsselungssnapshots

Sie können einen Snapshot kopieren, der mit einem KMS-Schlüssel verschlüsselt wurde. Wenn Sie einen verschlüsselten Snapshot kopieren, muss auch die Kopie des Snapshots verschlüsselt werden. Wenn Sie einen verschlüsselten Snapshot innerhalb desselben kopieren AWS-Region, können Sie die Kopie mit demselben KMS-Schlüssel wie den ursprünglichen Snapshot verschlüsseln. Oder Sie können einen anderen KMS-Schlüssel angeben.

Wenn Sie einen verschlüsselten Snapshot in andere Regionen kopieren, müssen Sie einen gültigen KMS-Schlüssel für die Ziel- AWS-Region angeben. Es kann ein regionsspezifischer KMS-Schlüssel oder ein multiregionaler Schlüssel sein. Weitere Informationen über multiregionale Schlüssel finden Sie unter Verwenden von multiregionalen Schlüsseln in AWS KMS.

Der Quell-Snapshot bleibt den gesamten Kopiervorgang über verschlüsselt. Weitere Informationen finden Sie unter Einschränkungen von Amazon RDS-verschlüsselten DB-Instances.

Die Kopie eines unverschlüsselten Snapshots kann verschlüsselt werden. Auf diese Weise können Sie zuvor unverschlüsselte DB-Instances schnell verschlüsseln. Dazu können Sie einen Snapshot Ihrer DB-Instance erstellen, wenn Sie bereit sind, sie zu verschlüsseln. Sie erstellen dann eine Kopie dieses Snapshots und geben einen KMS-Schlüssel an, um diese Snapshot-Kopie zu verschlüsseln. Anschließend können Sie eine verschlüsselte DB-Instance aus dem verschlüsselten Snapshot wiederherstellen.

> **Important**
>
> Wenn Sie explizite Deny-Statements für alle Ressourcen (*) in AWS KMS wichtigen Richtlinien mit verwalteten Services wie Amazon RDS verwenden, müssen Sie eine Bedingung angeben, um das Konto, das die Ressource besitzt, zuzulassen. Ohne diese

Bedingung kann der Kopiervorgang fehlschlagen, auch wenn die Ablehnungsregel Ausnahmen für Ihren IAM-Benutzer enthält.

Weitere Informationen zur AWS KMS Schlüsselverwaltung für Amazon RDS finden Sie unterAWS KMS key Verwaltung.

Überlegungen zum inkrementellen Kopieren von Snapshots

Ein inkrementeller Snapshot enthält nur die Daten, die sich nach dem letzten Snapshot derselben DB-Instance geändert haben. Das inkrementelle Kopieren von Snapshots ist schneller und verursacht geringere Speicherkosten als das vollständige Kopieren von Snapshots.

Ob eine Snapshot-Kopie inkrementell ist, hängt von der zuletzt abgeschlossenen Snapshot-Kopie und dem Quell-Snapshot ab. Wenn die letzte Snapshot-Kopie gelöscht wurde, ist die nächste Kopie eine vollständige Kopie und keine inkrementelle Kopie. Eine Snapshot-Kopie entspricht dem Typ des Quell-Snapshots. Wenn es sich bei dem Quell-Snapshot um einen inkrementellen Snapshot handelt, handelt es sich bei der Snapshot-Kopie um einen inkrementellen Snapshot.

Wenn Sie einen Snapshot kopieren AWS-Konten, ist die Kopie nur dann eine inkrementelle Kopie, wenn alle der folgenden Bedingungen erfüllt sind:

- Die neueste Snapshot-Kopie stammt von derselben Quell-DB-Instance und ist immer noch im Zielkonto vorhanden.
- Alle Kopien des Snapshots im Zielkonto sind entweder unverschlüsselt oder wurden mit demselben KMS-Schlüssel verschlüsselt.
- Wenn es sich bei der Quell-DB-Instance um eine Multi-AZ-Instance handelt, hat sie seit der Erstellung des letzten Snapshots von dieser Instance kein Failover auf eine andere AZ durchgeführt.

Die folgenden Beispiele veranschaulichen den Unterschied zwischen vollständigen und inkrementellen Snapshots. Sie gelten sowohl für freigegebene als auch für nicht freigegebene Snapshots.

Snapshot	Verschlüsselungsschlüssel	Vollständig oder inkrementell
S1	K1	Voll

Snapshot	Verschlüsselungsschlüssel	Vollständig oder inkrementell
S2	K1	Inkrementelle von S1
S3	K1	Inkrementelle von S2
S4	K1	Inkrementelle von S3
Kopie von S1 (S1C)	K2	Voll
Kopie von S2 (S2C)	K3	Voll
Kopie von S3 (S3C)	K3	Inkrementelle von S2C
Kopie von S4 (S4C)	K3	Inkrementelle von S3C
Kopie 2 von S4 (S4C2)	K4	Voll

> **Note**
>
> In diesen Beispielen sind Snapshots S2, S3 und S4 nur inkrementell, wenn der vorherige Snapshot noch vorhanden ist.
> Gleiches gilt für Kopien. Snapshot-Kopien S3C und S4C sind nur inkrementell, wenn die vorherige Kopie noch vorhanden ist.

Informationen zum Kopieren inkrementeller Snapshots zwischen den anderen finden Sie AWS-Regionen unter. Vollständige und inkrementelle Kopien

Überlegungen zum regionsübergreifenden Kopieren von Snapshots

Sie können DB-Snapshots über AWS-Regionen hinweg kopieren. Es gibt jedoch bestimmte Einschränkungen und Überlegungen für das Kopieren von regionsübergreifenden Snapshots.

Anfordern einer Regionsübergreifende Kopie für DB-Snapshots

Um mit der Quellregion zu kommunizieren, um eine regionsübergreifende DB-Snapshot-Kopie anzufordern, muss der Anforderer (IAM-Rolle oder IAM-Benutzer) Zugriff auf den Quell-DB-Snapshot und die Quellregion haben.

Bestimmte Bedingungen in der IAM-Richtlinie des Anforderers können dazu führen, dass die Anfrage fehlschlägt. In den folgenden Beispielen wird davon ausgegangen, dass Sie den DB-Snapshot von USA Ost (Ohio) zu US East (N. Virginia) kopieren. Diese Beispiele zeigen die Bedingungen in der IAM-Richtlinie des Anforderers, die dazu führen, dass die Anfrage fehlschlägt:

- Die Richtlinie des Anforderers hat eine Bedingung für `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

  Die Anfrage schlägt fehl, weil die Richtlinie den Zugriff auf die Quellregion nicht zulässt. Für eine erfolgreiche Anfrage geben Sie sowohl die Quell- als auch die Zielregion an.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- Die Richtlinie des Anforderers erlaubt keinen Zugriff auf den Quell-DB-Snapshot.

```
...
```

```
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot"
...
```

Für eine erfolgreiche Anfrage geben Sie sowohl die Quell- als auch die Ziel-Snapshots an.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot",
    "arn:aws:rds:us-east-2:123456789012:snapshot:source-snapshot"
]
...
```

- Die Richtlinie des Anforderers lehnt a `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

Die Kommunikation mit der Quellregion erfolgt über RDS im Namen des Anforderers. Lehnen Sie für eine erfolgreiche Anfrage keine Anrufe von AWS Diensten ab.

- Die Richtlinie des Anforderers hat eine Bedingung für `aws:SourceVpc` oder `aws:SourceVpce`.

Diese Anforderungen können fehlschlagen, da RDS den Aufruf an die entfernte Region nicht vom angegebenen VPC- oder VPC-Endpunkt ausführt.

Wenn Sie eine der vorherigen Bedingungen verwenden müssen, die dazu führen würde, dass eine Anfrage fehlschlägt, können Sie eine zweite Anweisung mit `aws:CalledVia` in Ihrer Richtlinie aufnehmen, um die Anfrage erfolgreich zu machen. Zum Beispiel können Sie `aws:CalledVia` mit `aws:SourceVpce` wie hier gezeigt verwenden:

```
...
"Effect": "Allow",
```

```
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CopyDBSnapshot"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Weitere Informationen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.

Autorisieren der Snapshot-Kopie

Nachdem eine regionsübergreifenden DB-Snapshot-Kopieranforderung `success` ausgibt, startet RDS die Kopie im Hintergrund. Es wird eine Autorisierung für RDS für den Zugriff auf den Quell-Snapshot erstellt. Diese Autorisierung verknüpft den Quell-DB-Snapshot mit dem Ziel-DB-Snapshot und ermöglicht es RDS, nur in den angegebenen Ziel-Snapshot zu kopieren.

Die Autorisierung wird von RDS unter Verwendung der `rds:CrossRegionCommunication`-Berechtigung in der serviceverknüpfte IAM-Rolle verifiziert. Wenn die Kopie autorisiert ist, kommuniziert RDS mit der Quellregion und schließt die Kopie ab.

RDS hat keinen Zugriff auf DB-Snapshots, die zuvor nicht von einer `CopyDBSnapshot`-Anfrage authorisiert wurden. Die Autorisierung wird widerrufen, wenn der Kopiervorgang abgeschlossen ist.

RDS verwendet die serviceverknüpfte Rolle, um die Autorisierung in der Quellregion zu überprüfen. Wenn Sie die serviceverknüpfte Rolle während des Kopiervorgangs löschen, schlägt die Kopie fehl.

Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch.

AWS Security Token Service Anmeldeinformationen verwenden

Sitzungstoken vom Endpunkt global AWS Security Token Service (AWS STS) sind nur in Bereichen gültig AWS-Regionen , die standardmäßig aktiviert sind (kommerzielle Regionen). Wenn Sie Anmeldeinformationen aus dem `assumeRole` API-Vorgang in verwenden AWS STS, verwenden Sie den regionalen Endpunkt, wenn es sich bei der Quellregion um eine Opt-in-Region handelt. Andernfalls schlägt die Anforderung fehl. Dies liegt daran, dass Ihre Anmeldeinformationen in beiden Regionen gültig sein müssen. Dies gilt nur für Opt-in-Regionen, wenn der regionale AWS STS Endpunkt verwendet wird.

Um den globalen Endpunkt zu verwenden, stellen Sie sicher, dass er für beide Regionen in den Vorgängen aktiviert ist. Stellen Sie den globalen Endpunkt `Valid in all AWS-Regionen` in den AWS STS Kontoeinstellungen auf ein.

Die gleiche Regel gilt für Anmeldeinformationen im vorsignierten URL-Parameter.

Weitere Informationen finden Sie unter AWS STS in an verwalten AWS-Region im IAM-Benutzerhandbuch.

Latenz- und mehrfache Kopieranfragen

Je nach Umfang und AWS-Regionen Menge der zu kopierenden Daten kann es Stunden dauern, bis eine regionsübergreifende Snapshot-Kopie abgeschlossen ist.

In einigen Fällen kann es eine große Anzahl von regionsübergreifenden Snapshot-Kopieranforderungen aus einer bestimmten AWS-Region geben. In solchen Fällen kann Amazon RDS neue regionsübergreifende Kopieranfragen von dieser Quelle AWS-Region in eine Warteschlange stellen, bis einige in Bearbeitung befindliche Kopien abgeschlossen sind. Zu Kopieranforderungen, die sich in der Warteschlange befinden, werden keine Fortschrittsinformationen angezeigt. Fortschrittsinformationen werden angezeigt, sobald das Kopieren beginnt.

Vollständige und inkrementelle Kopien

Wenn Sie einen Snapshot in einen anderen AWS-Region Snapshot als den Quell-Snapshot kopieren, ist die erste Kopie eine vollständige Snapshot-Kopie, auch wenn Sie einen inkrementellen Snapshot kopieren. Eine vollständige Snapshot-Kopie enthält alle Daten und Metadaten, die zur Wiederherstellung der DB-Instance erforderlich sind. Nach der ersten Snapshot-Kopie können Sie inkrementelle Snapshots derselben DB-Instance in dieselbe Zielregion innerhalb derselben kopieren.

AWS-Konto Weitere Informationen zu inkrementellen Snapshots finden Sie unter Überlegungen zum inkrementellen Kopieren von Snapshots.

Das inkrementelle Kopieren von Snapshots AWS-Regionen wird sowohl für unverschlüsselte als auch für verschlüsselte Snapshots unterstützt.

Wenn Sie einen Snapshot kopieren AWS-Regionen, handelt es sich bei der Kopie um eine inkrementelle Kopie, sofern die folgenden Bedingungen erfüllt sind:

- Der Snapshot wurde zuvor in die Zielregion kopiert.
- Die aktuelle Snapshot-Kopie existiert noch in der Zielregion.
- Alle Kopien des Snapshots in der Zielregion sind entweder unverschlüsselt oder wurden mit demselben KMS-Schlüssel verschlüsselt.

Überlegungen zu Optionsgruppen

DB-Optionsgruppen sind spezifisch für AWS-Region das, in dem sie erstellt wurden, und Sie können keine Optionsgruppe von einer Optionsgruppe AWS-Region in einer anderen verwenden AWS-Region.

Bei Oracle-Datenbanken können Sie die AWS CLI oder RDS-API verwenden, um die benutzerdefinierte DB-Optionsgruppe aus einem Snapshot zu kopieren, der für Sie freigegeben wurde AWS-Konto. Sie können nur Optionsgruppen innerhalb derselben AWS-Region kopieren. Die Optionsgruppe wird nicht kopiert, wenn sie bereits in das Zielkonto kopiert wurde und seit dem Kopieren keine Änderungen daran vorgenommen wurden. Wenn die Quelloptionsgruppe bereits kopiert wurde, sich aber seit dem Kopieren geändert hat, kopiert RDS die neue Version in das Zielkonto. Standardoptionsgruppen werden nicht kopiert.

Wenn Sie einen Snapshot regionsübergreifend kopieren, können Sie eine neue Optionsgruppe für den Snapshot angeben. Wir empfehlen, die neue Optionsgruppe vorzubereiten, bevor Sie den Snapshot kopieren. Erstellen Sie im Ziel AWS-Region eine Optionsgruppe mit denselben Einstellungen wie die ursprüngliche DB-Instance. Wenn in der neuen Version bereits eine vorhanden ist AWS-Region, können Sie diese verwenden.

In einigen Fällen können Sie einen Snapshot kopieren und keine neue Optionsgruppe für den Snapshot angeben. In diesen Fällen erhält die DB-Instance bei der Wiederherstellung des Snapshots die standardmäßige Optionsgruppe. Soll die neue DB-Instance die gleichen Optionen wie das Original erhalten, gehen Sie folgendermaßen vor:

1. Erstellen Sie im Ziel AWS-Region eine Optionsgruppe mit denselben Einstellungen wie in der ursprünglichen DB-Instance. Wenn in der neuen Version bereits eine vorhanden ist AWS-Region, können Sie diese verwenden.
2. Nachdem Sie den Snapshot im Ziel wiederhergestellt haben AWS-Region, ändern Sie die neue DB-Instance und fügen Sie die neue oder vorhandene Optionsgruppe aus dem vorherigen Schritt hinzu.

Überlegungen zur Parametergruppe

Wenn Sie einen Snapshot regionsübergreifend kopieren, enthält die Kopie nicht die von der ursprünglichen DB-Instance verwendete Parametergruppe. Wenn Sie einen Snapshot wiederherstellen, um eine neue DB-Instance zu erstellen, erhält diese DB-Instance die Standardparametergruppe für die, in der AWS-Region sie erstellt wurde. Soll die neue DB-Instance die gleichen Parameter wie das Original erhalten, gehen Sie folgendermaßen vor:

1. Erstellen Sie im Ziel AWS-Region eine DB-Parametergruppe mit denselben Einstellungen wie die ursprüngliche DB-Instance. Wenn in der neuen Version bereits eine vorhanden ist AWS-Region, können Sie diese verwenden.
2. Nachdem Sie den Snapshot im Ziel wiederhergestellt haben AWS-Region, ändern Sie die neue DB-Instance und fügen Sie die neue oder vorhandene Parametergruppe aus dem vorherigen Schritt hinzu.

Einen DB-Snapshot für Amazon teilen RDS

Mit Amazon RDS können Sie einen manuellen DB-Snapshot auf folgende Weise teilen:

- Durch die gemeinsame Nutzung eines manuellen DB-Snapshots, ob verschlüsselt oder unverschlüsselt, können autorisierte AWS-Konten Personen den Snapshot kopieren.
- Durch die gemeinsame Nutzung eines unverschlüsselten manuellen DB-Snapshots können autorisierte AWS-Konten Personen eine DB-Instance direkt aus dem Snapshot wiederherstellen, anstatt eine Kopie davon zu erstellen und von dort aus wiederherzustellen. Sie können eine DB-Instance jedoch nicht aus einem DB-Snapshot wiederherstellen, der zugleich freigegeben und verschlüsselt ist. Stattdessen können Sie eine Kopie des DB-Snapshots erstellen und die DB-Instance aus der Kopie wiederherstellen.

> **Note**
>
> Wenn Sie einen automatisierten DB-Snapshot freigeben möchten, erstellen Sie einen manuellen DB-Snapshot, indem Sie den automatisierten Snapshot kopieren und diese Kopie dann freigeben. Dieser Prozess gilt auch für durch AWS Backup generierte Ressourcen.

Weitere Informationen zum Kopieren von Snapshots finden Sie unter Kopieren eines DB-Snapshots für Amazon RDS. Weitere Informationen zum Wiederherstellen einer DB-Instance aus einem DB-Snapshot finden Sie unter Wiederherstellung auf einer DB-Instance.

Sie können einen manuellen Snapshot mit bis zu 20 anderen teilen. AWS-Konten

Beim Teilen manueller Schnappschüsse mit anderen AWS-Konten gelten die folgenden Einschränkungen:

- Wenn Sie eine DB-Instance aus einem gemeinsam genutzten Snapshot mithilfe von AWS Command Line Interface (AWS CLI) oder Amazon wiederherstellen RDSAPI, müssen Sie den Amazon-Ressourcennamen (ARN) des gemeinsam genutzten Snapshots als Snapshot-ID angeben.
- Sie können keinen DB-Snapshot freigeben, der eine Optionsgruppe mit permanenten oder dauerhaften Optionen verwendet, mit Ausnahme von Oracle DB-Instances, die über die `Timezone` Option `OLS` oder (oder beides) verfügen.

Eine permanente Option kann nicht aus einer Optionsgruppe entfernt werden. Optionsgruppen mit persistenten Optionen können nicht aus einer DB-Instance entfernt werden, nachdem die Optionsgruppe der DB-Instance zugewiesen wurde.

Die folgende Tabelle listet die permanenten und persistenten Optionen und die entsprechenden DB-Engines auf.

Optionsname	Persistent	Permanent	DB-Engine
TDE	Ja	Nein	Microsoft SQL Server Enterprise Edition
TDE	Ja	Ja	Oracle Enterprise Edition
Zeitzone	Ja	Ja	Oracle Enterprise Edition Oracle Standard Edition Oracle Standard Edition One Oracle Standard Ausgabe 2

Bei Oracle-DB-Instances können Sie gemeinsam genutzte DB-Snapshots mit der Option `Timezone` oder `OLS` (oder beiden) kopieren. Geben Sie dazu eine Zieloptionengruppe an, die diese Optionen enthält, wenn Sie den DB-Snapshot kopieren. Die OLS Option ist permanent und persistent nur für Oracle-DB-Instances, auf denen Oracle Version 12.2 oder höher ausgeführt wird. Weitere Informationen zu diesen Optionen finden Sie unter Oracle-Zeitzone und Oracle Label Security.

- Sie können einen Snapshot eines Multi-AZ-DB-Clusters nicht gemeinsam nutzen.

In den folgenden Themen finden Sie Informationen zum Teilen von öffentlichen Snapshots, zum Teilen verschlüsselter Snapshots und zum Beenden der Snapshot-Freigabe.

Themen

- Öffentliche Schnappschüsse für Amazon teilen RDS

- Verschlüsselte Snapshots für Amazon RDS teilen
- Snapshot-Sharing für Amazon beenden RDS

Freigeben eines Snapshots

Sie können einen DB-Snapshot mit dem AWS Management Console, dem oder dem AWS CLI teilen. RDS API

Konsole

Mithilfe der RDS Amazon-Konsole können Sie einen manuellen DB-Snapshot mit bis zu 20 Personen teilen AWS-Konten. Sie können die Konsole auch verwenden, um die Freigabe eines manuellen Snapshots für ein oder mehrere Konten zu beenden.

So teilen Sie einen manuellen DB-Snapshot mithilfe der RDS Amazon-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den manuellen Snapshot, den Sie freigeben möchten.
4. Wählen Sie unter Actions (Aktionen) die Option Share Snapshots (Snapshot freigeben).
5. Wählen Sie für DB Snapshot Visibility (Sichtbarkeit des DB-Snapshots) eine der folgenden Optionen.

   - Wenn die Quelle unverschlüsselt ist, wählen Sie Öffentlich, damit alle AWS Konten eine DB-Instance aus Ihrem manuellen DB-Snapshot wiederherstellen können, oder wählen Sie Privat, um nur den von Ihnen angegebenen Benutzern AWS-Konten die Wiederherstellung einer DB-Instance aus Ihrem manuellen DB-Snapshot zu gestatten.

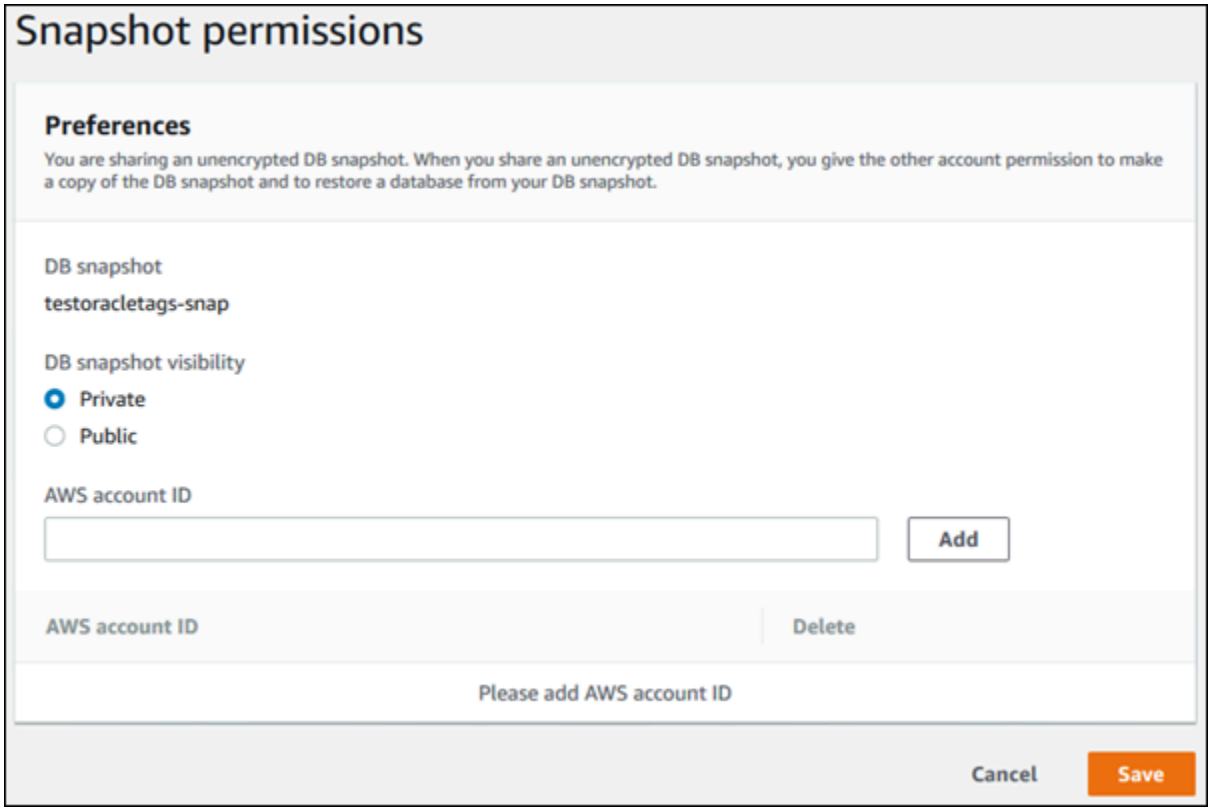
     > **Warning**
     >
     > Wenn Sie die Sichtbarkeit von DB-Snapshots auf Öffentlich setzen, AWS-Konten können alle eine DB-Instance aus Ihrem manuellen DB-Snapshot wiederherstellen und haben Zugriff auf Ihre Daten. Geben Sie keine manuellen DB-Snapshots als Public (Öffentlich) frei, die private Informationen enthalten.
     > Weitere Informationen finden Sie unter Öffentliche Schnappschüsse für Amazon teilen RDS.

- Ist die Quelle verschlüsselt, ist DB Snapshot Visibility (Sichtbarkeit des DB-Snapshots) auf Private (Privat) festgelegt, da verschlüsselte Snapshots nicht als öffentlich freigegeben werden können.

> **Note**
>
> Snapshots, die mit der Standardeinstellung verschlüsselt wurden, AWS KMS key können nicht geteilt werden. Informationen zur Umgehung dieses Problems finden Sie unterVerschlüsselte Snapshots für Amazon RDS teilen.

6. Geben Sie AWS unter Konto-ID die AWS-Konto Kennung für ein Konto ein, dem Sie die Wiederherstellung einer DB-Instance aus Ihrem manuellen Snapshot erlauben möchten, und wählen Sie dann Hinzufügen aus. Wiederholen Sie den Vorgang, um weitere AWS-Konto Identifikatoren (bis zu 20 AWS-Konten) hinzuzufügen.

   Wenn Ihnen beim Hinzufügen einer AWS-Konto Kennung zur Liste der zulässigen Konten ein Fehler unterläuft, können Sie sie aus der Liste löschen, indem Sie rechts neben der falschen AWS-Konto Kennung die Option Löschen wählen.



7. Nachdem Sie Kennungen für alle hinzugefügt haben AWS-Konten , denen Sie die Wiederherstellung des manuellen Snapshots erlauben möchten, wählen Sie Speichern, um Ihre Änderungen zu speichern.

AWS CLI

Verwenden Sie den Befehl `aws rds modify-db-snapshot-attribute`, um einen DB-Snapshot freizugeben. Verwenden Sie den `--values-to-add` Parameter, um eine Liste der Personen IDs hinzuzufügen AWS-Konten , die berechtigt sind, den manuellen Snapshot wiederherzustellen.

Example einen Snapshot mit einem einzigen Konto teilen

Im folgenden Beispiel wird AWS-Konto Identifier aktiviert`123456789012`, um den angegebenen DB-Snapshot wiederherzustellen`db7-snapshot`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier db7-snapshot \
--attribute-name restore \
--values-to-add 123456789012
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier db7-snapshot ^
--attribute-name restore ^
--values-to-add 123456789012
```

Example einen Snapshot mit mehreren Konten teilen

Im folgenden Beispiel werden zwei AWS-Konto Identifikatoren aktiviert, `111122223333` und`444455556666`, um den genannten `manual-snapshot1` DB-Snapshot wiederherzustellen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
```

```
--values-to-add {"111122223333","444455556666"}
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-add "[\"111122223333\",\"444455556666\"]"
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

Verwenden Sie den Befehl, um die AWS-Konten Optionen aufzulisten, die zur Wiederherstellung eines Snapshots aktiviert sind. `describe-db-snapshot-attributes` AWS CLI

RDS API

Sie können einen manuellen DB-Snapshot auch mit anderen teilen, AWS-Konten indem Sie Amazon verwenden RDSAPI. Rufen Sie dazu die Operation `ModifyDBSnapshotAttribute` auf. Geben Sie `restore` für an`AttributeName`, und verwenden Sie den `ValuesToAdd` Parameter, um eine Liste der IDs Personen hinzuzufügen AWS-Konten , die berechtigt sind, den manuellen Snapshot wiederherzustellen.

Um einen manuellen Snapshot öffentlich und für alle wiederherstellbar zu machen AWS-Konten, verwenden Sie den Wert`all`. Achten Sie jedoch darauf, den `all` Wert nicht für manuelle Snapshots hinzuzufügen, die private Informationen enthalten, die nicht für alle verfügbar sein sollen. AWS-Konten Darüber hinaus sollten Sie `all` nicht für verschlüsselte Snapshots angeben, da die öffentliche Bereitstellung dieser Snapshots nicht unterstützt wird.

Verwenden Sie den `DescribeDBSnapshotAttributes`APIVorgang, um alle Dateien aufzulisten, die zur Wiederherstellung eines Snapshots AWS-Konten berechtigt sind.

Öffentliche Schnappschüsse für Amazon teilen RDS

Sie können einen unverschlüsselten manuellen Snapshot als öffentlich freigeben, sodass der Snapshot für alle AWS-Konten verfügbar ist. Achten Sie bei der Freigabe eines Snapshots als öffentlich darauf, dass in Ihren öffentlichen Snapshots keine privaten Informationen enthalten sind.

Wenn ein Snapshot öffentlich geteilt wird, gibt er allen die AWS-Konten Erlaubnis, den Snapshot zu kopieren und daraus DB-Instances zu erstellen.

Die Speicherung von Backups öffentlicher Snapshots anderer Konten wird Ihnen nicht in Rechnung gestellt. Ihnen werden nur Ihre eigenen Snapshots berechnet.

Wenn Sie einen öffentlichen Snapshot kopieren, sind Sie der Eigentümer der Kopie. Ihnen wird die Speicherung von Backups Ihrer Snapshot-Kopie in Rechnung gestellt. Wenn Sie eine DB-Instance aus einem öffentlichen Snapshot erstellen, wird Ihnen diese DB-Instance in Rechnung gestellt. RDSPreisinformationen von Amazon finden Sie auf der RDSAmazon-Produktseite.

Sie können nur Ihre eigenen öffentlichen Snapshots löschen. Um einen geteilten oder öffentlichen Snapshot zu löschen, stellen Sie sicher, dass Sie sich bei demjenigen anmelden AWS-Konto , dem der Snapshot gehört.

Öffentliche Schnappschüsse anzeigen, die anderen gehören AWS-Konten

Sie können öffentliche Snapshots, die anderen Konten in einer bestimmten AWS Region gehören, auf der Seite Snapshots in der RDS Amazon-Konsole auf der Registerkarte Öffentlich anzeigen. Ihre Snapshots (die Ihrem Konto gehören) werden auf dieser Registerkarte nicht angezeigt.

So rufen Sie öffentliche Snapshots auf

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie die Registerkarte Public (Öffentlich).

   Die öffentlichen Snapshots werden angezeigt. Sie können in der Spalte Owner (Eigentümer) sehen, welches Konto einen öffentlichen Snapshot besitzt.

   > **Note**
   >
   > Möglicherweise müssen Sie die Seiteneinstellungen ändern, indem Sie das Zahnradsymbol oben rechts im Bereich Public Snapshots (Öffentliche Snapshots aufrufen, damit diese Spalte eingeblendet wird.

Aufrufen Ihrer eigenen öffentlichen Snapshots

Sie können den folgenden AWS CLI Befehl (nur Unix) verwenden, um die öffentlichen Snapshots anzuzeigen, die Ihnen AWS-Konto in einer bestimmten AWS Region gehören.

```
aws rds describe-db-snapshots --snapshot-type public --include-public |
 grep account_number
```

Die zurückgegebene Ausgabe ähnelt bei öffentlichen Snapshots dem folgenden Beispiel.

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot1",
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot2",
```

> **Note**
>
> Möglicherweise sehen Sie doppelte Einträge für `DBSnapshotIdentifier` oder `SourceDBSnapshotIdentifier`.

Teilen von öffentlichen Snapshots aus veralteten DB-Engine-Versionen

Das Wiederherstellen oder Kopieren von öffentlichen Snapshots aus veralteten DB-Engine-Versionen wird nicht unterstützt.

Die DB-Engines RDS für Oracle und RDS Postgre unterstützen das direkte Upgrade von SQL DB-Snapshot-Engine-Versionen. Sie können Ihre Snapshots aktualisieren und sie dann erneut öffentlich teilen. Weitere Informationen finden Sie hier:

- Aktualisieren eines Oracle-DB-Snapshots
- Aktualisierung einer SQL Postgre-DB-Snapshot-Engine-Version

Führen Sie für andere DB-Engines die folgenden Schritte aus, um Ihren vorhandenen, nicht unterstützten öffentlichen Snapshot zum Wiederherstellen oder Kopieren verfügbar zu machen:

1. Markieren Sie den Snapshot als privat.
2. Stellen Sie den Snapshot wieder her.
3. Führen Sie ein Upgrade der wiederhergestellten DB-Instance auf eine unterstützte Engine-Version durch.

4. Erstellen Sie einen neuen Snapshot.
5. Teilen Sie den Snapshot erneut öffentlich.

Verschlüsselte Snapshots für Amazon RDS teilen

Sie können DB-Snapshots freigeben, die während der Speicherung mittels des AES-256-Verschlüsselungsalgorithmus verschlüsselt wurden, wie in Verschlüsseln von Amazon RDS-Ressourcen beschrieben.

Die folgenden Einschränkungen gelten für die Freigabe verschlüsselter Snapshots:

- Sie können verschlüsselte Snapshots nicht als öffentlich freigeben.
- Sie können keine Oracle- oder Microsoft SQL Server-Snapshots freigeben, die mittels Transparent Data Encryption (TDE) verschlüsselt sind.
- Sie können keinen Snapshot teilen, der mit dem Standard-KMS-Schlüssel desjenigen verschlüsselt wurde AWS-Konto , der den Snapshot geteilt hat.

  Weitere Informationen zur AWS KMS Schlüsselverwaltung für Amazon RDS finden Sie unterAWS KMS key Verwaltung.

Um das Problem mit dem standardmäßigen KMS-Schlüssel zu umgehen, führen Sie die folgenden Aufgaben aus:

1. Erstellen Sie einen vom Kunden verwalteten Schlüssel und gewähren Sie Zugriff darauf.
2. Kopieren Sie den Snapshot aus dem Quellkonto und teilen Sie ihn.
3. Kopieren Sie den gemeinsam genutzten Snapshot in das Zielkonto.

Erstellen Sie einen vom Kunden verwalteten Schlüssel und gewähren Sie Zugriff darauf

Zuerst erstellen Sie einen benutzerdefinierten KMS-Schlüssel im selben Format AWS-Region wie der verschlüsselte DB-Snapshot. Bei der Erstellung des vom Kunden verwalteten Schlüssels gewähren Sie einem anderen Benutzer Zugriff darauf AWS-Konto.

> **Note**
>
> Sie können auch einen KMS-Schlüssel von einem anderen AWS Konto verwenden, wenn die Schlüsselrichtlinie Zugriff auf die Quell- und Zielkonten gewährt.

Um einen vom Kunden verwalteten Schlüssel zu erstellen und Zugriff darauf zu gewähren

1. Melden Sie sich AWS Management Console von der Quelle aus bei an AWS-Konto.
2. Öffnen Sie die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
3. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im Navigationsbereich auf Kundenverwaltete Schlüssel.
5. Klicken Sie auf Create key.
6. Gehen Sie auf der Seite „Schlüssel konfigurieren“ wie folgt vor:
    a. Wählen Sie als Schlüsseltyp die Option Symmetrisch aus.
    b. Wählen Sie unter Schlüsselverwendung die Option Verschlüsseln und entschlüsseln aus.
    c. Erweitern Sie Advanced options (Erweiterte Optionen).
    d. Wählen Sie für Herkunft des Schlüsselmaterials die Option KMS aus.
    e. Wählen Sie für Regionalität die Option Single Region Key aus.
    f. Wählen Sie Weiter aus.
7. Gehen Sie auf der Seite Labels hinzufügen wie folgt vor:
    a. Geben Sie für Alias einen Anzeigenamen für Ihren KMS-Schlüssel ein, z. **`share-snapshot`** B.
    b. (Optional) Geben Sie eine Beschreibung für Ihren KMS-Schlüssel ein.
    c. (Optional) Fügen Sie Ihrem KMS-Schlüssel Tags hinzu.
    d. Wählen Sie Weiter aus.
8. Klicken Sie auf der Seite Definieren wichtiger administrativer Berechtigungen auf Weiter.
9. Gehen Sie auf der Seite Schlüsselverwendungsberechtigungen definieren wie folgt vor:
    a. Wählen Sie für Andere AWS-Konten die Option Weitere hinzufügen aus AWS-Konto.
    b. Geben Sie die ID der Datei ein, AWS-Konto auf die Sie Zugriff gewähren möchten.

Sie können mehreren Zugriff gewähren AWS-Konten.

c. Wählen Sie Weiter aus.

10. Überprüfen Sie Ihren KMS-Schlüssel und wählen Sie dann Fertig stellen.

Kopieren Sie den Snapshot aus dem Quellkonto und teilen Sie ihn

Als Nächstes kopieren Sie den Quell-DB-Snapshot mithilfe des vom Kunden verwalteten Schlüssels in einen neuen Snapshot. Dann teilen Sie ihn mit dem Ziel AWS-Konto.

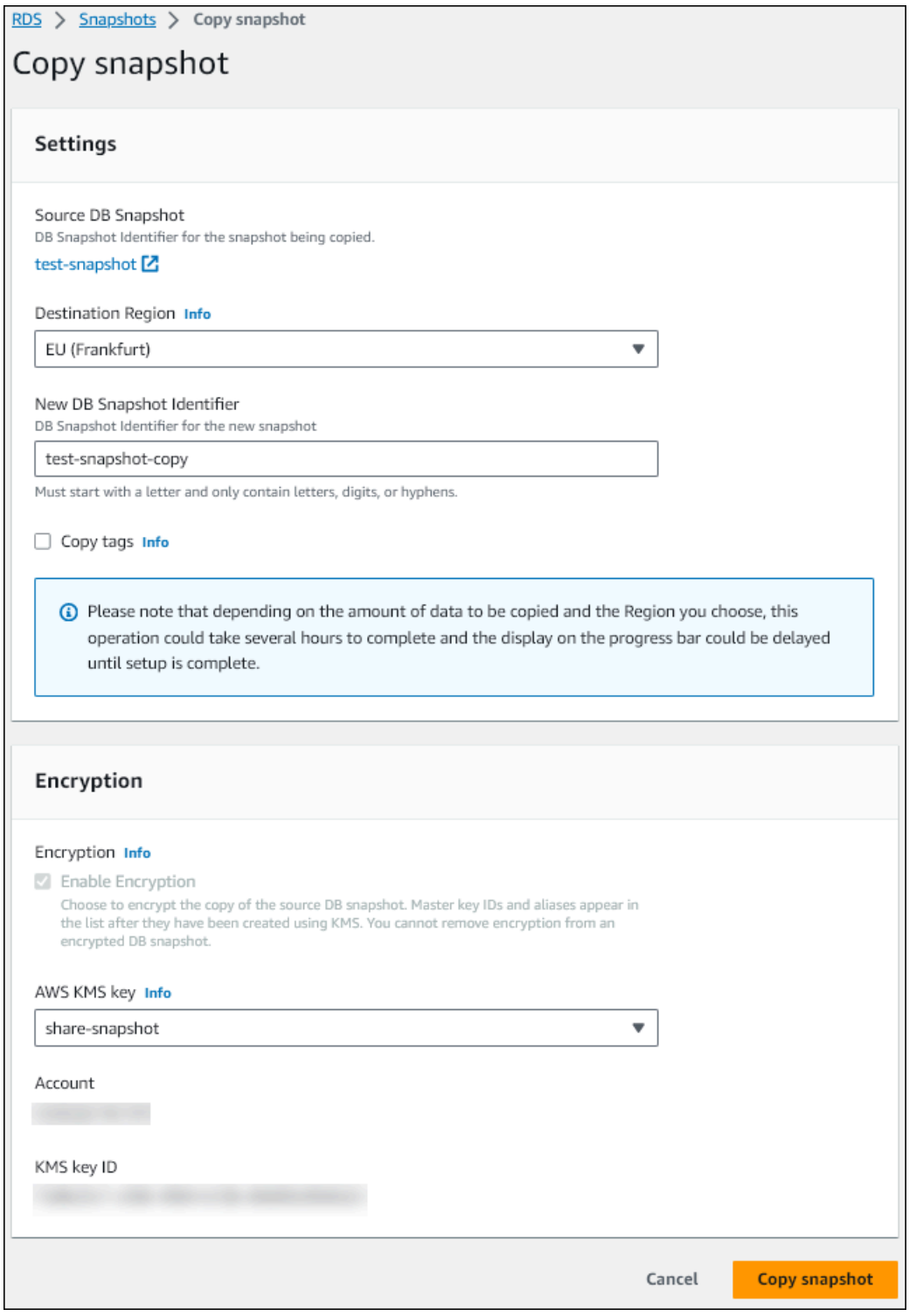
Um den Snapshot zu kopieren und zu teilen

1. Melden Sie sich bei der AWS Management Console von der Quelle aus an AWS-Konto.
2. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/
3. Wählen Sie im Navigationsbereich die Option Snapshots.
4. Wählen Sie den DB-Snapshot aus, den Sie kopieren möchten.
5. Wählen Sie für Actions (Aktionen) die Option Copy Snapshot (Snapshot kopieren).
6. Gehen Sie auf der Seite Snapshot kopieren wie folgt vor:

   a. Wählen Sie als Zielregion die Region aus, AWS-Region in der Sie den vom Kunden verwalteten Schlüssel im vorherigen Verfahren erstellt haben.

   b. Geben Sie den Namen der DB-Snapshot-Kopie in das Feld Neuer DB-Snapshot-Identifier ein.

   c. Wählen Sie für AWS KMS keyden vom Kunden verwalteten Schlüssel aus, den Sie erstellt haben.



d. Wählen Sie Copy Snapshot (Snapshot kopieren) aus.

7. Wenn die Snapshot-Kopie verfügbar ist, wählen Sie sie aus.
8. Wählen Sie unter Actions (Aktionen) die Option Share Snapshots (Snapshot freigeben).
9. Gehen Sie auf der Seite Snapshot-Berechtigungen wie folgt vor:

a. Geben Sie die AWS-Konto ID ein, mit der Sie die Snapshot-Kopie teilen, und wählen Sie dann Hinzufügen.

b. Wählen Sie Save aus.

Der Snapshot wird geteilt.

Kopieren Sie den gemeinsam genutzten Snapshot in das Zielkonto

Jetzt können Sie den gemeinsam genutzten Snapshot in das Ziel kopieren AWS-Konto.

Um den gemeinsam genutzten Snapshot zu kopieren

1. Melden Sie sich vom Ziel AWS Management Console aus beim an AWS-Konto.
2. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/
3. Wählen Sie im Navigationsbereich die Option Snapshots.
4. Wählen Sie den Tab Mit mir geteilt.
5. Wählen Sie den geteilten Snapshot aus.
6. Wählen Sie für Actions (Aktionen) die Option Copy Snapshot (Snapshot kopieren).
7. Wählen Sie Ihre Einstellungen für das Kopieren des Snapshots wie im vorherigen Verfahren, verwenden Sie jedoch eine AWS KMS key , die zum Zielkonto gehört.

   Wählen Sie Copy Snapshot (Snapshot kopieren) aus.

Snapshot-Sharing für Amazon beenden RDS

Um die gemeinsame Nutzung eines DB-Snapshots zu beenden, entziehen Sie dem Ziel die Berechtigung AWS-Konto.

Konsole

Um die gemeinsame Nutzung eines manuellen DB-Snapshots mit einem zu beenden AWS-Konto

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

3. Wählen Sie den manuellen Snapshot, für den Sie die Freigabe beenden möchten.
4. Wählen Sie die Option Actions (Aktionen) und anschließend Share Snapshot (Snapshot teilen) aus.
5. Um die Erlaubnis für ein zu entfernen AWS-Konto, wählen Sie Löschen als AWS Konto-ID für dieses Konto aus der Liste der autorisierten Konten.
6. Wählen Sie Speichern, um Ihre Änderungen zu speichern.

CLI

Verwenden Sie den `--values-to-remove` Parameter, um einen AWS-Konto Bezeichner aus der Liste zu entfernen.

Example das Stoppen der Snapshot-Freigabe

Das folgende Beispiel verhindert, dass AWS-Konto ID 444455556666 den Snapshot wiederherstellt.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-remove 444455556666
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-remove 444455556666
```

RDS API

Um die Freigabeberechtigung für eine zu entfernen AWS-Konto, verwenden Sie die `ModifyDBSnapshotAttribute`Operation mit `AttributeName` set to `restore` und dem `ValuesToRemove` Parameter. Um einen manuellen Snapshot als privat zu markieren, entfernen Sie den Wert `all` aus der Liste der Werte für das Attribut `restore`.

Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS

Sie können DB-Snapshot-Daten in einen Amazon S3-Bucket exportieren. Der Exportvorgang läuft im Hintergrund und hat keinen Einfluss auf die Leistung Ihrer aktiven Datenbank.

Wenn Sie einen DB-Snapshot exportieren, extrahiert Amazon RDS Daten aus dem Snapshot und speichert sie in einem Amazon-S3-Bucket in Ihrem Konto. Die Daten werden in einem Apache Parquet-Format gespeichert, das komprimiert und konsistent ist.

Sie können alle Arten von DB-Snapshots exportieren, einschließlich manueller Snapshots, automatisierter Systemsnapshots und Snapshots, die vom Service erstellt wurden. AWS Backup Standardmäßig werden alle Daten im Snapshot exportiert. Sie können sich jedoch dafür entscheiden, bestimmte Sätze von Datenbanken, Schemata oder Tabellen zu exportieren.

Nachdem die Daten exportiert wurden, können Sie die exportierten Daten direkt mit Tools wie Amazon Athena oder Amazon Redshift Spectrum analysieren. Weitere Informationen zur Verwendung von Athena zum Lesen von Parquet-Daten finden Sie unter Parquet SerDe im Amazon Athena Athena-Benutzerhandbuch. Weitere Informationen zur Verwendung von Redshift Spectrum zum Lesen von Parquet-Daten finden Sie unter COPY from columnar data formats im Amazon Redshift Database Developer Guide.

> **Warning**
>
> Sie können exportierte Snapshot-Daten nicht aus S3 in einer neuen DB-Instance wiederherstellen oder Snapshot-Daten aus S3 in eine bestehende DB-Instance importieren. Sie können die Daten jedoch mit Amazon Athena oder Redshift Spectrum zur Analyse verarbeiten. Darüber hinaus können Sie die Daten AWS Glue transformieren und dann mithilfe von Tools wie AWS DMS oder benutzerdefinierten Skripten in Amazon RDS importieren.

Weitere Informationen zum Exportieren von DB-Snapshots nach Amazon S3 finden Sie in den folgenden Themen.

Themen

- Überwachung von Snapshot-Exporten für Amazon RDS

- Abbrechen einer Snapshot-Exportaufgabe für Amazon RDS
- Fehlermeldungen für Amazon S3 S3-Exportaufgaben für Amazon RDS
- Fehlerbehebung RDS bei SQL Postgre-Berechtigungsfehlern
- Dateibenennungskonventionen für Exporte nach Amazon S3 für Amazon RDS
- Datenkonvertierung beim Export in einen Amazon S3 S3-Bucket für Amazon RDS

Übersicht über das Exportieren von Snapshot-Daten

Sie verwenden den folgenden Prozess, um DB-Snapshot-Daten in eine Amazon S3 einen Bucket zu exportieren. Weitere Informationen finden Sie in den folgenden Abschnitten.

1. Identifizieren Sie den zu exportierenden Snapshot.

   Verwenden Sie einen vorhandenen automatisierten oder manuellen Snapshot oder erstellen Sie einen manuellen Snapshot einer DB-Instance oder eines Multi-AZ-DB-Clusters.
2. Richten Sie den Zugriff auf den Amazon S3-Bucket ein.

   Ein Bucket ist ein Container für Amazon S3-Objekte oder -Dateien. Um die Informationen für den Zugriff auf einen Bucket bereitzustellen, führen Sie die folgenden Schritte aus:

   a. Identifizieren Sie den S3-Bucket, in den der Snapshot exportiert werden soll. Der S3-Bucket muss sich in derselben AWS Region wie der Snapshot befinden. Weitere Informationen finden Sie unter Identifizieren des Amazon S3-Buckets, in den exportiert werden soll.

   b. Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle, die der Snapshot-Exportaufgabe Zugriff auf den S3-Bucket gewährt. Weitere Informationen finden Sie unter Bereitstellen des Zugriffs auf einen Amazon S3-Bucket mit einer IAM-Rolle.
3. Erstellen Sie eine symmetrische Verschlüsselung AWS KMS key für die serverseitige Verschlüsselung. Der KMS-Schlüssel wird von der Snapshot-Exportaufgabe verwendet, um die AWS KMS serverseitige Verschlüsselung beim Schreiben der Exportdaten nach S3 einzurichten.

   Die KMS-Schlüsselrichtlinie muss die beiden Berechtigungen `kms:CreateGrant` und `kms:DescribeKey` enthalten. Weitere Informationen zur Verwendung von KMS-Schlüsseln in Amazon RDS finden Sie unter AWS KMS key Verwaltung.

   Wenn Ihre KMS-Schlüsselrichtlinie eine Deny-Anweisung enthält, stellen Sie sicher, dass Sie den AWS Dienstprinzipal `export.rds.amazonaws.com` explizit ausschließen.

Sie können einen KMS-Schlüssel in Ihrem AWS Konto oder einen kontoübergreifenden KMS-Schlüssel verwenden. Weitere Informationen finden Sie unter Verwendung eines Cross-Kontos AWS KMS key für die Verschlüsselung von Amazon S3 S3-Exporten.

4. Exportieren Sie den Snapshot mit der Konsole oder dem CLI-Befehl `start-export-task` nach Amazon S3. Weitere Informationen finden Sie unter Exportieren eines Snapshots in einen Amazon-S3-Bucket.
5. Um auf Ihre exportierten Daten im Amazon S3-Bucket zuzugreifen, siehe Hochladen, Herunterladen und Verwalten von Objekten im Amazon Simple Storage Service User Guide.

Einrichten des Zugriffs auf einen Amazon S3-Bucket

Um DB-Snapshot-Daten in eine Amazon S3-Datei zu exportieren, geben Sie zunächst dem Snapshot die Berechtigung für den Zugriff auf den Amazon S3-Bucket. Dann erstellen Sie eine IAM-Rolle, um dem Amazon-RDS-Service zu erlauben, in den Amazon S3-Bucket zu schreiben.

Themen

- Identifizieren des Amazon S3-Buckets, in den exportiert werden soll
- Bereitstellen des Zugriffs auf einen Amazon S3-Bucket mit einer IAM-Rolle
- Einen kontoübergreifenden Amazon-S3-Bucket verwenden
- Verwendung eines Cross-Kontos AWS KMS key für die Verschlüsselung von Amazon S3 S3-Exporten

Identifizieren des Amazon S3-Buckets, in den exportiert werden soll

Identifizieren Sie den Amazon S3-Bucket, in den der DB-Snapshot exportiert werden soll. Verwenden Sie einen vorhandenen S3-Bucket oder erstellen Sie einen neuen S3-Bucket.

> **Note**
>
> Der S3-Bucket, in den exportiert werden soll, muss sich in derselben AWS Region wie der Snapshot befinden.

Weitere Informationen zur Arbeit mit Amazon S3-Buckets finden Sie im Amazon Simple Storage Service User Guide:

- Wie zeige ich die Eigenschaften für einen S3-Bucket an?
- Wie aktiviere ich die Standardverschlüsselung für einen Amazon S3-Bucket?
- Wie erstelle ich einen S3-Bucket?

Bereitstellen des Zugriffs auf einen Amazon S3-Bucket mit einer IAM-Rolle

Bevor Sie DB-Snapshot-Daten nach Amazon S3 exportieren, geben Sie den Snapshot-Exportaufgaben Schreibzugriffsrechte auf den Amazon S3-Bucket.

Zum Gewähren dieser Berechtigung erstellen Sie eine IAM-Richtlinie, die Zugriff auf den Bucket ermöglicht. Erstellen Sie dann eine IAM-Rolle und fügen Sie der Rolle die Richtlinie an. Die IAM-Rolle weisen Sie später Ihrer Snapshot-Exportaufgabe zu.

Informationen zu anderen Amazon S3-Zugriffsverwaltungstools finden Sie unter Zugriffskontrolle in Amazon S3 im Amazon S3 S3-Benutzerhandbuch.

> **Important**
>
> Wenn Sie planen, Ihren Snapshot AWS Management Console zu exportieren, können Sie festlegen, dass die IAM-Richtlinie und die Rolle beim Exportieren des Snapshots automatisch erstellt werden. Detaillierte Anweisungen finden Sie unter Exportieren eines Snapshots in einen Amazon-S3-Bucket.

So geben Sie DB-Snapshot-Aufgaben Zugriff auf Amazon S3

1. Erstellen Sie eine IAM-Richtlinie. Diese Richtlinie bietet die Berechtigungen für den Bucket und die Objekte, die den Zugriff auf Ihre Snapshot-Exportaufgabe ermöglichen Amazon S3.

   Nehmen Sie die folgenden erforderlichen Aktionen in die Richtlinie auf, um die Übertragung von Dateien aus Amazon RDS in einen S3-Bucket zu ermöglichen:

   - `s3:PutObject*`
   - `s3:GetObject*`
   - `s3:ListBucket`
   - `s3:DeleteObject*`
   - `s3:GetBucketLocation`

Fügen Sie in die Richtlinie die folgenden Ressourcen zur Identifizierung des S3-Buckets und der Objekte im Bucket ein. Die folgende Liste von Ressourcen zeigt das ARN-Format (Amazon-Ressourcenname) für den Amazon S3-Zugriff.

- `arn:aws:s3:::amzn-s3-demo-bucket`
- `arn:aws:s3:::amzn-s3-demo-bucket/*`

Weitere Informationen zur Erstellung einer IAM-Richtlinie für Amazon RDS finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff. Siehe auch Tutorial: Erstellen und Anfügen Ihrer ersten vom Kunden verwalteten Richtlinie im IAM-Benutzerhandbuch.

Mit dem folgenden AWS CLI Befehl wird eine IAM-Richtlinie `ExportPolicy` mit diesen Optionen erstellt. Sie gewährt Zugriff auf einen Bucket namens *amzn-s3-demo-bucket*.

> **Note**
>
> Nachdem Sie die Richtlinie erstellt haben, notieren Sie den ARN der Richtlinie. Sie benötigen den ARN für einen nachfolgenden Schritt, in dem Sie die Richtlinie an eine IAM-Rolle anhängen.

```
aws iam create-policy  --policy-name ExportPolicy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ExportPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

```
            ]
        }
    ]
}'
```

2. Erstellen Sie eine IAM-Rolle, damit Amazon RDS diese IAM-Rolle in Ihrem Namen übernehmen kann, um auf Ihre Amazon-S3-Buckets zuzugreifen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.

   Das folgende Beispiel zeigt, wie der AWS CLI Befehl verwendet wird, um eine Rolle mit dem Namen `rds-s3-export-role` zu erstellen.

```
aws iam create-role  --role-name rds-s3-export-role  --assume-role-policy-document
 '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
           "Service": "export.rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole"
      }
    ]
  }'
```

3. Fügen Sie die erstellte IAM-Richtlinie der IAM-Rolle an, die Sie erstellt haben.

   Mit dem folgenden AWS CLI Befehl wird die zuvor erstellte Richtlinie an die angegebene `rds-s3-export-role` Rolle angehängt. Ersetzen Sie *`your-policy-arn`* durch den Richtlinien-ARN, den Sie in einem früheren Schritt notiert haben.

```
aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-
export-role
```

Einen kontoübergreifenden Amazon-S3-Bucket verwenden

Sie können Amazon S3 S3-Buckets AWS kontenübergreifend verwenden. Um einen kontoübergreifenden Bucket zu verwenden, fügen Sie eine Bucket-Richtlinie hinzu, um den Zugriff auf

die IAM-Rolle zu erlauben, die Sie für die S3-Exporte verwenden. Weitere Informationen finden Sie unter Beispiel 2: Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen.

Fügen Sie eine Bucket-Richtlinie an Ihren Bucket an, wie im folgenden Beispiel gezeigt.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/Admin"
            },
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket",
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        }
    ]
}
```

Verwendung eines Cross-Kontos AWS KMS key für die Verschlüsselung von Amazon S3 S3-Exporten

Sie können ein Cross-Konto verwenden AWS KMS key , um Amazon S3 S3-Exporte zu verschlüsseln. Zuerst fügen Sie dem lokalen Konto eine Schlüsselrichtlinie hinzu und fügen dann IAM-Richtlinien im externen Konto hinzu. Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung eines KMS-Schlüssels erlauben.

So verwenden Sie einen kontoübergreifenden KMS-Schlüssel

1. Fügen Sie dem lokalen Konto eine Schlüsselrichtlinie hinzu.

   Das folgende Beispiel gibt `ExampleRole` und `ExampleUser` im externen Konto 444455556666 Berechtigungen im lokalen Konto 123456789012.

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::444455556666:role/ExampleRole",
            "arn:aws:iam::444455556666:user/ExampleUser"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "*"
}
```

2. Fügen Sie IAM-Richtlinien im externen Konto hinzu.

   Die folgende IAM-Richtlinie erlaubt es dem Prinzipal, den KMS-Schlüssel im Konto 123456789012 für kryptografische Operationen zu erlauben. Um diese Berechtigung an `ExampleRole` und `ExampleUser` zu erteilen, fügen Sie die Richtlinie zu ihnen im Konto 444455556666 an.

```
{
    "Sid": "Allow use of KMS key in account 123456789012",
    "Effect": "Allow",
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
```

```
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Exportieren eines Snapshots in einen Amazon-S3-Bucket

Pro Person können bis zu fünf DB-Snapshot-Exportaufgaben gleichzeitig ausgeführt werden. AWS-Konto

> **Note**
>
> Das Exportieren von RDS-Snapshots kann je nach Datenbanktyp und -größe eine Weile dauern. Die Exportaufgabe stellt zuerst die gesamte Datenbank wieder her und skaliert sie, bevor die Daten in Amazon S3 extrahiert werden. Der Fortschritt des Vorgangs während dieser Phase wird als Starting (Startet) angezeigt. Wenn die Aufgabe zum Exportieren von Daten zu S3 wechselt, wird der Fortschritt als In progress (In Bearbeitung) angezeigt.
> Die Zeit, die für den Export benötigt wird, hängt von den in der Datenbank gespeicherten Daten ab. Beispielsweise exportieren Tabellen mit gut verteilten numerischen Primärschlüssel- oder Indexspalten am schnellsten. Bei Tabellen, die keine Spalte enthalten, die für die Partitionierung geeignet ist, und bei Tabellen mit nur einem Index für eine auf Zeichenfolgen basierende Spalte, dauert dies länger, da der Export einen langsameren Single-Thread-Prozess verwendet. Diese längere Exportzeit tritt auf, da der Export einen langsameren Single-Thread-Prozess verwendet.

Sie können einen DB-Snapshot mit der AWS Management Console, der oder der RDS-API nach Amazon S3 exportieren. AWS CLI Verwenden Sie die AWS CLI oder die RDS-API, um einen DB-Snapshot in einen kontoübergreifenden Amazon S3 S3-Bucket zu exportieren.

Wenn Sie eine Lambda-Funktion zum Exportieren eines Snapshots verwenden, fügen Sie die Aktion `kms:DescribeKey` der Lambda-Funktionsrichtlinie hinzu. Weitere Informationen finden Sie unter AWS Lambda Berechtigungen.

Konsole

Die Konsolenoption Export to Amazon S3 (Zu Amazon S3-Konsole exportieren) wird nur für Snapshots angezeigt, die zu Amazon S3 exportiert werden können. Ein Snapshot ist aus folgenden Gründen möglicherweise nicht für den Export verfügbar:

- Die DB-Engine wird für den S3-Export nicht unterstützt.
- Die DB-Engine-Version wird für den S3-Export nicht unterstützt.
- Der S3-Export wird in der AWS Region, in der der Snapshot erstellt wurde, nicht unterstützt.

So exportieren Sie einen DB-Snapshot:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie auf den Registerkarten die Art des Snapshots aus, den Sie exportieren möchten.
4. Wählen Sie in der Liste der Snapshots den Snapshot aus, den Sie exportieren möchten.
5. Wählen Sie für Actions (Aktionen) die Option Export to Amazon S3 (Nach Amazon S3 exportieren) aus.

   Das Fenster Export to Amazon S3 (Nach Amazon S3 exportieren) erscheint.
6. Geben Sie für Export identifier (Export-ID) einen Namen ein, um die Exportaufgabe zu identifizieren. Dieser Wert wird auch für den Namen der im S3-Bucket erstellten Datei verwendet.
7. Wählen Sie die zu exportierenden Daten aus:

   - Wählen Sie All (Alle), um alle Daten im Snapshot zu exportieren.
   - Wählen Sie Partial (Teilweise), um bestimmte Teile des Snapshots zu exportieren. Um zu ermitteln, welche Teile des Snapshots exportiert werden sollen, geben Sie eine oder mehrere Datenbanken, Schemas oder Tabellen für Bezeichner ein, getrennt durch Leerzeichen.

     Verwenden Sie das folgende Format:

     ```
     database[.schema][.table] database2[.schema2][.table2] ... databasen[.schemaN]
     [.tablen]
     ```

     Zum Beispiel:

```
mydatabase mydatabase2.myschema1 mydatabase2.myschema2.mytable1
 mydatabase2.myschema2.mytable2
```

8. Wählen Sie für S3 bucket (S3-Bucket) den Bucket aus, in den exportiert werden soll.

   Um die exportierten Daten einem Ordnerpfad im S3-Bucket zuzuordnen, geben Sie den optionalen Pfad für S3 prefix (S3-Präfix) ein.

9. Wählen Sie für die IAM-Rolle entweder eine Rolle, die Ihnen Schreibzugriff auf den gewählten S3-Bucket gewährt, oder erstellen Sie eine neue Rolle.

   - Wenn Sie eine Rolle durch Befolgen der Schritte in Bereitstellen des Zugriffs auf einen Amazon S3-Bucket mit einer IAM-Rolle erstellt haben, wählen Sie diese Rolle.
   - Wenn Sie keine Rolle erstellt haben, die Ihnen Schreibzugriff auf den von Ihnen gewählten S3-Bucket gewährt, wählen Sie Create a new role (Neue Rolle erstellen) aus, um die Rolle automatisch zu erstellen. Geben Sie dann unter IAM role name (IAM-Rollenname) einen Namen für die Rolle ein.

10. Für AWS KMS key geben Sie den ARN für den Schlüssel ein, der für die Verschlüsselung der exportierten Daten verwendet werden soll.
11. Wählen Sie Export to Amazon S3 (Nach Amazon S3 exportieren) aus.

AWS CLI

Um einen DB-Snapshot mit dem nach Amazon S3 zu exportieren AWS CLI, verwenden Sie den start-export-taskBefehl mit den folgenden erforderlichen Optionen:

- `--export-task-identifier`
- `--source-arn`
- `--s3-bucket-name`
- `--iam-role-arn`
- `--kms-key-id`

In den folgenden Beispielen wird die Snapshot-Exportaufgabe benannt*`my-snapshot-export`*, die einen Snapshot in einen S3-Bucket mit dem Namen exportiert*`amzn-s3-demo-bucket`*.

Example

Für LinuxmacOS, oderUnix:

```
aws rds start-export-task \
    --export-task-identifier my-snapshot-export \
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name \
    --s3-bucket-name amzn-s3-demo-bucket \
    --iam-role-arn iam-role \
    --kms-key-id my-key
```

Windows:

```
aws rds start-export-task ^
    --export-task-identifier my-snapshot-export ^
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name ^
    --s3-bucket-name amzn-s3-demo-bucket ^
    --iam-role-arn iam-role ^
    --kms-key-id my-key
```

Beispiel für eine Ausgabe folgt.

```
{
    "Status": "STARTING",
    "IamRoleArn": "iam-role",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "my-export-bucket",
    "PercentProgress": 0,
    "KmsKeyId": "my-key",
    "ExportTaskIdentifier": "my-snapshot-export",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name"
}
```

Um einen Ordnerpfad im S3-Bucket für den Snapshot-Export anzugeben, fügen Sie die `--s3-prefix` Option in den start-export-taskBefehl ein.

RDS-API

Um einen DB-Snapshot mithilfe der Amazon RDS-API nach Amazon S3 zu exportieren, verwenden Sie den StartExportTaskVorgang mit den folgenden erforderlichen Parametern:

- `ExportTaskIdentifier`
- `SourceArn`
- `S3BucketName`
- `IamRoleArn`
- `KmsKeyId`

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Versions- und Regionsverfügbarkeit beim Exportieren von Snapshots nach S3 finden Sie unter Unterstützte Regionen und DB-Engines für den Export von Snapshots nach S3 in Amazon RDS.

Einschränkungen

Das Exportieren von DB-Snapshot-Daten nach Amazon S3 hat die folgenden Einschränkungen:

- Sie können nicht mehrere Exportaufgaben für denselben DB-Snapshot gleichzeitig ausführen. Dies gilt sowohl für vollständige als auch Teilexporte.
- Das Exportieren von Snapshots aus Datenbanken, die Magnetspeicher verwenden, wird nicht unterstützt.
- Exporte nach S3 unterstützen keine S3-Präfixe, die einen Doppelpunkt (:) enthalten.
- Die folgenden Zeichen im S3-Dateipfad werden während des Exports in Unterstriche (_) konvertiert:

```
\ ` " (space)
```

- Wenn eine Datenbank, ein Schema oder eine Tabelle andere Zeichen als den folgenden enthält, wird ein teilweiser Export nicht unterstützt. Sie können jedoch den gesamten DB-Snapshot exportieren.
  - Lateinische Buchstaben (A–Z)
  - Ziffern (0–9)
  - Dollar-Symbol ($)
  - Unterstrich (_)

- Leerzeichen ( ) und bestimmte andere Zeichen werden in den Spaltennamen von Datenbanktabellen nicht unterstützt. Tabellen mit den folgenden Zeichen in Spaltennamen werden beim Export übersprungen:

  ```
  , ; { } ( ) \n \t = (space)
  ```

- Tabellen mit Schrägstrichen (/) im Namen werden beim Export übersprungen.
- Temporäre und nicht protokollierte Tabellen von RDS für PostgreSQL werden beim Export übersprungen.
- Wenn die Daten ein großes Objekt wie BLOB oder CLOB mit einer Größe von 500 MB oder mehr enthalten, schlägt der Export fehl.
- Wenn eine Zeile in einer Tabelle ca. 2 GB groß oder noch größer ist, wird die Tabelle beim Export übersprungen.
- Bei Teilexporten hat die `ExportOnly` Liste eine maximale Größe von 200 KB.
- Es wird dringend empfohlen, für jede Exportaufgabe einen eindeutigen Namen zu verwenden. Wenn Sie keinen eindeutigen Aufgabennamen verwenden, erhalten Sie möglicherweise die folgende Fehlermeldung:

  ExportTaskAlreadyExistsFault: Beim Aufrufen der StartExportTask Operation ist ein Fehler aufgetreten (ExportTaskAlreadyExists): Die Exportaufgabe mit der ID ist *xxxxx* bereits vorhanden.
- Sie können einen Snapshot löschen, während Sie seine Daten nach S3 exportieren, aber die Speicherkosten für diesen Snapshot werden Ihnen so lange in Rechnung gestellt, bis die Exportaufgabe abgeschlossen ist.
- Sie können exportierte Snapshot-Daten aus S3 nicht in einer neuen DB-Instance wiederherstellen oder Snapshot-Daten aus S3 in eine bestehende DB-Instance importieren.
- Pro Person AWS-Konto können bis zu fünf DB-Snapshot-Exportaufgaben gleichzeitig ausgeführt werden.
- Um einen DB-Snapshot in einen kontoübergreifenden Amazon S3 S3-Bucket zu exportieren, müssen Sie die AWS CLI oder die RDS-API verwenden.
- Nachdem Amazon RDS eine Exportaufgabe abgeschlossen hat, müssen Sie möglicherweise eine kurze Zeit warten, bis Sie eine weitere Exportaufgabe aus demselben DB-Snapshot starten können.
- Sie können keine Ansichten oder materialisierte Ansichten exportieren.
- RDS Export to S3 unterstützt keine tagbasierte Zugriffskontrolle für. GuardDuty Malware Protection for S3

Überwachung von Snapshot-Exporten für Amazon RDS

Sie können DB-Snapshot-Exporte mit dem AWS Management Console AWS CLI, dem oder dem überwachen RDSAPI.

Konsole

So überwachen Sie DB-Snapshot-Exporte:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Um die Liste der Snapshot-Exporte anzuzeigen, wählen Sie die Registerkarte Exports in Amazon S3 (Exporte in Amazon S3) aus.
4. Um Informationen über einen bestimmten Snapshot-Export anzuzeigen, wählen Sie die Exportaufgabe aus.

AWS CLI

Verwenden Sie den describe-export-tasksBefehl AWS CLI, um DB-Snapshot-Exporte mit dem zu überwachen.

Das folgende Beispiel zeigt, wie Sie aktuelle Informationen über alle Ihre Snapshot-Exporte anzeigen können.

Example

```
aws rds describe-export-tasks

{
    "ExportTasks": [
        {
            "Status": "CANCELED",
            "TaskEndTime": "2019-11-01T17:36:46.961Z",
            "S3Prefix": "something",
            "ExportTime": "2019-10-24T20:23:48.364Z",
            "S3Bucket": "amzn-s3-demo-bucket",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/
bPxRfiCYEXAMPLEKEY",
            "ExportTaskIdentifier": "anewtest",
```

```
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-25T19:10:58.885Z",
            "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:parameter-
groups-test"
        },
{
            "Status": "COMPLETE",
            "TaskEndTime": "2019-10-31T21:37:28.312Z",
            "WarningMessage": "{\"skippedTables\":[],\"skippedObjectives\":[],\"general
\":[{\"reason\":\"FAILED_TO_EXTRACT_TABLES_LIST_FOR_DATABASE\"}]}",
            "S3Prefix": "",
            "ExportTime": "2019-10-31T06:44:53.452Z",
            "S3Bucket": "amzn-s3-demo-bucket1",
            "PercentProgress": 100,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "thursday-events-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 263,
            "TaskStartTime": "2019-10-31T20:58:06.998Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-31-06-44"
        },
        {
            "Status": "FAILED",
            "TaskEndTime": "2019-10-31T02:12:36.409Z",
            "FailureCause": "The S3 bucket edgcuc-export isn't located in the current
 AWS Region. Please, review your S3 bucket name and retry the export.",
            "S3Prefix": "",
            "ExportTime": "2019-10-30T06:45:04.526Z",
            "S3Bucket": "amzn-s3-demo-bucket2",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "wednesday-afternoon-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-30T22:43:40.034Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-30-06-45"
        }
    ]
```

```
}
```

Um Informationen über einen bestimmten Snapshot-Export anzuzeigen, fügen Sie die Option `--export-task-identifier` mit dem Befehl `describe-export-tasks` ein. Um die Ausgabe zu filtern, fügen Sie die Option `--Filters` ein. Weitere Optionen finden Sie im describe-export-tasksBefehl.

RDS API

Verwenden Sie den DescribeExportTasksVorgang, um Informationen zu DB-Snapshot-Exporten mithilfe von Amazon RDS API anzuzeigen.

Um den Abschluss des Exportworkflows zu verfolgen oder einen anderen Workflow zu initiieren, können Sie Themen von Amazon Simple Notification Service abonnieren. Weitere Informationen zu Amazon finden SNS Sie unterMit RDS Amazon-Event-Benachrichtigungen arbeiten.

Abbrechen einer Snapshot-Exportaufgabe für Amazon RDS

Sie können eine DB-Snapshot-Exportaufgabe mit dem AWS Management Console AWS CLI, dem oder dem abbrechen RDSAPI.

> **Note**
>
> Das Abbrechen einer Snapshot-Exportaufgabe entfernt keine Daten, die nach Amazon S3 exportiert wurden. Informationen zum Löschen der Daten über die Konsole finden Sie unter Wie lösche ich Objekte aus einem S3-Bucket?. Um die Daten mit dem zu löschenCLI, verwenden Sie den Befehl delete-object.

Konsole

So brechen Sie eine Aufgabe zum Export von Snapshots ab:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie die Registerkarte Exports in Amazon S3 (Exporte in Amazon S3) aus.
4. Wählen Sie die Snapshot-Exportaufgabe aus, die Sie abbrechen möchten.
5. Klicken Sie auf Abbrechen.

6. Wählen Sie die Bestätigungsseite Cancel export task (Exportaufgabe abbrechen) aus.

AWS CLI

Um eine Snapshot-Exportaufgabe mit dem abzubrechen AWS CLI, verwenden Sie den cancel-export-taskBefehl. Der Befehl erfordert die Option `--export-task-identifier`.

Example

```
aws rds cancel-export-task --export-task-identifier my_export
{
    "Status": "CANCELING",
    "S3Prefix": "",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "amzn-s3-demo-bucket",
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "ExportTaskIdentifier": "my_export",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:export-example-1"
}
```

RDS API

Um eine Snapshot-Exportaufgabe über Amazon abzubrechen RDSAPI, verwenden Sie den CancelExportTaskVorgang mit dem `ExportTaskIdentifier` Parameter.

Fehlermeldungen für Amazon S3 S3-Exportaufgaben für Amazon RDS

In der folgenden Tabelle werden die Nachrichten beschrieben, die zurückgegeben werden, wenn Amazon-S3-Exportaufgaben fehlschlagen.

Fehlernachricht	Beschreibung
Ein unbekannter interner Fehler ist aufgetreten.	Die Aufgabe ist fehlgeschlagen, da eine unbekannte Störung, eine Ausnahme oder ein Fehler aufgetreten ist.

Fehlernachricht	Beschreibung
Beim Schreiben der Metadaten der Exportaufgabe in den S3-Bucket [Bucket-Name] ist ein unbekannter interner Fehler aufgetreten.	Die Aufgabe ist fehlgeschlagen, da eine unbekannte Störung, eine Ausnahme oder ein Fehler aufgetreten ist.
Beim RDS Export konnten die Metadaten der Exportaufgabe nicht geschrieben werden, da er die IAM Rolle [RolleARN] nicht annehmen kann.	Die Exportaufgabe übernimmt Ihre IAM Rolle bei der Überprüfung, ob sie Metadaten in Ihren S3-Bucket schreiben darf. Wenn die Aufgabe Ihre IAM Rolle nicht übernehmen kann, schlägt sie fehl.
Beim RDS Export konnten die Metadaten der Exportaufgabe mithilfe der Rolle [IAMRolle] mit dem Schlüssel [KMSSchlüssel-ID] nicht in den S3-Bucket [Bucketname] geschrieben werden. ARN Fehlercode: [Fehlercode]	Eine oder mehrere Berechtigungen fehlen, sodass die Exportaufgabe nicht auf den S3-Bucket zugreifen kann. Diese Fehlermeldung wird ausgelöst, wenn einer der folgenden Fehlercodes empfangen wird: • `AWSSecurityTokenServiceException` mit dem Fehlercode `AccessDenied` • `AmazonS3Exception` mit dem Fehlercode `NoSuchBucket` , `AccessDenied` , `KMS.KMSInvalidStateException` , `403 Forbidden` , oder `KMS.DisabledException` Diese Fehlercodes weisen darauf hin, dass die Einstellungen für die IAM Rolle, den S3-Bucket oder den KMS Schlüssel falsch konfiguriert sind.
Die IAM Rolle [RolleARN] ist nicht berechtigt, [S3-Aktion] im S3-Bucket [Bucket-Name] aufzurufen. Überprüfen Sie Ihre Berechtigungen und versuchen Sie den Export erneut.	Die IAM Richtlinie ist falsch konfiguriert. Die Berechtigung für die spezifische S3-Aktion für den S3-Bucket fehlt, was zu einem Fehler der Exportaufgabe führt.

Fehlernachricht	Beschreibung
KMSDie Schlüsselüberprüfung ist fehlgeschlagen. Überprüfen Sie die Anmeldeinformationen auf Ihrem KMS Schlüssel und versuchen Sie es erneut.	Die Überprüfung der KMS Schlüsselanmeldedaten ist fehlgeschlagen.
Die Überprüfung der S3-Anmeld einformation ist fehlgeschlagen. Überprüfen Sie die Berechtigungen für Ihren S3-Bucket und Ihre IAM Richtlinie.	Die Überprüfung der S3-Anmeldeinformation ist fehlgeschlagen.
Der S3-Bucket [Bucket-Name] ist ungültig. Entweder befindet es sich nicht in der aktuellen AWS Region oder es gibt sie nicht. Überprüfen Sie Ihren S3-Bucket-Namen und versuchen Sie den Export erneut.	Der S3-Bucket ist ungültig.
Der S3-Bucket [Bucket-Name] befindet sich nicht in der aktuellen AWS Region. Überprüfen Sie Ihren S3-Bucket-Namen und versuchen Sie den Export erneut.	Der S3-Bucket befindet sich in der falschen AWS Region.

Fehlerbehebung RDS bei SQL Postgre-Berechtigungsfehlern

Beim Exportieren von SQL Postgre-Datenbanken nach Amazon S3 wird möglicherweise ein `PERMISSIONS_DO_NOT_EXIST` Fehler angezeigt, der besagt, dass bestimmte Tabellen übersprungen wurden. Dieser Fehler tritt normalerweise auf, wenn der Superuser, den Sie beim Erstellen der Datenbank angegeben haben, nicht berechtigt ist, auf diese Tabellen zuzugreifen.

Führen Sie den folgenden Befehl aus, um diesen Fehler zu beheben:

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema_name TO superuser_name
```

Weitere Informationen zu Superuser-Berechtigungen finden Sie unter Berechtigungen von Hauptbenutzerkonten.

Dateibenennungskonventionen für Exporte nach Amazon S3 für Amazon RDS

Exportierte Daten für bestimmte Tabellen werden im Format *base_prefix*/*files* gespeichert, wobei das Basispräfix folgendes ist:

```
export_identifier/database_name/schema_name.table_name/
```

Zum Beispiel:

```
export-1234567890123-459/rdststdb/rdststdb.DataInsert_7ADB5D19965123A2/
```

Es gibt zwei Konventionen für die Benennung von Dateien.

- Aktuelle Konvention:

  ```
  batch_index/part-partition_index-random_uuid.format-based_extension
  ```

  Der Batchindex ist eine Sequenznummer, die einen aus der Tabelle gelesenen Datenstapel darstellt. Wenn wir Ihre Tabelle nicht in kleine Teile partitionieren können, die parallel exportiert werden sollen, wird es mehrere Batch-Indizes geben. Das Gleiche passiert, wenn Ihre Tabelle in mehrere Tabellen partitioniert ist. Es wird mehrere Batch-Indizes geben, einen für jede der Tabellenpartitionen Ihrer Haupttabelle.

  Wenn wir Ihre Tabelle in kleine Teile partitionieren können, die parallel gelesen werden sollen, wird es nur den 1 Batch-Index-Ordner geben.

  Im Batchindex-Ordner befinden sich eine oder mehrere Parquet-Dateien, die die Daten Ihrer Tabelle enthalten. Das Präfix des Parquet-Dateinamens ist`part-`*partition_index*. Wenn Ihre Tabelle partitioniert ist, gibt es mehrere Dateien, die mit dem Partitionsindex `00000` beginnen.

  Es kann Lücken in der Reihenfolge des Partitionsindexes geben. Dies liegt daran, dass jede Partition aus einer Bereichsabfrage in Ihrer Tabelle abgerufen wird. Wenn sich im Bereich dieser Partition keine Daten befinden, wird diese Sequenznummer übersprungen.

  Nehmen wir beispielsweise an, dass die `id` Spalte der Primärschlüssel der Tabelle ist und ihre Minimal- und Maximalwerte `100` und `1000` sind. Wenn wir versuchen, diese Tabelle mit neun Partitionen zu exportieren, lesen wir sie mit parallel Abfragen wie den folgenden:

```
SELECT * FROM table WHERE id <= 100 AND id < 200
SELECT * FROM table WHERE id <= 200 AND id < 300
```

Dies sollte neun Dateien von `part-00000-random_uuid.gz.parquet` bis erzeugen`part-00008-random_uuid.gz.parquet`. Wenn es jedoch keine Zeilen IDs zwischen `200` und gibt`350`, ist eine der fertigen Partitionen leer und es wird keine Datei dafür erstellt. Im vorherigen Beispiel wurde `part-00001-random_uuid.gz.parquet` nicht erstellt.

- Ältere Konvention:

```
part-partition_index-random_uuid.format-based_extension
```

Dies entspricht der aktuellen Konvention, jedoch ohne das *batch_index* Präfix, zum Beispiel:

```
part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet
part-00001-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet
part-00002-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

Die Namenskonvention für Dateien kann geändert werden. Daher empfehlen wir beim Lesen von Zieltabellen, dass Sie alles innerhalb des Basispräfixes für die Tabelle lesen.

Datenkonvertierung beim Export in einen Amazon S3 S3-Bucket für Amazon RDS

Wenn Sie einen DB-Snapshot in einen Amazon S3-Bucket exportieren, konvertiert Amazon RDS Daten in das Parquet-Format, exportiert Daten darin und speichert Daten im Parquet-Format. Weitere Informationen über Parquet finden Sie auf der Website Apache Parquet.

Parquet speichert alle Daten als einen der folgenden primitiven Typen:

- BOOLEAN
- INT32
- INT64
- INT96
- FLOAT
- DOUBLE

- BYTE_ARRAY – Ein Byte-Array mit variabler Länge, auch bekannt als Binary
- FIXED_LEN_BYTE_ARRAY – Ein Byte-Array fester Länge, das verwendet wird, wenn die Werte eine konstante Größe haben

Es gibt nur wenige Parquet-Datentypen, um die Komplexität beim Lesen und Schreiben des Formats zu reduzieren. Parquet bietet logische Typen zur Erweiterung primitiver Typen. Ein logischer Typ ist als Annotation, bei der Daten in einem `LogicalType`-Metadatenfeld implementiert sind. Die logische Typannotation beschreibt, wie der primitive Typ zu interpretieren ist.

Wenn der logische Typ `STRING` einen `BYTE_ARRAY`-Typ annotiert, gibt er an, dass das Byte-Array als UTF-8-kodierte Zeichenfolge interpretiert werden soll. Nach Abschluss einer Exportaufgabe informiert Sie Amazon RDS, wenn eine Zeichenfolgenkonvertierung stattgefunden hat. Die zugrunde liegenden exportierten Daten entsprechen immer den Daten aus der Quelle. Aufgrund des Kodierungsunterschieds in UTF-8 können jedoch einige Zeichen beim Einlesen von Tools wie Athena anders als in der Quelle erscheinen.

Weitere Informationen finden Sie unter Logische Typdefinitionen für Parquet in der Parquet-Dokumentation.

Themen

- MySQL- und MariaDB-Datentyp-Mapping zu Parquet
- PostgreSQL-Datentyp-Mapping zu Parquet

MySQL- und MariaDB-Datentyp-Mapping zu Parquet

Die folgende Tabelle zeigt das Mapping von MySQL- und MariaDB-Datentypen zu Parquet-Datentypen, wenn die Daten konvertiert und nach Amazon S3 exportiert werden.

Quelldatentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zur Konvertierung
Numerische Datentypen			
BIGINT	INT64		
BIGINT UNSIGNED	FIXED_LEN_BYTE_ARRAY(9)	DECIMAL(20,0)	Parquet unterstützt nur signierte Typen,

Quelldatentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zur Konvertierung
			daher erfordert das Mapping ein zusätzliches Byte (8 plus 1), um den Typ BIGINT_UNSIGNED zu speichern.
BIT	BYTE_ARRAY		
DECIMAL	INT32	DECIMAL (p,s)	Wenn der Quellwert kleiner als 2^{31} ist, wird er als gespeichert INT32.
	INT64	DECIMAL (p,s)	Wenn der Quellwert 2^{31} oder größer, aber kleiner als 2^{63} ist, wird er als gespeichertINT64.
	FIX_LEN_BYTE_ARRAY(N)	DECIMAL (p,s)	Wenn der Quellwert 2^{63} oder größer ist, wird er als FIXED_LEN_BYTE_ARRAY(N) gespeichert.
	BYTE_ARRAY	STRING	Parquet unterstützt maximal 38 Dezimalstellen. Der Dezimalwert wird in eine Zeichenfolge vom Typ BYTE_ARRAY konvertiert und als codiert. UTF8

Quelldatentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zur Konvertierung
DOUBLE	DOUBLE		
FLOAT	DOUBLE		
INT	INT32		
INT UNSIGNED	INT64		
MEDIUMINT	INT32		
MEDIUMINT UNSIGNED	INT64		
NUMERIC	INT32	DECIMAL (p,s)	Wenn der Quellwert kleiner als 2^{31} ist, wird er als gespeichert. INT32
	INT64	DECIMAL (p,s)	Wenn der Quellwert 2^{31} oder größer, aber kleiner als 2^{63} ist, wird er als gespeichertINT64.
	FIXED_LEN_ARRAY(N)	DECIMAL (p,s)	Wenn der Quellwert 2^{63} oder größer ist, wird er als FIXED_LEN_BYTE_ARRAY(N) gespeichert.

Quelldatentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zur Konvertierung
	BYTE_ARRAY	STRING	Parquet unterstützt keine numerische Genauigkeit größer als 38. Dieser numerische Wert wird in eine Zeichenfolge vom Typ BYTE_ARRAY konvertiert und als codiert. UTF8
SMALLINT	INT32		
SMALLINT UNSIGNED	INT32		
TINYINT	INT32		
TINYINT UNSIGNED	INT32	INT (16, wahr)	
Zeichenfolgen-Datentypen			
BINARY	BYTE_ARRAY		
BLOB	BYTE_ARRAY		
CHAR	BYTE_ARRAY		
ENUM	BYTE_ARRAY	STRING	
LINESTRING	BYTE_ARRAY		
LONGBLOB	BYTE_ARRAY		
LONGTEXT	BYTE_ARRAY	STRING	
MEDIUMBLOB	BYTE_ARRAY		

Quelldatentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zur Konvertierung
MEDIUMTEXT	BYTE_ARRAY	STRING	
MULTILINESTRING	BYTE_ARRAY		
SET	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TINYBLOB	BYTE_ARRAY		
TINYTEXT	BYTE_ARRAY	STRING	
VARBINARY	BYTE_ARRAY		
VARCHAR	BYTE_ARRAY	STRING	
Datums- und Uhrzeit-Datentypen			
DATUM	BYTE_ARRAY	STRING	Ein Datum wird in eine Zeichenfolge vom Typ BYTE_ARRAY konvertiert und als codiert. UTF8
DATETIME	INT64	TIMESTAMP_MICROS	
TIME	BYTE_ARRAY	STRING	Ein TIME-Typ wird in eine Zeichenfolge in einem BYTE_ARRAY konvertiert und als codiert. UTF8
TIMESTAMP (ZEITSTEMPEL)	INT64	TIMESTAMP_MICROS	

Quelldatentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zur Konvertierung
JAHR	INT32		
Geometrische Datentypen			
GEOMETRY	BYTE_ARRAY		
GEOMETRYCOLLECTION	BYTE_ARRAY		
MULTIPOINT	BYTE_ARRAY		
MULTIPOLYGON	BYTE_ARRAY		
POINT	BYTE_ARRAY		
POLYGON	BYTE_ARRAY		
JSON-Datentyp			
JSON	BYTE_ARRAY	STRING	

PostgreSQL-Datentyp-Mapping zu Parquet

Die folgende Tabelle zeigt das Mapping von PostgreSQL-Datentypen zu Parquet-Datentypen, wenn Daten konvertiert und nach Amazon S3 exportiert werden.

PostgreSQL-Datentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zum Mapping
Numerische Datentypen			
BIGINT	INT64		
BIGSERIAL	INT64		
DECIMAL	BYTE_ARRAY	STRING	Ein DECIMAL-Typ wird in eine

PostgreSQL-Datentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zum Mapping
			Zeichenfolge in einem BYTE_ARRAY-Typ konvertiert und als codiert. UTF8 Diese Konvertierung soll Komplikationen aufgrund von Datengenauigkeit und Datenwerten, die keine Zahlen sind (NaN), vermeiden.
DOUBLE PRECISION	DOUBLE		
INTEGER	INT32		
MONEY	BYTE_ARRAY	STRING	
REAL	FLOAT		
SERIAL	INT32		
SMALLINT	INT32	INT (16, wahr)	
SMALLSERIAL	INT32	INT (16, wahr)	
Zeichenfolgen- und verwandte Datentypen			

PostgreSQL-Datentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zum Mapping
ARRAY	BYTE_ARRAY	STRING	Ein Array wird in eine Zeichenfolge umgewandelt und als BINARY (UTF8) codiert. Diese Konvertierung dient dazu, Komplikationen aufgrund von Datengenauigkeit, Datenwerten, die keine Zahl sind (NaN), und Zeitdatenwerten zu vermeiden.
BIT	BYTE_ARRAY	STRING	
BIT VARYING	BYTE_ARRAY	STRING	
BYTEA	BINARY		
CHAR	BYTE_ARRAY	STRING	
CHAR(N)	BYTE_ARRAY	STRING	
ENUM	BYTE_ARRAY	STRING	
NAME	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TEXT SEARCH	BYTE_ARRAY	STRING	
VARCHAR(N)	BYTE_ARRAY	STRING	
XML	BYTE_ARRAY	STRING	

PostgreSQL-Datentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zum Mapping
Datums- und Uhrzeit-Datentypen			
DATUM	BYTE_ARRAY	STRING	
INTERVAL	BYTE_ARRAY	STRING	
TIME	BYTE_ARRAY	STRING	
TIME WITH TIME ZONE	BYTE_ARRAY	STRING	
TIMESTAMP	BYTE_ARRAY	STRING	
TIMESTAMP WITH TIME ZONE	BYTE_ARRAY	STRING	
Geometrische Datentypen			
BOX	BYTE_ARRAY	STRING	
CIRCLE	BYTE_ARRAY	STRING	
LINE	BYTE_ARRAY	STRING	
LINESEGMENT	BYTE_ARRAY	STRING	
PATH	BYTE_ARRAY	STRING	
POINT	BYTE_ARRAY	STRING	
POLYGON	BYTE_ARRAY	STRING	
JSON-Datentypen			
JSON	BYTE_ARRAY	STRING	
JSONB	BYTE_ARRAY	STRING	
Weitere Datentypen			

PostgreSQL-Datentyp	Parquet-Primitiv-Typ	Logische Typannotation	Anmerkungen zum Mapping
BOOLEAN	BOOLEAN		
CIDR	BYTE_ARRAY	STRING	Network-Datentyp
COMPOSITE	BYTE_ARRAY	STRING	
DOMAIN	BYTE_ARRAY	STRING	
INET	BYTE_ARRAY	STRING	Network-Datentyp
MACADDR	BYTE_ARRAY	STRING	
OBJECT IDENTIFIER	–		
PG_LSN	BYTE_ARRAY	STRING	
RANGE	BYTE_ARRAY	STRING	
UUID	BYTE_ARRAY	STRING	

Wird AWS Backup zur Verwaltung automatisierter Backups für Amazon verwendet RDS

AWS Backup ist ein vollständig verwalteter Backup-Service, der es einfach macht, die Sicherung von Daten zwischen AWS Diensten in der Cloud und vor Ort zu zentralisieren und zu automatisieren. Sie können Backups Ihrer RDS Amazon-Datenbanken in verwalten AWS Backup.

> **Note**
>
> Von verwaltete Backups AWS Backup gelten als manuelle DB-Snapshots, werden aber nicht auf das DB-Snapshot-Kontingent für RDS angerechnet. Backups, die mit erstellt wurden, AWS Backup haben Namen, die auf `awsbackup:backup-job-number` enden.

Weitere Informationen zu AWS Backup finden Sie im AWS Backup Entwicklerhandbuch.

Um Backups anzuzeigen, die verwaltet werden von AWS Backup

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie die Registerkarte Backup-Service.

   Ihre AWS Backup Backups werden unter Backup-Dienst-Snapshots aufgeführt.

Metriken in einer RDS Amazon-Instance überwachen

In den folgenden Abschnitten finden Sie einen Überblick über die RDS Amazon-Überwachung und eine Erklärung, wie Sie auf Metriken zugreifen können. Weitere Informationen zum Überwachen von Ereignissen, Protokollen und Datenbankaktivitäts-Streams finden Sie unter Überwachung von Ereignissen, Protokollen und Streams in einem Amazon RDS Amazon.

Themen

- Überwachungsplan
- Leistungsbasislinie
- Richtlinien zur Leistung
- Überwachungstools für Amazon RDS
- Status der anzeigen
- Empfehlungen von Amazon RDS
- Metriken in der RDS Amazon-Konsole anzeigen
- Kombinierte Metriken mit dem Performance Insights Insights-Dashboard anzeigen
- Überwachung von Amazon RDS mit Amazon CloudWatch
- Überwachung von Amazon RDS mit CloudWatch Database Insights
- Überwachung mit Performance Insights auf Amazon RDS
- Analysieren von Leistungsanomalien mit Amazon DevOps Guru für Amazon RDS
- Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung)
- Metrik-Referenz für Amazon RDS

Überwachungsplan

Bevor Sie mit der Überwachung von Amazon RDS beginnen, erstellen Sie einen Überwachungsplan. Dieser Plan sollte die folgenden Fragen beantworten:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen möchten Sie überwachen?

- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungs-Tools möchten Sie verwenden?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Leistungsbasislinie

Um Ihre Überwachungsziele zu erreichen, müssen Sie eine Basislinie erstellen. Messen Sie dazu die Leistung unter verschiedenen Lastbedingungen zu verschiedenen Zeiten in Ihrer Amazon RDS Amazon . Sie können Metriken wie die folgenden überwachen:

- Netzwerkdurchsatz
- Client-Verbindungen
- I/O für Lese-, Schreib- oder Metadatenoperationen
- Burst-Guthaben für Ihre DB-Instances

Wir empfehlen, historische Leistungsdaten für Amazon RDS Amazon zu speichern. Mithilfe der gespeicherten Daten können Sie die aktuelle Leistung mit früheren Trends vergleichen. Sie können so auch normale Leistungsmuster von Anomalien unterscheiden und Methoden zur Behebung von Problemen entwickeln.

Richtlinien zur Leistung

Die zulässigen Werte für Leistungsmetriken sind im Allgemeinen davon abhängig, wie die Basislinie aussieht, und wofür die Anwendung gedacht ist. Prüfen Sie, ob dauerhafte oder tendenzielle Abweichungen von Ihrer Ausgangsbasis vorliegen. Die folgenden Metriken sind häufig die Ursache von Leistungsproblemen:

- Hoch CPU oder RAM Verbrauch — Hohe Werte für CPU oder RAM Verbrauch können angemessen sein, wenn sie Ihren Zielen für Ihre Anwendung (wie Durchsatz oder Parallelität) entsprechen und erwartet werden.
- Nutzung des Datenträgerplatzes – Überprüfen Sie die Nutzung des Datenträgerplatzes, wenn konsistent 85 Prozent oder mehr des gesamten Datenträgerplatzes belegt werden. Prüfen Sie, ob Daten in der Instance gelöscht oder auf einem anderen System archiviert werden können, um Speicherplatz freizugeben.

- Netzwerkdatenverkehr – Wenden Sie sich an Ihren Systemadministrator, um zu erfahren, welcher Durchsatz für Ihr Domänennetzwerk und Ihre Internetverbindung erwartet wird. Überprüfen Sie den Netzwerkdatenverkehr, wenn der Durchsatz dauerhaft unter dem erwarteten Wert liegt.
- Datenbankverbindungen – Sie sollten eine Einschränkung der Datenbankverbindungen in Betracht ziehen, wenn eine große Anzahl von Benutzerverbindungen sowie auch eine Abnahme der Instance-Leistung und -Reaktionszeit zu erkennen sind. Die optimale Anzahl der Benutzerverbindungen für Ihre DB-Instance ist von der Instance-Klasse und der Komplexität der ausgeführten Operationen abhängig. Um die Anzahl der Datenbankverbindungen zu bestimmen, ordnen Sie Ihre DB-Instance einer Parametergruppe zu, für die der Parameter `User Connections` auf einen anderen Wert als „0“ (unbegrenzt) gesetzt ist. Sie können eine entweder eine vorhandene Parametergruppe verwenden oder eine neue erstellen. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
- IOPSMetriken — Die erwarteten Werte für IOPS Messwerte hängen von der Festplattenspezifikation und der Serverkonfiguration ab. Verwenden Sie also Ihre Ausgangswerte, um herauszufinden, was typisch ist. Prüfen Sie, ob dauerhafte Abweichungen von den Werten Ihrer Ausgangsbasis vorliegen. Um eine optimale IOPS Leistung zu erzielen, sollten Sie sicherstellen, dass Ihr typisches Arbeitssatz in den Arbeitsspeicher passt, um Lese- und Schreibvorgänge zu minimieren.

Wenn die Leistung außerhalb der festgelegten Baseline liegt, müssen Sie möglicherweise Änderungen vornehmen, um die Datenbankverfügbarkeit für Ihre Workload zu optimieren. Beispielsweise müssen Sie möglicherweise die Instance-Klasse Ihrer DB-Instance ändern. Oder Sie müssen möglicherweise die Anzahl der DB-Instance und Read Replicas ändern, die für Clients verfügbar sind.

Überwachungstools für Amazon RDS

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon RDS und Ihren anderen AWS Lösungen. AWS bietet verschiedene Überwachungstools, um Amazon RDS Amazon zu beobachten, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen.

Themen

- Automatisierte Überwachungstools
- Manuelle Überwachungstools

Automatisierte Überwachungstools

Wir empfehlen, dass Sie die Überwachungsaufgaben möglichst automatisieren.

Themen

- RDSAmazon-Instance und Empfehlungen
- CloudWatch Amazon-Metriken für Amazon RDS
- Amazon RDS Performance Insights und Betriebssystemüberwachung
- Integrierte Services

RDSAmazon-Instance und Empfehlungen

Sie können die folgenden automatisierten Tools verwenden, um Amazon RDS Amazon anzusehen und zu melden, wenn etwas nicht stimmt:

- Amazon RDS — Details zum aktuellen Status Ihres anzeigen, indem Sie die RDS Amazon-Konsole verwenden, AWS CLI, oder der RDSAPI.
- Amazon RDS Amazon — Reagieren Sie auf automatisierte Empfehlungen für Datenbankressourcen wie DB-Instances, Read Replicas und . Weitere Informationen finden Sie unter Empfehlungen von Amazon RDS.

CloudWatch Amazon-Metriken für Amazon RDS

Amazon RDS lässt sich CloudWatch für zusätzliche Überwachungsfunktionen in Amazon integrieren.

- Amazon CloudWatch — Dieser Service überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen AWS in Echtzeit. Sie können die folgenden CloudWatch Amazon-Funktionen mit Amazon RDS Amazon verwenden:
  - CloudWatch Amazon-Metriken — RDSAmazon sendet automatisch Metriken an CloudWatch jede Minute für jede aktive Datenbank. Es fallen keine zusätzlichen Gebühren für RDS Amazon-Metriken an CloudWatch. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mit Amazon CloudWatch.
  - CloudWatch Amazon-Alarme — Sie können sich eine einzelne Amazon RDS Amazon über einen bestimmten Zeitraum ansehen. Anschließend können Sie eine oder mehrere Aktionen basierend auf dem Wert der Metrik relativ zu einem von Ihnen festgelegten Schwellenwert ausführen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mit Amazon CloudWatch.

Amazon RDS Performance Insights und Betriebssystemüberwachung

Sie können die folgenden automatisierten Tools verwenden, um die Leistung von Amazon RDS zu überwachen:

- Amazon RDS Performance Insights — Beurteilen Sie die Auslastung Ihrer Datenbank und legen Sie fest, wann und wo Maßnahmen ergriffen werden müssen. Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.
- Amazon RDS Enhanced Monitoring — Sehen Sie sich Metriken für das Betriebssystem in Echtzeit an. Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Integrierte Services

Folgendes AWS Dienste sind in Amazon RDS Amazon integriert:

- Amazon EventBridge ist ein serverloser Event-Bus-Service, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. Weitere Informationen finden Sie unter Überwachung von Amazon-Ereignissen.
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien von Amazon RDS Amazon und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS Amazon.

- AWS CloudTrailerfasst API Anrufe und damit verbundene Ereignisse, die von oder im Namen Ihres AWS-Konto und übermittelt die Protokolldateien an einen Amazon S3 S3-Bucket, den Sie angeben. Weitere Informationen finden Sie unter überwachen Amazon RDS API ruft an AWS CloudTrail.
- Database Activity Streams ist eine Amazon RDS Amazon , die einen near-real-time Stream der Aktivitäten in Ihrer bereitstellt. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mithilfe von Datenbankaktivitätsstreams.

Manuelle Überwachungstools

Sie müssen die Elemente, die von den CloudWatch Alarmen nicht abgedeckt werden, manuell überwachen. Der Amazon RDS CloudWatch, AWS Trusted Advisor und andere AWS Konsolen-Dashboards bieten einen at-a-glance Überblick über den Status Ihres AWS Umgebung. Wir empfehlen, auch die Protokolldateien auf Ihrer DB-Instance zu überprüfen.

- Von der RDS Amazon-Konsole aus können Sie die folgenden Elemente für Ihre Ressourcen überwachen:
  - Die Anzahl der Verbindungen zu einer DB-Instance
  - Die Anzahl der Lese- und Schreibvorgänge in einer DB-Instance
  - Der von einer DB-Instance aktuell verwendete Speicherplatz
  - Die Größe des Speichers und die Menge, die für eine DB-Instance verwendet CPU wird
  - Der Netzwerkdatenverkehr von und zu einer DB-Instance
- Aus dem Trusted Advisor Auf dem Dashboard können Sie die folgenden Prüfungen zur Kostenoptimierung, Sicherheit, Fehlertoleranz und Leistungsverbesserung überprüfen:
  - Amazon RDS Idle DB-Instances
  - Zugriffsrisiko RDS für Amazon Security Group
  - RDSAmazon-Backups
  - Amazon RDS Multi-AZ

  Weitere Informationen zu diesen Prüfungen finden Sie unter Trusted Advisor – bewährte Methoden (Prüfungen).
- CloudWatch Die Startseite zeigt:
  - Aktuelle Alarme und Status
  - Diagramme mit Alarmen und Ressourcen
  - Servicestatus

Darüber hinaus können CloudWatch Sie Folgendes verwenden:

- Erstellen Sie benutzerdefinierte Dashboards zur Überwachung der Services, die Ihnen wichtig sind.
- Aufzeichnen von Metrikdaten, um Probleme zu beheben und Trends zu erkennen.
- Suchen und durchsuchen Sie all Ihre AWS Ressourcen-Metriken.
- Erstellen und Bearbeiten von Alarmen, um über Probleme benachrichtigt zu werden.

Status der anzeigen

Mit der Amazon RDS-Konsole können Sie schnell auf den Status Ihrer zugreifen.

Themen

- Anzeigen von Amazon RDS DB-Instance-Status

Anzeigen von Amazon RDS DB-Instance-Status

Der Status einer DB-Instance zeigt den Zustand der DB-Instance an. Sie können die folgenden Verfahren verwenden, um den DB-Instance-Status in der Amazon RDS-Konsole, im AWS CLI Befehl oder im API-Vorgang anzuzeigen.

> **Note**
>
> Amazon RDS führt daneben noch einen anderen Statuswert, der als Wartungsstatus bezeichnet wird, und der in der Spalte Wartung in der Amazon RDS-Konsole angezeigt wird. Dieser Wert gibt den Status der Wartungspatches an, die auf der DB-Instance installiert werden müssen. Der Wartungsstatus ist von dem Status der DB-Instance unabhängig. Weitere Informationen zum Wartungsstatus finden Sie unter Updates auf einen anwenden.

In der folgenden Tabelle finden Sie die möglichen Statuswerte für DB-Instances. Diese Tabelle zeigt auch, ob Ihnen die DB-Instance und der Speicher in Rechnung gestellt werden, die nur für Speicher in Rechnung gestellt oder nicht in Rechnung gestellt werden. Für alle DB-Instance-Status wird immer die Sicherungsnutzung berechnet.

DB-Instance-Status	Berechnet	Beschreibung
available	Berechnet	Die DB-Instance ist fehlerfrei und verfügbar.
backing-up	Berechnet	Die DB-Instance wird derzeit gesichert.
configuring-enhanced-monitoring	Berechnet	Für diese DB-Instance wird „Enhanced Monitoring“ (Erweiterte Überwachung) aktiviert oder deaktiviert.
configuring-iam-database-auth	Berechnet	AWS Identity and Access Management Die (IAM-) Datenbankauthentifizierung wird für diese DB-Instance aktiviert oder deaktiviert.
configuring-log-exports	Berechnet	Das Veröffentlichen von Protokolldateien in Amazon CloudWatch Logs wird für diese DB-Instance aktiviert oder deaktiviert.

DB-Instance-Status	Berechnet	Beschreibung
converting-to-vpc	Berechnet	Die sich nicht in einer Amazon Virtual Private Cloud (Amazon VPC) befindliche DB-Instance wird in eine DB-Instance umgewandelt, die sich in einer Amazon VPC befindet.
creating	Nicht in Rechnung gestellt (kein PITR) Abgerechnet (nur PITR)	Die DB-Instance wird gerade erstellt. Während die DB-Instance erstellt wird, kann nicht auf sie zugegriff en werden. Wenn Sie eine Datenbank während der point-in-time Wiederherstellung (PITR) wiederherstellen, wird Ihnen die Rechnung in Rechnung gestellt, während sich die Datenbank im Erstellungszustand befindet. Dies ist das einzige Szenario, in dem für den Erstellungsstatus Gebühren anfallen.
delete-precheck	Nicht berechnet	Amazon RDS überprüft, ob Lesereplikate fehlerfrei sind und sicher gelöscht werden können.
deleting	Nicht berechnet	Die DB-Instance wird gerade gelöscht.
failed	Nicht berechnet	Die DB-Instance befindet sich im Fehlerzustand und Amazon RDS kann sie nicht wiederherstellen. Führen Sie eine point-in-time Wiederherstellung auf den letzten wiederherstellbaren Zeitpunkt der DB-Instance durch, um die Daten wiederherzustellen.
inaccessible-encry ption-credentials	Nicht berechnet	Die zum Verschlüsseln oder Entschlüsseln der DB-Instance AWS KMS key verwendete Datei kann nicht abgerufen oder wiederhergestellt werden.

DB-Instance-Status	Berechnet	Beschreibung
inaccessible-encryption-credentials-recoverable	Berechnet für Speicher	Es ist kein Zugriff auf den KMS-Schlüssel möglich, der zum Ver- oder Entschlüsseln der DB-Instance verwendet wird. Wenn der KMS-Schlüssel jedoch aktiv ist, kann ein Neustart der DB-Instance ihn wiederherstellen. Weitere Informationen finden Sie unter Verschlüsseln einer DB-Instance.
inkompatibel-erstellen	Nicht berechnet	Amazon RDS versucht, eine DB-Instance zu erstellen, kann dies jedoch nicht tun, da Ressourcen mit Ihrer DB-Instance nicht kompatibel sind. Dieser Status kann beispielsweise auftreten, wenn das Instance-Profil für Ihre DB-Instance nicht über die richtigen Berechtigungen verfügt.
incompatible-network	Nicht berechnet	Amazon RDS versucht, auf der DB-Instance eine Wiederherstellungsaktion durchzuführen, was bislang gescheitert ist, weil der Status der VPC die Durchführung der Aktion verhindert. Dieser Status kann beispielsweise eintreten, wenn alle IP-Adressen in einem Subnetz belegt sind und Amazon RDS keine IP-Adresse für die DB-Instance abrufen kann.
incompatible-option-group	Berechnet	Amazon RDS hat versucht, eine Optionsgruppenänderung zu übernehmen, konnte den Vorgang jedoch nicht abschließen, und Amazon RDS kann keinen Rollback auf den vorherigen Optionsgruppenstatus durchführen. Weitere Informationen enthält die Liste Recent Events (Aktuelle Ereignisse) für die DB-Instance. Dieser Status kann beispielsweise eintreten, wenn eine Optionsgruppe eine Option wie TDE und die DB-Instance keine verschlüsselten Daten enthält.

DB-Instance-Status	Berechnet	Beschreibung
incompatible-parameters	Berechnet	Amazon RDS kann die DB-Instance nicht hochfahren, weil in der DB-Parametergruppe der DB-Instance angegebene Parameter nicht mit der DB-Instance kompatibel sind. Machen Sie die letzten Änderungen an den Parametern der DB-Instance rückgängig, um wieder auf die DB-Instance zugreifen zu können. Beachten Sie für weitere Informationen zu den inkompatiblen Parametern die Liste Recent Events (Aktuelle Ereignisse) für die DB-Instance.
incompatible-restore	Nicht berechnet	Amazon RDS kann keine point-in-time Wiederherstellung durchführen. Häufige Ursachen für diesen Status sind die Verwendung temporärer Tabellen die Verwendung von MyISAM-Tabellen mit MySQLoder die Verwendung von Aria-Tabellen mit MariaDB.
unzureichende Kapazität	Nicht berechnet	Amazon RDS kann Ihre Instance nicht erstellen, da derzeit nicht genügend Kapazität verfügbar ist. Um Ihre DB-Instance in derselben AZ mit demselben Instance-Typ zu erstellen, löschen Sie Ihre DB-Instance, warten Sie ein paar Stunden und versuchen Sie erneut zu erstellen. Alternativ können Sie eine neue Instance mit einer anderen Instance-Klasse oder AZ erstellen.
maintenance	Berechnet	Amazon RDS installiert auf der DB-Instance ein Wartungsupdate. Dieser Status wird bei Wartungen auf der Ebene der Instance verwendet, die von RDS lange im Voraus geplant werden.
modifying	Berechnet	Die DB-Instance wird aufgrund einer Kundenanfrage geändert.
moving-to-vpc	Berechnet	Die DB-Instance wird in eine neue Amazon Virtual Private Cloud (Amazon VPC) verschoben.

DB-Instance-Status	Berechnet	Beschreibung
rebooting	Berechnet	Die DB-Instance wird aufgrund einer Kundenanfrage oder eines Amazon RDS-Prozesses neu gestartet.
resetting-master-credentials	Berechnet	Die Masteranmeldeinformationen für die DB-Instance werden auf eine Kundenanfrage hin zurückgesetzt.
renaming	Berechnet	Die DB-Instance wird aufgrund einer Kundenanfrage umbenannt.
restore-error	Berechnet	Bei dem Versuch, einen Snapshot point-in-time oder aus einem Snapshot wiederherzustellen, ist bei der DB-Instance ein Fehler aufgetreten.
starting	Berechnet für Speicher	Die DB-Instance wird gestartet.
stopped	Berechnet für Speicher	Die DB-Instance ist angehalten.
Wird angehalten …	Berechnet für Speicher	Die DB-Instance wird gestoppt.
storage-config-upgrade	Berechnet	Die Konfiguration des Speicherdateisystems der DB-Instance wird aktualisiert. Dieser Status gilt nur für grüne Datenbanken innerhalb einer Blau/Grün-Bereitstellung oder für DB-Instance-Lesereplikate.
storage-full	Berechnet	Die DB-Instance hat die Speicherkapazitäts zuteilung erreicht. Dies ist ein kritischer Status, wir empfehlen deshalb eine umgehende Behebung dieses Problems. Vergrößern Sie zu diesem Zweck den verfügbaren Speicher, indem Sie die DB-Instance ändern. Um diese Situation zu vermeiden, stellen Sie CloudWatch Amazon-Alarme so ein, dass Sie gewarnt werden, wenn der Speicherplatz knapp wird.

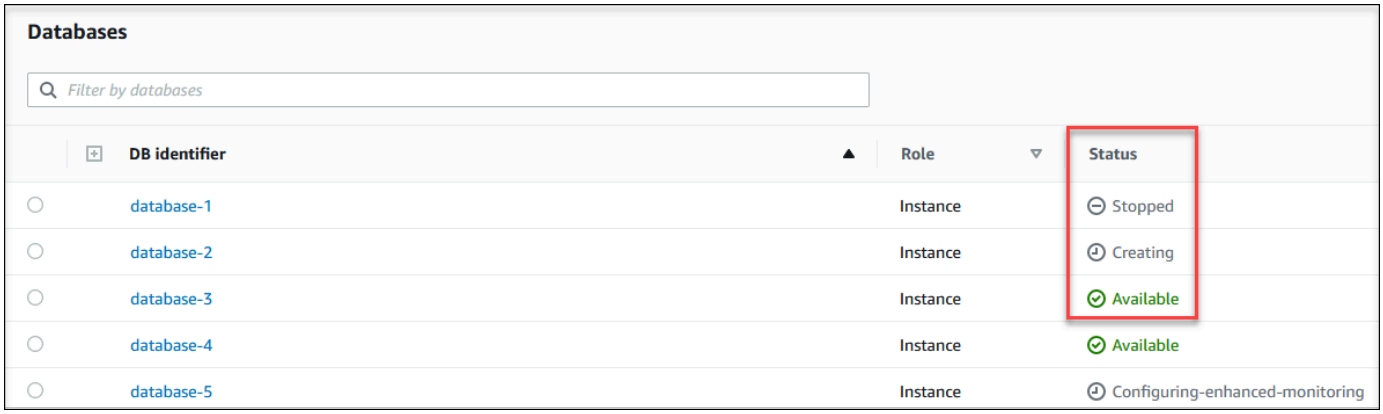
DB-Instance-Status	Berechnet	Beschreibung
Speicherinitialisierung	Berechnet	Die DB-Instance lädt Datenblöcke aus Amazon S3, um die Volume-Leistung nach der Wiederherstellung aus einem Snapshot zu optimieren. Sie bleibt für den Betrieb verfügbar, aber die Leistung ist möglicher weise erst dann voll ausgeschöpft, wenn die Initialis ierung abgeschlossen ist.
Speicheroptimierung	Berechnet	Amazon RDS optimiert den Speicher Ihrer DB-Instan ce. Der Prozess der Speicheroptimierung ist in der Regel kurz, kann aber manchmal bis zu und sogar über 24 Stunden dauern. Während der Speicheroptimierung bleibt die DB-Instance verfügbar. Die Speicheroptimierung ist ein Hintergrundprozess, der die Verfügbarkeit der Instance nicht beeinträchtigt.
upgrading	Berechnet	Die Version der Datenbank-Engine oder des Betriebssystems wird aktualisiert.

Konsole

So zeigen Sie den Status einer DB-Instance an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.

   Die Seite „Datenbanken“ wird zusammen mit der Liste der DB-Instances angezeigt. Für jede DB-Instance wird der Statuswert angezeigt.



CLI

Verwenden Sie den describe-db-instancesBefehl, um die AWS CLI DB-Instance und ihre Statusinformationen mithilfe von anzuzeigen. Der folgende AWS CLI Befehl listet beispielsweise alle Informationen zu DB-Instances auf.

```
aws rds describe-db-instances
```

Um eine bestimmte DB-Instance und ihren Status anzuzeigen, rufen Sie den describe-db-instancesBefehl mit der folgenden Option auf:

- `DBInstanceIdentifier` – der Name der DB-Instance

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Um nur den Status aller DB-Instances anzuzeigen, verwenden Sie die folgende Abfrage in AWS CLI.

```
aws rds describe-db-instances --query 'DBInstances[*].
[DBInstanceIdentifier,DBInstanceStatus]' --output table
```

API

Rufen Sie den DBInstances Vorgang Describe auf, um den Status der DB-Instance mithilfe der Amazon RDS-API anzuzeigen.

Empfehlungen von Amazon RDS

Amazon RDS bietet automatisierte Empfehlungen für Datenbankressourcen wie DB-Instances, Read Replicas und DB-Parametergruppen. Diese Empfehlungen bieten Anleitungen nach bewährten Methoden, indem sie die DB-Instance-Konfiguration, die Nutzung und die Leistungsdaten analysieren.

Amazon RDS Performance Insights überwacht bestimmte Kennzahlen und erstellt automatisch Schwellenwerte, indem analysiert wird, welche Werte für eine bestimmte Ressource als potenziell problematisch angesehen werden. Wenn neue Metrikwerte über einen bestimmten Zeitraum einen vordefinierten Schwellenwert überschreiten, generiert Performance Insights eine proaktive Empfehlung. Diese Empfehlung trägt dazu bei, future Auswirkungen auf die Datenbankleistung zu verhindern. Beispielsweise wird die Empfehlung „Idle In Transaction“ RDSfür Postgre SQL generiert, wenn die mit der Datenbank verbundenen Sitzungen keine aktive Arbeit verrichten, aber Datenbankressourcen blockiert bleiben können. Um proaktive Empfehlungen zu erhalten, müssen Sie Performance Insights mit einer kostenpflichtigen Aufbewahrungsfrist aktivieren. Informationen zum Aktivieren von Performance Insights finden Sie unterPerformance Insights für Amazon RDS ein- und ausschalten. Informationen zu Preisen und Datenspeicherung für Performance Insights finden Sie unterPreisgestaltung und Datenspeicherung für Performance Insights.

DevOpsGuru for RDS überwacht bestimmte Messwerte, um zu erkennen, wann das Verhalten der Kennzahl äußerst ungewöhnlich oder ungewöhnlich wird. Diese Anomalien werden als reaktive Erkenntnisse mit Empfehlungen gemeldet. DevOpsGuru for RDS könnte Ihnen beispielsweise empfehlen, eine Erhöhung der CPU Kapazität in Betracht zu ziehen oder Warteereignisse zu untersuchen, die zur Datenbankauslastung beitragen. DevOpsGuru for bietet RDS auch proaktive Empfehlungen, die auf Schwellenwerten basieren. Für diese Empfehlungen müssen Sie DevOps Guru für aktivierenRDS. Informationen zum Aktivieren von DevOps Guru für RDS finden Sie unter DevOpsGuru einschalten und die Ressourcenabdeckung angeben.

Empfehlungen erhalten einen der folgenden Status: „Aktiv“, „Abgelehnt“, „Ausstehend“ oder „Gelöst“. Gelöste Empfehlungen sind 365 Tage lang verfügbar.

Sie können die Empfehlungen ansehen oder verwerfen. Sie können eine konfigurationsbasierte aktive Empfehlung sofort anwenden, sie im nächsten Wartungsfenster planen oder sie verwerfen. Für auf Schwellenwerten basierende proaktive und auf maschinellem Lernen basierende reaktive Empfehlungen müssen Sie die vorgeschlagene Ursache des Problems überprüfen und dann die empfohlenen Maßnahmen zur Behebung des Problems ergreifen.

Empfehlungen werden in den folgenden Bereichen unterstützt AWS-Regionen:

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asia Pacific (Mumbai)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irland)
- Europe (London)
- Europe (Paris)
- Europa (Stockholm)
- Südamerika (São Paulo)

In den folgenden Abschnitten erfahren Sie, wie Sie Empfehlungen von Amazon RDS Amazon einsehen, anwenden, ablehnen und ändern können.

Themen

- Amazon RDS Amazon anzeigen
- Anwendung der Amazon RDS Amazon
- Amazon RDS Amazon werden abgelehnt
- Abgelehnte Amazon RDS in aktive Empfehlungen umwandeln
- Empfehlungen von Amazon RDS Amazon Reference

Amazon RDS Amazon anzeigen

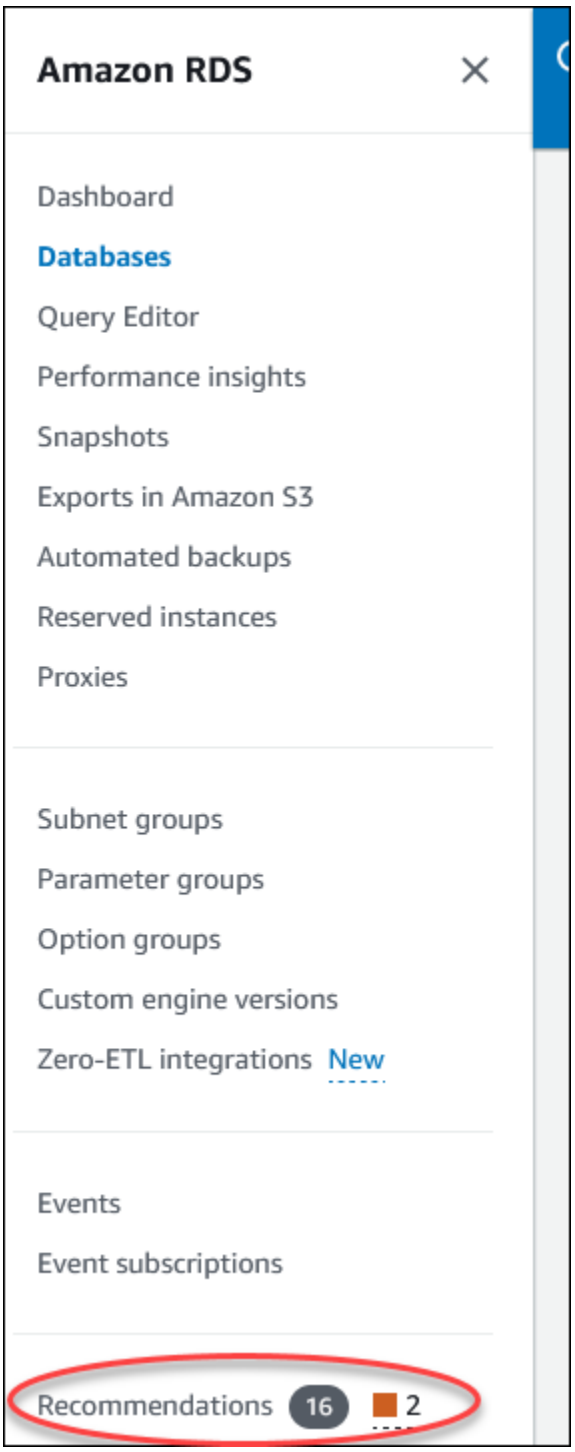
Mithilfe der RDS Amazon-Konsole können Sie Amazon RDS Amazon für Ihre Datenbankressourcen einsehen.

Konsole

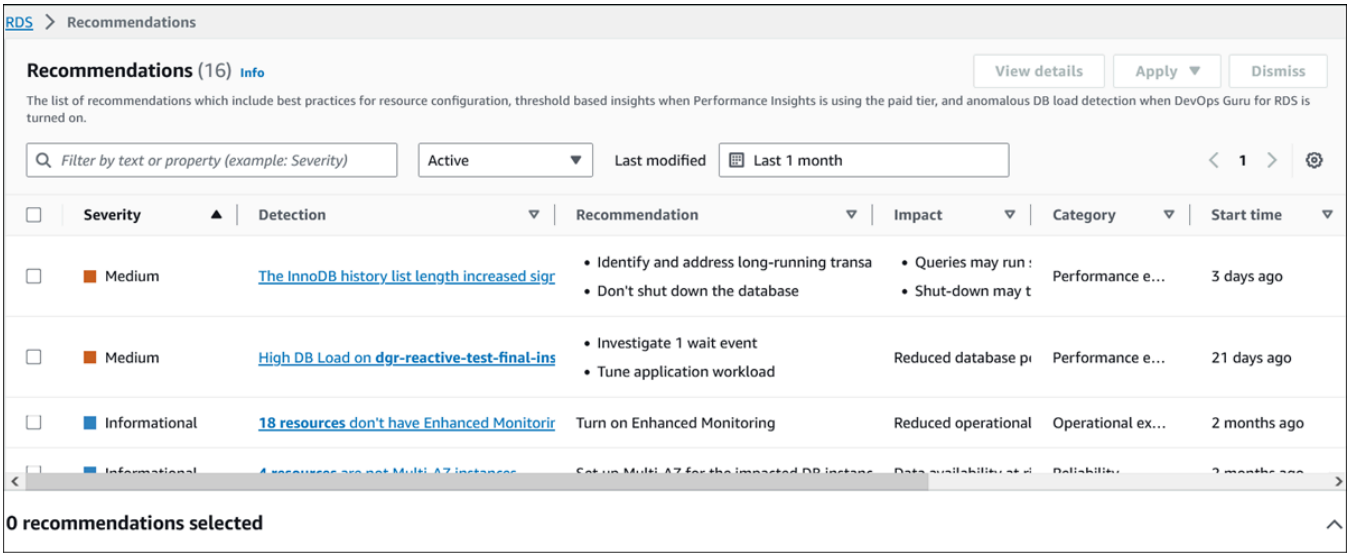
Um die Amazon RDS Amazon

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:

   - Wählen Sie Empfehlungen aus. Die Anzahl der aktiven Empfehlungen für Ihre Ressourcen und die Anzahl der Empfehlungen mit dem höchsten Schweregrad, die im letzten Monat generiert wurden, sind neben Empfehlungen verfügbar. Um die Anzahl der aktiven Empfehlungen für jeden Schweregrad zu ermitteln, wählen Sie die Zahl aus, die den höchsten Schweregrad anzeigt.

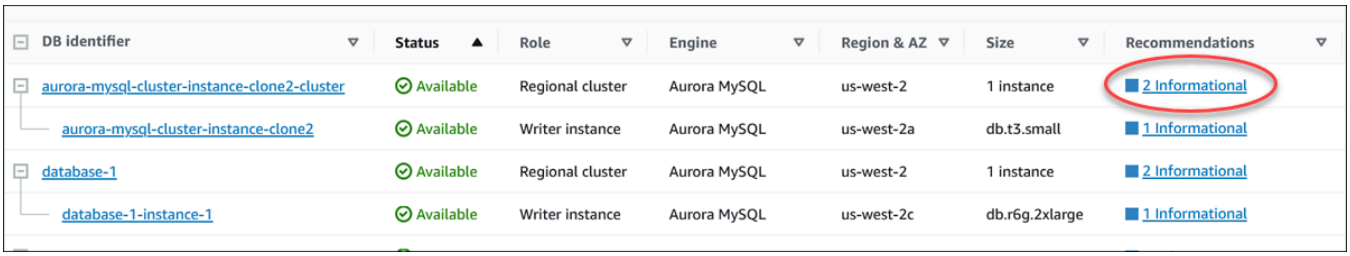


Standardmäßig wird auf der Seite „Empfehlungen“ eine Liste der neuen Empfehlungen des letzten Monats angezeigt. Amazon RDS gibt Empfehlungen für alle Ressourcen in Ihrem Konto und sortiert die Empfehlungen nach ihrem Schweregrad.

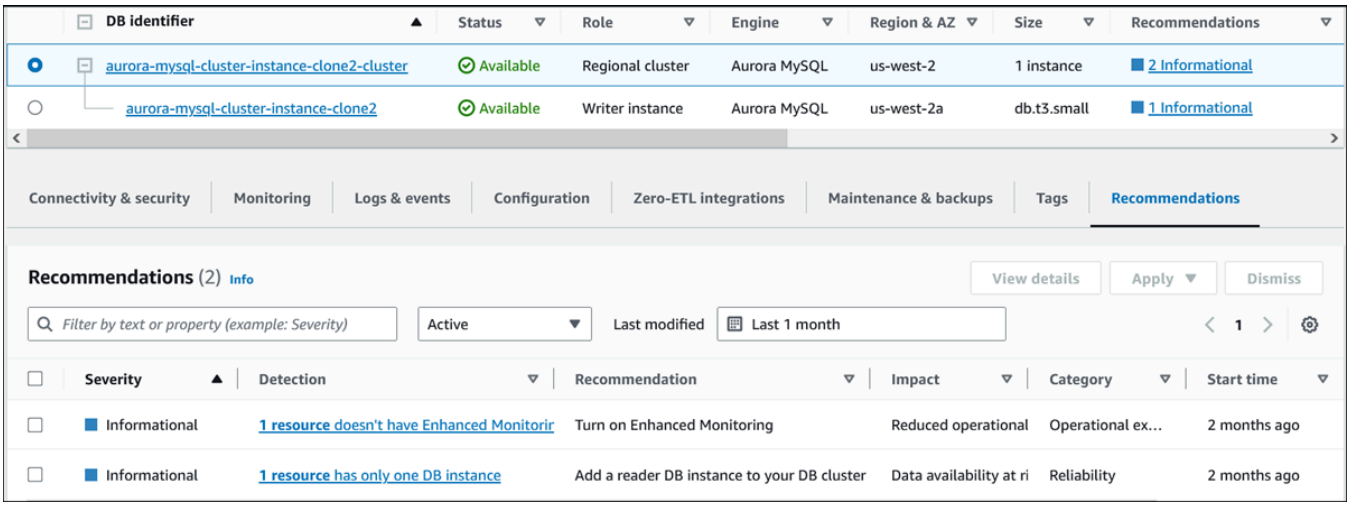


Sie können eine Empfehlung auswählen, um unten auf der Seite einen Abschnitt zu sehen, der die betroffenen Ressourcen und Einzelheiten zur Umsetzung der Empfehlung enthält.

- Wählen Sie auf der Seite Datenbanken die Option Empfehlungen für eine Ressource aus.



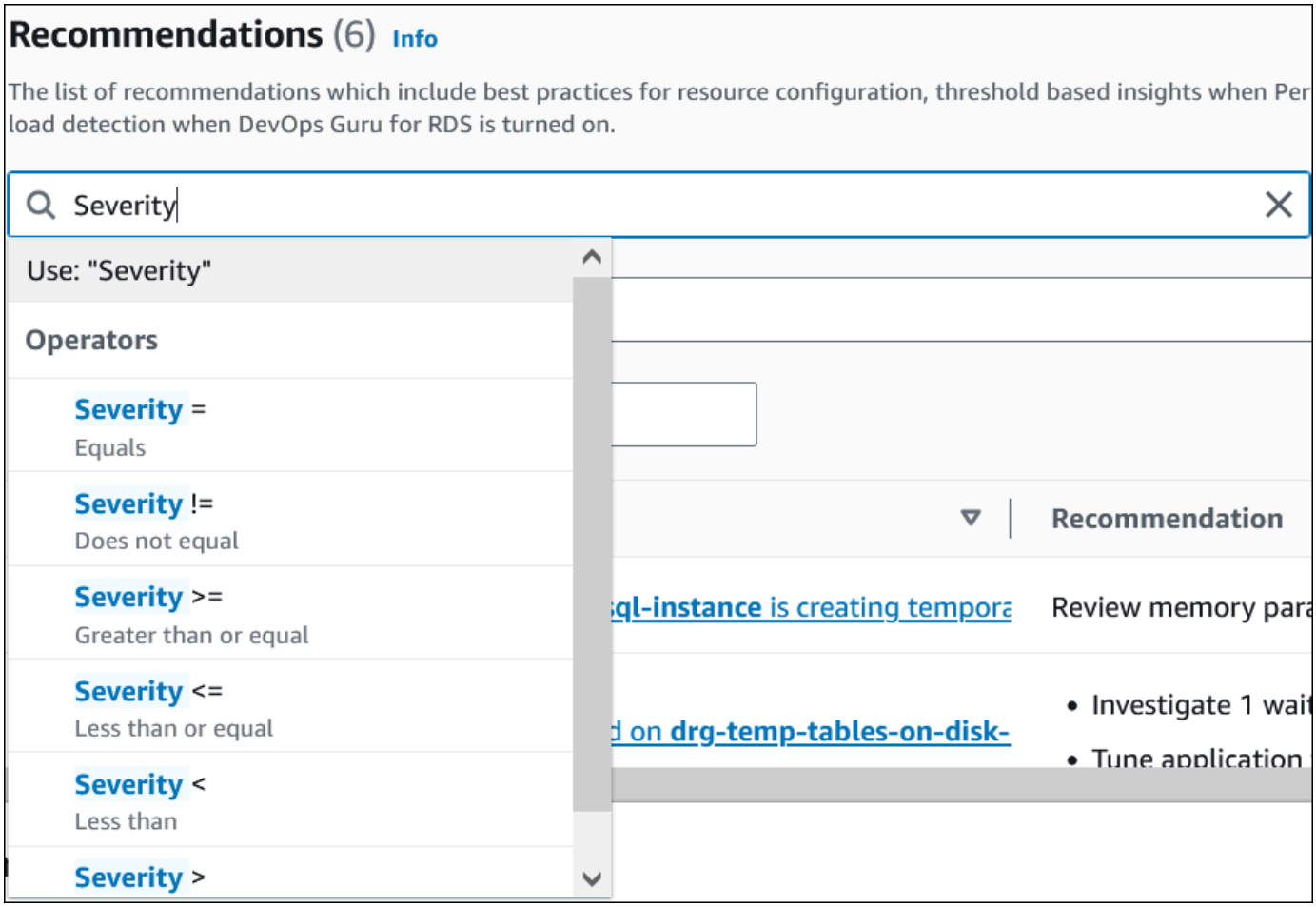
Auf der Registerkarte Empfehlungen werden die Empfehlungen und ihre Details für die ausgewählte Ressource angezeigt.



Die folgenden Informationen sind für die Empfehlungen verfügbar:

- Schweregrad — Das Ausmaß der Auswirkungen des Problems. Die Schweregrade lauten „Hoch“, „Mittel“, „Niedrig“ und „Informativ“.
- Erkennung — Die Anzahl der betroffenen Ressourcen und eine kurze Beschreibung des Problems. Wählen Sie diesen Link, um die Empfehlung und die Analysedetails anzuzeigen.
- Empfehlung — Eine kurze Beschreibung der empfohlenen Maßnahme, die angewendet werden soll.
- Auswirkung — Eine kurze Beschreibung der möglichen Auswirkungen, wenn die Empfehlung nicht angewendet wird.
- Kategorie — Die Art der Empfehlung. Die Kategorien sind Leistungseffizienz, Sicherheit, Zuverlässigkeit, Kostenoptimierung, betriebliche Exzellenz und Nachhaltigkeit.
- Status — Der aktuelle Status der Empfehlung. Die möglichen Status sind Alle, Aktiv, Abgelehnt, Gelöst und Ausstehend.
- Startzeit — Die Zeit, zu der das Problem begann. Zum Beispiel vor 18 Stunden.
- Letzte Änderung — Der Zeitpunkt, zu dem die Empfehlung aufgrund einer Änderung des Schweregrads zuletzt vom System aktualisiert wurde, oder der Zeitpunkt, zu dem Sie auf die Empfehlung geantwortet haben. Zum Beispiel vor 10 Stunden.
- Endzeit — Der Zeitpunkt, zu dem das Problem beendet wurde. Bei anhaltenden Problemen wird die Uhrzeit nicht angezeigt.
- Ressourcen-ID — Der Name einer oder mehrerer Ressourcen.

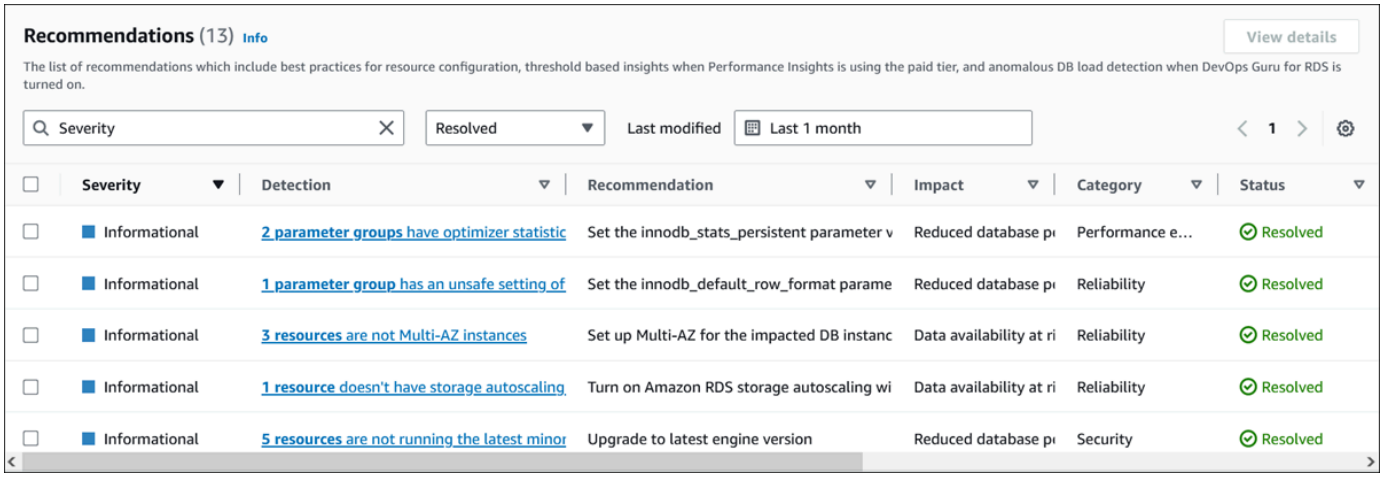
3. (Optional) Wählen Sie in dem Feld die Operatoren Schweregrad oder Kategorie aus, um die Liste der Empfehlungen zu filtern.



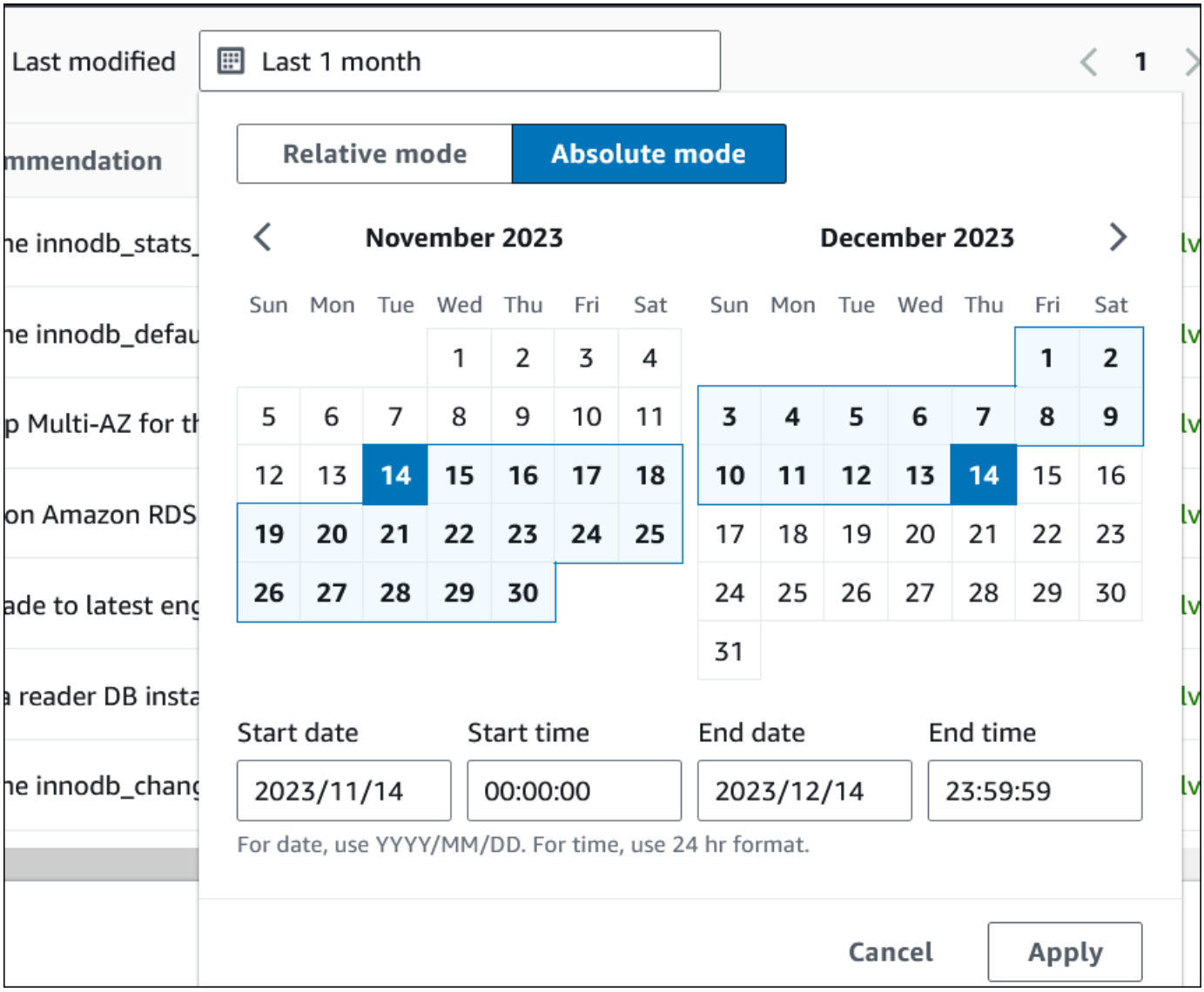
Die Empfehlungen für den ausgewählten Vorgang werden angezeigt.

4. (Optional) Wählen Sie einen der folgenden Empfehlungsstatus:

   - Aktiv (Standard) — Zeigt die aktuellen Empfehlungen an, die Sie anwenden, für das nächste Wartungsfenster planen oder ablehnen können.
   - Alle — Zeigt alle Empfehlungen mit dem aktuellen Status an.
   - Abgelehnt — Zeigt die abgelehnten Empfehlungen an.
   - Gelöst — Zeigt die Empfehlungen an, die gelöst wurden.
   - Ausstehend — Zeigt die Empfehlungen an, deren empfohlene Maßnahmen derzeit ausgeführt werden oder für das nächste Wartungsfenster geplant sind.



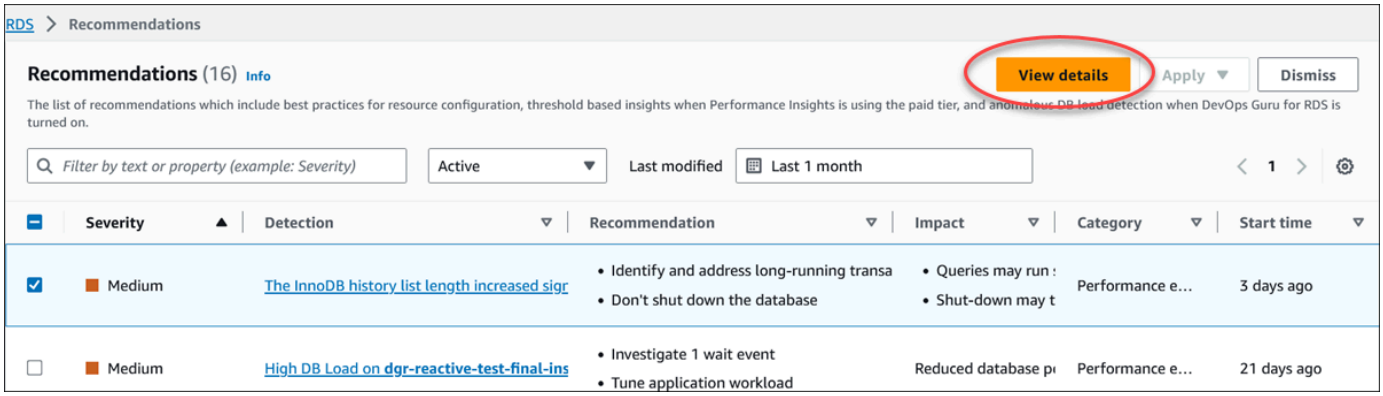
5. (Optional) Wählen Sie unter Letzte Änderung den relativen Modus oder den absoluten Modus, um den Zeitraum zu ändern. Auf der Seite „Empfehlungen“ werden die Empfehlungen angezeigt, die in dem Zeitraum generiert wurden. Der Standardzeitraum ist der letzte Monat. Im Modus Absolut können Sie den Zeitraum wählen oder die Uhrzeit in die Felder Startdatum und Enddatum eingeben.



Die Empfehlungen für den eingestellten Zeitraum werden angezeigt.

Beachten Sie, dass Sie alle Empfehlungen für Ressourcen in Ihrem Konto sehen können, indem Sie den Bereich auf Alle setzen.

6. (Optional) Wählen Sie auf der rechten Seite Einstellungen aus, um die anzuzeigenden Details anzupassen. Sie können ein Seitenformat wählen, die Textzeilen umbrechen und die Spalten zulassen oder ausblenden.
7. (Optional) Wählen Sie eine Empfehlung und dann Details anzeigen aus.



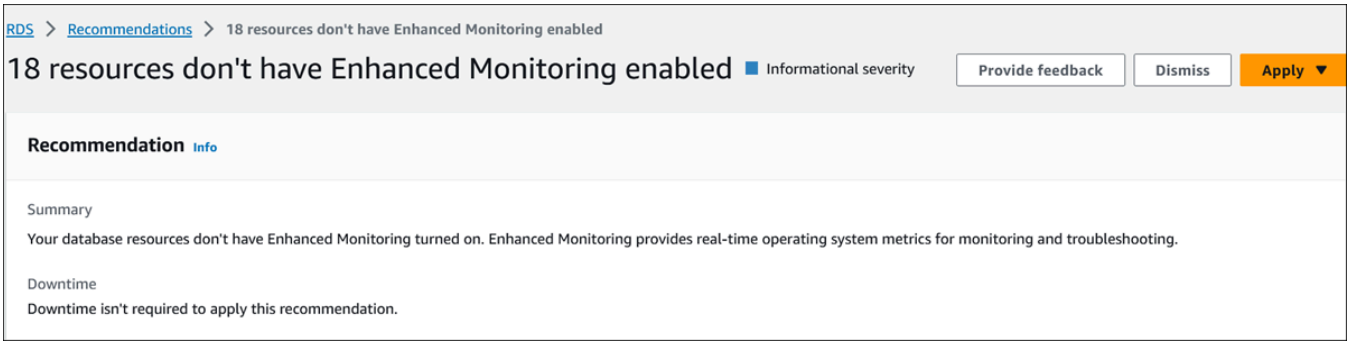
Die Seite mit den Empfehlungsdetails wird angezeigt. Der Titel gibt die Gesamtzahl der Ressourcen an, bei denen das Problem erkannt wurde, und den Schweregrad.

Informationen zu den Komponenten auf der Detailseite für eine auf Anomalien basierende reaktive Empfehlung finden Sie unter Reaktive Anomalien anzeigen im Amazon DevOps Guru-Benutzerhandbuch.

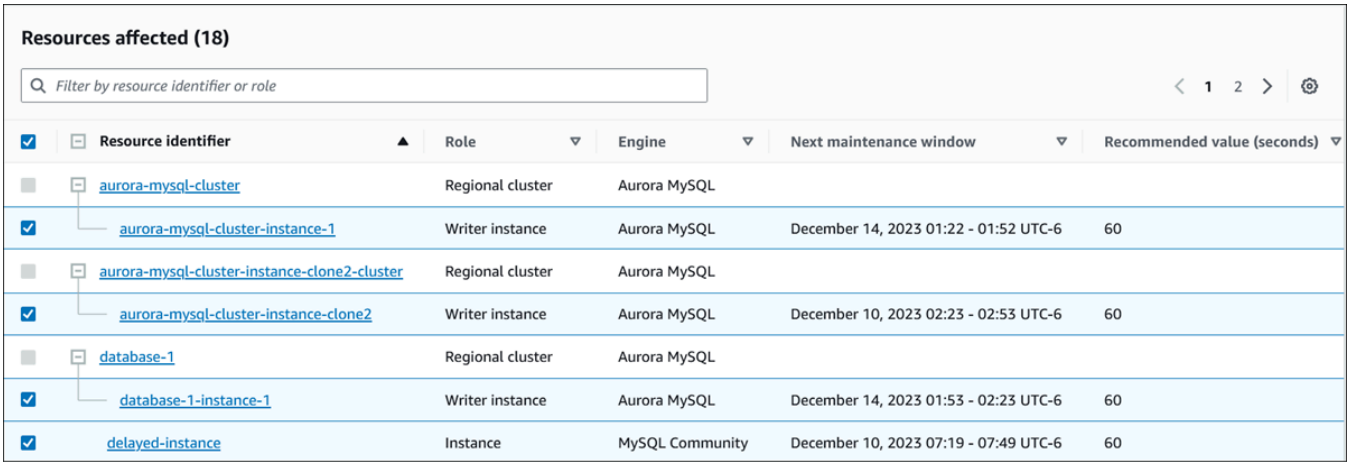
Informationen zu den Komponenten auf der Detailseite für eine auf Schwellenwerten basierende proaktive Empfehlung finden Sie unter. Proaktive Empfehlungen von Performance Insights anzeigen

Bei den anderen automatisierten Empfehlungen werden auf der Seite mit den Empfehlungsdetails die folgenden Komponenten angezeigt:

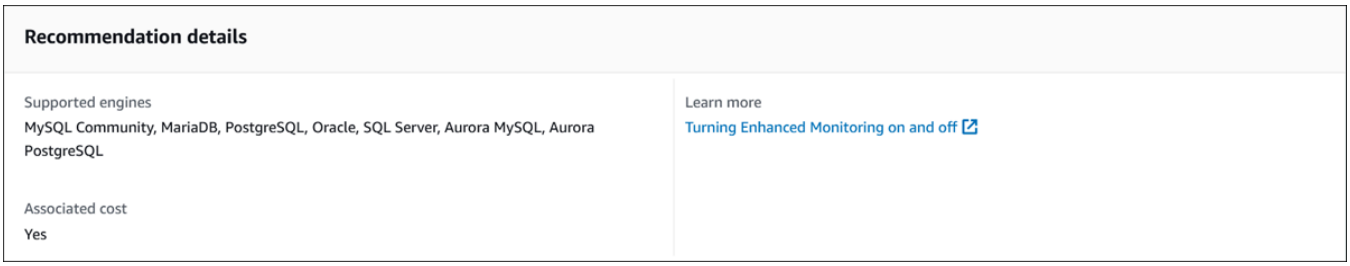
- Empfehlung — Eine Zusammenfassung der Empfehlung und der Angabe, ob Ausfallzeiten erforderlich sind, um die Empfehlung umzusetzen.



- Betroffene Ressourcen — Einzelheiten zu den betroffenen Ressourcen.



- Einzelheiten zur Empfehlung — Informationen zum unterstützten Modul, alle erforderlichen Kosten für die Umsetzung der Empfehlung und Link zur Dokumentation mit weiteren Informationen.



CLI

Verwenden Sie den folgenden Befehl in, um RDS Amazon-Empfehlungen für die anzuzeigen AWS CLI.

```
aws rds describe-db-recommendations
```

RDS API

Verwenden Sie die escribeDBRecommendations Operation D RDS API, um RDS Amazon-Empfehlungen mithilfe von Amazon anzuzeigen.

Anwendung der Amazon RDS Amazon

Um Amazon RDS Amazon über die RDS Amazon-Konsole anzuwenden, wählen Sie auf der Detailseite eine konfigurationsbasierte Empfehlung oder eine betroffene Ressource aus. Wählen Sie dann, ob die Empfehlung sofort angewendet werden soll, oder planen Sie sie für das nächste

Wartungsfenster ein. Die Ressource muss möglicherweise neu gestartet werden, damit die Änderung wirksam wird. Für einige Empfehlungen zu DB-Parametergruppen müssen Sie möglicherweise die Ressourcen neu starten.

Die auf Schwellenwerten basierenden proaktiven oder anomaliebasierten reaktiven Empfehlungen verfügen nicht über die Option Anwenden und müssen möglicherweise zusätzlich überprüft werden.

Konsole

Um eine konfigurationsbasierte Empfehlung anzuwenden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:

   - Wählen Sie Empfehlungen aus.

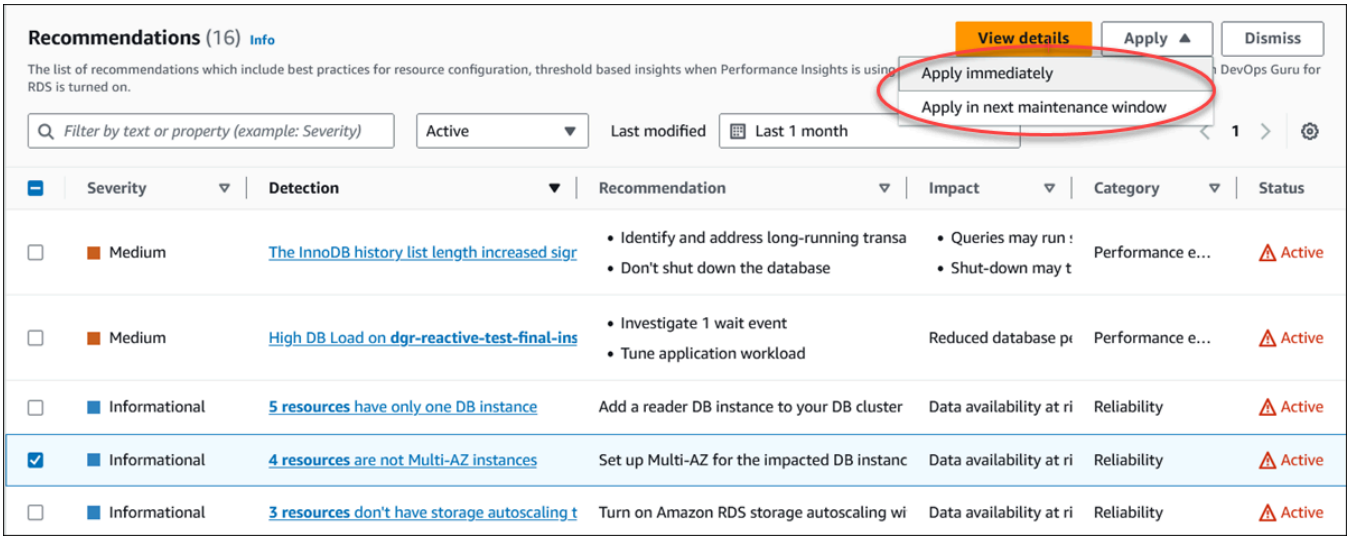
     Die Seite mit den Empfehlungen wird mit der Liste aller Empfehlungen angezeigt.
   - Wählen Sie Datenbanken und dann auf der Datenbankseite Empfehlungen für eine Ressource aus.

     Die Details werden auf der Registerkarte Empfehlungen für die ausgewählte Empfehlung angezeigt.
   - Wählen Sie Erkennung für eine aktive Empfehlung auf der Seite „Empfehlungen“ oder auf der Registerkarte „Empfehlungen“ auf der Datenbankseite aus.

     Die Seite mit den Empfehlungsdetails wird angezeigt.
3. Wählen Sie auf der Seite mit den Empfehlungsdetails eine Empfehlung oder eine oder mehrere betroffene Ressourcen aus und führen Sie einen der folgenden Schritte aus:

   - Wählen Sie „Anwenden“ und anschließend „Sofort anwenden“, um die Empfehlung sofort anzuwenden.
   - Wählen Sie „Anwenden“ und anschließend „Im nächsten Wartungsfenster anwenden“, um einen Termin für das nächste Wartungsfenster festzulegen.

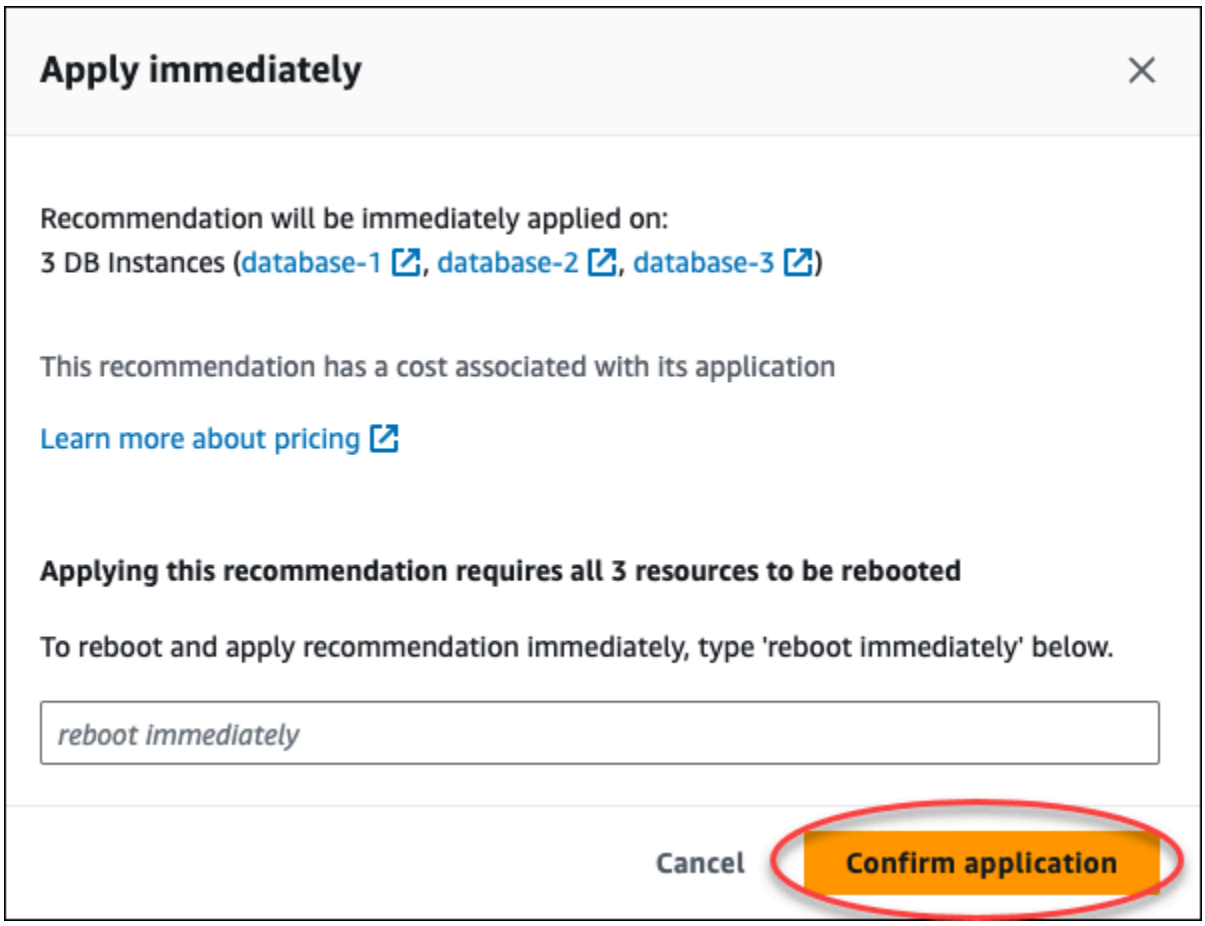
     Der ausgewählte Empfehlungsstatus wird bis zum nächsten Wartungsfenster auf „Ausstehend“ aktualisiert.



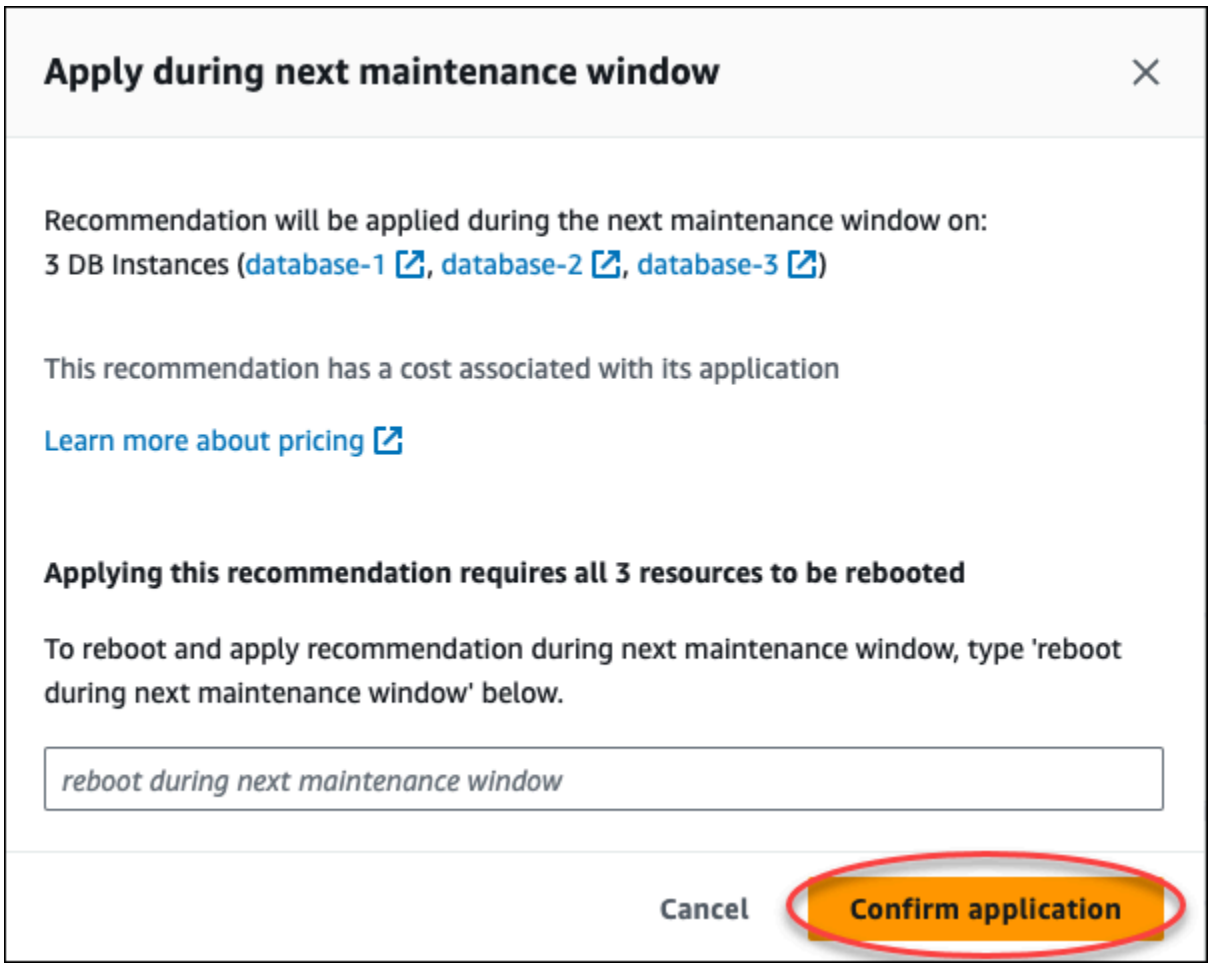
Ein Bestätigungsfenster wird angezeigt.

4. Wählen Sie Bewerbung bestätigen, um die Empfehlung zu übernehmen. In diesem Fenster wird bestätigt, ob die Ressourcen automatisch oder manuell neu gestartet werden müssen, damit die Änderungen wirksam werden.

   Das folgende Beispiel zeigt das Bestätigungsfenster, in dem die Empfehlung sofort angewendet werden kann.



Das folgende Beispiel zeigt das Bestätigungsfenster, um die Anwendung der Empfehlung im nächsten Wartungsfenster zu planen.



Ein Banner zeigt eine Meldung an, wenn die angewendete Empfehlung erfolgreich war oder fehlgeschlagen ist.

Das folgende Beispiel zeigt das Banner mit der erfolgreichen Nachricht.



Das folgende Beispiel zeigt das Banner mit der Fehlermeldung.



RDS API

Um eine konfigurationsbasierte RDS über Amazon anzuwenden RDS API

1. Verwenden Sie die escribeDBRecommendationsD-Operation. Die `RecommendedActions` Ausgabe kann eine oder mehrere empfohlene Aktionen enthalten.
2. Verwenden Sie das RecommendedActionObjekt für jede empfohlene Aktion aus Schritt 1. Die Ausgabe enthält `Operation` und`Parameters`.

   Das folgende Beispiel zeigt die Ausgabe mit einer empfohlenen Aktion.

```
   "RecommendedActions": [
       {
           "ActionId": "0b19ed15-840f-463c-a200-b10af1b552e3",
           "Title": "Turn on auto backup", // localized
           "Description": "Turn on auto backup for my-mysql-instance-1", //
localized
           "Operation": "ModifyDbInstance",
           "Parameters": [
               {
                   "Key": "DbInstanceIdentifier",
                   "Value": "my-mysql-instance-1"
               },
               {
                   "Key": "BackupRetentionPeriod",
                   "Value": "7"
               }
           ],
           "ApplyModes": ["immediately", "next-maintenance-window"],
           "Status": "applied"
       },
       ... // several others
   ],
```

3. Verwenden Sie `operation` für jede empfohlene Aktion aus der Ausgabe in Schritt 2 den Wert und geben Sie die `Parameters` Werte ein.
4. Nachdem der Vorgang in Schritt 2 erfolgreich war, verwenden Sie den odifyDBRecommendation Vorgang M, um den Empfehlungsstatus zu ändern.

Amazon RDS Amazon werden abgelehnt

Sie können eine oder mehrere Amazon RDS Amazon über die RDS Amazon-Konsole oder Amazon RDS API ablehnen. AWS CLI

Konsole

Um eine oder mehrere Empfehlungen abzulehnen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:

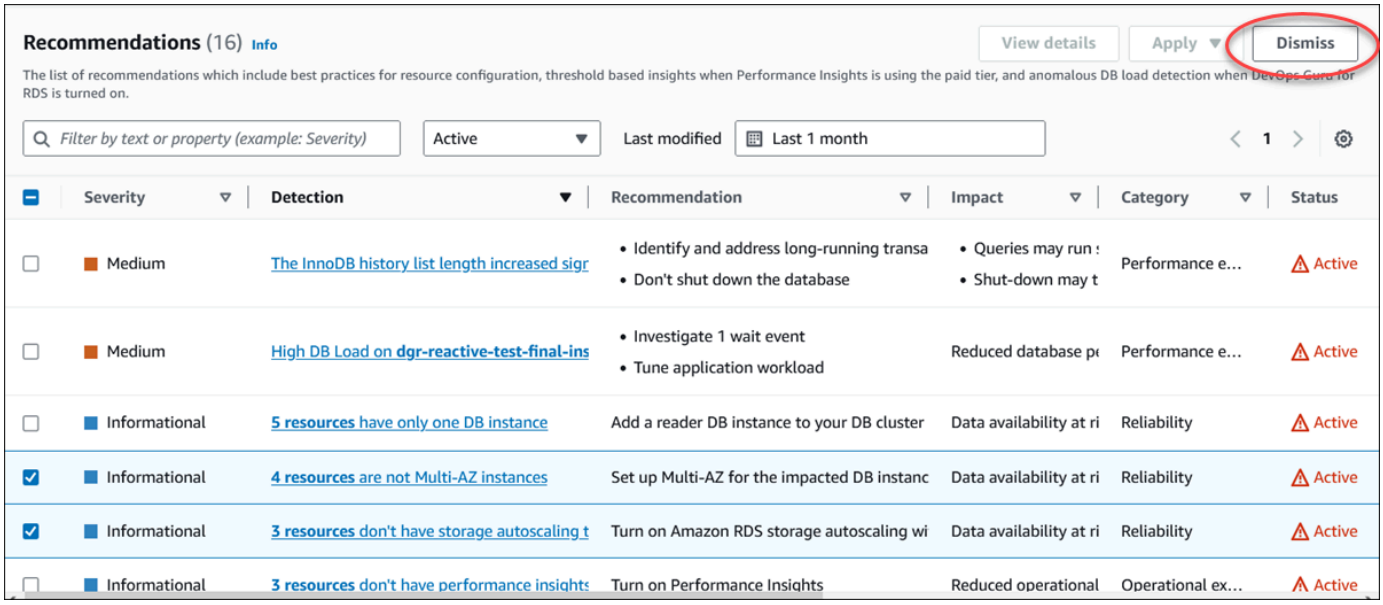
   - Wählen Sie Empfehlungen aus.

     Die Seite mit den Empfehlungen wird mit der Liste aller Empfehlungen angezeigt.
   - Wählen Sie Datenbanken und dann auf der Datenbankseite Empfehlungen für eine Ressource aus.

     Die Details werden auf der Registerkarte Empfehlungen für die ausgewählte Empfehlung angezeigt.
   - Wählen Sie Erkennung für eine aktive Empfehlung auf der Seite „Empfehlungen“ oder auf der Registerkarte „Empfehlungen“ auf der Datenbankseite aus.

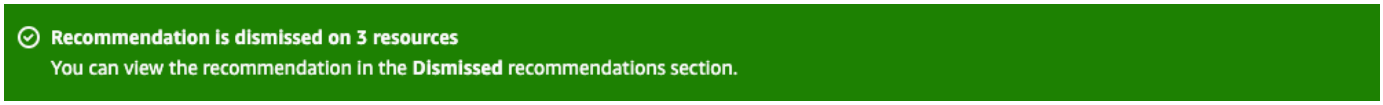
     Auf der Seite mit den Empfehlungsdetails wird die Liste der betroffenen Ressourcen angezeigt.
3. Wählen Sie auf der Seite mit den Empfehlungsdetails eine oder mehrere Empfehlungen oder eine oder mehrere betroffene Ressourcen aus und klicken Sie dann auf Ablehnen.

   Das folgende Beispiel zeigt die Seite mit den Empfehlungen, auf der mehrere aktive Empfehlungen zum Ablehnen ausgewählt wurden.



Ein Banner zeigt eine Meldung an, wenn die ausgewählten oder mehrere Empfehlungen abgelehnt werden.

Das folgende Beispiel zeigt das Banner mit der erfolgreichen Nachricht.



Das folgende Beispiel zeigt das Banner mit der Fehlermeldung.



CLI

Um RDSeine zurückzuweisen, verwenden Sie den AWS CLI

1. Führen Sie den Befehl `aws rds describe-db-recommendations --filters "Name=status,Values=active"` aus.

   Die Ausgabe enthält eine Liste mit Empfehlungen im `active` Status.
2. Suchen Sie `recommendationId` nach der Empfehlung, die Sie aus Schritt 1 verwerfen möchten.

3. Führen Sie den Befehl `>aws rds modify-db-recommendation --status dismissed --recommendationId <ID>` mit dem `recommendationId` aus Schritt 2 aus, um die Empfehlung zu verwerfen.

RDS API

Verwenden Sie die odifyDBRecommendationM-Operation, um RDSeine abzulehnen RDSAPI, die Amazon verwendet.

Abgelehnte Amazon RDS in aktive Empfehlungen umwandeln

Sie können eine oder mehrere abgelehnte Amazon RDS mithilfe der RDS Amazon-Konsole oder Amazon in aktive Empfehlungen verschieben RDSAPI. AWS CLI

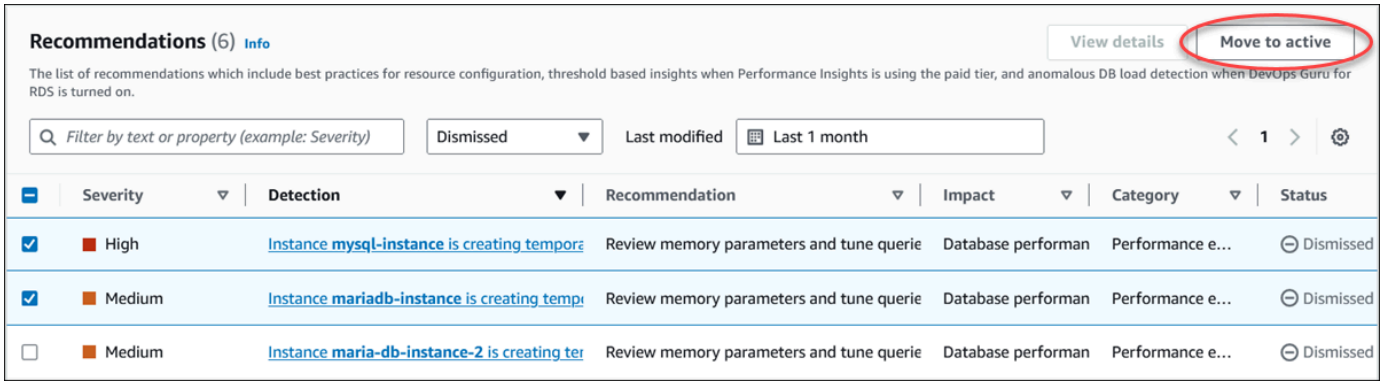
Konsole

Um eine oder mehrere abgelehnte Empfehlungen in aktive Empfehlungen umzuwandeln

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:
   - Wählen Sie Empfehlungen aus.

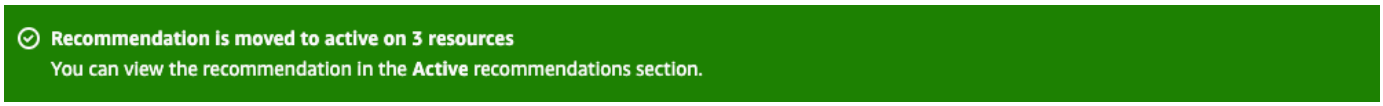
     Auf der Seite „Empfehlungen“ wird eine nach Schweregrad sortierte Liste mit Empfehlungen für alle Ressourcen in Ihrem Konto angezeigt.
   - Wählen Sie Datenbanken und dann auf der Datenbankseite Empfehlungen für eine Ressource aus.

     Auf der Registerkarte Empfehlungen werden die Empfehlungen und ihre Details für die ausgewählte Ressource angezeigt.
3. Wählen Sie eine oder mehrere abgelehnte Empfehlungen aus der Liste aus und klicken Sie dann auf Zur aktiven Empfehlung verschieben.



Ein Banner zeigt eine Erfolgs- oder Fehlschlagsmeldung an, wenn die ausgewählten Empfehlungen vom Status „Abgelehnt“ in den Status „Aktiv“ verschoben werden.

Das folgende Beispiel zeigt das Banner mit der Erfolgsmeldung.



Das folgende Beispiel zeigt das Banner mit der Fehlermeldung.



CLI

Um eine abgelehnte RDS in eine aktive Empfehlung umzuwandeln, verwenden Sie AWS CLI

1. Führen Sie den Befehl `aws rds describe-db-recommendations --filters "Name=status,Values=dismissed"` aus.

   Die Ausgabe enthält eine Liste mit Empfehlungen im `dismissed` Status.
2. Suchen Sie `recommendationId` nach der Empfehlung, deren Status Sie aus Schritt 1 ändern möchten.
3. Führen Sie den Befehl `>aws rds modify-db-recommendation --status active --recommendationId <ID>` mit der Empfehlung `recommendationId` aus Schritt 2 aus, um zur aktiven Empfehlung zu wechseln.

RDS API

Um eine abgelehnte RDS über Amazon in eine aktive Empfehlung RDS API umzuwandeln, verwenden Sie die odifyDBRecommendationM-Operation.

Empfehlungen von Amazon RDS Amazon Reference

Amazon RDS generiert Empfehlungen für eine Ressource, wenn die Ressource erstellt oder geändert wird. In der folgenden Tabelle finden Sie Beispiele für Empfehlungen von RDSAmazon .

Typ	Beschreibung	Empfehlung	Ausfall: it erforde ich	Zusätzliche Informati onen
Magnetisches Volumen wird verwendet	Ihre DB-Instan ces verwenden Magnetspeicher. Magnetspeicher wird für die meisten DB-Instances nicht empfohlen. Wählen Sie einen anderen Speichertyp: Allgemein (SSD) oder Bereitgestellt. IOPS	Wählen Sie einen anderen Speichertyp: Allgemein (SSD) oder IOPS Bereitgestellt.	Ja	Bände der vorherigen Generation in der EC2 Amazon-Dokumentati on.
Automatis ierte Ressource nsicherungen sind deaktivie rt	Automatisierte Backups sind für Ihre DB-Instances nicht aktiviert. Automatis ierte Backups werden empfohlen, da sie die point-in-time Wiederherstellung Ihrer DB-Instances ermöglichen.	Aktivieren Sie automatische Backups mit einer Aufbewahrungsfrist von bis zu 14 Tagen.	Ja	Aktivieren von automatisierten Backups Entmystifizierung der RDS Amazon-Ba ckup-Speicherkoste n im Datenbank-Blog AWS

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
Ein Upgrade der Engine-Nebenversion ist erforderlich	Auf Ihren Datenbankressourcen wird nicht die neueste Nebenversion der DB-Engine ausgeführt. Die neueste Nebenversion enthält die neuesten Sicherheitsupdates und andere Verbesserungen.	Führen Sie ein Upgrade auf die neueste Engine-Version durch.	Ja	Upgrade der Engine-Version für eine DB-Instance
Enhanced Monitoring ist ausgeschaltet	Für Ihre Datenbankressourcen ist Enhanced Monitoring nicht aktiviert. Erweiterte Überwachung bietet Echtzeit-Betriebssystemmetriken für die Überwachung und Fehlerbehebung.	Aktivieren Sie die erweiterte Überwachung.	Nein	Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung)

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
Die Speicherverschlüsselung ist ausgeschaltet	Amazon RDS unterstützt Verschlüsselung im Ruhezustand für alle Datenbank-Engines mithilfe der Schlüssel, die Sie im AWS Key Management Service (AWS KMS) verwalten. Auf einer aktiven DB-Instance mit RDS Amazon-Verschlüsselung werden die im Speicher gespeicherten Daten verschlüsselt, ähnlich wie bei automatisierten Backups, Read Replicas und Snapshots. Wenn die Verschlüsselung beim Erstellen einer DB-Instance nicht aktiviert ist, müssen Sie eine verschlüsselte Kopie des entschlüsselten Snapshots der DB-Instance erstellen und wiederherstellen, bevor Sie	Aktivieren Sie die Verschlüsselung von Daten im Ruhezustand für Ihre DB-Instance.	Ja	Sicherheit in Amazon RDS Kopieren eines DB-Snapshots für Amazon RDS

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
	die Verschlüsselung aktivieren.			
Performance Insights ist ausgeschaltet	Performance Insights überwacht die Auslastung Ihrer DB-Instance, um Sie bei der Analyse und Lösung von Datenbankleistungsproblemen zu unterstützen. Wir empfehlen, Performance Insights zu aktivieren.	Aktivieren Sie Performance Insights.	Nein	Überwachung mit Performance Insights auf Amazon RDS
Bei DB-Instances ist die automatische Speicherskalierung deaktiviert	Die automatische Speicherskalierung ist für Ihre DB-Instance nicht aktiviert. Wenn die Datenbank-Arbeitslast zunimmt, skaliert die automatische RDS Speicherskalierung die Speicherkapazität automatisch und ohne Ausfallzeiten.	Automatische Skalierung des RDS Amazon-Speichers mit einem angegebenen maximalen Speicherschwellenwert aktivieren	Nein	Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSRessourcen: Aktualisierung der Hauptversionen ist erforderlich	Datenbanken mit der aktuellen Hauptversion für die DB-Engine werden nicht unterstützt. Wir empfehlen Ihnen, auf die neueste Hauptversion zu aktualisieren, die neue Funktionen und Verbesserungen enthält.	Führen Sie ein Upgrade auf die neueste Hauptversion für die DB-Engine durch.	Ja	Upgrade der Engine-Version für eine DB-Instance Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates
RDSEin Update der Ressourceninstanzklasse ist erforderlich	Auf Ihrer DB-Instance wird eine DB-Instance-Klasse einer früheren Generation ausgeführt. Wir haben DB-Instance-Klassen einer früheren Generation durch DB-Instance-Klassen mit besseren Kosten, besserer Leistung oder beidem ersetzt. Wir empfehlen, dass Sie Ihre DB-Instance mit einer DB-Instance-Klasse einer neueren Generation ausführen.	Aktualisieren Sie die DB-Instance-Klasse.	Ja	Unterstützte DB-Engines für DB-Instance-Klassen

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSRessourcen, die die End-of-Support-Engine-Edition im Rahmen der im Lieferumfang enthaltenen Lizenz verwenden	Wir empfehlen Ihnen, die Hauptversion auf die neueste von Amazon unterstützte Engine-Version zu aktualisierenRDS, um mit der aktuellen Lizenzunterstützung fortzufahren. Die Engine-Version Ihrer Datenbank wird mit der aktuellen Lizenz nicht unterstützt.	Wir empfehlen Ihnen, Ihre Datenbank auf die neueste unterstützte Version von Amazon zu aktualisierenRDS, um das lizenzierte Modell weiterhin verwenden zu können.	Ja	Upgrades der Oracle-Hauptversion

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
DB-Instances, die keine Multi-AZ-Bereitstellung verwenden	Wir empfehlen Ihnen, die Multi-AZ-Bereitstellung zu verwenden. Die Multi-AZ-Bereitstellungen verbessern die Verfügbarkeit und Dauerhaftigkeit der DB-Instance.	Richten Sie Multi-AZ für die betroffenen DB-Instances ein	Nein Während dieser Änderung treten keine Ausfallzeiten auf. Es kann jedoch zur Beeinträchtigung der Leistung kommen. Weitere Informationen finden Sie unter Konvertierung einer DB-Instance in eine	Preise für Amazon RDS Multi-AZ

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
			Multi-AZ-Bereitstellung für Amazon RDS	
Die DB-Speicherparameter weichen vom Standard ab	Die Speicherparameter der DB-Instances unterscheiden sich erheblich von den Standardwerten. Diese Einstellungen können sich auf die Leistung auswirken und zu Fehlern führen. Wir empfehlen, die benutzerdefinierten Speicherparameter für die DB-Instance auf ihre Standardwerte in der DB-Parametergruppe zurückzusetzen.	Setzen Sie die Speicherparameter auf ihre Standardwerte zurück.	Nein	Bewährte Methoden für die Konfiguration von Leistungsparametern RDS für Amazon for My SQL on the AWS Database Blog

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`InnoDB_Change_Buffering` Parameter, der weniger als den optimalen Wert verwendet	Durch die Pufferung von Änderungen kann eine My SQL DB-Instance einige Schreibvorgänge zurückstellen, die zur Verwaltung sekundärer Indizes erforderlich sind. Diese Funktion war in Umgebungen mit langsamen Festplatten nützlich. Die geänderte Pufferkonfiguration verbesserte die Leistung der Datenbank geringfügig, führte jedoch zu Verzögerungen bei der Wiederherstellung nach einem Absturz und zu langen Shutdown-Zeiten während des Upgrades. In meiner SQL Version 8.4 `OFF` standardmäßig auf eingestellt.	Stellen Sie den `InnoDB_Change_Buffering` Parameterwert `NONE` in Ihren DB-Parametergruppen auf ein.	Nein	Bewährte Methoden für die Konfiguration von Leistungsparametern RDS für Amazon for My SQL on the AWS Database Blog

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
Der Abfrage-Cache-Parameter ist aktiviert	Wenn Änderungen erfordern, dass Ihr Abfrage-Cache gelöscht wird, scheint Ihre DB-Instance zum Stillstand zu kommen. Die meisten Workloads profitieren nicht von einem Abfrage-Cache. Der Abfrage-Cache wurde aus My SQL 8.0 und höheren Versionen entfernt. Wir empfehlen, den Parameter query_cache_type auf 0 zu setzen.	Setzen Sie den `query_cache_type` Parameterwert `0` in Ihren DB-Parametergruppen auf.	Ja	Bewährte Methoden für die Konfiguration von Leistungsparametern RDS für Amazon for My SQL on the AWS Database Blog

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`log_output` Der Parameter ist auf eine Tabelle gesetzt	Wenn auf gesetzt `log_output` ist`TABLE`, wird mehr Speicherplatz verwendet als wenn auf eingestellt `log_output` ist`FILE`. Wir empfehlen, den Parameter auf zu setzen`FILE`, um zu verhindern, dass die Speichergrößenbeschränkung erreicht wird. In My SQL 8.4 und höheren Versionen `FILE` standardmäßig auf eingestellt.	Stellen Sie den `log_output` Parameterwert `FILE` in Ihren DB-Parametergruppen auf ein.	Nein	MySQL-Datenbank-Protokolldateien

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
Parametergruppen verwenden keine riesigen Seiten	Große Seiten können die Skalierbarkeit der Datenbank erhöhen, aber Ihre DB-Instance verwendet keine großen Seiten. Wir empfehlen, dass Sie den `use_large_pages` Parameterwert `ONLY` in der DB-Parametergruppe für Ihre DB-Instance aufsetzen.	Stellen Sie den `use_large_pages` Parameterwert `ONLY` in Ihren DB-Parametergruppen auf ein.	Ja	Aktivieren von HugePages für eine Instance von RDS für Oracle
`autovacuum` Der Parameter ist ausgeschaltet	Der Autovacuum-Parameter ist für die ausgeschaltet. Wenn Sie die automatische Bereinigung deaktivieren, werden Tabelle und Index größer und die Leistung beeinträchtigt. Wir empfehlen, dass Sie Autovacuum in Ihren DB-Parametergruppen aktivieren.	Aktivieren Sie den Autovacuum-Parameter in Ihren .	Nein	Grundlegendes zum Thema Autovakuum in Amazon RDS für SQL Postgre-Umgebungen im Datenbank-Blog AWS

Typ	Beschreibung	Empfehlung	Ausfall: it erforde ich	Zusätzliche Informati onen
`synchrono us_commit` Der Parameter ist ausgeschaltet	Wenn der `synchrono us_commit` Parameter ausgescha ltet ist, können Daten bei einem Datenbank absturz verloren gehen. Die Haltbarke it der Datenbank ist gefährdet. Wir empfehlen Ihnen, den `synchrono us_commit` -Parameter zu aktivieren.	Aktivieren Sie `synchrono us_commit` Parameter in Ihren DB-Parametergruppe n.	Ja	Amazon Aurora SQL Postgre-Parameter: Replikation, Sicherheit und Protokollierung im AWS Datenbank-Blog
`track_cou nts` Der Parameter ist ausgeschaltet	Wenn der `track_counts` Parameter ausgescha ltet ist, sammelt die Datenbank keine Statistiken zur Datenbankaktivität. Autovacuum erfordert, dass diese Statistiken korrekt funktionieren. Wir empfehlen , den Parameter `track_counts` auf 1 zu setzen.	Setzen Sie `track_counts` den Parameter auf1.	Nein	Laufzeitstatistiken für Postgre SQL

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`enable_indexonlyscan` Der Parameter ist ausgeschaltet	Der Abfrageplaner oder Optimierer kann den Plantyp „Nur Index“ für den Scanplan nicht verwenden, wenn er deaktiviert ist. Es wird empfohlen, den `enable_indexonlyscan` Parameterwert auf festzulegen. 1	Setzen Sie den `enable_indexonlyscan` Parameterwert auf1.	Nein	Planner-Methodenkonfiguration für PostgreSQL
`enable_indexscan` Der Parameter ist ausgeschaltet	Der Abfrageplaner oder Optimierer kann den Plantyp Indexscan nicht verwenden, wenn er ausgeschaltet ist. Wir empfehlen, den `enable_indexscan` Wert auf festzulegen. 1	Setzen Sie den `enable_indexscan` Parameterwert auf1.	Nein	Planner-Methodenkonfiguration für PostgreSQL

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`innodb_flush_log_at_trx` Der Parameter ist ausgeschaltet	Der Wert des `innodb_flush_log_at_trx` Parameters Ihrer DB-Instance ist kein sicherer Wert. Dieser Parameter steuert die Persistenz von Commit-Operationen auf der Festplatte. Wir empfehlen, den Parameter `innodb_flush_log_at_trx` auf 1 zu setzen.	Setzen Sie den `innodb_flush_log_at_trx` Parameterwert auf1.	Nein	Bewährte Methoden für die Konfiguration von Leistungsparametern RDS für Amazon for My SQL on the AWS Database Blog
`sync_binlog` Der Parameter ist ausgeschaltet	Die Synchronisation des Binärprotokolls mit der Festplatte wird erst erzwungen, wenn die Transaktions-Commits in Ihrer DB-Instance bestätigt wurden. Wir empfehlen, den `sync_binlog` Parameterwert auf zu setzen. 1	Setzen Sie den `sync_binlog` Parameterwert auf1.	Nein	Bewährte Methoden für die Konfiguration von Replikationsparametern RDS für Amazon for My SQL on the AWS Database Blog

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`innodb_stats_persistent` Der Parameter ist ausgeschaltet	Ihre DB-Instance ist nicht dafür konfiguriert, die InnoDB-Statistiken auf der Festplatte zu speichern. Wenn die Statistiken nicht gespeichert werden, werden sie bei jedem Neustart der Instanz und jedem Zugriff auf die Tabelle neu berechnet. Dies führt zu Abweichungen im Abfrageausführungsplan. Sie können den Wert dieses globalen Parameters auf Tabellenebene ändern. Es wird empfohlen, den `innodb_stats_persistent` Parameterwert auf festzulegen`ON`.	Setzen Sie den `innodb_stats_persistent` Parameterwert auf`ON`.	Nein	Bewährte Methoden für die Konfiguration von Leistungsparametern RDS für Amazon for My SQL on the AWS Database Blog

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`innodb_open_files` Der Parameter ist niedrig	Der `innodb_open_files` Parameter steuert die Anzahl der Dateien, die InnoDB gleichzeitig öffnen kann. InnoDB öffnet alle Protokoll- und System-Tablespace-Dateien, wenn mysqld läuft. Ihre DB-Instance hat einen niedrigen Wert für die maximale Anzahl von Dateien, die InnoDB gleichzeitig öffnen kann. Wir empfehlen, den Parameter `innodb_open_files` auf einen Mindestwert von 65 zu setzen.	Setzen Sie den `innodb_open_files` Parameter auf einen Mindestwert von. 65	Ja	InnoDB öffnet Dateien für My SQL

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`max_user_connections` Der Parameter ist niedrig	Ihre DB-Instance hat einen niedrigen Wert für die maximale Anzahl gleichzeitiger Verbindungen für jedes Datenbankkonto. Wir empfehlen, den `max_user_connections` Parameter auf eine Zahl größer als zu setzen`5`.	Erhöhen Sie den Wert des `max_user_connections` Parameters auf eine Zahl größer als`5`.	Ja	Festlegung von Kontoressourcenlimits für My SQL
Read Replicas sind im schreibbaren Modus geöffnet	Ihre DB-Instance verfügt über eine Read Replica im schreibbaren Modus, der Updates von Clients ermöglicht. Wir empfehlen, den `read_only` Parameter auf zu setzen, `TrueIfReplica` damit sich die Read Replicas nicht im schreibbaren Modus befinden.	Setzen Sie den `read_only` Parameterwert auf. `TrueIfReplica`	Nein	Bewährte Methoden für die Konfiguration von Replikationsparametern RDS für Amazon for My SQL on the AWS Database Blog

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`innodb_default_row_format` Die Parametereinstellung ist unsicher	In Ihrer DB-Instance ist ein bekanntes Problem aufgetreten: Auf eine Tabelle, die in einer SQL Version unter Meine Version 8.0.26 mit der `row_format` Einstellung auf `COMPACT` oder `REDUNDANT` erstellt wurde, kann nicht zugegriffen werden und sie kann nicht wiederhergestellt werden, wenn der Index 767 Byte überschreitet. Wir empfehlen, den Parameterwert auf zu setzen. `innodb_default_row_format` `DYNAMIC`	Setzen Sie den `innodb_default_row_format` Parameterwert auf`DYNAMIC`.	Nein	Änderungen in MySQL 8.0.26

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
`general_logging` Der Parameter ist aktiviert	Die allgemeine Protokollierung ist für Ihre DB-Instance aktiviert. Diese Einstellung ist nützlich bei der Behebung von Datenbankproblemen. Das Aktivieren der allgemeinen Protokollierung erhöht jedoch die Anzahl der I/O-Operationen und den zugewiesenen Speicherplatz, was zu Konflikten und Leistungseinbußen führen kann. Prüfen Sie Ihre Anforderungen für die allgemeine Nutzung der Protokollierung. Wir empfehlen, den `general_logging` Parameterwert auf zu setzen`0`.	Prüfen Sie Ihre Anforderungen für die allgemeine Nutzung der Protokollierung. Wenn dies nicht verpflichtend ist, empfehlen wir Ihnen, den `general_logging` Parameterwert auf zu setzen`0`.	Nein	Überblick über RDS-for-MySQL-Datenbankprotokolle

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSDie Instance ist in Bezug auf die Systemspeicherkapazität nicht ausreichend ausgestattet	Wir empfehlen, dass Sie Ihre Abfragen so anpassen, dass sie weniger Speicher verwenden oder einen DB-Instance-Typ mit mehr zugewiesenem Speicher verwenden. Wenn der Instance nur noch wenig Arbeitsspeicher zur Verfügung steht, wird die Datenbankleistung beeinträchtigt.	Verwenden Sie eine DB-Instance mit höherer Speicherkapazität	Ja	Vertikale und horizontale Skalierung Ihrer RDS Amazon-Instance im AWS Datenbank-Blog RDSAmazon-Instance-Typen RDSAmazon-Preisgestaltung
RDSInstance mit unzureichender Systemkapazität CPU	Wir empfehlen Ihnen, Ihre Abfragen so zu optimieren, dass sie weniger verbrauchen, CPU oder Ihre DB-Instance so zu modifizieren, dass sie eine DB-Instance-Klasse mit einer höheren Zuweisung verwendet. vCPUs Die Datenbankleistung kann sinken, wenn der Speicherplatz einer DB-Instance knapp wirdCPU.	Verwenden Sie eine DB-Instance mit höherer CPU Kapazität	Ja	Vertikale und horizontale Skalierung Ihrer RDS Amazon-Instance im AWS Datenbank-Blog RDSAmazon-Instance-Typen RDSAmazon-Preisgestaltung

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSRessourcen nutzen das Verbindungspooling nicht richtig	Wir empfehlen Ihnen, Amazon RDS Proxy zu aktivieren, um bestehende Datenbankverbindungen effizient zu bündeln und gemeinsam zu nutzen. Wenn Sie bereits einen Proxy für Ihre Datenbank verwenden, konfigurieren Sie ihn korrekt, um das Verbindungspooling und den Lastenausgleich über mehrere DB-Instances hinweg zu verbessern. RDS Proxy kann dazu beitragen, das Risiko von Verbindungsausfällen und Verbindungsausfällen zu verringern und gleichzeitig die Verfügbarkeit und Skalierbarkeit zu verbessern.	Aktivieren Sie den RDS Proxy oder ändern Sie Ihre bestehende Proxykonfiguration	Nein	Vertikale und horizontale Skalierung Ihrer RDS Amazon-Instance im AWS Datenbank-Blog Amazon RDS-Proxy Amazon RDS Proxy-Preise

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSInstances erstellen zu viele temporäre Objekte	Wir empfehlen Ihnen, Ihre Arbeitslast zu optimieren, um zu verhindern, dass zu viele temporäre Objekte erstellt werden, oder zu RDS Instanzklassen zu wechseln, die optimierte Lesevorgänge unterstützen. RDS Optimized Reads verbessert die Datenbankleistung für Workloads mit einer großen Anzahl temporärer Objekte und/oder großer temporärer Objekte. Bewerten Sie Ihre Arbeitslast, um festzustellen, ob die Verwendung einer Instanz mit RDS optimierten Lesevorgängen Ihrer Datenbank-Arbeitslast zugute kommt.	Verwenden Sie einen DB-Instance-Typ mit RDS optimierten Lesevorgängen	Ja	RDSAmazon-Instance-Typen Verbesserung der Abfrageleistung RDS für My SQL mit Amazon RDS Optimized Reads Verbesserung der Abfrageleistung RDS für MariaDB mit Amazon RDS Optimized Reads Verbesserung der Abfrageleistung RDS für Postgre SQL mit Amazon RDS Optimized Reads

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSInstances sind im Hinblick auf die Systemkapazität nicht ausreichend ausgestattet IOPS	Wir empfehlen, die Datenbank-Arbeitslast zu optimieren, um die DB-Instance zu reduzieren IOPS oder auf einen Typ mit einem höheren IOPS Standardlimit zu skalieren. Die aktuelle DB-Instance kann Provisioned nicht unterstützenIOPS, oder der Datenbank-Workload ist stark IOPS ausgelastet.	Verwenden Sie einen DB-Instance-Typ mit höheren Standardlimits IOPS	Ja	RDSAmazon-Instance-Typen Amazon RDS DB-Instance-Speicher Datenbanklast
RDSInstances haben unzureichend bereitgestellte Amazon-Volumes EBS	Wir empfehlen, die Datenbank-Arbeitslast zu optimieren, um die IOPS für die Datenbank bereitgestellten Daten zu reduzieren IOPS oder zu erhöhen. Wenn sich die IOPS Auslastung dem BereitstellungswertnähertIOPS, kann die Datenbankleistung sinken.	Stellen Sie mehr IOPS für die DB-Instance bereit	Ja	RDSAmazon-Instance-Typen Amazon RDS DB-Instance-Speicher Datenbanklast

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSDie Durchsatzkapazität der Instances ist unzureichend bereitgestellt	Wir empfehlen, den Datenbank-Workload zu optimieren, um den Durchsatz zu reduzieren oder den bereitgestellten Durchsatz für die Datenbank zu erhöhen. Wenn sich die Durchsatznutzung dem bereitgestellten Durchsatz nähert, kann die Datenbankleistung beeinträchtigt werden.	Stellen Sie mehr Durchsatz für die DB-Instance bereit	Ja	RDSAmazon-Instance-Typen Amazon RDS DB-Instance-Speicher Datenbanklast

Typ	Beschreibung	Empfehlung	Ausfallzeit erforderlich	Zusätzliche Informationen
RDSInstances sind nicht ausreichend für I/O bereitgestellt EBS	Wir empfehlen, die Datenbank-Arbeitslast zu optimieren, um I/O-Operationen zu reduzieren, oder die DB-Instance so zu modifizieren, dass sie Amazon RDS io2 Block Express-Volumes verwendet, die für Datenbank-Workloads konzipiert sind, die hohe Leistung, hohen Durchsatz und geringe Latenz erfordern. Bei der aktuellen Arbeitslast ist die Datenbank möglicherweise nicht in der Lage, I/O-Operationen mit der erforderlichen Geschwindigkeit zu verarbeiten, was zu Leistungseinbußen führen kann.	Verwenden Sie Amazon RDS io2 Block Express-Volumes für die Instance RDS	Nein	Amazon RDS DB-Instance-Speicher CloudWatchAmazon-Metriken für Amazon RDS Bereitgestellte IOPS SSD Volumes im EBS Amazon-Benutzerhandbuch

Metriken in der RDS Amazon-Konsole anzeigen

Amazon ist in Amazon RDS integriert CloudWatch , um eine Vielzahl von in der RDS Konsole anzuzeigen. Für Beschreibungen these metrics (dieser Metriken) (der Instance-Ebene- und Cluster-Ebene-Metriken), finden Sie unter Metrik-Referenz für Amazon RDS.

Für Ihre DB-Instance werden die folgenden Metrikkategorien überwacht:

- CloudWatch— Zeigt die CloudWatch Amazon-Metriken für an RDS, auf die Sie in der RDS Konsole zugreifen können. Sie können auch in der CloudWatch Konsole auf diese Metriken zugreifen. Jede Metrik beinhaltet ein Diagramm, das die überwachte Metrik für ein bestimmtes Zeitintervall anzeigt. Eine Liste der CloudWatch Messwerte finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS.
- Verbesserte Überwachung — Zeigt eine Zusammenfassung der Betriebssystemmetriken an, wenn Ihre Enhanced Monitoring aktiviert hat. RDSübermittelt die Metriken von Enhanced Monitoring an Ihr Amazon CloudWatch Logs-Konto. Jede Betriebssystem-Metrik beinhaltet ein Diagramm, das die überwachte Metrik für ein bestimmtes Zeitintervall anzeigt. Eine Übersicht finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung). Eine Liste der Metriken für „Enhanced Monitoring“·(Erweiterte·Überwachung) finden Sie unter Betriebssystemmetriken in „Erweiterte Überwachung“.
- Betriebssystem-Prozessliste – Zeigt Detailinformationen über alle Prozesse, die innerhalb Ihres DB-Instance- ausgeführt werden.
- Performance Insights — Öffnet das Amazon RDS Performance Insights Insights-Dashboard für eine DB-Instance . Eine Übersicht über Performance Insights finden Sie unter Überwachung mit Performance Insights auf Amazon RDS. Eine Liste der Peformance-Insights-Metriken finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS Performance Insights.

Amazon bietet RDS jetzt eine konsolidierte Ansicht von Performance Insights und CloudWatch Metriken im Performance Insights Insights-Dashboard. Performance Insights muss aktiviert sein, damit Ihre DB-Instance diese Ansicht verwenden kann. Sie können die neue Überwachungsansicht auf der Registerkarte Überwachung oder Performance Insights im Navigationsbereich auswählen. Anweisungen zur Auswahl dieser Ansicht finden Sie unter Kombinierte Metriken mit dem Performance Insights Insights-Dashboard anzeigen.

Kombinierte Metriken mit dem Performance Insights Insights-Dashboard anzeigen

Important

AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

Amazon RDS bietet im Performance Insights-Dashboard eine konsolidierte Ansicht von Performance Insights und CloudWatch Metriken für Ihre DB-Instance. Sie können das vorkonfigurierte Dashboard verwenden oder ein benutzerdefiniertes Dashboard erstellen. Das vorkonfigurierte Dashboard enthält die am häufigsten verwendeten Metriken zur Diagnose von Leistungsproblemen für eine Datenbank-Engine. Alternativ können Sie ein benutzerdefiniertes Dashboard mit den Metriken für eine Datenbank-Engine erstellen, die Ihren Analyseanforderungen entsprechen. Verwenden Sie dann dieses Dashboard für alle DB-Instances dieses Datenbank-Engine-Typs in Ihrem AWS Konto.

Sie können die Überwachungsansicht auf der Registerkarte Überwachung oder Performance Insights im Navigationsbereich auswählen.

Performance Insights muss für Ihre DB-Instance aktiviert sein, um die kombinierten Metriken im Performance-Insights-Dashboard anzeigen zu können. Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten.

In den folgenden Abschnitten erfahren Sie, wie Sie Performance Insights und CloudWatch Metriken anzeigen.

Themen

- Auswahl der neuen Überwachungsansicht auf der Registerkarte Überwachung
- Auswahl der neuen Überwachungsansicht auf der Seite Performance Insights
- Erstellen eines benutzerdefinierten Dashboards mit Performance Insights
- Auswahl des vorkonfigurierten Dashboards mit Performance Insights

Auswahl der neuen Überwachungsansicht auf der Registerkarte Überwachung

Important

AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden.

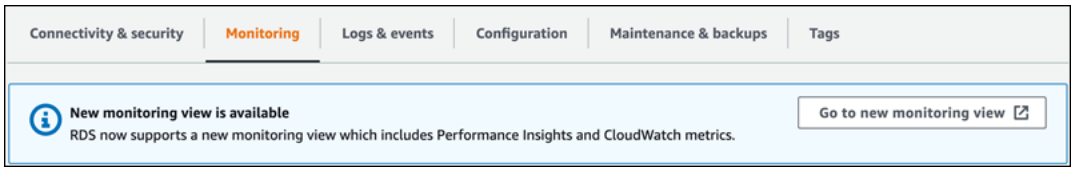
Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

In der Amazon RDS-Konsole können Sie die neue Überwachungsansicht wählen, um Performance Insights und CloudWatch Metriken für Ihre DB-Instance anzuzeigen.

Um die neue Überwachungsansicht auf der Registerkarte Überwachung auszuwählen

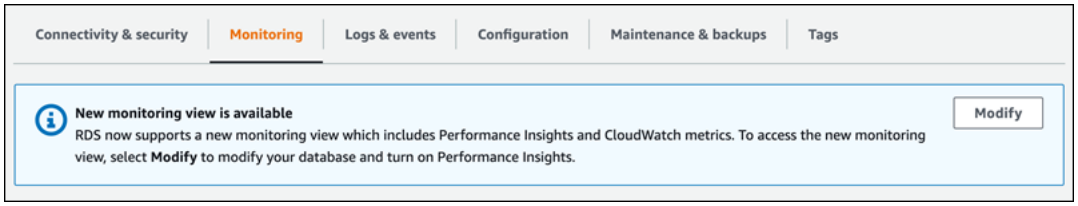
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Datenbanken aus.
3. Wählen Sie die DB-Instance zur Überwachung aus.
4. Scrollen Sie nach unten und wählen Sie die Registerkarte Überwachung aus.

   Es erscheint ein Banner mit der Option, die neue Überwachungsansicht auszuwählen. Das folgende Beispiel veranschaulicht das Banner zur Auswahl der neuen Überwachungsansicht.



5. Wählen Sie Gehe zu neuer Überwachungsansicht, um das Performance Insights-Dashboard mit Performance Insights und CloudWatch Metriken für Ihre zu öffnen.
6. (Optional) Wenn Performance Insights für Ihre DB-Instance deaktiviert ist, wird ein Banner mit der Option angezeigt, Ihren DB-Cluster zu ändern und Performance Insights zu aktivieren.

   Das folgende Beispiel veranschaulicht das Banner zum Ändern des DB-Clusters auf der Registerkarte Überwachung.



Wählen Sie Ändern aus, um Ihren DB-Cluster zu ändern und Performance Insights zu aktivieren. Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten.

Auswahl der neuen Überwachungsansicht auf der Seite Performance Insights

Important

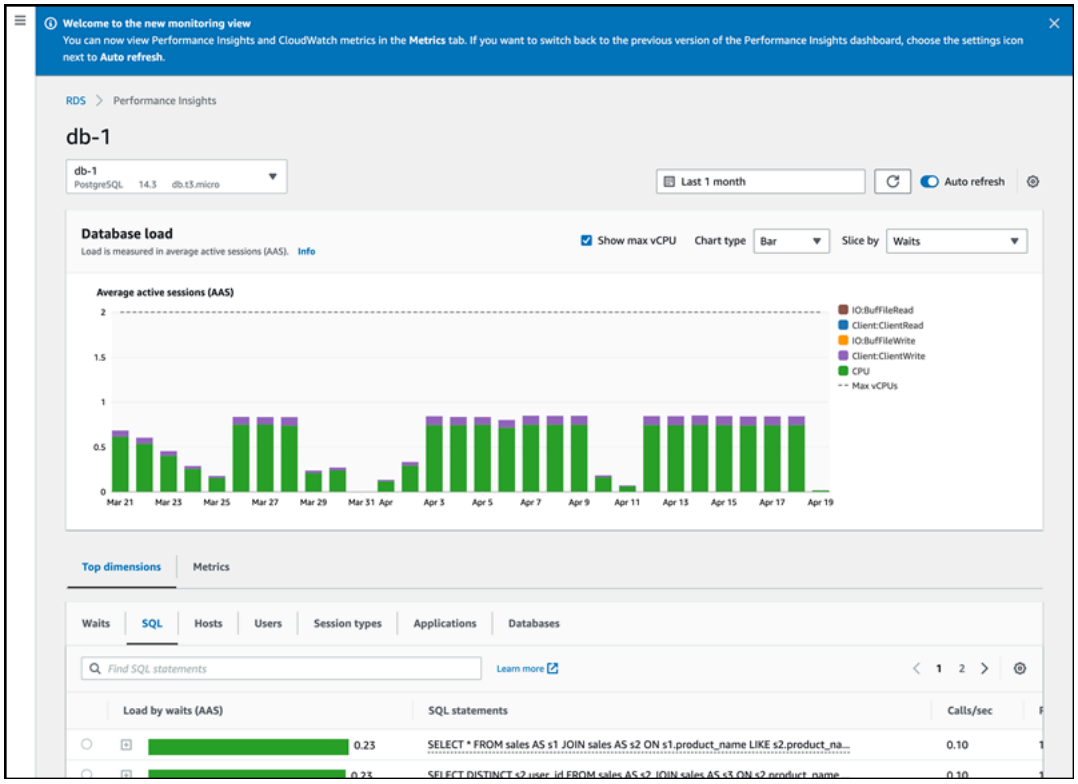
AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit

CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

In der Amazon RDS-Konsole können Sie die neue Überwachungsansicht wählen, um Performance Insights und CloudWatch Metriken für Ihre DB-Instance anzuzeigen.

Um die neue Überwachungsansicht mit Performance Insights im Navigationsbereich auszuwählen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus, um das Performance Insights-Dashboard aufzurufen, das sowohl Performance Insights als auch CloudWatch Metriken für Ihre DB-Instance anzeigt.



Erstellen eines benutzerdefinierten Dashboards mit Performance Insights

> **Important**
>
> AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
> Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
> Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
> Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

In der neuen Überwachungsansicht können Sie ein benutzerdefiniertes Dashboard mit den Metriken erstellen, die Sie für Ihre Analyseanforderungen benötigen.

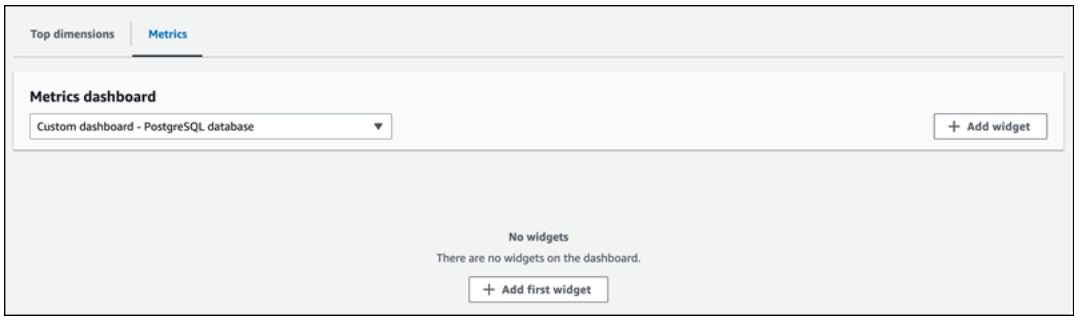
Sie können ein benutzerdefiniertes Dashboard erstellen, indem Sie Performance Insights und CloudWatch Metriken für Ihre DB-Instance auswählen. Sie können dieses benutzerdefinierte Dashboard für andere DB-Instances desselben Datenbank-Engine-Typs in Ihrem AWS Konto verwenden.

> **Note**
>
> Das benutzerdefinierte Dashboard unterstützt bis zu 50 Metriken.

Verwenden Sie das Widget-Einstellungsmenü, um das Dashboard zu bearbeiten oder zu löschen und das Widget-Fenster zu verschieben oder in der Größe zu ändern.

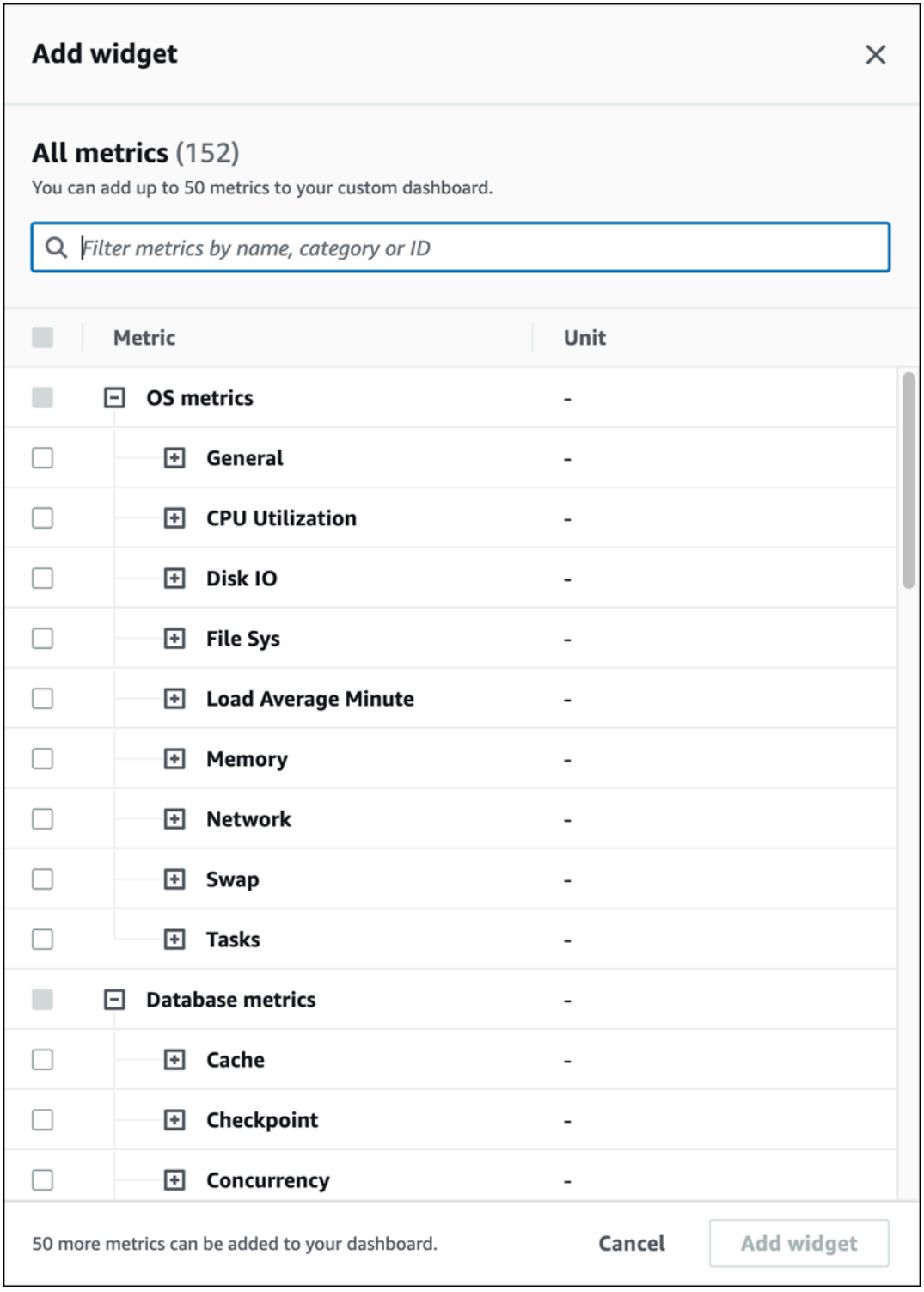
So erstellen Sie ein benutzerdefiniertes Dashboard mit Performance Insights im Navigationsbereich

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.
4. Scrollen Sie im Fenster nach unten zur Registerkarte Metriken.
5. Wählen Sie das benutzerdefinierte Dashboard aus der Dropdown-Liste aus. Das folgende Beispiel veranschaulicht die Erstellung eines benutzerdefinierten Dashboards.



6. Wählen Sie Widget hinzufügen aus, um das Fenster Widget hinzufügen zu öffnen. Sie können die verfügbaren Betriebssystem- (OS) -Metriken, Datenbank-Metriken und CloudWatch Metriken im Fenster öffnen und anzeigen.

   Das folgende Beispiel veranschaulicht das Fenster Widget hinzufügen mit den Metriken.



7. Wählen Sie die Metriken aus, die Sie im Dashboard anzeigen möchten. Klicken Sie dann auf Widget hinzufügen. Sie können über das Suchfeld eine bestimmte Metrik suchen.

   Die ausgewählten Metriken werden in Ihrem Dashboard angezeigt.

8. (Optional) Wenn Sie Ihr Dashboard ändern oder löschen möchten, wählen Sie das Einstellungensymbol oben rechts im Widget und dann eine der folgenden Aktionen im Menü aus.

   - Bearbeiten – Ändern Sie die Metrikenliste im Fenster. Wählen Sie Widget aktualisieren aus, nachdem Sie die Metriken für Ihr Dashboard ausgewählt haben.
   - Löschen – Löscht das Widget. Wählen Sie im Bestätigungsfenster Löschen aus.

Auswahl des vorkonfigurierten Dashboards mit Performance Insights

> **Important**
>
> AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
> Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
> Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
> Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

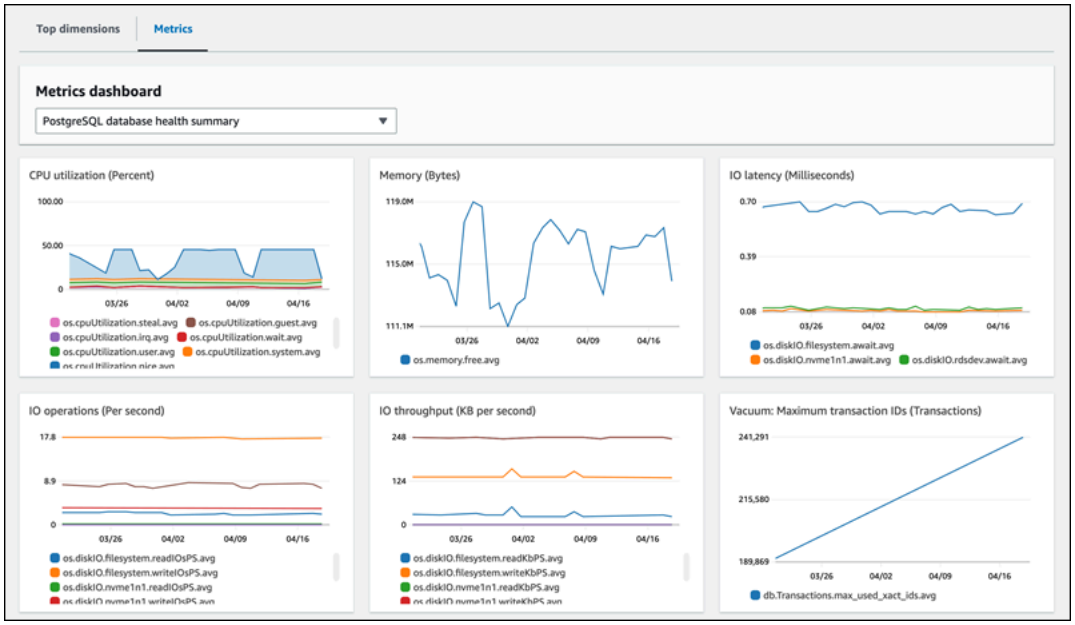
Sie können die am häufigsten verwendeten Metriken über das vorkonfigurierte Dashboard anzeigen. Dieses Dashboard hilft bei der Diagnose von Leistungsproblemen mit einer Datenbank-Engine und reduziert die durchschnittliche Wiederherstellungszeit von Stunden auf Minuten.

> **Note**
>
> Dieses Dashboard kann nicht bearbeitet werden.

Um das vorkonfigurierte Dashboard mit Performance Insights im Navigationsbereich auszuwählen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.
4. Scrollen Sie im Fenster nach unten zur Registerkarte Metriken.
5. Wählen Sie ein vorkonfiguriertes Dashboard aus der Dropdown-Liste aus.

   Sie können die Metriken für die DB-Instance im Dashboard anzeigen. Das folgende Beispiel veranschaulicht ein vorkonfiguriertes Metriken-Dashboard.



Überwachung von Amazon RDS mit Amazon CloudWatch

Amazon CloudWatch ist ein Metrik-Repository. Das Repository erfasst und verarbeitet Rohdaten aus Amazon RDS, um nahezu in Echtzeit lesbare Metriken bereitzustellen. Eine vollständige Liste der Amazon RDS-Metriken, an die gesendet wurden CloudWatch, finden Sie unter Metrik-Referenz für Amazon RDS-Metrik-Referenz .

Themen

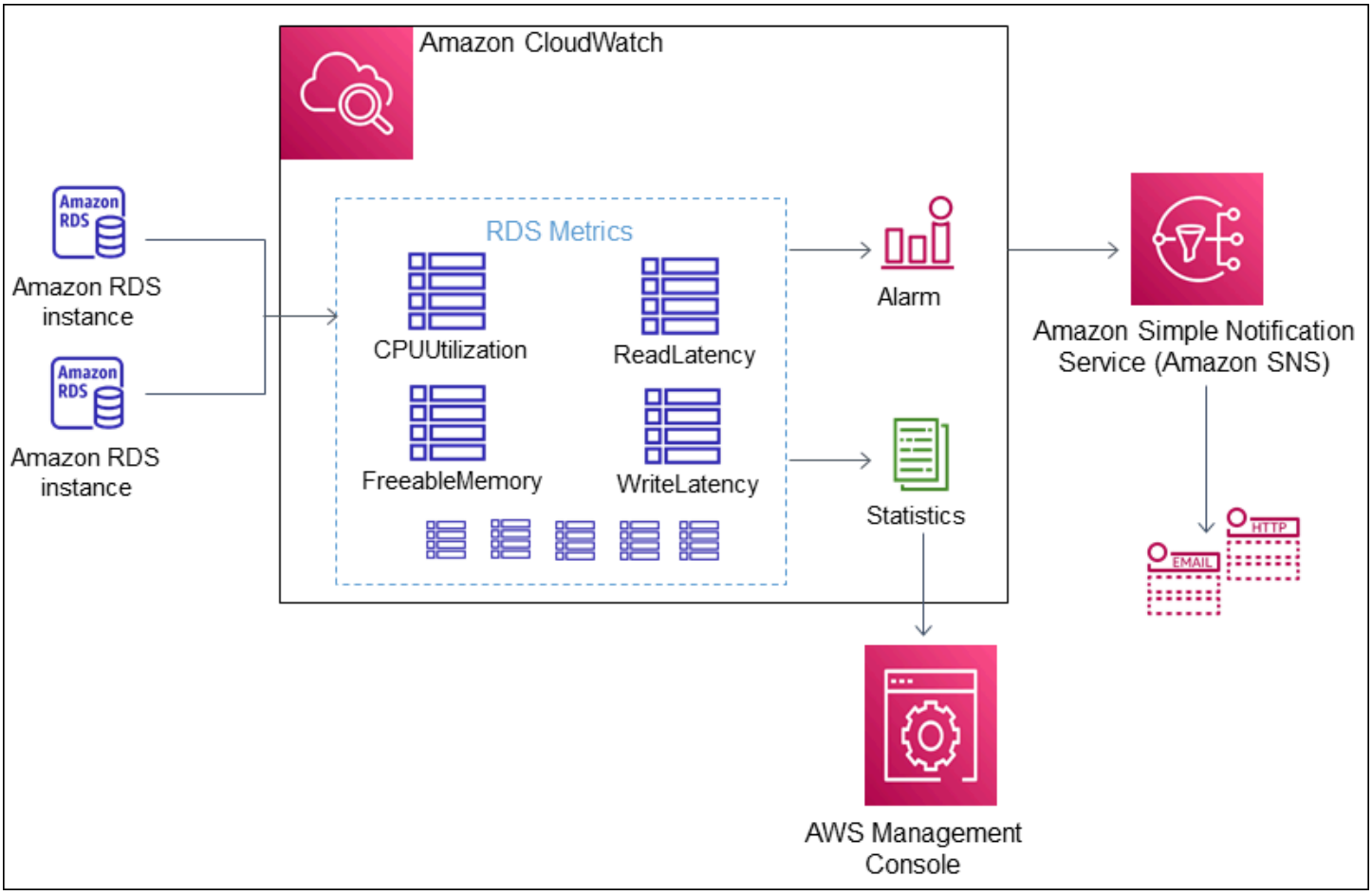
- Überblick über Amazon RDS, und Amazon CloudWatch
- Metriken des in der CloudWatch Konsole anzeigen und AWS CLI
- Performance Insights Insights-Metriken exportieren nach CloudWatch
- CloudWatch Alarme zur Überwachung von Amazon RDS Amazon erstellen
- Tutorial: Erstellen eines CloudWatch Amazon-Alarms für Multi-AZ-DB-Cluster-Replikatverzögerungen für Amazon RDS

Überblick über Amazon RDS, und Amazon CloudWatch

Standardmäßig sendet Amazon RDS Amazon automatisch Metrikdaten innerhalb von 1 Minute an CloudWatch . Die Metrik `CPUUtilization` zeichnet z. B. den Prozentsatz der CPU-Auslastung für eine DB-Instance im Laufe der Zeit auf. Datenpunkte mit einem Zeitraum von 60 Sekunden (1 Minute) stehen 15 Tage lang zur Verfügung Dies bedeutet, dass Sie auf die Verlaufsdaten zugreifen und sehen können, wie die Leistung Ihrer Webanwendung oder Ihres Service ist.

Sie können jetzt Performance Insights Insights-Metrik-Dashboards von Amazon RDS nach Amazon CloudWatch exportieren. Sie können entweder die vorkonfigurierten oder die benutzerdefinierten Metrik-Dashboards als neues Dashboard exportieren oder sie zu einem vorhandenen Dashboard hinzufügen. CloudWatch Das exportierte Dashboard kann in der CloudWatch Konsole angezeigt werden. Weitere Informationen zum Exportieren der Performance Insights Insights-Metrik-Dashboards in CloudWatch finden Sie unterPerformance Insights Insights-Metriken exportieren nach CloudWatch.

Wie im folgenden Diagramm dargestellt, können Sie Alarme für Ihre CloudWatch Metriken einrichten. Beispielsweise könnten Sie einen Alarm erstellen, der signalisiert, wenn die CPU-Auslastung für eine Instanz über 70 % beträgt. Sie können so konfigurieren Amazon Simple Notification Service , dass Sie eine E-Mail erhalten, wenn der Schwellenwert überschritten wird.



Amazon RDS veröffentlicht die folgenden Arten von Metriken auf Amazon CloudWatch:

- Metriken für Ihre RDS-DB-Instances

  Eine Tabelle dieser Metriken finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS.
- Performance-Insights-Metriken

  Eine Tabelle dieser Metriken finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS Performance Insights und Performance-Insights-Zählermetriken.
- Verbesserte Monitoring-Metriken (in Amazon CloudWatch Logs veröffentlicht)

  Eine Tabelle dieser Metriken finden Sie unter Betriebssystemmetriken in „Erweiterte Überwachung“.
- Nutzungsmetriken für die Amazon RDS-Servicekontingente in Ihrem AWS-Konto

Eine Tabelle dieser Metriken finden Sie unter CloudWatch Amazon-Nutzungsmetriken für Amazon RDS. Weitere Informationen über Amazon-RDS-Kontingente finden Sie unter Kontingente und Beschränkungen für Amazon RDS.

Weitere Informationen zu CloudWatch finden Sie unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch. Weitere Informationen zur Aufbewahrung von CloudWatch Kennzahlen finden Sie unter Aufbewahrung von Metriken.

Metriken des in der CloudWatch Konsole anzeigen und AWS CLI

Im Folgenden finden Sie Informationen zum Anzeigen von Metriken für Ihre DB-Instance mithilfe von CloudWatch. Informationen zur Überwachung von Metriken für das Betriebssystem Ihrer DB-Instance in Echtzeit mithilfe von CloudWatch Logs finden Sie unterÜberwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Wenn Sie Amazon RDS Amazon verwenden, sendet RDSAmazon CloudWatch jede Minute Metriken und Dimensionen an Amazon.

Sie können jetzt Performance Insights Insights-Metrik-Dashboards von Amazon RDS nach Amazon exportieren CloudWatch und diese Metriken in der CloudWatch Konsole anzeigen. Weitere Informationen zum Exportieren der Performance Insights Insights-Metrik-Dashboards in CloudWatch finden Sie unterPerformance Insights Insights-Metriken exportieren nach CloudWatch.

Gehen Sie wie folgt vor, um die Metriken für Amazon RDS in der CloudWatch Konsole und anzuzeigenCLI.

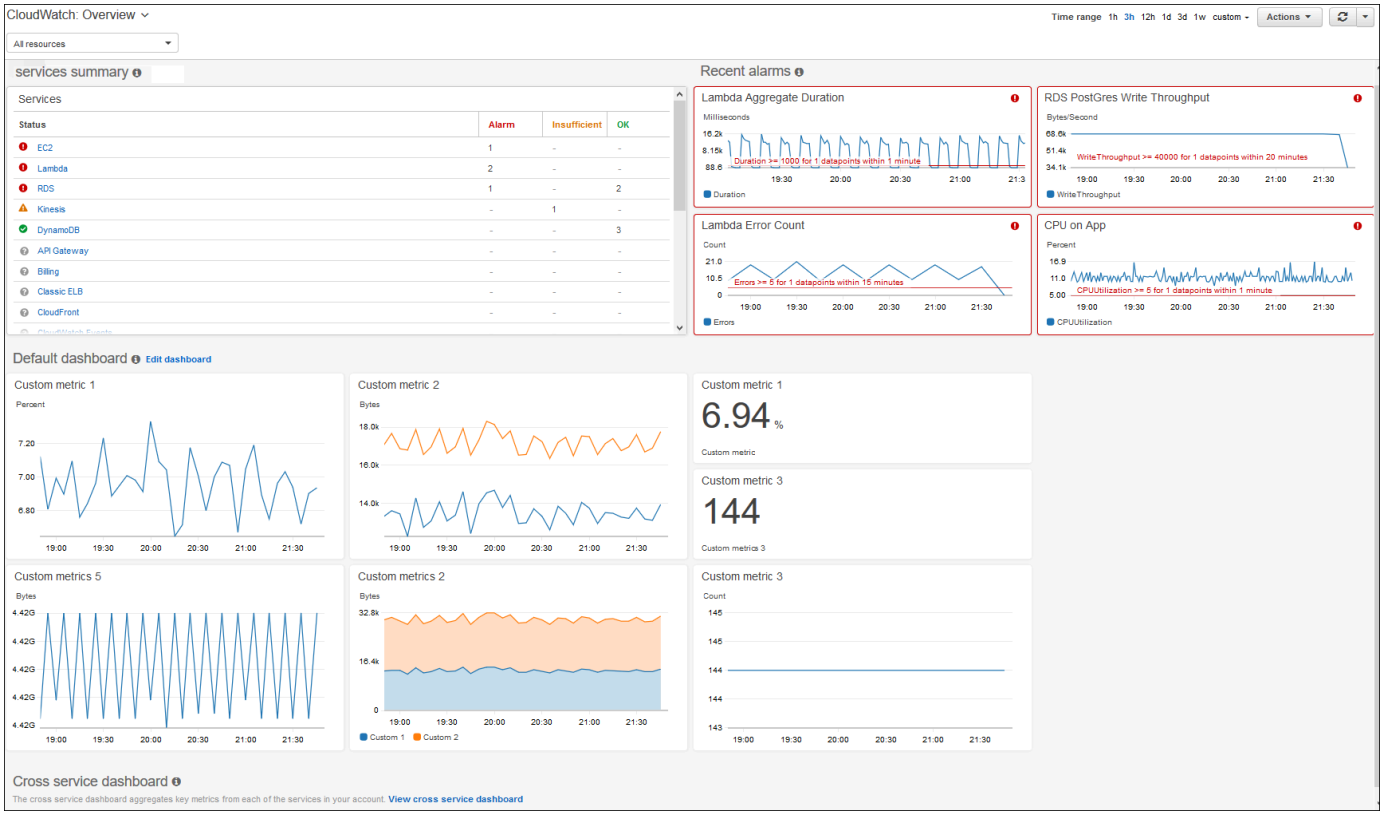
Konsole

So zeigen Sie Metriken mit der CloudWatch Amazon-Konsole an

Metriken werden zunächst nach dem Service-Namespace und anschließend nach den verschiedenen Dimensionskombinationen in den einzelnen Namespaces gruppiert.

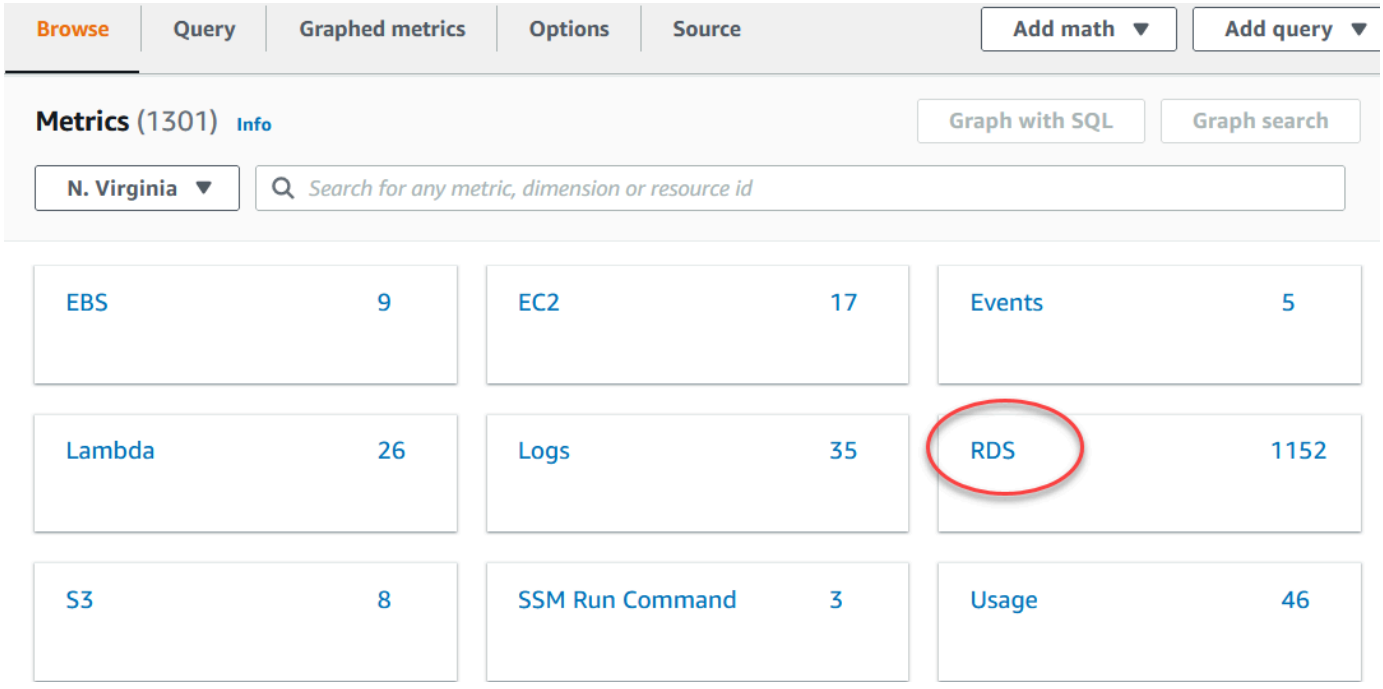
1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

   Die CloudWatch Übersichts-Startseite wird angezeigt.



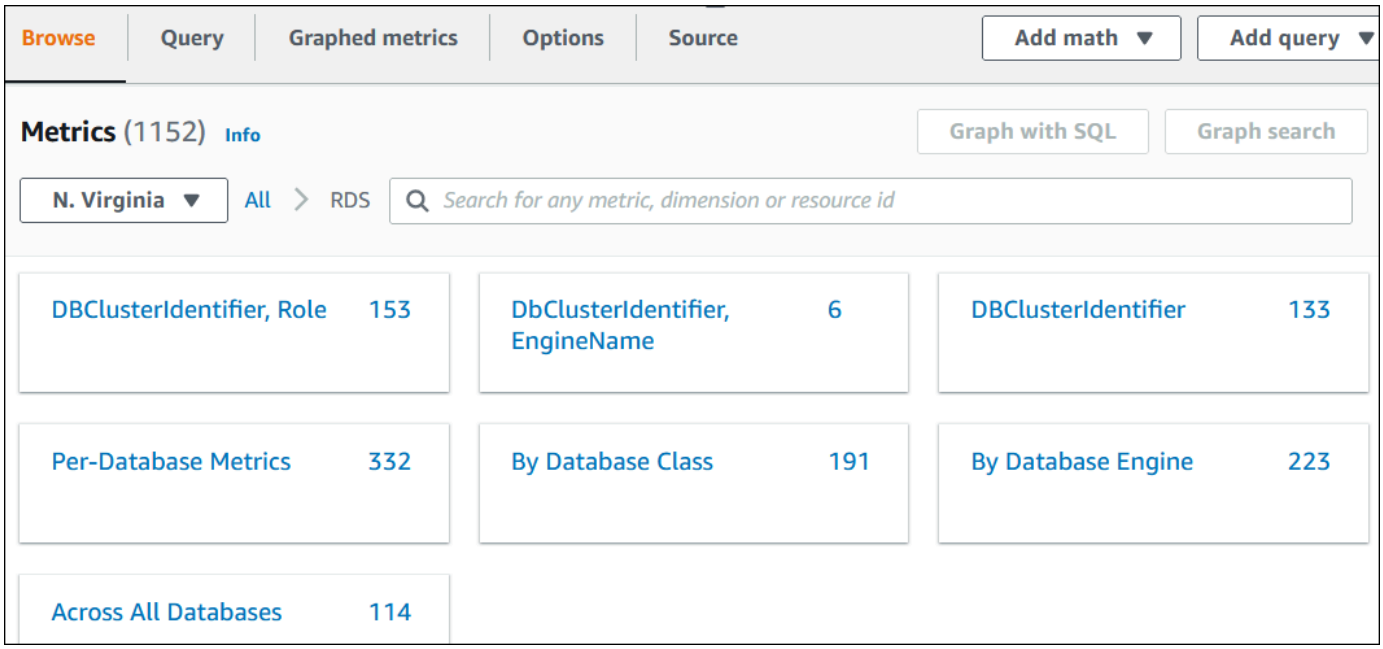
2. Ändern Sie, falls erforderlich, die AWS-Region. Wählen Sie in der Navigationsleiste aus, AWS-Region wo sich Ihre AWS Ressourcen befinden. Weitere Informationen finden Sie unter Regionen und Endpunkte.

3. Wählen Sie im Navigationsbereich Metrics (Metriken) und dann All metrics (Alle Metriken) aus.

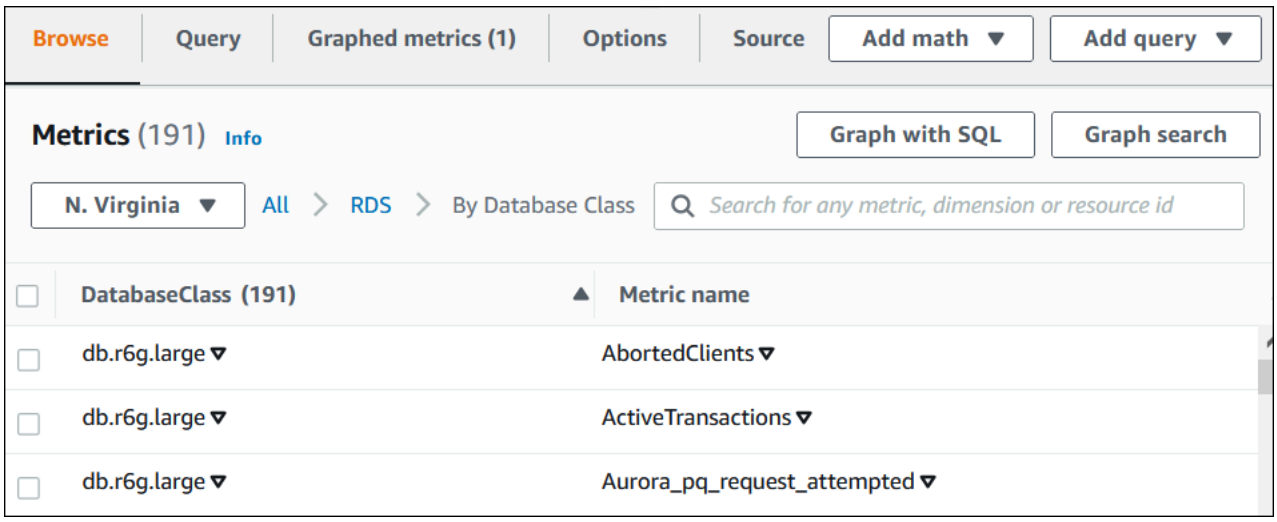


4. Scrollen Sie nach unten und wählen Sie den RDSMetrik-Namespace aus.

   Auf der Seite werden die Amazon RDS Amazon angezeigt. Beschreibungen dieser Dimensionen finden Sie unter CloudWatch Amazon-Abmessungen für Amazon RDS.



5. Wählen Sie eine Metrikdimension, z. B. By Database Class (Nach Datenbanken-Klasse).

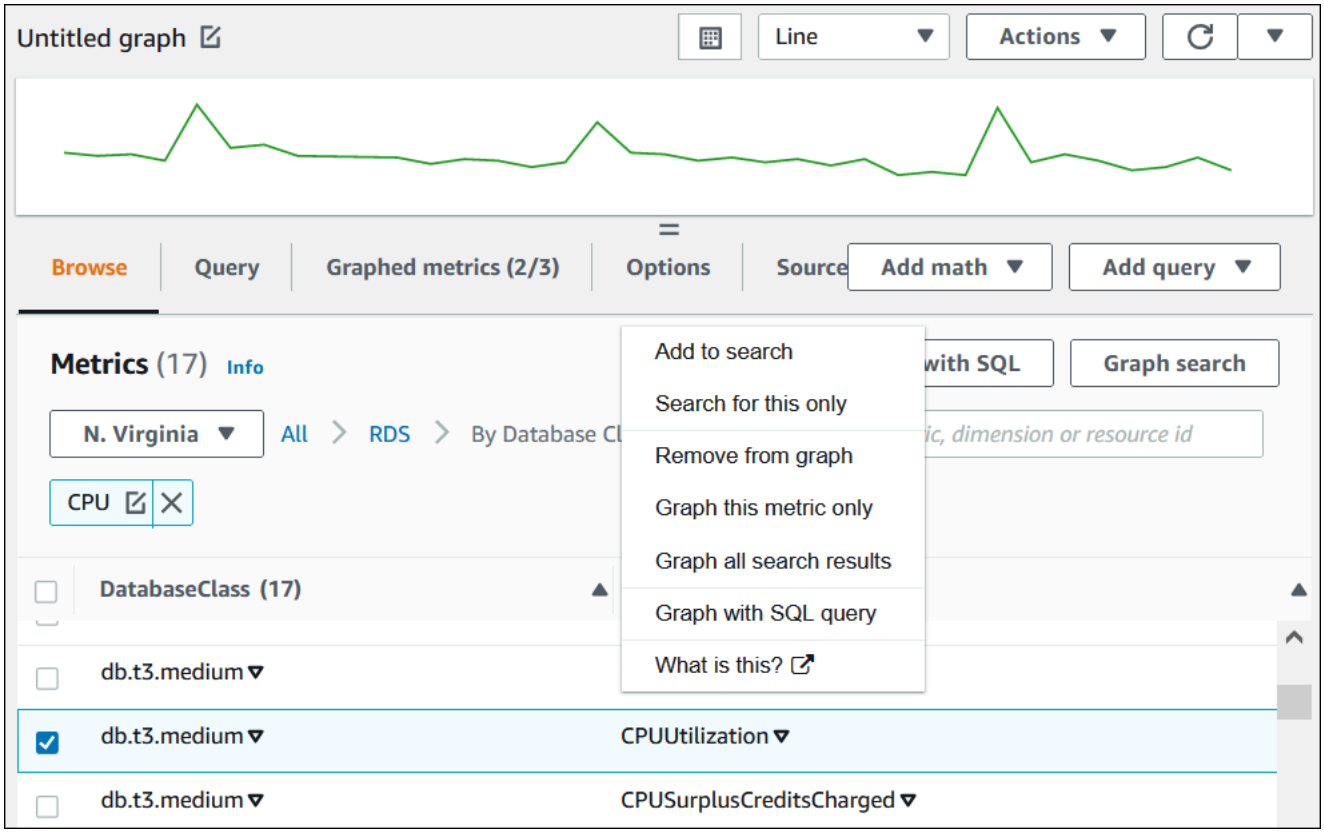


6. Führen Sie eine der folgenden Aktionen aus:

   - Verwenden Sie die Spaltenüberschrift, um die Metriken zu sortieren.
   - Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren.

- Um nach Ressource zu filtern, müssen Sie zunächst die Ressourcen-ID und dann die Option Add to search (Zu Suche hinzufügen) wählen.
- Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Add to search (Zu Suche hinzufügen) wählen.

Im folgenden Beispiel wird nach der Klasse db.t3.medium gefiltert und die Metrik grafisch dargestellt. CPUUtilization



AWS CLI

Verwenden Sie den CloudWatch Befehl AWS CLI`list-metrics`, um Metrikinformationen mithilfe von abzurufen. Im folgenden Beispiel listen Sie alle Metriken im `AWS/RDS`-Namespace auf.

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

Verwenden Sie den Befehl, um metrische Daten abzurufen `get-metric-data`.

Im folgenden Beispiel werden `CPUUtilization` Statistiken zum Beispiel `my-instance` über einen bestimmten Zeitraum von 24 Stunden mit einer Granularität von 5 Minuten abgerufen.

Erstellen Sie eine JSON Datei `CPU_metric.json` mit dem folgenden Inhalt.

```
{
   "StartTime" : "2023-12-25T00:00:00Z",
   "EndTime" : "2023-12-26T00:00:00Z",
   "MetricDataQueries" : [{
     "Id" : "cpu",
     "MetricStat" : {
    "Metric" : {
        "Namespace" : "AWS/RDS",
        "MetricName" : "CPUUtilization",
        "Dimensions" : [{ "Name" : "DBInstanceIdentifier" , "Value" : my-instance}]
    },
       "Period" : 360,
       "Stat" : "Minimum"
     }
   }]
}
```

Example

Für LinuxmacOS, oderUnix:

```
aws cloudwatch get-metric-data \
    --cli-input-json file://CPU_metric.json
```

Windows:

```
aws cloudwatch get-metric-data ^
     --cli-input-json file://CPU_metric.json
```

Beispielausgabe:

```
{
    "MetricDataResults": [
        {
            "Id": "cpu",
            "Label": "CPUUtilization",
            "Timestamps": [
                "2023-12-15T23:48:00+00:00",
                "2023-12-15T23:42:00+00:00",
```

```
                "2023-12-15T23:30:00+00:00",
                "2023-12-15T23:24:00+00:00",
                ...
            ],
            "Values": [
                13.299778337027714,
                13.677507543049558,
                14.24976250395827,
                13.02521708695145,
                ...
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Weitere Informationen finden Sie unter Statistiken für eine Metrik abrufen im CloudWatch Amazon-Benutzerhandbuch.

Performance Insights Insights-Metriken exportieren nach CloudWatch

Mit Performance Insights können Sie das vorkonfigurierte oder benutzerdefinierte Metrik-Dashboard für Ihre DB-Instance nach Amazon CloudWatch exportieren. Sie können das Metrik-Dashboard als neues Dashboard exportieren oder es zu einem vorhandenen CloudWatch Dashboard hinzufügen. Wenn Sie das Dashboard zu einem vorhandenen CloudWatch Dashboard hinzufügen möchten, können Sie eine Kopfzeile erstellen, sodass die Metriken in einem separaten Abschnitt im CloudWatch Dashboard angezeigt werden.

Sie können das Dashboard für exportierte Metriken in der CloudWatch Konsole anzeigen. Wenn Sie einem Performance Insights Insights-Metrik-Dashboard nach dem Export neue Metriken hinzufügen, müssen Sie dieses Dashboard erneut exportieren, um die neuen Metriken in der CloudWatch Konsole anzuzeigen.

Sie können auch ein Metrik-Widget im Performance Insights Insights-Dashboard auswählen und die Metrikdaten in der CloudWatch Konsole anzeigen.

Weitere Informationen zum Anzeigen der Metriken in der CloudWatch Konsole finden Sie unterMetriken des in der CloudWatch Konsole anzeigen und AWS CLI.

Exportieren Sie in den folgenden Abschnitten Performance Insights Insights-Metriken in ein neues oder vorhandenes Dashboard und zeigen Sie Performance Insights Insights-Metriken in an CloudWatch. CloudWatch

Themen

- Performance Insights Insights-Metriken als neues Dashboard exportieren nach CloudWatch
- Hinzufügen von Performance Insights Insights-Metriken zu einem vorhandenen CloudWatch Dashboard
- Anzeige eines Performance Insights Insights-Metrik-Widgets in CloudWatch

Performance Insights Insights-Metriken als neues Dashboard exportieren nach CloudWatch

Wählen Sie im Performance Insights Insights-Dashboard ein vorkonfiguriertes oder benutzerdefiniertes Metrik-Dashboard aus und exportieren Sie es als neues Dashboard nach CloudWatch. Sie können das exportierte Dashboard in der CloudWatch Konsole anzeigen.

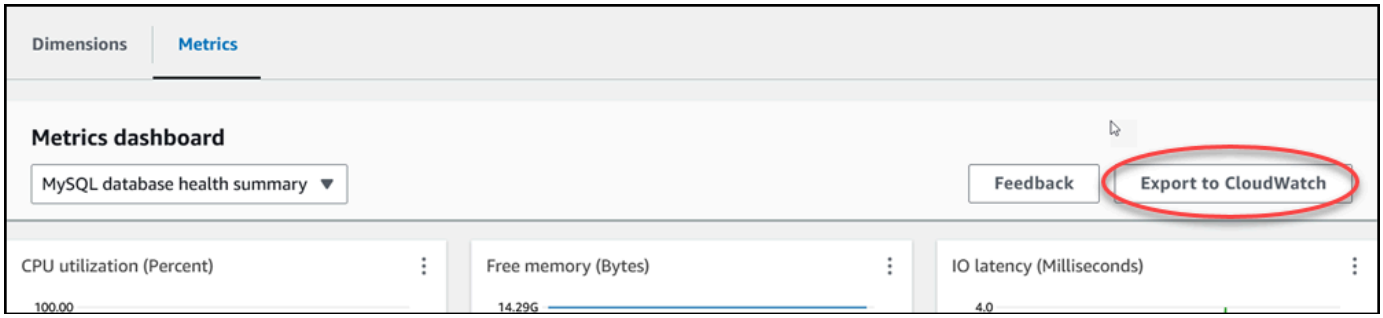
Um ein Performance Insights Insights-Metrik-Dashboard als neues Dashboard zu exportieren CloudWatch

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.

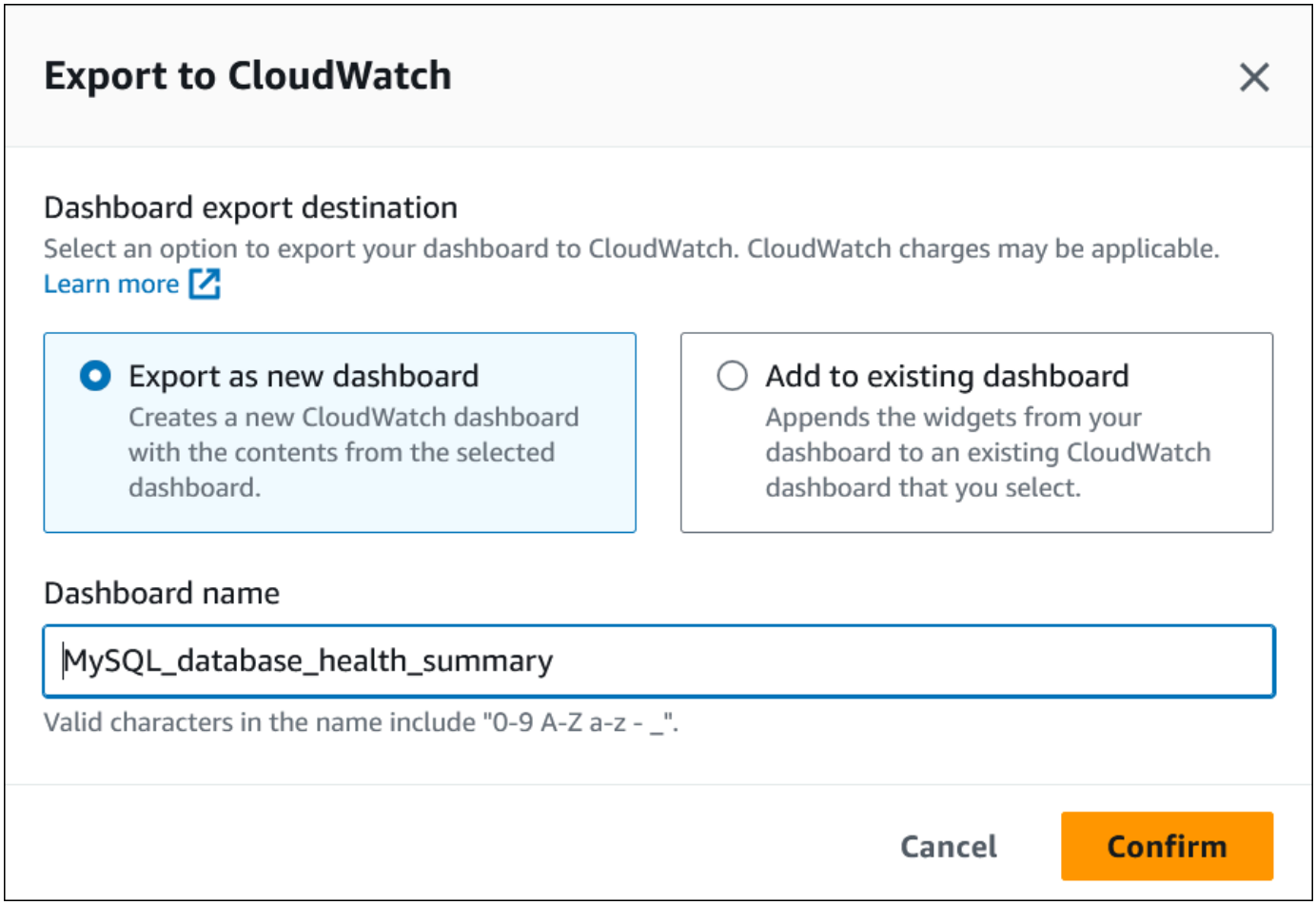
   Das Performance-Insights-Dashboard wird für die DB-Instance angezeigt.
4. Scrollen Sie nach unten und wählen Sie Metriken aus.

   Standardmäßig wird das vorkonfigurierte Dashboard mit Performance-Insights-Metriken angezeigt.
5. Wählen Sie ein vorkonfiguriertes oder benutzerdefiniertes Dashboard und wählen Sie dann Exportieren nach CloudWatch.

   Das CloudWatch Fenster Exportieren nach wird angezeigt.

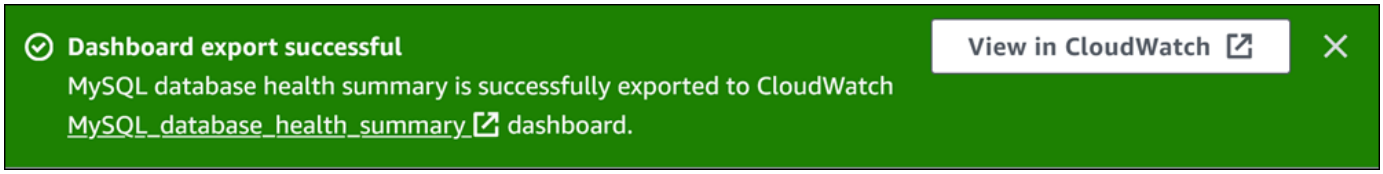


6. Wählen Sie Als neues Dashboard exportieren aus.



7. Geben Sie im Feld Dashboard-Name einen Namen für das neue Dashboard ein und wählen Sie Bestätigen aus.

   In einem Banner wird eine Meldung angezeigt, nachdem der Dashboard-Export erfolgreich war.



8. Wählen Sie im Banner den Link oder „Anzeigen CloudWatch in“, um das Metrik-Dashboard in der CloudWatch Konsole aufzurufen.

Hinzufügen von Performance Insights Insights-Metriken zu einem vorhandenen CloudWatch Dashboard

Fügen Sie einem vorhandenen Dashboard ein vorkonfiguriertes oder benutzerdefiniertes CloudWatch Metrik-Dashboard hinzu. Sie können dem Metrik-Dashboard eine Bezeichnung hinzufügen, die in einem separaten Abschnitt im CloudWatch Dashboard angezeigt wird.

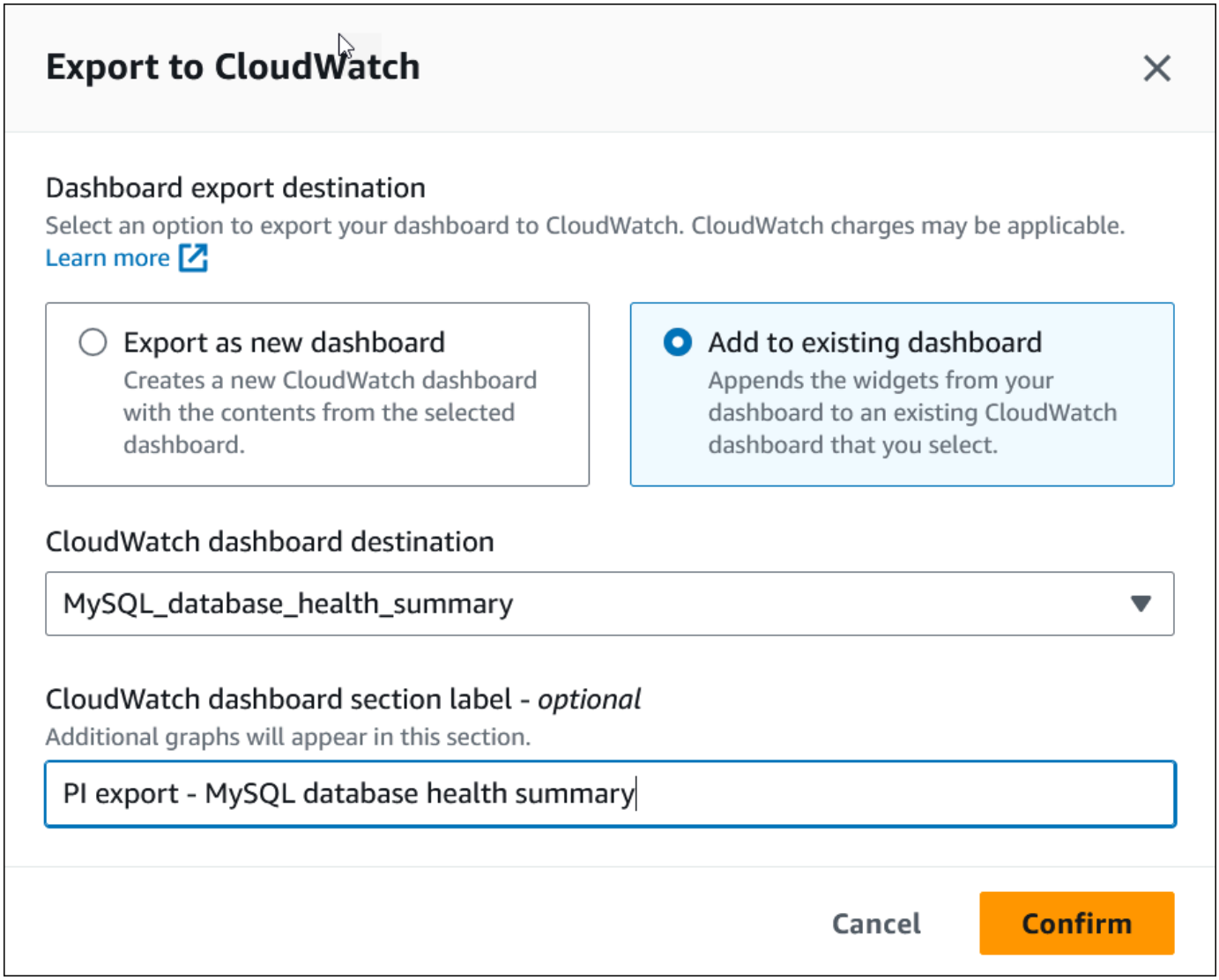
Um die Metriken in ein vorhandenes CloudWatch Dashboard zu exportieren

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.

   Das Performance-Insights-Dashboard wird für die DB-Instance angezeigt.
4. Scrollen Sie nach unten und wählen Sie Metriken aus.

   Standardmäßig wird das vorkonfigurierte Dashboard mit Performance-Insights-Metriken angezeigt.
5. Wählen Sie das vorkonfigurierte oder benutzerdefinierte Dashboard und dann Exportieren nach CloudWatch.

   Das CloudWatch Fenster Exportieren nach wird angezeigt.
6. Wählen Sie Zu vorhandenem Dashboard hinzufügen aus.



7. Geben Sie das Ziel und die Bezeichnung des Dashboards an und wählen Sie dann Bestätigen aus.

   - CloudWatch Dashboard-Ziel — Wählen Sie ein vorhandenes CloudWatch Dashboard aus.
   - CloudWatch Bezeichnung des Dashboard-Abschnitts — optional — Geben Sie einen Namen für die Performance Insights Insights-Metriken ein, die in diesem Abschnitt im CloudWatch Dashboard angezeigt werden sollen.

   In einem Banner wird eine Meldung angezeigt, nachdem der Dashboard-Export erfolgreich war.

8. Wählen Sie im Banner den Link oder „Anzeigen CloudWatch in“, um das Metrik-Dashboard in der CloudWatch Konsole aufzurufen.

Anzeige eines Performance Insights Insights-Metrik-Widgets in CloudWatch

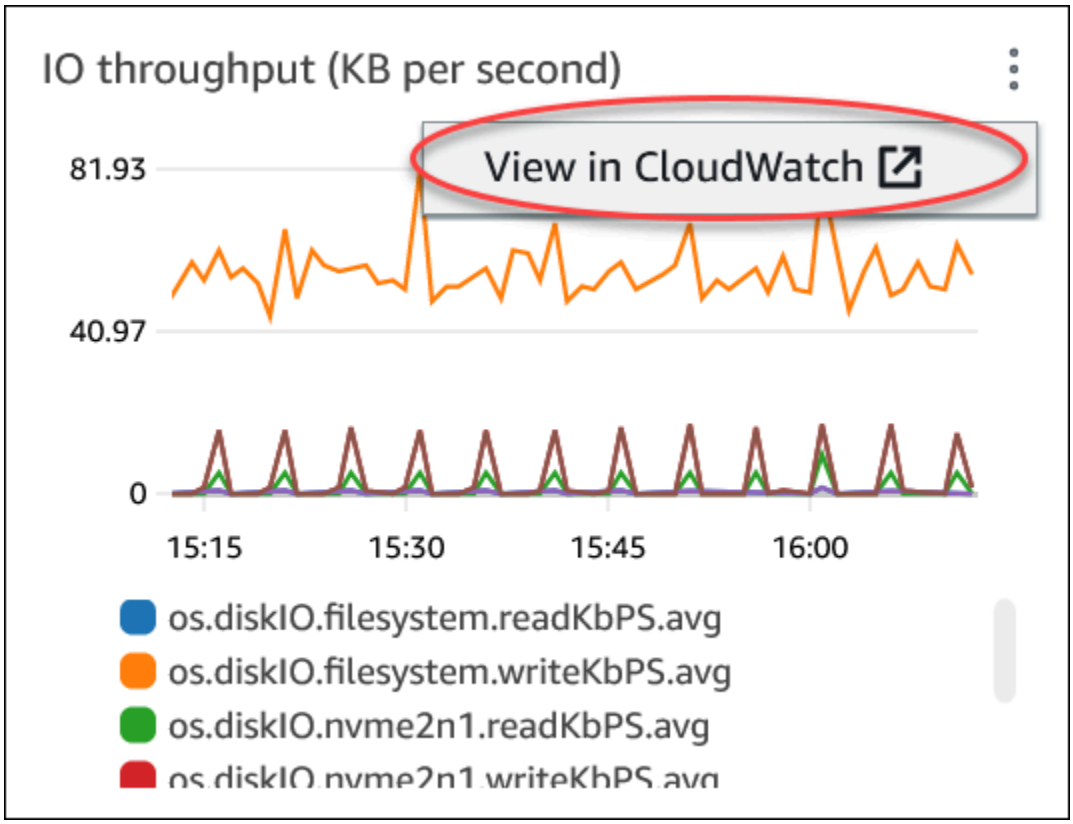
Wählen Sie im Amazon Performance Insights Insights-Dashboard ein RDS Performance Insights Insights-Metrik-Widget aus und sehen Sie sich die Metrikdaten in der CloudWatch Konsole an.

Um ein Metrik-Widget zu exportieren und die Metrikdaten in der CloudWatch Konsole anzuzeigen

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.

   Das Performance-Insights-Dashboard wird für die DB-Instance angezeigt.
4. Scrollen Sie nach unten zu Metriken.

   Standardmäßig wird das vorkonfigurierte Dashboard mit Performance-Insights-Metriken angezeigt.
5. Wählen Sie ein Metrik-Widget und dann im Menü die Option Anzeigen CloudWatch in.



   Die Metrikdaten werden in der CloudWatch Konsole angezeigt.

CloudWatch Alarme zur Überwachung von Amazon RDS Amazon erstellen

Sie können einen CloudWatch Alarm erstellen, der eine SNS Amazon-Nachricht sendet, wenn sich der Zustand des Alarms ändert. Ein Alarm überwacht eine Metrik über einen bestimmten, von Ihnen definierten Zeitraum. Der Alarm kann auch eine oder mehrere Aktionen durchführen, die vom Wert der Metrik im Vergleich zu einem gegebenen Schwellenwert in einer Reihe von Zeiträumen abhängt. Die Aktion ist eine Benachrichtigung, die an ein SNS Amazon-Thema oder eine Amazon EC2 Auto Scaling Scaling-Richtlinie gesendet wird.

Alarme lösen nur Aktionen für anhaltende Statusänderungen aus. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Zustand befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

Sie können die mathematische Funktion DB_ PERF _ INSIGHTS metric in der CloudWatch Konsole verwenden, um Amazon RDS nach Performance Insights Insights-Zählermetriken abzufragen. Die INSIGHTS Funktion DB_ PERF _ beinhaltet auch die DBLoad Metrik in Intervallen unter einer Minute. Sie können CloudWatch Alarme für diese Metriken einrichten.

Weitere Informationen zum Erstellen eines Alarms finden Sie unter Erstellen eines Alarms für Performance Insights Insights-Zählermetriken aus einer AWS Datenbank.

Um einen Alarm einzurichten, verwenden Sie AWS CLI

- Rufen Sie die folgende Seite auf `put-metric-alarm`. Weitere Informationen finden Sie in der AWS CLI -Befehlsreferenz.

Um einen Alarm einzustellen mit dem CloudWatch API

- Rufen Sie die folgende Seite auf `PutMetricAlarm`. Weitere Informationen finden Sie unter Amazon CloudWatch API Reference

Weitere Informationen zum Einrichten von SNS Amazon-Themen und zum Erstellen von Alarmen finden Sie unter CloudWatchAmazon-Alarme verwenden.

Tutorial: Erstellen eines CloudWatch Amazon-Alarms für Multi-AZ-DB-Cluster-Replikatverzögerungen für Amazon RDS

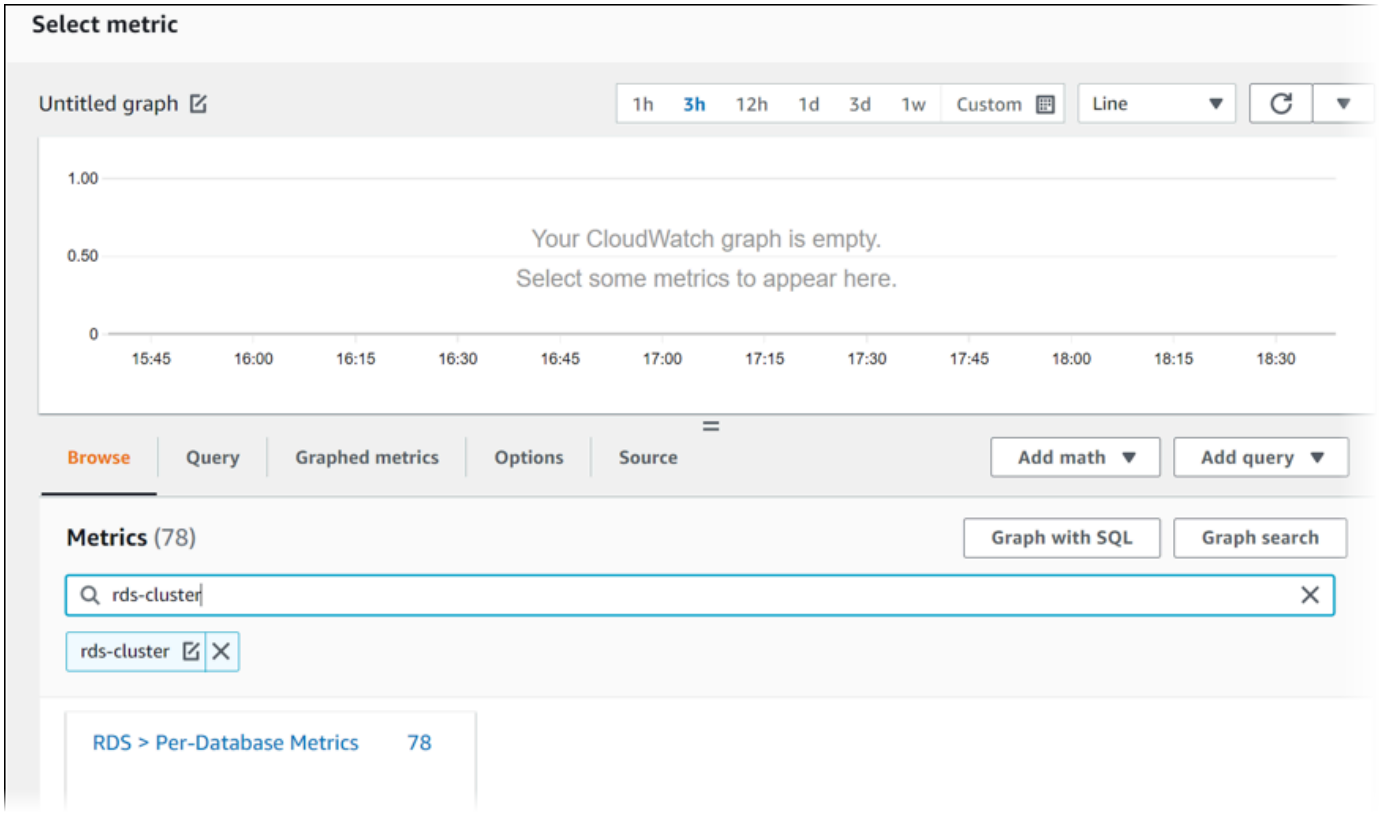
Sie können einen CloudWatch Amazon-Alarm erstellen, der eine SNS Amazon-Nachricht sendet, wenn die Replikatverzögerung für einen Multi-AZ-DB-Cluster einen Schwellenwert überschritten hat.

Ein Alarm überwacht die `ReplicaLag`-Metrik über einen bestimmten, von Ihnen definierten Zeitraum. Die Aktion ist eine Benachrichtigung, die an ein SNS Amazon-Thema oder eine Amazon EC2 Auto Scaling Scaling-Richtlinie gesendet wird.

Um einen CloudWatch Alarm für Multi-AZ-DB-Cluster-Replikatverzögerungen einzurichten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudWatch Konsole unter. https://console.aws.amazon.com/cloudwatch/
2. Wählen Sie im Navigationsbereich Alarms (Alarme) und All alarms (Alle Alarme) aus.
3. Wählen Sie Create alarm (Alarm erstellen) aus.
4. Wählen Sie auf der Seite Specify metric and conditions (Metrik und Bedingungen angeben) die Option Select metric (Metrik auswählen) aus:
5. Geben Sie im Suchfeld den Namen des Multi-AZ-DB-Clusters ein und drücken Sie die Eingabetaste.

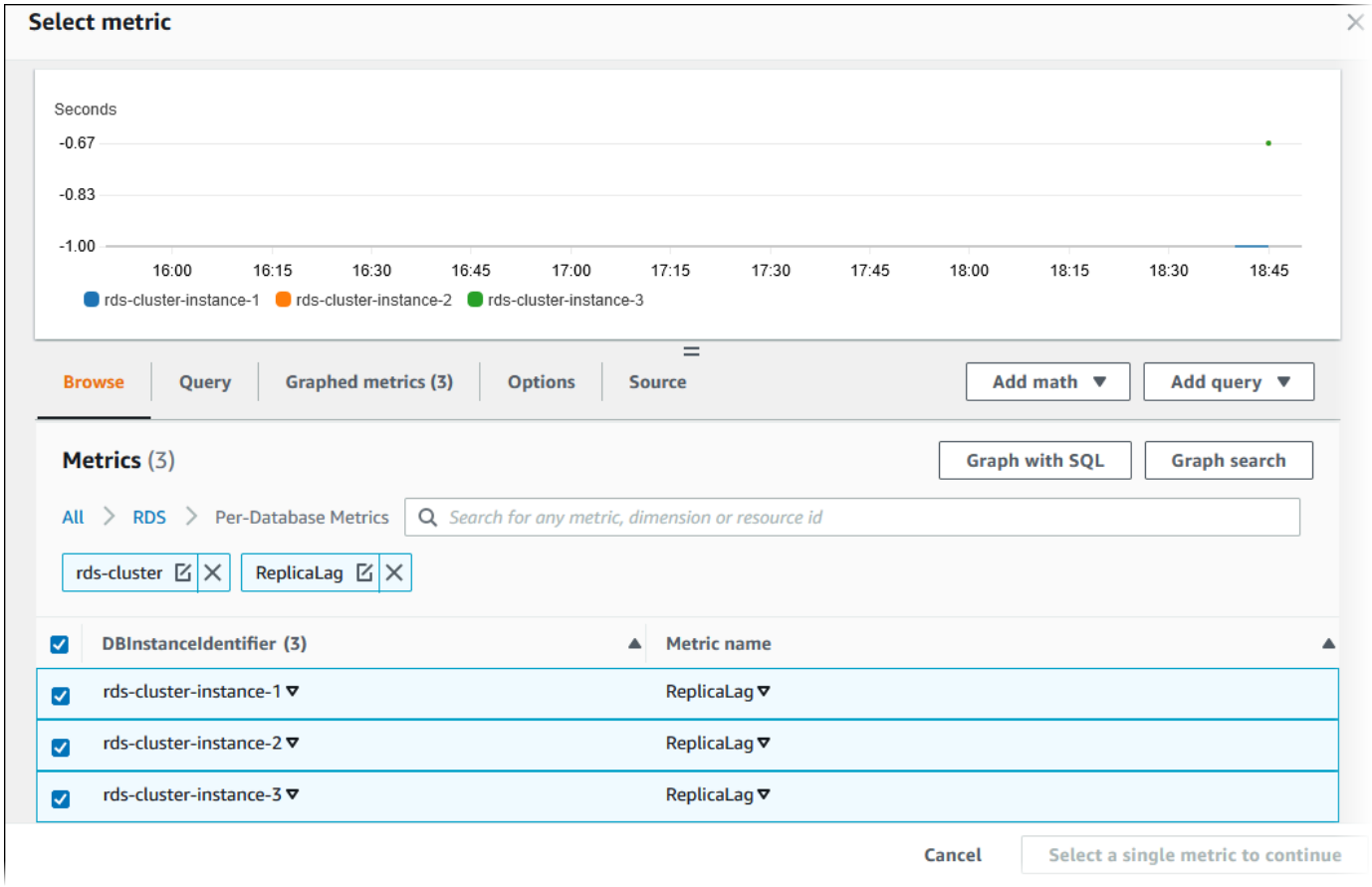
   Die folgende Abbildung zeigt die Seite Select metric (Metrik auswählen), auf der ein Multi-AZ-DB-Cluster namens `rds-cluster` eingegeben ist.



6. Wählen Sie RDSMetriken pro Datenbank.

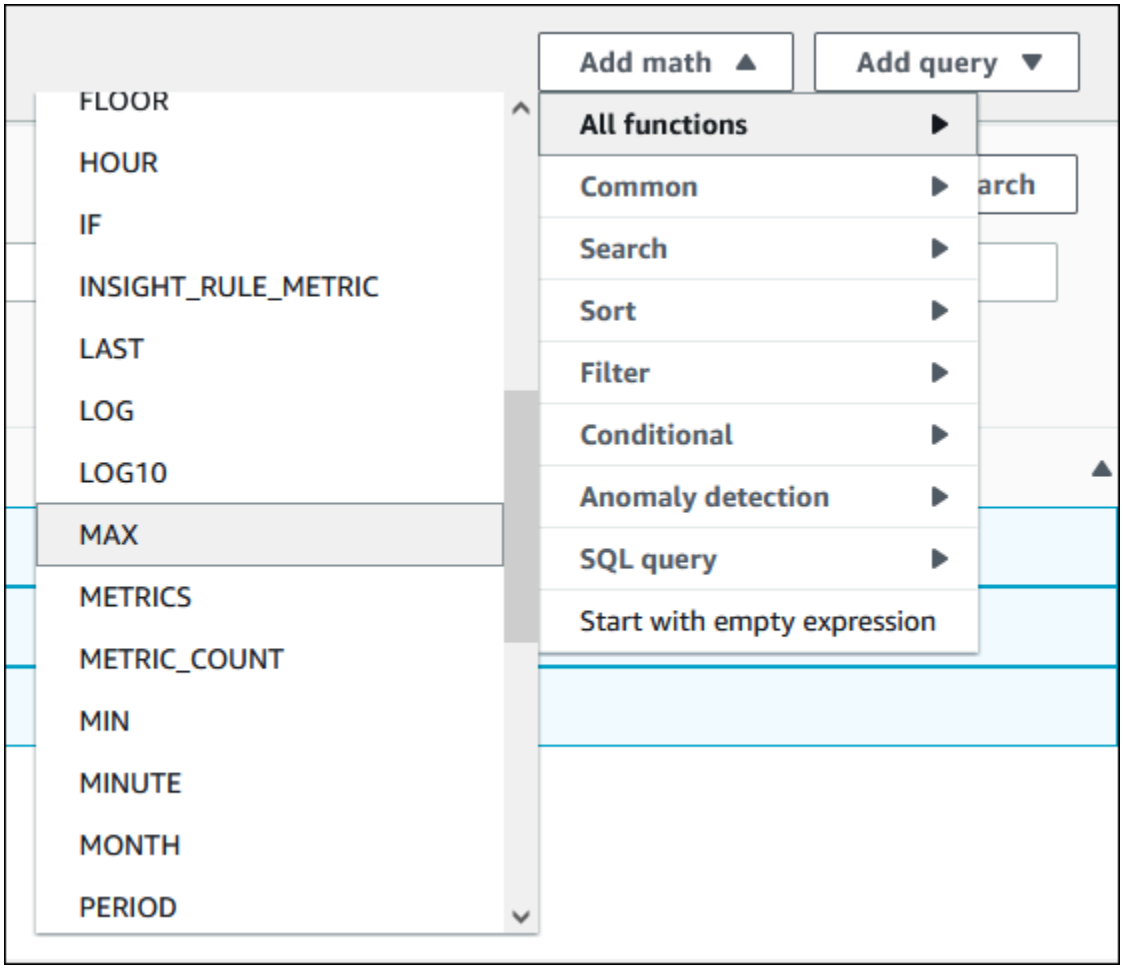
7. Geben Sie im Suchfeld **`ReplicaLag`** ein und drücken Sie die Eingabetaste und wählen Sie dann jede DB-Instance im DB-Cluster aus.

   Die folgende Abbildung zeigt die Seite „Metrik auswählen“ mit den für die ReplicaLagMetrik ausgewählten DB-Instances.



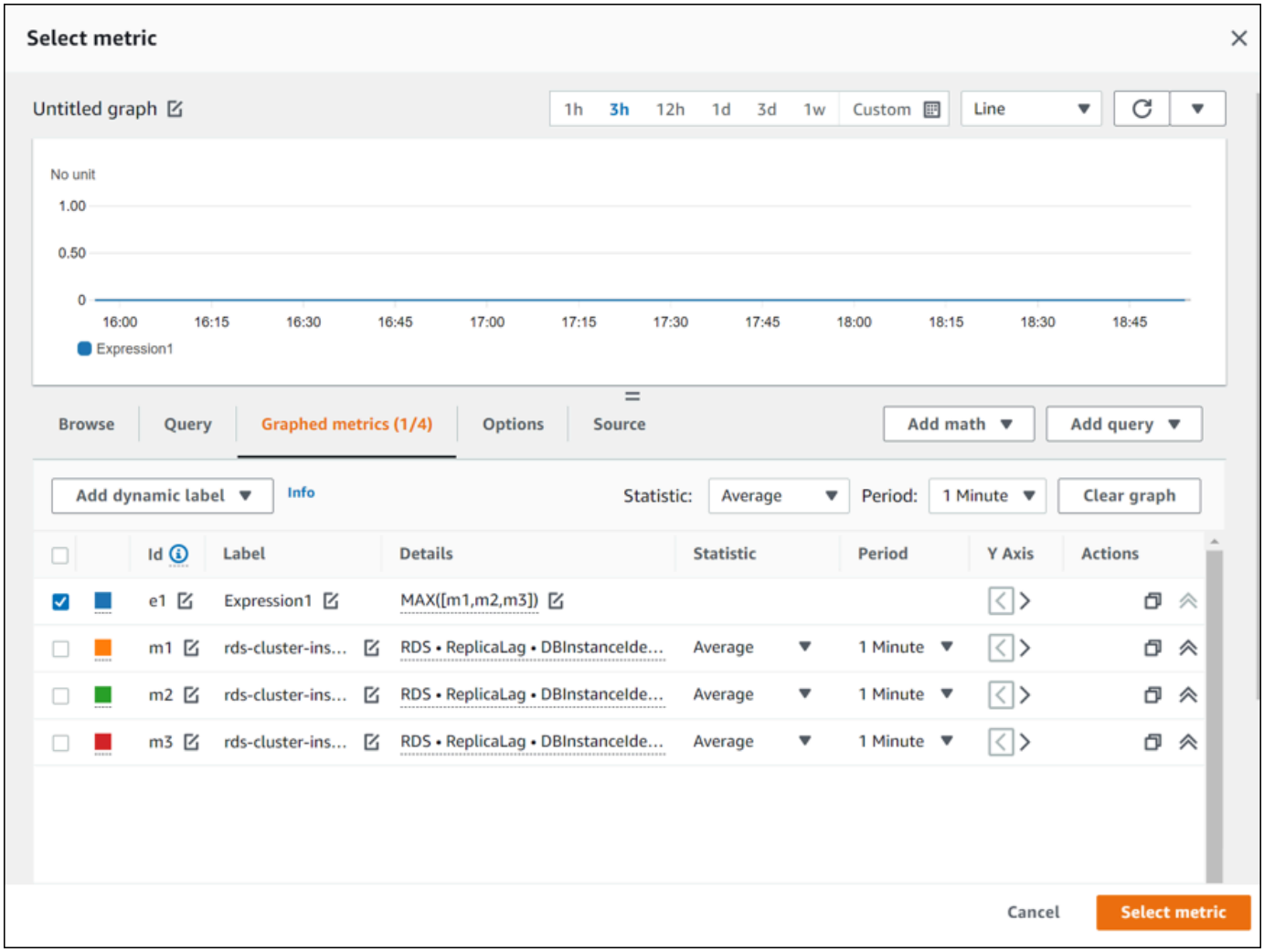
   Dieser Alarm berücksichtigt die Verzögerung der Replikate für alle drei DB-Instances im Multi-AZ-DB-Cluster. Der Alarm reagiert, wenn eine DB-Instance den Schwellenwert überschreitet. Er verwendet einen mathematischen Ausdruck, der den Maximalwert der drei Metriken zurückgibt. Sortieren Sie zunächst nach dem Metriknamen und wählen Sie dann alle drei ReplicaLagMetriken aus.

8. Wählen Sie unter Mathematik hinzufügen die Option Alle Funktionen, aus MAX.



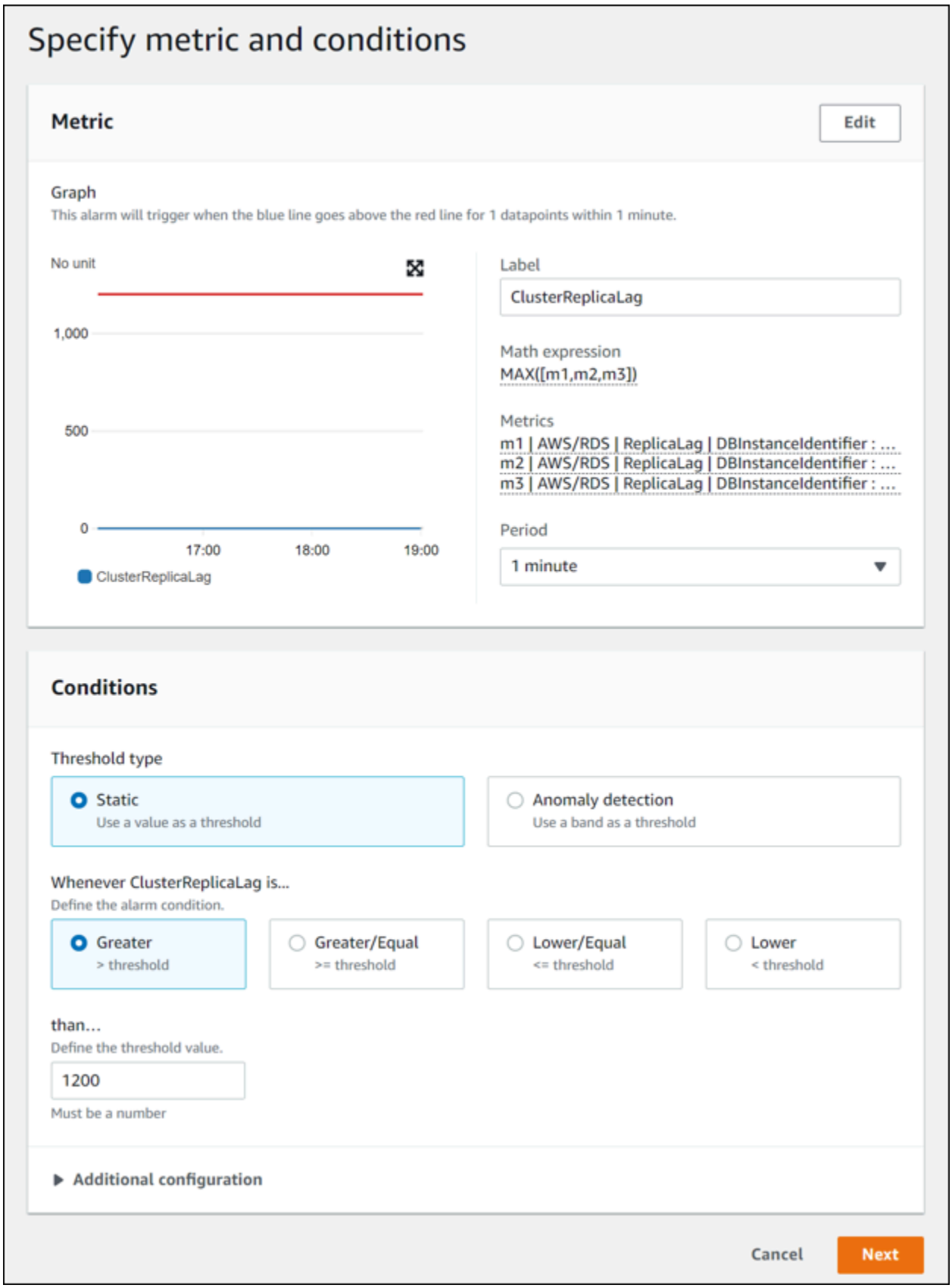
9. Wählen Sie die Registerkarte Graphed metrics (Grafische Metriken) aus und bearbeiten Sie die Details für Expression1 bis **`MAX([m1,m2,m3])`**.
10. Ändern Sie für alle drei ReplicaLagMesswerte den Zeitraum auf 1 Minute.
11. Löschen Sie die Auswahl aus allen Metriken mit Ausnahme von Expression1.

    Die Seite Select metric (Metrik auswählen) sollte ähnlich wie in der folgenden Abbildung gezeigt aussehen.

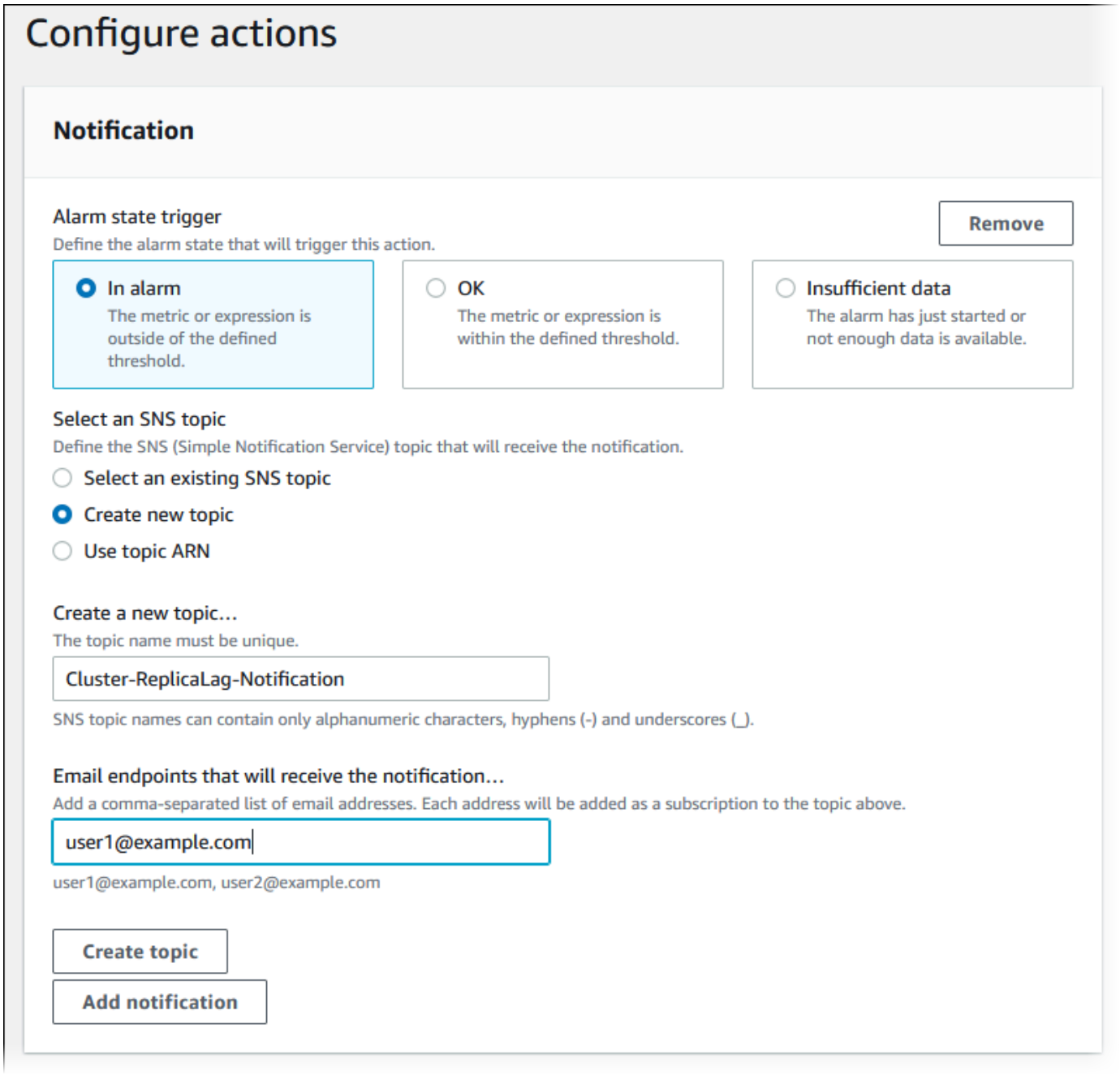


12. Wählen Sie Select metric (Metrik auswählen) aus.
13. Ändern Sie auf der Seite Specify metric and conditions (Metrik und Bedingungen angeben) das Label in einen aussagekräftigen Namen wie **`ClusterReplicaLag`**, und geben Sie eine Anzahl von Sekunden in Define the threshold value (Den Schwellenwert definieren) ein. Geben Sie für dieses Tutorial **`1200`** Sekunden (20 Minuten) ein. Sie können diesen Wert an Ihre Workload-Anforderungen anpassen.

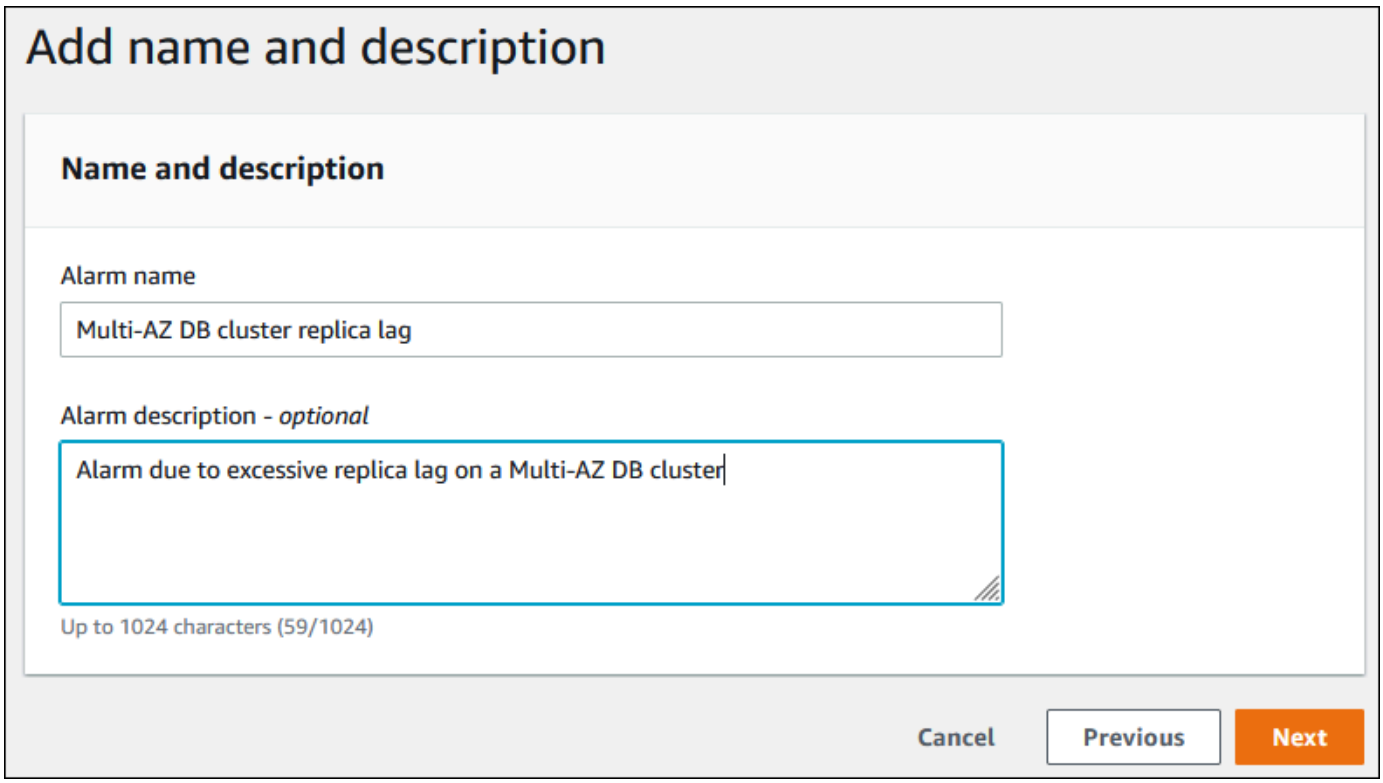
    Die Seite Specify metric and conditions (Metrik und Bedingungen angeben) sollte folgender Abbildung ähneln.


Specify metric and conditions
Metric
Edit
Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 1 minute.
No unit
1,000
500
0
17:00
18:00
19:00
ClusterReplicaLag
Label
ClusterReplicaLag
Math expression
MAX([m1,m2,m3])
Metrics
m1 | AWS/RDS | ReplicaLag | DBInstanceIdentifier : ...
m2 | AWS/RDS | ReplicaLag | DBInstanceIdentifier : ...
m3 | AWS/RDS | ReplicaLag | DBInstanceIdentifier : ...
Period
1 minute
Conditions
Threshold type
Static
Use a value as a threshold
Anomaly detection
Use a band as a threshold
Whenever ClusterReplicaLag is...
Define the alarm condition.
Greater
> threshold
Greater/Equal
>= threshold
Lower/Equal
<= threshold
Lower
< threshold
than...
Define the threshold value.
1200
Must be a number
Additional configuration
Cancel
Next

14. Wählen Sie Next (Weiter) aus, und die Seite Configure actions (Aktionen konfigurieren) wird angezeigt.
15. Lassen Sie In alarm (Im Alarm) ausgewählt, wählen Sie Create new topic (Neues Thema erstellen) aus und geben Sie den Themennamen und eine gültige E-Mail-Adresse ein.



16. Wählen Sie Create topic (Thema erstellen) und danach Next (Weiter).
17. Geben Sie auf der Seite Add name and description (Name und Beschreibung hinzufügen) den Alarm name (Alarmnamen) und die Alarm description (Alarmbeschreibung) ein und wählen Sie Next (Weiter) aus.



18. Zeigen Sie eine Vorschau des Alarms an, den Sie im Bereich Preview and create (Vorschau erstellen und erstellen) erstellen möchten, und wählen Sie dann Create alarm (Alarm erstellen).

Überwachung von Amazon RDS mit CloudWatch Database Insights

Überwachen Sie die Datenbanklast (DB Load) für Ihre Flotte von Amazon RDS Amazon mit Database Insights. DB Load misst den Grad der Sitzungsaktivität in Ihrer Datenbank. Sie können Database Insights verwenden, um die Leistung Ihrer Amazon RDS Amazon skalierbar zu analysieren und Fehler zu beheben.

Mit Database Insights können Sie die DB-Auslastung Ihrer Datenbankflotte visualisieren und die Auslastung nach Wartezeiten, SQL-Anweisungen, Hosts oder Benutzern filtern.

Standardmäßig aktiviert RDS den Standardmodus von Database Insights für Ihre Amazon RDS-Datenbanken.

Informationen zur Verwendung von Database Insights in der CloudWatch Amazon-Konsole finden Sie unter CloudWatch Database Insights im CloudWatch Amazon-Benutzerhandbuch.

Preisgestaltung

Informationen zur Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Themen

- Amazon RDS, Region und Instance-Klassenunterstützung für Database Insights
- Einschalten des erweiterten Modus von Database Insights für Amazon RDS
- Einschalten des Standardmodus von Database Insights für Amazon RDS
- Konfiguration Ihrer Datenbank zur Überwachung langsamer SQL-Abfragen mit Database Insights for Amazon RDS
- Überlegungen zu Database Insights für Amazon RDS

Amazon RDS, Region und Instance-Klassenunterstützung für Database Insights

Die folgende Tabelle enthält Amazon RDS Amazon , die Database Insights unterstützen.

Amazon-RDS-DB-Engine	Unterstützte Engine-Versionen und Regionen	Beschränkungen für Instance-Klasse
Amazon RDS für MariaDB	Weitere Informationen zur Version und regionalen Verfügbarkeit von Database Insights with RDS for MariaDB finden Sie unter. Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS	Database Insights wird für die folgenden Instanzklassen nicht unterstützt: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.klein
RDS for MySQL	Weitere Informationen zur Version und regionalen Verfügbarkeit von Database Insights with RDS for MySQL finden Sie unterUnterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	Database Insights wird für die folgenden Instanzklassen nicht unterstützt: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.klein

Amazon-RDS-DB-Engine	Unterstützte Engine-Versionen und Regionen	Beschränkungen für Instance-Klasse
Amazon RDS for Microsoft SQL Server	Weitere Informationen zur Verfügbarkeit von Database Insights mit RDS für SQL Server in Version und Region finden Sie unterUnterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	N/A
Amazon RDS für PostgreSQL	Weitere Informationen zur Version und regionalen Verfügbarkeit von Database Insights with RDS for PostgreSQL finden Sie unter. Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS	N/A
Amazon RDS für Oracle	Weitere Informationen zur Verfügbarkeit von Database Insights with RDS for Oracle in Version und Region finden Sie unter. Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS	N/A

Amazon RDS DB-Engine-, Regions- und Instance-Klassenunterstützung für Database Insights-Funktionen

Die folgende Tabelle enthält Amazon RDS Amazon , die Database Insights-Funktionen unterstützen.

Funktion	Preisstufe	Unterstützte Regionen	Unterstützte DB-Engines	Unterstützte Instance-Klassen
SQL-Statistiken für Performance Insights	Alle	Alle	Alle	Alle
Analysieren von Oracle-Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS	Alle	Alle	RDS für Oracle	Alle
Analysieren der Datenbankleistung für einen bestimmten Zeitraum	Nur kostenpflichtige Stufe	Alle	• RDS for MariaDB • RDS for MySQL • RDS for PostgreSQL	Alle
Proaktive Empfehlungen von Performance Insights anzeigen	Nur kostenpflichtige Stufe	• US East (Ohio) • USA Ost (Nord-Virginia) • USA West (Nordkalifornien) • USA West (Oregon) • Asia Pacific (Mumbai)	Alle	Alle

Funktion	Preisstufe	Unterstützte Regionen	Unterstützte DB-Engines	Unterstützte Instance-Klassen
		• Asia Pacific (Seoul) • Asien-Pazifik (Singapur) • Asien-Pazifik (Sydney) • Asien-Pazifik (Tokio) • Canada (Central) • Europe (Frankfurt) • Europa (Irland) • Europa (London) • Europe (Paris) • Europa (Stockholm) • Südamerika (São Paulo)		

Amazon RDS — Unterstützung der für Database Insights

Amazon RDS unterstützt Database Insights in den folgenden Bereichen AWS-Regionen.

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Africa (Cape Town)
- Asien-Pazifik (Hongkong)

- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Jakarta)
- Asien-Pazifik (Malaysia)
- Asien-Pazifik (Melbourne)
- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Osaka)
- Asien-Pazifik (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Kanada West (Calgary)
- Europe (Frankfurt)
- Europa (Irland)
- Europa (London)
- Europa (Milan)
- Europa (Paris)
- Europa (Spain)
- Europa (Stockholm)
- Europa (Zürich)
- Israel (Tel Aviv)
- Naher Osten (Bahrain)
- Naher Osten (VAE)
- Südamerika (São Paulo)
- AWS GovCloud (US-Ost)
- AWS GovCloud (US-West)

Einschalten des erweiterten Modus von Database Insights für Amazon RDS

Gehen Sie wie folgt vor, um den erweiterten Modus von Database Insights für Amazon RDS zu aktivieren.

Aktivieren Sie den erweiterten Modus von Database Insights, wenn Sie eine Datenbank für Amazon RDS erstellen.

Console

In der Konsole können Sie den erweiterten Modus von Database Insights aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen.

Um den erweiterten Modus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster mithilfe der Konsole erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie Datenbank erstellen aus.
4. Wählen Sie im Bereich Database Insights den Modus Erweitert aus. Wählen Sie dann die folgenden Optionen aus:
   - Retention (Aufbewahrung): die Zeit, wie lange Performance Insight-Daten aufbewahrt werden Die Aufbewahrungsfrist für den erweiterten Modus von Database Insights muss 15 Monate betragen.
   - AWS KMS key – Geben Sie Ihren KMS-Schlüssel an. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
5. Wählen Sie Datenbank erstellen aus.

AWS CLI

Um den erweiterten Modus von Database Insights beim Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters zu aktivieren, rufen Sie den create-db-cluster AWS CLI Befehl create-db-instanceor auf und geben Sie die folgenden Werte ein:

- `--database-insights-mode advancedum` den erweiterten Modus von Database Insights zu aktivieren.
- `--engine`— Die Datenbank-Engine für den .

- `--db-instance-identifier`— Der Bezeichner für die DB-Instance oder `--db-cluster-identifier` — Der Identifier für den Multi-AZ-DB-Cluster.
- `--enable-performance-insights`um Performance Insights für Database Insights zu aktivieren.
- `--performance-insights-retention-period`— Die Aufbewahrungsfrist für Daten für Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster. Um Database Insights zu aktivieren, muss der Aufbewahrungszeitraum mindestens 465 Tage betragen.

Im folgenden Beispiel wird der erweiterte Modus von Database Insights aktiviert, wenn ein erstellt wird.

Für Linux, macOS oder Unix:

```
aws rds create-db-instance \
    --database-insights-mode advanced \
    --engine postgresql \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 465
```

Für Windows:

```
aws rds create-db-instance ^
    --database-insights-mode advanced ^
    --engine postgresql ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 465
```

RDS API

Um den erweiterten Modus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen, geben Sie die folgenden Parameter für Ihren Vorgang Create DBInstance oder Create DBCluster Amazon RDS API an.

- `DatabaseInsightsMode` auf `advanced`
- `EnablePerformanceInsights` auf `True`
- `PerformanceInsightsRetentionPeriod`bis mindestens 465 Tage

Aktivieren Sie Database Insights, wenn Sie eine Datenbank für Amazon RDS ändern. Wenn Sie eine ändern, um den erweiterten Modus von Database Insights zu aktivieren, führt dies nicht zu Ausfallzeiten.

> **Note**
>
> Um Database Insights zu aktivieren, muss jede DB-Instance in einem dieselben Performance Insights- und Enhanced Monitoring-Einstellungen haben.

Console

In der Konsole können Sie den erweiterten Modus von Database Insights aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster ändern.

Um den erweiterten Modus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster mithilfe der Konsole ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie eine DB-Instance oder ein Multi-AZ-DB-Cluster, und wählen Sie Modify (Ändern) aus.
4. Wählen Sie im Bereich Database Insights den Modus Erweitert aus. Wählen Sie dann die folgenden Optionen aus:
   - Retention (Aufbewahrung): die Zeit, wie lange Performance Insight-Daten aufbewahrt werden Die Aufbewahrungsfrist für den erweiterten Modus von Database Insights muss 15 Monate betragen.
   - AWS KMS key – Geben Sie Ihren KMS-Schlüssel an. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
5. Klicken Sie auf Continue (Fortfahren).
6. Wählen Sie für Einplanung von Änderungen die Option Sofort anwenden aus. Wenn Sie während des nächsten geplanten Wartungsfensters „Anwenden“ wählen, ignoriert Ihre

Datenbank diese Einstellung und aktiviert sofort den erweiterten Modus von Database Insights.

7. Wählen Sie modifizieren aus.

AWS CLI

Um den erweiterten Modus von Database Insights beim Ändern einer DB-Instance oder eines Multi-AZ-DB-Clusters zu aktivieren, rufen Sie den modify-db-cluster AWS CLI Befehl modify-db-instanceor auf und geben Sie die folgenden Werte ein:

- `--database-insights-mode advanced`um den erweiterten Modus von Database Insights zu aktivieren.
- `--db-instance-identifier`— Der Bezeichner für die DB-Instance oder `--db-cluster-identifier` — Der Identifier für den Multi-AZ-DB-Cluster.
- `--enable-performance-insights`um Performance Insights für Database Insights zu aktivieren.
- `--performance-insights-retention-period`— Die Aufbewahrungsfrist für Daten für Ihre . Um den erweiterten Modus von Database Insights zu aktivieren, muss der Aufbewahrungszeitraum mindestens 465 Tage betragen.

Im folgenden Beispiel wird der erweiterte Modus von Database Insights aktiviert, wenn eine geändert wird.

Für Linux, macOS oder Unix:

```
aws rds modify-db-instance \
    --database-insights-mode advanced \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 465
```

Für Windows:

```
aws rds modify-db-instance ^
    --database-insights-mode advanced ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
```

```
    --performance-insights-retention-period 465
```

RDS API

Um den erweiterten Modus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen ändern, geben Sie die folgenden Parameter für Ihren DBCluster Amazon RDS-API-Vorgang „Ändern“ DBInstance oder „Ändern“ an.

- `DatabaseInsightsMode` auf `advanced`
- `EnablePerformanceInsights` auf `True`
- `PerformanceInsightsRetentionPeriod`bis mindestens 465 Tage

Einschalten des Standardmodus von Database Insights für Amazon RDS

Gehen Sie wie folgt vor, um den Standardmodus von Database Insights für Amazon RDS zu aktivieren.

Aktivieren Sie den Standardmodus von Database Insights, wenn Sie eine Datenbank für Amazon RDS erstellen.

Console

In der Konsole können Sie den Standardmodus von Database Insights aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen.

Um den Standardmodus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster mithilfe der Konsole erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie Datenbank erstellen aus.
4. Wählen Sie im Bereich Database Insights die Option Standardmodus aus. Wählen Sie dann aus den folgenden Optionen, um Performance Insights ein- oder auszuschalten:
    - Um Performance Insights zu deaktivieren, deaktivieren Sie die Option Performance Insights aktivieren.

- Um Performance Insights zu aktivieren, wählen Sie Performance Insights aktivieren aus. Um Performance Insights zu konfigurieren, geben Sie die folgenden Optionen an:
  - Retention (Aufbewahrung): die Zeit, wie lange Performance Insight-Daten aufbewahrt werden Die Aufbewahrungsfrist muss mindestens 7 Tage betragen.
  - AWS KMS key – Geben Sie Ihren KMS-Schlüssel an. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.

5. Wählen Sie Datenbank erstellen aus.

AWS CLI

Um den Standardmodus von Database Insights beim Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters zu aktivieren, rufen Sie den create-db-cluster AWS CLI Befehl create-db-instanceor auf und geben Sie die folgenden Werte ein:

- `--database-insights-mode standard`um den Standardmodus von Database Insights zu aktivieren.
- `--engine`— Die Datenbank-Engine für den .
- `--db-instance-identifier`— Der Bezeichner für die DB-Instance oder `--db-cluster-identifier` — Der Identifier für den Multi-AZ-DB-Cluster.
- `--enable-performance-insights`oder `--no-enable-performance-insights` um Performance Insights ein- oder auszuschalten. Wenn Sie dies angeben`--enable-performance-insights`, müssen Sie auch den `--performance-insights-retention-period` — Aufbewahrungszeitraum für Daten für Ihre angeben. Die Aufbewahrungsfrist muss mindestens 7 Tage betragen.

Das folgende Beispiel aktiviert den Standardmodus von Database Insights und Performance Insights beim Erstellen einer .

Für Linux, macOS oder Unix:

```
aws rds create-db-instance \
    --database-insights-mode standard \
    --engine postgresql \
    --db-instance-identifier sample-db-identifier \
```

```
    --enable-performance-insights \
    --performance-insights-retention-period 7
```

Für Windows:

```
aws rds create-db-instance ^
    --database-insights-mode standard ^
    --engine postgresql ^
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 7
```

Das folgende Beispiel aktiviert den Standardmodus von Database Insights und deaktiviert Performance Insights beim Erstellen einer .

Für Linux, macOS oder Unix:

```
aws rds create-db-instance \
    --database-insights-mode standard \
    --engine postgresql \
    --db-instance-identifier sample-db-identifier \
    --no-enable-performance-insights
```

Für Windows:

```
aws rds create-db-instance ^
    --database-insights-mode standard ^
    --engine postgresql ^
    --db-instance-identifier sample-db-identifier ^
    --no-enable-performance-insights
```

RDS API

Um den Standardmodus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen erstellen, geben Sie die folgenden Parameter für Ihren Vorgang Create DBInstance oder Create DBCluster Amazon RDS API an.

- `DatabaseInsightsMode` auf `standard`
- `EnablePerformanceInsights`zu `True` oder`False`. Wenn Sie `EnablePerformanceInsights` auf einstellen`True`, müssen Sie `PerformanceInsightsRetentionPeriod` auf mindestens 7 Tage einstellen.

Aktivieren Sie den Standardmodus von Database Insights, wenn Sie eine Datenbank für Amazon RDS ändern. Wenn Sie eine ändern, um den Standardmodus von Database Insights zu aktivieren, führt dies nicht zu Ausfallzeiten.

> **Note**
>
> Um Database Insights zu aktivieren, muss jede DB-Instance in einem dieselben Performance Insights- und Enhanced Monitoring-Einstellungen haben.

Console

In der Konsole können Sie den Standardmodus von Database Insights aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster ändern.

Um den Standardmodus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster mithilfe der Konsole ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie eine DB-Instance oder ein Multi-AZ-DB-Cluster, und wählen Sie Modify (Ändern) aus.
4. Wählen Sie im Bereich Database Insights die Option Standardmodus aus. Wählen Sie dann aus den folgenden Optionen:

   - Um Performance Insights zu deaktivieren, deaktivieren Sie die Option Performance Insights aktivieren.
   - Um Performance Insights zu aktivieren, wählen Sie Performance Insights aktivieren aus. Um Performance Insights zu konfigurieren, geben Sie die folgenden Optionen an:
     - Retention (Aufbewahrung): die Zeit, wie lange Performance Insight-Daten aufbewahrt werden Die Aufbewahrungsfrist muss mindestens 7 Tage betragen.
     - AWS KMS key – Geben Sie Ihren KMS-Schlüssel an. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.

5. Klicken Sie auf Continue (Fortfahren).
6. Wählen Sie für Einplanung von Änderungen die Option Sofort anwenden aus. Wenn Sie während des nächsten geplanten Wartungsfensters Anwenden wählen, ignoriert Ihre Datenbank diese Einstellung und aktiviert sofort den Standardmodus von Database Insights.
7. Wählen Sie modifizieren aus.

AWS CLI

Um den Standardmodus von Database Insights bei der Änderung einer DB-Instance oder eines Multi-AZ-DB-Clusters zu aktivieren, rufen Sie den modify-db-cluster AWS CLI Befehl modify-db-instanceor auf und geben Sie die folgenden Werte ein:

- `--database-insights-mode standard`um den Standardmodus von Database Insights zu aktivieren.
- `--db-instance-identifier`— Der Bezeichner für die DB-Instance oder `--db-cluster-identifier` — Der Identifier für den Multi-AZ-DB-Cluster.
- `--enable-performance-insights`oder `--no-enable-performance-insights` um Performance Insights ein- oder auszuschalten. Wenn Sie dies angeben`--enable-performance-insights`, müssen Sie auch den `--performance-insights-retention-period` — Aufbewahrungszeitraum für Daten für Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster angeben. Die Aufbewahrungsfrist muss mindestens 7 Tage betragen.

Das folgende Beispiel aktiviert den Standardmodus von Database Insights und aktiviert Performance Insights, wenn eine geändert wird.

Für Linux, macOS oder Unix:

```
aws rds modify-db-instance \
    --database-insights-mode standard \
    --db-instance-identifier sample-db-identifier \
    --enable-performance-insights \
    --performance-insights-retention-period 7
```

Für Windows:

```
aws rds modify-db-instance ^
    --database-insights-mode standard ^
```

```
    --db-instance-identifier sample-db-identifier ^
    --enable-performance-insights ^
    --performance-insights-retention-period 7
```

Das folgende Beispiel aktiviert den Standardmodus von Database Insights und deaktiviert Performance Insights, wenn eine geändert wird.

Für Linux, macOS oder Unix:

```
aws rds modify-db-instance \
    --database-insights-mode standard \
    --db-instance-identifier sample-db-identifier \
    --no-enable-performance-insights
```

Für Windows:

```
aws rds modify-db-instance ^
    --database-insights-mode standard ^
    --db-instance-identifier sample-db-identifier ^
    --no-enable-performance-insights
```

RDS API

Um den Standardmodus von Database Insights zu aktivieren, wenn Sie eine DB-Instance oder einen ändern, geben Sie die folgenden Parameter für Ihren DBCluster Amazon RDS-API-Vorgang „Ändern“ DBInstance oder „Ändern“ an.

- `DatabaseInsightsMode` auf `standard`
- `EnablePerformanceInsights`zu `True` oder`False`. Wenn Sie `EnablePerformanceInsights` auf einstellen`True`, müssen Sie `PerformanceInsightsRetentionPeriod` auf mindestens 7 Tage einstellen.

Konfiguration Ihrer Datenbank zur Überwachung langsamer SQL-Abfragen mit Database Insights for Amazon RDS

Um langsame SQL-Abfragen für Ihre Datenbank zu überwachen, können Sie den Abschnitt Langsame SQL-Abfragen im Database Insights-Dashboard verwenden. Bevor Sie Ihre Datenbank für die Überwachung langsamer SQL-Abfragen konfigurieren, ist der Abschnitt Langsame SQL-Abfragen leer.

Weitere Informationen zur Überwachung langsamer SQL-Abfragen im Database Insights-Dashboard finden Sie unter Viewing the Database Instance Dashboard for CloudWatch Database Insights im CloudWatch Amazon-Benutzerhandbuch.

Gehen Sie wie folgt vor, um Ihre Datenbank für die Überwachung langsamer SQL-Abfragen mit Database Insights zu konfigurieren:

1. Aktivieren Sie Protokollexporte in CloudWatch Logs.
2. Erstellen oder ändern Sie die DB-Parametergruppe für Ihre DB-Instance.

Informationen zur Konfiguration von Protokollexporten finden Sie unter Veröffentlichen von CloudWatch Datenbankprotokollen in Amazon Logs im Amazon RDS-Benutzerhandbuch .

Informationen zum Erstellen oder Ändern Ihrer DB−Parametergruppe finden Sie in den folgenden Themen.

- Erstellen einer DB-Parametergruppe in Amazon RDS
- Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS

RDS for MariaDB

Um Ihre RDS for MariaDB-DB-Instance für die Überwachung langsamer SQL-Abfragen zu konfigurieren, können Sie die folgende Parameterkombination als Beispiel verwenden:

- `log_slow_query`— eingestellt auf `1`
- `log_slow_query_time`— eingestellt auf `1.0`
- `log_output`— eingestellt auf `FILE`

Dies ist eine mögliche Konfiguration. Eine umfassende Anleitung zu den Protokollparametern für langsame Abfragen von MariaDB und zusätzlichen Konfigurationsoptionen finden Sie in der MariaDB-Dokumentation für das Slow Query Log.

RDS for MySQL

Um Ihre RDS for MySQL-DB-Instance für die Überwachung langsamer SQL-Abfragen zu konfigurieren, können Sie die folgende Parameterkombination als Beispiel verwenden:

- `slow_query_log`— eingestellt auf `1`

- `long_query_time`— eingestellt auf `1.0`
- `log_output`— eingestellt auf `FILE`

Dies ist eine mögliche Konfiguration. Eine umfassende Anleitung zu den Logparametern für langsame Abfragen von MySQL und zusätzlichen Konfigurationsoptionen finden Sie in der MySQL-Dokumentation für das Slow Query Log.

RDS for PostgreSQL

Um Ihre RDS for PostgreSQL-DB-Instance für die Überwachung langsamer SQL-Abfragen zu konfigurieren, können Sie die folgende Parameterkombination als Beispiel verwenden. Beachten Sie, dass die Einstellung dieser Parameter die Leistung Ihrer DB-Instance beeinträchtigen kann.

- `log_min_duration_statement`— eingestellt auf `1000`
- `log_statement`— eingestellt auf `none`
- `log_destination`— eingestellt auf `stderr`

Dies ist eine mögliche Konfiguration. Eine umfassende Anleitung zu PostgreSQL-Logging-Parametern und zusätzlichen Konfigurationsoptionen finden Sie in der PostgreSQL-Dokumentation zur Logging-Konfiguration.

> **Note**
>
> Für RDS for MySQL können Sie den Parameter `long_query_time` mit einer Granularität von 1 Mikrosekunde konfigurieren. Sie können diesen Parameter beispielsweise auf festlegen. `0.000001` Abhängig von der Anzahl der Abfragen auf der DB-Instance `long_query_time` kann der Wert des Parameters die Leistung verringern. Beginnen Sie mit dem Wert `1.0` und passen Sie ihn an Ihre Arbeitslast an. Wenn Sie diesen Parameter auf setzen`0`, protokolliert Database Insights alle Abfragen.

Informationen zu RDS für MariaDB, RDS für MySQL und RDS für PostgreSQL-Logs finden Sie im Folgenden.

- MariaDB-Datenbank-Protokolldateien
- MySQL-Datenbank-Protokolldateien

- Datenbank-Protokolldateien von RDS für PostgreSQL

Überlegungen zu Database Insights für Amazon RDS

Im Folgenden finden Sie Überlegungen zu Database Insights für Amazon RDS .

- Sie können Database Insights für eine DB-Instance in einem DB-Cluster, , nicht verwalten.
- Um den erweiterten Modus von Database Insights zu aktivieren, müssen Sie Performance Insights aktivieren und die Aufbewahrungsfrist für Performance Insights auf mindestens 465 Tage (15 Monate) festlegen. Für die Festlegung der Aufbewahrungsfrist von Performance Insights auf 15 Monate fallen neben den Kosten für Database Insights keine zusätzlichen Kosten an. Informationen zu den Preisen für Database Insights finden Sie unter CloudWatch Amazon-Preise.
- Um Database Insights zu aktivieren, muss jede DB-Instance in einem dieselben Performance Insights- und Enhanced Monitoring-Einstellungen haben.
- Wenn Sie eine ändern, um einen der beiden Modi von Database Insights zu aktivieren, führt dies nicht zu Ausfallzeiten.

Überwachung mit Performance Insights auf Amazon RDS

Important

AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

Performance Insights erweitert vorhandene Amazon-RDS-Überwachungsfunktionen, um die Leistung Ihres Datenbank zu veranschaulichen und ihre Analyse zu unterstützen. Mit dem Performance Insights-Dashboard können Sie die Datenbanklast Ihrer Amazon-RDS-DB-Instance visualisieren und die Last nach Wartezeiten, SQL-Anweisungen, Hosts oder Benutzern filtern. Weitere Informationen zur Verwendung von Performance Insights mit Amazon DocumentDB finden Sie im Amazon-DocumentDB-Entwicklerhandbuch.

Themen

- Übersicht über Performance-Insights für Amazon RDS

- Performance Insights für Amazon RDS ein- und ausschalten
- Überblick über das Leistungsschema für Performance Insights auf , Amazon RDS for MariaDB oder MySQL
- Konfigurieren von Zugriffsrichtlinien für Performance Insights
- Analyse der Metriken mit dem Performance Insights-Dashboard
- Proaktive Empfehlungen von Performance Insights anzeigen
- Metriken mit Performance Insights API für Amazon abrufen RDS
- Protokollieren von Performance Insights Insights-Aufrufen mit AWS CloudTrail
- Performance Insights API und VPC Schnittstellenendpunkte ()AWS PrivateLink

Übersicht über Performance-Insights für Amazon RDS

Important

AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit

CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

Standardmäßig aktiviert RDS Performance Insights im Assistenten zur Konsolenerstellung für alle Amazon RDS-Engines. Wenn mehr als eine Datenbank auf einer DB-Instance vorhanden ist, aggregiert Performance Insights Leistungsdaten.

Das folgende Video gibt eine Übersicht über Performance Insights für Amazon RDS.

Verwenden von Performance Insights, um die Leistung von Amazon Aurora-PostgreSQL zu analysieren

Important

In diesem Handbuch wird die Verwendung von Amazon RDS Performance Insights mit anderen als Aurora-DB-Engines beschrieben. Informationen zum Verwenden von Amazon RDS Performance Insights mit Amazon Aurora finden Sie unter Verwenden von Amazon RDS Performance Insights im Amazon Aurora-Benutzerhandbuch.

Themen

- Datenbanklast
- Maximal CPU
- DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance Insights
- Preisgestaltung und Datenspeicherung für Performance Insights

Datenbanklast

Die Datenbanklast (DB-Last) misst den Grad der Sitzungsaktivität in Ihrer Datenbank. `DBLoad`ist die wichtigste Metrik in Performance Insights, und Performance Insights erfasst jede Sekunde die Datenbanklast.

Themen

- Aktive Sitzungen

- Durchschnittliche aktive Sitzungen
- Durchschnittliche aktive Ausführungen
- Dimensionen

Aktive Sitzungen

Eine Datenbank-Sitzung repräsentiert den Dialog einer Anwendung mit einer relationalen Datenbank. Eine aktive Sitzung ist eine Verbindung, die Arbeit an die DB-Engine gesendet hat und auf eine Antwort wartet.

Eine Sitzung ist aktiv, wenn sie entweder auf der CPU läuft oder darauf wartet, dass eine Ressource verfügbar wird, damit sie fortfahren kann. Beispielsweise kann eine aktive Sitzung warten, bis eine Seite (oder ein Block) in den Speicher eingelesen wird, und verbraucht dann CPU, während sie Daten von der Seite liest.

Durchschnittliche aktive Sitzungen

Die durchschnittlich aktive Sitzungen (AAS) ist die Einheit für die `DBLoad`-Metrik in Performance Insights. Es wird gemessen, wie viele Sitzungen gleichzeitig in der Datenbank aktiv sind.

Jede Sekunde ruft Performance-Insights eine Stichprobe der Anzahl der Sitzungen ab, die gleichzeitig eine Abfrage ausführen. Für jede aktive Sitzung sammelt Performance Insights die folgenden Daten:

- SQL-Anweisung
- Sitzungsstatus (läuft auf der CPU oder wartet)
- Host
- Benutzer, der SQL ausführt

Performance Insights berechnet die AAS durch Dividieren der Gesamtzahl der Sitzungen durch die Gesamtzahl der Stichproben für einen bestimmten Zeitraum. Die folgende Tabelle zeigt beispielsweise 5 aufeinander folgende Stichproben einer laufenden Abfrage, die in Intervallen von 1 Sekunde abgefragt wurden.

Beispiel	Anzahl der Sitzungen, die eine Abfrage ausführen	AAS	Berechnung
1	2	2	2 Sitzungen insgesamt / 1 Stichprobe
2	0	1	2 Sitzungen insgesamt / 2 Stichproben
3	4	2	6 Sitzungen insgesamt / 3 Stichproben
4	0	1.5	6 Sitzungen insgesamt / 4 Stichproben
5	4	2	10 Sitzungen insgesamt / 5 Stichproben

Im vorherigen Beispiel beträgt die DB-Last für das Zeitintervall 2 AAS. Diese Messung bedeutet, dass im Durchschnitt 2 Sitzungen gleichzeitig während des Zeitraums aktiv waren, in dem die 5 Stichproben erfasst wurden.

Durchschnittliche aktive Ausführungen

Die durchschnittlichen aktiven Ausführungen (AAE) pro Sekunde stehen im Zusammenhang mit AAS. Um die AAE zu berechnen, teilt Performance Insights die Gesamtausführungszeit einer Abfrage durch das Zeitintervall. Die folgende Tabelle zeigt die AAE-Berechnung für dieselbe Abfrage in der vorherigen Tabelle.

Verstrichene Zeit	Gesamtausführungszeit	AAE	Berechnung
60	120	2	120 Durchführungssekunden/60 verstrichene Sekunden

Verstrichene Zeit	Gesamtausführungszeit	AAE	Berechnung
120	120	1	120 Durchführungssekunden/120 verstrichene Sekunden
180	380	2.11	380 Ausführungssekunden/180 verstrichene Sekunden
240	380	1.58	380 Ausführungssekunden/240 verstrichene Sekunden
300	600	2	600 Durchführungssekunden/300 verstrichene Sekunden

In den meisten Fällen sind die AAS und die AAE für eine Abfrage ungefähr gleich. Da es sich bei den Eingaben zu den Berechnungen jedoch um unterschiedliche Datenquellen handelt, variieren die Berechnungen häufig geringfügig.

Dimensionen

Die`db.load` Metrik unterscheidet sich von den anderen Zeitreihenmetriken, da Sie sie in Unterkomponenten aufteilen können, die als Dimensionen bezeichnet werden. Sie können sich Dimensionen als „Aufteilungs“-Kategorien für die verschiedenen Merkmale der `DBLoad`-Metrik vorstellen.

Wenn Sie Leistungsprobleme diagnostizieren, sind die folgenden Dimensionen oft am nützlichsten:

Themen

- Warteereignisse
- Haupt-SQL
- Plans (Pläne)

Eine vollständige Liste der Dimensionen für die Amazon-RDS-Engines finden Sie unter DB-Last aufgeteilt nach Dimensionen.

Warteereignisse

Ein Warteereignis bewirkt, dass eine SQL-Anweisung wartet, bis ein bestimmtes Ereignis eintritt, bevor sie mit der Ausführung fortfahren kann. Warteereignisse sind eine wichtige Dimension oder Kategorie für die DB-Last, da sie angeben, wo die Arbeit behindert wird.

Jede aktive Sitzung läuft entweder auf der CPU oder wartet. Sitzungen verbrauchen beispielsweise CPU, wenn sie Speicher nach einem Puffer suchen, eine Berechnung durchführen oder Prozeduralcode ausführen. Wenn Sitzungen keine CPU verbrauchen, warten sie möglicherweise darauf, dass ein Speicherpuffer frei wird, eine Datendatei gelesen oder ein Protokoll geschrieben wird. Je mehr Zeit eine Sitzung auf Ressourcen wartet, desto weniger Zeit läuft sie auf der CPU.

Wenn Sie eine Datenbank tunen, versuchen Sie oft, die Ressourcen herauszufinden, auf die Sitzungen warten. Beispielsweise könnten zwei oder drei Warteereignisse 90 Prozent der DB-Last ausmachen. Diese Maßnahme bedeutet, dass aktive Sitzungen im Durchschnitt die meiste Zeit damit verbringen, auf eine kleine Anzahl von Ressourcen zu warten. Wenn Sie die Ursache dieser Wartezeiten herausfinden können, können Sie eine Lösung versuchen.

Die Warteereignisse variieren je nach DB-Engine:

- Informationen zu allen MariaDB- und MySQL-Warteereignissen finden Sie unter Wait Event Summary Tables in der MySQL-Dokumentation.
- Informationen zu allen PostgreSQL-Warteereignissen finden Sie unter Der Statistikkollektor > Warteereignistabellen in der PostgreSQL-Dokumentation.
- Informationen zu allen Oracle-Warteereignissen finden Sie unter Descriptions of Wait Events in der Oracle-Dokumentation.
- Informationen zu allen SQL-Warteereignissen finden Sie unter Types of Wait in der SQL Server-Dokumentation.

> **Note**
>
> Bei Oracle arbeiten Hintergrundprozesse manchmal ohne eine verknüpfte SQL-Anweisung. In diesen Fällen meldet Performance Insights die Art des Hintergrundprozesses, der mit einem Doppelpunkt und der diesem Hintergrundprozess zugeordneten Warteklasse verknüpft ist. Zu Arten des Hintergrundprozesses gehören `LGWR`, `ARC0`, `PMON` usw.

Wenn das Archivierungsprogramm beispielsweise I/O ausführt, ist der Performance Insights-Bericht ähnlic `ARC1:System I/O`. Gelegentlich fehlt auch der Hintergrundprozesstyp, und Performance Insights meldet nur die Warteklasse, z. B. `:System I/O`.

Haupt-SQL

Während Warteereignisse Engpässe zeigen, zeigt Top-SQL an, welche Abfragen am meisten zur DB-Last beitragen. Beispielsweise könnten derzeit viele Abfragen gleichzeitig in der Datenbank ausgeführt werden, aber eine einzelne Abfrage könnte 99 % der DB-Last verbrauchen. In diesem Fall könnte die hohe Belastung auf ein Problem mit der Abfrage hinweisen.

Standardmäßig zeigt die Performance Insights-Konsole die wichtigsten SQL-Abfragen an, die zur Datenbanklast beitragen. Die Konsole zeigt auch relevante Statistiken für jede Anweisung an. Um Leistungsprobleme für eine bestimmte Anweisung zu diagnostizieren, können Sie deren Ausführungsplan untersuchen.

Plans (Pläne)

Ein Ausführungsplan, auch einfach als Plan bezeichnet, ist eine Abfolge von Schritten, mit denen auf Daten zugegriffen wird. Ein Plan zum Verbinden der Tabellen `t1` und `t2` könnte beispielsweise alle Zeilen in `t1` durchlaufen und jede Zeile mit einer Zeile in `t2` vergleichen. In einer relationalen Datenbank ist ein Optimierer integrierter Code, der den effizientesten Plan für eine SQL-Abfrage ermittelt.

Für DB-Instances sammelt Performance Insights automatisch Ausführungspläne. Um SQL-Leistungsprobleme zu diagnostizieren, untersuchen Sie die erfassten Pläne für SQL-Abfragen mit hohem Ressourcenbedarf. Die Pläne zeigen, wie die Datenbank Abfragen analysiert und ausgeführt hat.

Informationen zur Analyse der Datenbanklast mithilfe von Plänen finden Sie unter:

- Oracle: Analysieren von Oracle-Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS
- SQL Server: Analysieren von SQL Serverausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS

Erfassung der Pläne

Alle fünf Minuten identifiziert Performance Insights die ressourcenintensivsten Abfragen und erfasst deren Pläne. So müssen Sie nicht eine riesige Zahl von Plänen manuell erfassen und verwalten. Stattdessen können Sie die Registerkarte Top SQL (Top-SQL) verwenden, um sich auf die Pläne für die problematischsten Abfragen zu konzentrieren.

> **Note**
>
> Performance Insights erfasst keine Pläne für Abfragen, deren Text die Obergrenze für erfassbaren Abfragetext überschreitet. Weitere Informationen finden Sie unter Zugriff auf mehr SQL-Text im Performance-Insights-Dashboard.

Der Aufbewahrungszeitraum für Ausführungspläne ist der gleiche wie für Ihre Performance-Insights-Daten. Die Aufbewahrungseinstellung ist Standard (7 Tage). Um Ihre Leistungsdaten länger aufzubewahren, geben Sie 1–24 Monate an. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.

Digest-Abfragen

Auf der Registerkarte Top SQL (Top-SQL) werden standardmäßig Digest-Abfragen angezeigt. Eine Digest-Abfrage hat selbst keinen Plan, aber alle Abfragen, die Literalwerte verwenden, haben Pläne. Eine Digest-Abfrage könnte beispielsweise den Text `WHERE \`email\`=?` enthalten. Das Digest kann zwei Abfragen enthalten, eine mit dem Text `WHERE email=user1@example.com` und eine mit `WHERE email=user2@example.com`. Jede dieser Literalabfragen kann mehrere Pläne umfassen.

Wenn Sie eine Digest-Abfrage auswählen, zeigt die Konsole alle Pläne für untergeordnete Anweisungen der ausgewählten Zusammenfassung an. So müssen Sie nicht alle untergeordneten Anweisungen durchsehen, um den Plan zu finden. Möglicherweise sehen Sie Pläne, die nicht in der angezeigten Liste der Top 10 der untergeordneten Anweisungen enthalten sind. Die Konsole zeigt Pläne für alle untergeordneten Abfragen an, für die Pläne erfasst wurden, unabhängig davon, ob sich die Abfragen unter den Top 10 befinden.

Maximal CPU

Das Diagramm Datenbanklast im Dashboard dient zum Erfassen, Aggregieren und Anzeigen von Sitzungsinformationen. Um festzustellen, ob aktive Sitzungen das Maximum überschreitenCPU, sehen Sie sich ihr Verhältnis zur Max CPU v-Zeile an. Performance Insights bestimmt den Max CPU v-Wert anhand der Anzahl der v CPU (virtuellenCPU) Kerne für Ihre DB-Instance.

Ein Prozess kann jeweils auf einem v CPU ausgeführt werden. Wenn die Anzahl der Prozesse die Anzahl von überschreitetvCPUs, werden die Prozesse in die Warteschlange gestellt. Wenn die Warteschlange zunimmt, nimmt die Datenbankleistung ab. Wenn die Datenbanklast häufig über der Max CPU V-Linie liegt und der primäre Wartestatus gleich istCPU, CPU ist der überlastet. In diesem Fall möchten Sie möglicherweise die Verbindungen zur Instance drosseln, alle SQL Abfragen mit hoher CPU Last optimieren oder eine größere Instance-Klasse in Betracht ziehen. Hohe und konsistente Instances von Wartezuständen deuten darauf hin, dass es möglicherweise Engpässe oder Probleme mit Ressourcenkonflikten gibt, die behoben werden müssen. Dies kann auch dann der Fall sein, wenn die DB-Last die CPUMax-V-Grenze nicht überschreitet.

DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance Insights

Important

AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit

CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

Die folgende Tabelle enthält DB-Engines von Amazon RDS, die Performance Insights unterstützen.

> **Note**
>
> Informationen zu Amazon Aurora finden Sie unter Amazon-Aurora-DB-Engine-Unterstützung für Performance Insights im Amazon-Aurora-Benutzerhandbuch.

Amazon-RDS-DB-Engine	Unterstützte Engine-Versionen und Regionen	Beschränkungen für Instance-Klasse
Amazon RDS für MariaDB	Weitere Informationen zur Versions- und Regionsverfügbarkeit von Performance Insights with RDS for MariaDB finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	Performance Insights für wird in den folgenden Instance-Klassen nicht unterstützt: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.klein
RDS for MySQL	Weitere Informationen zur Versions- und Regionsverfügbarkeit von Performance Insights with RDS for MySQL finden Sie unter Unterstützte Regionen und DB-	Performance Insights für wird in den folgenden Instance-Klassen nicht unterstützt: • db.t2.micro •

Amazon-RDS-DB-Engine	Unterstützte Engine-Versionen und Regionen	Beschränkungen für Instance-Klasse
	Engines für Performance Insights in Amazon RDS.	db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.klein
Amazon RDS for Microsoft SQL Server	Weitere Informationen zur Versions- und Regionsverfügbarkeit von Performance Insights with RDS for SQL Server finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	N/A
Amazon RDS für PostgreSQL	Weitere Informationen zur Versions- und Regionsverfügbarkeit von Performance Insights with RDS for PostgreSQL finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	N/A
Amazon RDS für Oracle	Weitere Informationen zur Versions- und Regionsverfügbarkeit von Performance Insights with RDS for Oracle finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	N/A

DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen

Die folgende Tabelle enthält DB-Engines von Amazon RDS, die Performance-Insights-Funktionen unterstützen.

Funktion	Preisstufe	Unterstützte Regionen	Unterstützte DB-Engines	Unterstützte Instance-Klassen
SQL-Statistiken für Performance Insights	Alle	Alle	Alle	Alle
Analysieren von Oracle-Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS	Alle	Alle	RDS für Oracle	Alle
Analysieren der Datenbankleistung für einen bestimmten Zeitraum	Nur kostenpflichtige Stufe	Alle	RDS for PostgreSQL	Alle
Proaktive Empfehlungen von Performance Insights anzeigen	Nur kostenpflichtige Stufe	• US East (Ohio) • USA Ost (Nord-Virginia) • USA West (Nordkalifornien) • USA West (Oregon)	Alle	Alle

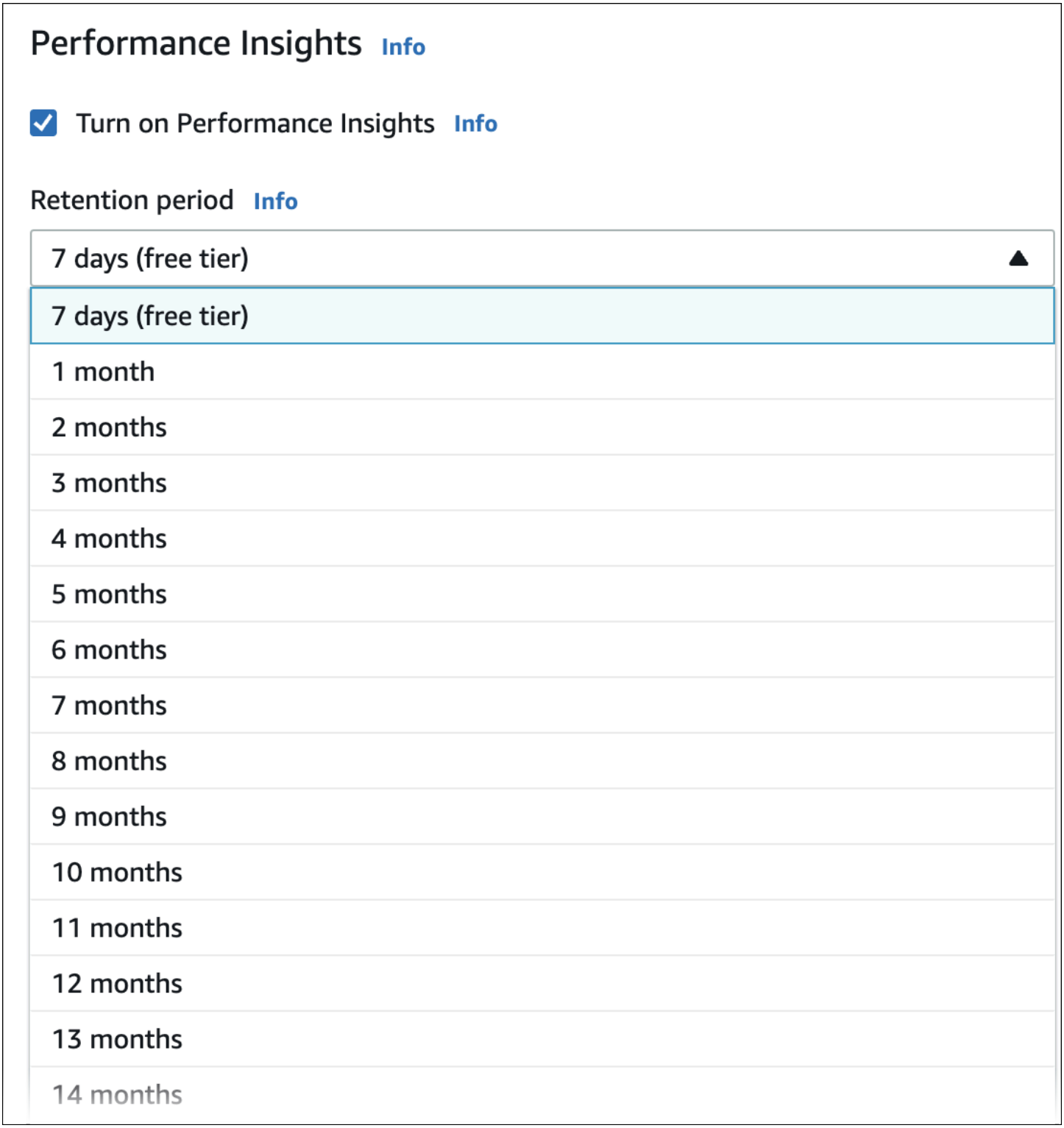
Funktion	Preisstufe	Unterstützte Regionen	Unterstützte DB-Engines	Unterstützte Instance-Klassen
		• Asia Pacific (Mumbai) • Asia Pacific (Seoul) • Asien-Pazifik (Singapur) • Asien-Pazifik (Sydney) • Asien-Pazifik (Tokio) • Canada (Central) • Europe (Frankfurt) • Europa (Irland) • Europa (London) • Europe (Paris) • Europa (Stockholm) • Südamerika (São Paulo)		

Preisgestaltung und Datenspeicherung für Performance Insights

Standardmäßig umfasst Performance Insights einen Leistungsdatenverlauf von 7 Tagen und 1 Million API-Anfragen pro Monat. Sie können auch längere Aufbewahrungsfristen erwerben. Umfassende Informationen zur Preisgestaltung finden Sie unter Performance Insights – Preise.

In der RDS-Konsole können Sie einen der folgenden Aufbewahrungszeiträume für Ihre Performance Insights Insights-Daten auswählen:

- Standard (7 Tage)
- ***n***Monate, wobei ***n***eine Zahl zwischen 1 und 24 liegt



Informationen zum Einstellen einer Aufbewahrungsfrist mithilfe von finden Sie AWS CLI unterPerformance Insights für Amazon RDS ein- und ausschalten.

> **Note**
>
> Das Stoppen einer DB-Instance oder eines mit aktiviertem Performance Insights hat keine Auswirkungen auf die Datenspeicherung. Solange eine DB-Instance oder ein gestoppt ist, sammelt Performance Insights keine Daten.

Performance Insights für Amazon RDS ein- und ausschalten

> **Important**
>
> AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
> Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
> Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
> Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

Sie können Performance Insights für Ihre oder Ihren Multi-AZ-DB-Cluster aktivieren, wenn Sie ihn erstellen. Bei Bedarf können Sie es später deaktivieren, indem Sie Ihre von der Konsole aus ändern. Das Aktivieren und Deaktivieren von Performance Insights führt nicht zu Ausfallzeiten, einem Neustart oder einem Failover.

> **Note**
>
> Das Leistungsschema ist ein optionales Leistungstool, das von Amazon RDS for MariaDB oder MySQL verwendet wird. Nach dem Aktivieren oder Deaktivieren des Leistungsschemas müssen Sie neu starten. Wenn Sie jedoch Performance Insights aktivieren oder deaktivieren, müssen Sie keinen Neustart durchführen. Weitere Informationen finden Sie unter Überblick über das Leistungsschema für Performance Insights auf , Amazon RDS for MariaDB oder MySQL.

Der Performance Insights-Agent verbraucht eine begrenzte Menge an CPU und Arbeitsspeicher auf dem DB-Host. Wenn die DB-Last hoch ist, begrenzt der Agent die Auswirkungen auf die Leistung, indem Daten seltener erfasst werden.

Console

In der Konsole können Sie Performance Insights aktivieren oder deaktivieren, wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen oder ändern.

Performance Insights beim Erstellen einer DB-Instance oder eines ein- oder ausschalten

Nach der Erstellung einer neuen DB-Instance oder eines aktiviert Amazon RDS standardmäßig Performance Insights. Um Performance Insights zu deaktivieren, wählen Sie die Option Database Insights — Standard und deaktivieren Sie die Option Performance Insights aktivieren.

Weitere Informationen finden Sie unter den folgenden Themen.

- Um eine DB-Instance zu erstellen, befolgen Sie die Anweisungen für Ihre spezifische DB-Engine in Erstellen einer Amazon RDS-DB-Instance.
- Befolgen Sie die Anweisungen für Ihre DB-Engine unter Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen, um einen Multi-AZ-DB-Cluster zu erstellen.

Der folgende Screenshot zeigt den Abschnitt Performance-Insights.



Wenn Sie Performance-Insights aktivieren wählen, haben Sie die folgenden Optionen:

- Aufbewahrung (nur für den Standardmodus von Database Insights) — Die Dauer der Aufbewahrung von Performance Insights Insights-Daten. Die Aufbewahrungseinstellung ist Standard (7 Tage). Um Ihre Leistungsdaten länger aufzubewahren, geben Sie 1–24 Monate an. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.
- AWS KMS key— Geben Sie Ihre an AWS KMS key. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Änderung einer AWS KMS Richtlinie für Performance Insights.

Performance Insights beim Ändern einer ein- oder ausschalten

In der Konsole können Sie oder einen Multi-AZ-DB-Cluster ändern, um Performance Insights zu verwalten.

So verwalten Sie Performance Insights für eine DB-Instance oder einen mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbanken aus.
3. Wählen Sie eine oder einen Multi-AZ-DB-Cluster und wählen Sie Modify.

4. Um Performance Insights zu aktivieren, wählen Sie Performance Insights aktivieren aus. Um Performance Insights zu deaktivieren, wählen Sie die Option Database Insights — Standard und deaktivieren Sie die Option Performance Insights aktivieren.

   Wenn Sie Performance-Insights aktivieren wählen, haben Sie die folgenden Optionen:

   - Aufbewahrung (nur für den Standardmodus von Database Insights) — Die Dauer der Aufbewahrung von Performance Insights Insights-Daten. Die Aufbewahrungseinstellung ist Standard (7 Tage). Um Ihre Leistungsdaten länger aufzubewahren, geben Sie 1–24 Monate an. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.
   - AWS KMS key – Geben Sie Ihren KMS-Schlüssel an. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
5. Klicken Sie auf Weiter.
6. Wählen Sie für Einplanung von Änderungen die Option Sofort anwenden aus. Wenn Sie im nächsten geplanten Wartungsfenster die Option Anwenden wählen, ignoriert Ihre Instance diese Einstellung und aktiviert Performance Insights sofort.
7. Wählen Sie Modify instance (Instance ändern).

AWS CLI

Wenn Sie den create-db-instance AWS CLI Befehl verwenden, aktivieren Sie Performance Insights, indem Sie entweder `advanced` oder angeben `--enable-performance-insights` und `--database-insights-mode` darauf setzen`standard`. Um Performance Insights zu deaktivieren, geben Sie an `--no-enable-performance-insights` und stellen Sie `database-insights-mode` es auf ein`standard`.

Sie können diese Werte auch mit den folgenden AWS CLI Befehlen angeben:

- create-db-cluster
- modify-db-cluster
- create-db-instance-read-replikat
- modify-db-instance
- restore-db-instance-from-3

Wenn Sie Performance Insights in der CLI aktivieren, können Sie mit der `--performance-insights-retention-period`-Option optional die Anzahl der Tage angeben, die Performance Insights-Daten aufbewahrt werden sollen. Sie können *`month`* * 31 (wobei *`month`* es sich um eine Zahl zwischen 1—23 handelt) oder 731 angeben7. Wenn Sie beispielsweise Ihre Leistungsdaten 3 Monate lang aufbewahren möchten, geben Sie 93 an, was 3 * 31 ist. Der Standardwert ist 7 Tage. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.

Das folgende Beispiel aktiviert Performance Insights für `sample-db-cluster` und legt fest, dass Performance-Insights-Daten für 93 Tage (3 Monate) aufbewahrt werden.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-cluster \
 --database-insights-mode standard \
    --db-cluster-identifier sample-db-instance \
    --enable-performance-insights \
    --performance-insights-retention-period 93
```

Windows:

```
aws rds modify-db-cluster ^
 --database-insights-mode standard ^
    --db-cluster-identifier sample-db-instance ^
    --enable-performance-insights ^
    --performance-insights-retention-period 93
```

Wenn Sie eine Aufbewahrungsfrist wie 94 Tage angeben, was kein gültiger Wert ist, gibt RDS einen Fehler aus.

```
An error occurred (InvalidParameterValue) when calling the CreateDBInstance
 operation:
Invalid Performance Insights retention period. Valid values are: [7, 31, 62, 93,
 124, 155, 186, 217,
248, 279, 310, 341, 372, 403, 434, 465, 496, 527, 558, 589, 620, 651, 682, 713, 731]
```

> **Note**
>
> Sie können Performance Insights nur für eine Instance in einem DB-Cluster umschalten, in der Performance Insights nicht auf Clusterebene verwaltet wird.

RDS API

Wenn Sie eine neue DB-Instance mithilfe des Amazon RDS-API-Vorgangs „DBInstanceVorgang erstellen“ erstellen, aktivieren Sie Performance Insights, indem Sie `EnablePerformanceInsights` auf einstellen`True`. Um Performance Insights zu deaktivieren, setzen Sie `EnablePerformanceInsights` auf `False` und stellen Sie `DatabaseInsightsMode` auf ein`standard`.

Sie können den Wert für `EnablePerformanceInsights` auch mittels der folgenden API-Operationen angeben:

- Erstellen DBCluster (Multi-AZ-DB-Cluster)
- Ändern DBCluster (Multi-AZ-DB-Cluster)
- Modifizieren SieDBInstance
- CreateDBInstanceReadReplica
- Aus S3 wiederherstellen DBInstance

Wenn Sie Performance Insights aktivieren, können Sie optional mit dem Parameter `PerformanceInsightsRetentionPeriod` den Zeitraum in Tagen angeben, wie lange Performance-Insights-Daten gespeichert werden sollen. Sie können angeben`7`, *`month`* * 31 (wobei *`month`* es sich um eine Zahl zwischen 1—23 handelt) oder. `731` Wenn Sie beispielsweise Ihre Leistungsdaten 3 Monate lang aufbewahren möchten, geben Sie `93` an, was 3 * 31 ist. Der Standardwert ist 7 Tage. Weitere Informationen zum Aufbewahrungszeitraum finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights.

Überblick über das Leistungsschema für Performance Insights auf , Amazon RDS for MariaDB oder MySQL

Das Leistungsschema ist eine optionale Funktion zur Überwachung der Laufzeitleistung von Amazon RDS for MariaDB oder MySQL auf niedriger Detailebene. Das Leistungsschema ist so konzipiert, dass es minimale Auswirkungen auf die Datenbankleistung hat. Performance Insights ist eine separate Funktion, die Sie mit oder ohne Leistungsschema verwenden können.

Themen

- Überblick über das Leistungsschema-Objekt
- Performance Insights und das Performance-Schema

- Automatische Verwaltung des Leistungsschemas durch Performance Insights
- Auswirkung eines Neustarts auf das Leistungsschema
- Feststellen, ob Performance Insights das Leistungsschema verwaltet
- Schalten Sie das Leistungsschema für , Amazon RDS for MariaDB oder MySQL ein.

Überblick über das Leistungsschema-Objekt

Das Leistungsschema überwacht Ereignisse in MariaDB und MySQL-Datenbanken. Ein Ereignis ist eine Datenbankserver-Aktion, die Zeit in Anspruch nimmt und instrumentiert wurde, damit Timing-Informationen erfasst werden können. Nachfolgend finden Sie Beispiele für Ereignisse:

- Funktionsaufrufe
- Wartet auf das Betriebssystem
- Phasen der SQL-Ausführung
- Gruppen von SQL-Anweisungen

Die `PERFORMANCE_SCHEMA`-Speicher-Engine ist ein Mechanismus zur Implementierung der Leistungsschema-Funktion. Diese Engine sammelt Ereignisdaten mithilfe der Instrumentierung im Quellcode der Datenbank. Die Engine speichert Ereignisse in Arbeitsspeichertabellen in der `performance_schema`-Datenbank. Sie können `performance_schema` genauso abfragen, wie Sie andere Tabellen abfragen können. Weitere Informationen finden Sie unter MySQL-Leistungsschema im MySQL-Referenzhandbuch.

Performance Insights und das Performance-Schema

Performance Insights und das Leistungsschema sind separate Funktionen, die jedoch verbunden sind. Das Verhalten von Performance Insights für Amazon RDS für MariaDB oder MySQL hängt davon ab, ob das Performance-Schema aktiviert ist und wenn ja, ob Performance Insights das Performance-Schema automatisch verwaltet. Die folgende Tabelle beschreibt das Verhalten.

Performance-Schema ist aktiviert	Performance Insights-Verwaltungsmodus	Performance Insights-Verhalten
Ja	Automatisch	•

Performance-Schema ist aktiviert	Performance Insights-Verwaltungsmodus	Performance Insights-Verhalten
		Sammelt detaillierte Überwachungsinformationen auf niedriger Ebene • Sammelt jede Sekunde aktive Sitzungsmetriken • Zeigt die DB-Last kategorisiert nach detaillierten Warteereignissen an, mit denen Sie Engpässe identifizieren können
Ja	Manuell	• Erfasst Warteereignisse und Metriken pro SQL • Sammelt jede Sekunde aktive Sitzungsmetriken • Meldet Benutzerzustände wie Einfügen und Senden, mit denen Sie Engpässe nicht identifizieren können
Nein	N/A	• Erfasst keine Warteereignisse, Metriken pro SQL oder andere detaillierte Überwachungsinformationen auf niedriger Ebene • Sammelt aktive Sitzungsmetriken alle fünf Sekunden statt jede Sekunde • Meldet Benutzerzustände wie Einfügen und Senden, mit denen Sie Engpässe nicht identifizieren können

Automatische Verwaltung des Leistungsschemas durch Performance Insights

Wenn Sie eine Amazon-RDS-for-MariaDB- oder MySQL-DB-Instance erstellen und Performance Insights aktiviert ist, wird das Leistungsschema ebenfalls aktiviert. In diesem Fall verwaltet

Performance Insights Ihre Parameter des Leistungsschemas automatisch. Dies ist die empfohlene Konfiguration.

Wenn Performance Insights das Leistungsschema automatisch verwaltet, `performance_schema` ist die Quelle von`System default`.

> **Note**
>
> Die automatische Verwaltung des Leistungsschemas wird für die Instance-Klasse t4g.medium nicht unterstützt.

Wenn Sie den Parameterwert `performance_schema` manuell ändern und später zur automatischen Verwaltung zurückkehren möchten, finden Sie weitere Informationen unter Schalten Sie das Leistungsschema für , Amazon RDS for MariaDB oder MySQL ein..

> **Important**
>
> Beim Aktivieren des Leistungsschemas durch Performance Insights werden die Parametergruppenwerte nicht geändert. Die Werte werden jedoch für die DB-Instances geändert, die ausgeführt werden. Die geänderten Werte können als einzige Möglichkeit über den Befehl `SHOW GLOBAL VARIABLES` eingesehen werden.

Auswirkung eines Neustarts auf das Leistungsschema

Performance Insights und das Leistungsschema unterscheiden sich in ihren Anforderungen für Neustarts der DB-Instance:

Leistungsschema

Sie müssen die DB-Instance neu starten, um diese Funktion ein- oder auszuschalten.

Performance Insights

Sie müssen die DB-Instance nicht neu starten, um diese Funktion ein- oder auszuschalten.

Wenn das Leistungsschema derzeit nicht aktiviert ist und Sie Performance Insights aktivieren, ohne die DB-Instance neu zu starten, wird das Leistungsschema nicht aktiviert.

Feststellen, ob Performance Insights das Leistungsschema verwaltet

In der folgenden Tabelle finden Sie heraus, ob Performance Insights derzeit das Leistungsschema für alle unterstützten Engine-Hauptversionen verwaltet.

Einstellung des performance_schema-Parameters	Einstellung der Spalte „Source“ (Quelle)	Performance Insights verwaltet das Leistungsschema?
`0`	`System default`	Ja
`0` oder `1`	`Modified`	Nein

Im folgenden Verfahren stellen Sie fest, ob Performance Insights das Leistungsschema automatisch verwaltet.

So stellen Sie fest, ob Performance Insights das Leistungsschema automatisch verwaltet

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Parameter groups (Parametergruppen).
3. Wählen Sie den Namen der Parametergruppe für Ihre DB-Instance.
4. Geben Sie im Suchfeld **`performance_schema`** ein.
5. Prüfen Sie, ob Source die Standardeinstellung des Systems ist und ob der Wert 0 ist. Ist das der Fall, verwaltet Performance Insights das Leistungsschema automatisch.

   In dem hier gezeigten Beispiel verwaltet Performance Insights das Performance-Schema nicht automatisch.



Schalten Sie das Leistungsschema für , Amazon RDS for MariaDB oder MySQL ein.

Angenommen, Performance Insights ist für Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster aktiviert, verwaltet aber derzeit nicht das Leistungsschema. Wenn Sie zulassen möchten, dass Performance Insights das Leistungsschema automatisch verwaltet, führen Sie die folgenden Schritte aus.

So konfigurieren Sie das Leistungsschema für die automatische Verwaltung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Parameter groups (Parametergruppen).
3. Wählen Sie den Namen der Parametergruppe für Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster.
4. Wählen Sie Bearbeiten aus.
5. Geben Sie im Suchfeld **`performance_schema`** ein.
6. Wählen Sie den `performance_schema`-Parameter aus.
7. Wählen Sie Auf Standardwert festlegen.
8. Bestätigen Sie, indem Sie „Werte auf Standard setzen“ wählen.
9. Wählen Sie Save Changes.
10. Starten Sie die DB-Instance oder den Multi-AZ-DB-Cluster neu.

> **Important**
>
> Wenn Sie das Leistungsschema aktivieren oder deaktivieren, müssen Sie die DB-Instance oder den Multi-AZ-DB-Cluster unbedingt neu starten.

Weitere Informationen zum Ändern von Instance-Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS. Weitere Informationen zu den Seiten im Dashboard finden Sie unter Analyse der Metriken mit dem Performance Insights-Dashboard. Weitere Informationen zum MySQL-Leistungsschema finden Sie in der MySQL-Dokumentation unter MySQL Performance Schema (für 8.0) und MySQL Performance Schema (für 8.4).

Konfigurieren von Zugriffsrichtlinien für Performance Insights

Um auf Performance Insights zugreifen zu können, muss ein Principal über die entsprechenden Berechtigungen von AWS Identity and Access Management (IAM) verfügen. Sie können den Zugriff wie folgt erteilen:

- Fügen Sie die `AmazonRDSPerformanceInsightsReadOnly`-verwaltete Richtlinie an einen Berechtigungssatz oder eine Rolle an, um auf alle schreibgeschützten Operationen der Performance-Insights-API zuzugreifen. Fügen Sie die folgenden CloudWatch Berechtigungen hinzu:`GetMetricStatistics`, und. `ListMetrics` `GetMetricData` Weitere Informationen zu CloudWatch Berechtigungen finden Sie in der CloudWatch Amazon-Berechtigungsreferenz.
- Fügen Sie die `AmazonRDSPerformanceInsightsFullAccess`-verwaltete Richtlinie an einen Berechtigungssatz oder eine Rolle an, um auf alle Operationen der Performance-Insights-API zuzugreifen. Hängen Sie die folgenden CloudWatch Berechtigungen an:`GetMetricStatistics`,, und`ListMetrics`. `GetMetricData` Weitere Informationen zu CloudWatch Berechtigungen finden Sie in der CloudWatch Amazon-Berechtigungsreferenz.
- Erstellen Sie eine benutzerdefinierte IAM-Richtlinie und fügen Sie diese an einen Berechtigungssatz oder eine Rolle an.

Wenn Sie bei der Aktivierung von Performance Insights einen vom Kunden verwalteten Schlüssel angegeben haben, stellen Sie sicher, dass die Benutzer in Ihrem Konto über die `kms:GenerateDataKey` Berechtigungen `kms:Decrypt` und für die verfügen AWS KMS key.

In den folgenden Abschnitten fügen Sie einem IAM-Prinzipal eine AWS verwaltete Richtlinie hinzu, erstellen eine benutzerdefinierte IAM-Richtlinie, ändern eine AWS KMS Richtlinie und gewähren detaillierten Zugriff für Performance Insights.

Themen

- Anhängen der AmazonRDSPerformanceInsightsReadOnly Richtlinie an einen IAM-Prinzipal
- Anhängen der AmazonRDSPerformanceInsightsFullAccess Richtlinie an einen IAM-Prinzipal
- Eine benutzerdefinierte IAM Richtlinie für Performance Insights erstellen
- Änderung einer AWS KMS Richtlinie für Performance Insights
- Gewährung eines detaillierten Zugriffs für Performance Insights

Anhängen der AmazonRDSPerformanceInsightsReadOnly Richtlinie an einen IAM-Prinzipal

`AmazonRDSPerformanceInsightsReadOnly`ist eine AWS verwaltete Richtlinie, die Zugriff auf alle schreibgeschützten Operationen der Amazon RDS Performance Insights-API gewährt.

Wenn Sie `AmazonRDSPerformanceInsightsReadOnly` an einen Berechtigungssatz oder eine Rolle anfügen, kann der Empfänger Performance Insights mit anderen Konsolenfunktionen verwenden.

Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsReadOnly.

Anhängen der AmazonRDSPerformanceInsightsFullAccess Richtlinie an einen IAM-Prinzipal

`AmazonRDSPerformanceInsightsFullAccess`ist eine AWS verwaltete Richtlinie, die Zugriff auf alle Operationen der Amazon RDS Performance Insights API gewährt.

Wenn Sie `AmazonRDSPerformanceInsightsFullAccess` an einen Berechtigungssatz oder eine Rolle anfügen, kann der Empfänger Performance Insights mit anderen Konsolenfunktionen verwenden.

Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsFullAccess.

Eine benutzerdefinierte IAM Richtlinie für Performance Insights erstellen

Benutzern, die nicht über die `AmazonRDSPerformanceInsightsFullAccess` Richtlinie `AmazonRDSPerformanceInsightsReadOnly` oder verfügen, können Sie Zugriff auf Performance Insights gewähren, indem Sie eine benutzerverwaltete IAM Richtlinie erstellen oder ändern. Wenn Sie die Richtlinie an einen IAM Berechtigungssatz oder eine Rolle anhängen, kann der Empfänger Performance Insights verwenden.

Erstellen eine benutzerdefinierten Richtlinie

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.

4. Wählen Sie auf der Seite „Richtlinie erstellen“ die JSONOption aus.
5. Kopieren Sie den Text aus dem Abschnitt mit dem JSONRichtliniendokument im Referenzhandbuch für AWS verwaltete Richtlinien und fügen Sie ihn ein AmazonRDSPerformanceInsightsReadOnly oder AmazonRDSPerformanceInsightsFullAccessRichtlinie.
6. Wählen Sie Richtlinie prüfen.
7. Geben Sie einen Namen und optional eine Beschreibung für die Richtlinie an und wählen Sie dann Create policy (Richtlinie erstellen) aus.

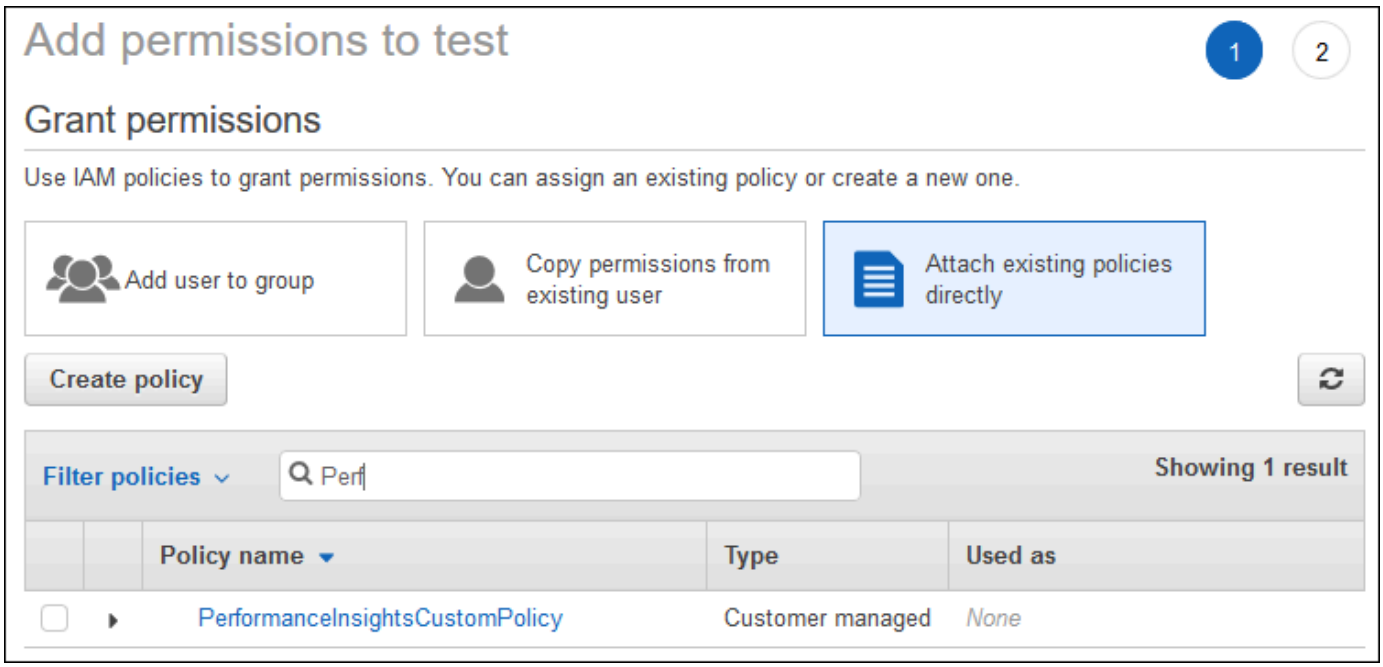
Sie können die Richtlinie nun an einen Berechtigungssatz oder eine Rolle anfügen. Das folgende Verfahren setzt voraus, dass Sie für diesen Zweck bereits einen Benutzer zur Verfügung haben.

So fügen Sie die Richtlinie an einen Benutzer an

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Klicken Sie im Navigationsbereich auf Users (Benutzer).
3. Wählen Sie einen vorhandenen Benutzer aus der Liste aus.

> **Important**
>
> Um Performance Insights verwenden zu können, stellen Sie sicher, dass Sie RDS zusätzlich zur benutzerdefinierten Richtlinie Zugriff auf Amazon haben. Die `AmazonRDSPerformanceInsightsReadOnly` vordefinierte Richtlinie bietet beispielsweise Lesezugriff auf Amazon. RDS Weitere Informationen finden Sie unter Verwalten des Zugriffs mit Richtlinien.

4. Wählen Sie auf der Seite Übersicht die Option Add permissions (Berechtigungen hinzufügen) aus.
5. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen). Geben Sie für die Suche die ersten Zeichen Ihres Richtliniennamens ein, wie in der folgenden Abbildung gezeigt.



6. Wählen Sie Ihre Richtlinie und wählen Sie anschließend Nächster Schritt: Prüfen.
7. Wählen Sie Add permissions (Berechtigungen hinzufügen) aus.

Änderung einer AWS KMS Richtlinie für Performance Insights

Performance Insights verwendet an AWS KMS key , um sensible Daten zu verschlüsseln. Wenn Sie Performance Insights über die API oder die Konsole aktivieren, haben Sie folgende Möglichkeiten:

- Wählen Sie die Standardeinstellung Von AWS verwalteter Schlüssel.

  Amazon RDS verwendet die Von AWS verwalteter Schlüssel für Ihre neue DB-Instance. Amazon RDS erstellt einen Von AWS verwalteter Schlüssel für Ihr AWS-Konto. Ihr AWS-Konto hat für jeden einen anderen Von AWS verwalteter Schlüssel für Amazon RDS AWS-Region.
- Wählen Sie einen kundenverwalteten Schlüssel.

  Wenn Sie einen vom Kunden verwalteten Schlüssel angeben, benötigen Benutzer in Ihrem Konto, die die Performance Insights API aufrufen, die Berechtigungen `kms:Decrypt` und `kms:GenerateDataKey` für den KMS-Schlüssel. Sie können diese Berechtigungen über IAM-Richtlinien konfigurieren. Wir empfehlen jedoch, dass Sie diese Berechtigungen über Ihre KMS-Schlüsselrichtlinie verwalten. Weitere Informationen finden Sie unter Schlüsselrichtlinien in AWS KMS im Entwicklerhandbuch für AWS Key Management Service .

Example

Das folgende Beispiel zeigt, wie Sie Ihrer KMS-Schlüsselrichtlinie Anweisungen hinzufügen können. Diese Anweisungen erlaubt den Zugriff auf Performance Insights. Je nachdem, wie Sie den KMS-Schlüssel verwenden, möchten Sie möglicherweise einige Einschränkungen ändern. Bevor Sie Ihrer Richtlinie Anweisungen hinzufügen, entfernen Sie alle Kommentare.

JSON

```
{
    "Version" : "2012-10-17",
    "Id" : "your-policy",
    "Statement" : [
        {
            "Sid" : "AllowViewingRDSPerformanceInsights",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::444455556666:role/Role1"
                ]
                },
             "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
                ],
            "Resource": "*",
            "Condition" : {
            "StringEquals" : {
                "kms:ViaService" : "rds.us-east-1.amazonaws.com"
                },
            "ForAnyValue:StringEquals": {
                "kms:EncryptionContext:aws:pi:service": "rds",
                "kms:EncryptionContext:service": "pi",
                "kms:EncryptionContext:aws:rds:db-id": "db-
AAAAABBBBBCCCCDDDDDEEEEE"
                }
            }
        }
    ]
}
```

So verwendet Performance Insights vom AWS KMS Kunden verwaltete Schlüssel

Performance Insights verwendet vom Kunden verwaltete Schlüssel, um vertrauliche Daten zu verschlüsseln. Wenn Sie Performance Insights aktivieren, können Sie einen AWS KMS -Schlüssel über die API bereitstellen. Performance Insights erstellt KMS-Berechtigungen für diesen Schlüssel. Das Feature verwendet den Schlüssel und führt die erforderlichen Operationen aus, um vertrauliche Daten zu verarbeiten. Zu den vertraulichen Daten gehören Felder wie Benutzer, Datenbank, Anwendung und SQL-Abfragetext. Performance Insights stellt sicher, dass die Daten sowohl im Ruhezustand als auch während der Übertragung verschlüsselt bleiben.

So arbeitet Performance Insights IAM mit AWS KMS

IAM erteilt Berechtigungen für bestimmte. APIs Performance Insights hat die folgenden öffentlichen Bereiche APIs, die Sie mithilfe von IAM-Richtlinien einschränken können:

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetadata`
- `GetResourceMetrics`
- `ListAvailableResourceDimensions`
- `ListAvailableResourceMetrics`

Sie können die folgenden API-Abfragen verwenden, um vertrauliche Daten abzurufen.

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetrics`

Wenn Sie die API verwenden, um vertrauliche Daten abzurufen, nutzt Performance Insights die Anmeldeinformationen des Aufrufers. Diese Überprüfung stellt sicher, dass der Zugriff auf vertrauliche Daten auf Benutzer beschränkt ist, die Zugriff auf den KMS-Schlüssel haben.

Wenn Sie diese aufrufen APIs, benötigen Sie Berechtigungen zum Aufrufen der API über die IAM-Richtlinie und Berechtigungen zum Aufrufen der `kms:decrypt` Aktion über die AWS KMS Schlüsselrichtlinie.

Die API `GetResourceMetrics` kann sowohl vertrauliche als auch nicht vertrauliche Daten zurückgeben. Die Anforderungsparameter bestimmen, ob die Antwort vertrauliche Daten enthalten soll. Die API gibt vertrauliche Daten zurück, wenn die Anfrage eine vertrauliche Dimension im Filter- oder Group-by-Parameter enthält.

Weitere Informationen zu den Dimensionen, die Sie mit der `GetResourceMetrics` API verwenden können, finden Sie unter. DimensionGroup

Example Beispiele

Im folgenden Beispiel werden die vertraulichen Daten für die Gruppe `db.user` angefordert:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
  "MetricQueries": [
    {
      "Metric": "db.load.avg",
      "GroupBy": {
        "Group": "db.user",
        "Limit": 2
      }
    }
  ],
  "StartTime": 1693872000,
  "EndTime": 1694044800,
  "PeriodInSeconds": 86400
}
```

Example

Im folgenden Beispiel werden die nicht vertraulichen Daten für die Metrik `db.load.avg` angefordert:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "MetricQueries": [
        {
            "Metric": "db.load.avg"
        }
    ],
    "StartTime": 1693872000,
    "EndTime": 1694044800,
    "PeriodInSeconds": 86400
}
```

Gewährung eines detaillierten Zugriffs für Performance Insights

Eine differenzierte Zugriffskontrolle bietet zusätzliche Möglichkeiten, den Zugriff auf Performance Insights zu kontrollieren. Diese Zugriffskontrolle kann den Zugriff auf einzelne Dimensionen für `GetResourceMetrics`, `DescribeDimensionKeys`, und `GetDimensionKeyDetails` Performance Insights Insights-Aktionen zulassen oder verweigern. Um einen differenzierten Zugriff zu verwenden, geben Sie Dimensionen in der IAM-Richtlinie mithilfe von Bedingungsschlüsseln an. Die Bewertung des Zugriffs folgt der Bewertungslogik der IAM-Richtlinie. Weitere Informationen finden Sie unter Logik der Richtlinienauswertung im Handbuch für -IAM-Benutzer. Wenn in der IAM-Richtlinienanweisung keine Dimension angegeben ist, steuert die Anweisung den Zugriff auf alle Dimensionen für die angegebene Aktion. Eine Liste der verfügbaren Dimensionen finden Sie unter DimensionGroup.

Um herauszufinden, auf welche Dimensionen Ihre Anmeldeinformationen zugreifen dürfen, verwenden Sie den `AuthorizedActions` Parameter in `ListAvailableResourceDimensions` und geben Sie die Aktion an. Die zulässigen Werte für `AuthorizedActions` lauten wie folgt:

- `GetResourceMetrics`
- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`

Wenn Sie den `AuthorizedActions` Parameter beispielsweise angeben`GetResourceMetrics`, wird die Liste der Dimensionen `ListAvailableResourceDimensions` zurückgegeben, auf die die `GetResourceMetrics` Aktion zugreifen darf. Wenn Sie im `AuthorizedActions` Parameter mehrere Aktionen angeben, wird ein Schnittpunkt von Dimensionen `ListAvailableResourceDimensions` zurückgegeben, auf die diese Aktionen zugreifen dürfen.

Example

Das folgende Beispiel bietet Zugriff auf die angegebenen Dimensionen `GetResourceMetrics` und `DescribeDimensionKeys` Aktionen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowToDiscoverDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:ListAvailableResourceDimensions"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
        },
        {
            "Sid": "SingleAllow",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
```

```
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "pi:Dimensions": [
                        "db.sql_tokenized.id",
                        "db.sql_tokenized.statement"
                    ]
                }
            }
        }


    ]
}
```

Im Folgenden finden Sie die Antwort für die angeforderte Dimension:

```
 // ListAvailableResourceDimensions API
// Request
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["DescribeDimensionKeys"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
```

```
                //  { "Identifier": "db.sql_tokenized.db_id" }, // not included
 because not allows in the IAM Policy
                  { "Identifier": "db.sql_tokenized.statement" }
              ]
           }

       ] }
   ]
}
```

Im folgenden Beispiel wird für die Dimensionen eine Option „Zulassen“ und zwei „Zugriff verweigern“ angegeben.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
          {
            "Sid": "AllowToDiscoverDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:ListAvailableResourceDimensions"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
          },

          {
            "Sid": "O01AllowAllWithoutSpecifyingDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
        },
```

```
        {
            "Sid": "O01DenyAppDimensionForAll",
            "Effect": "Deny",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "pi:Dimensions": [
                        "db.application.name"
                    ]
                }
            }
        },

        {
            "Sid": "O01DenySQLForGetResourceMetrics",
            "Effect": "Deny",
            "Action": [
                "pi:GetResourceMetrics"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "pi:Dimensions": [
                        "db.sql_tokenized.statement"
                    ]
                }
            }
        }
    ]
}
```

Im Folgenden finden Sie die Antworten für die angeforderten Dimensionen:

```
   // ListAvailableResourceDimensions API
// Request
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["GetResourceMetrics"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.application",
                "Dimensions": [

                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.application.name" }
                ]
            },
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                    { "Identifier": "db.sql_tokenized.db_id" },

                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.sql_tokenized.statement" }
                ]
            },
            ...
        ] }
    ]
}
```

```
// ListAvailableResourceDimensions API
// Request
```

```
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["DescribeDimensionKeys"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.application",
                "Dimensions": [
                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.application.name" }
                ]
            },
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                    { "Identifier": "db.sql_tokenized.db_id" },

                  // allowed for DescribeDimensionKeys because our IAM Policy
                  // denies it only for GetResourceMetrics
                    { "Identifier": "db.sql_tokenized.statement" }
                ]
            },
            ...
        ] }
    ]
}
```

Analyse der Metriken mit dem Performance Insights-Dashboard

> **Important**
>
> AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen

Preise nicht mehr. Die Performance Insights API wird weiterhin ohne Preisänderungen existieren. Die Kosten für die Performance Insights Insights-API werden zusammen mit den Kosten für CloudWatch Database Insights in Ihrer AWS Rechnung ausgewiesen.
Wir empfehlen, dass Sie alle , die die kostenpflichtige Stufe von Performance Insights verwenden, vor dem 30. November 2025 auf den erweiterten Modus von Database Insights aktualisieren. Informationen zum Upgrade auf den erweiterten Modus von Database Insights finden Sie unterEinschalten des erweiterten Modus von Database Insights für Amazon RDS.
Wenn Sie keine Maßnahmen ergreifen, verwenden DB-Instances, die Performance Insights verwenden, standardmäßig den Standardmodus von Database Insights. Im Standardmodus von Database Insights verlieren Sie möglicherweise den Zugriff auf den Leistungsdatenverlauf von mehr als 7 Tagen und können Ausführungspläne und On-Demand-Analysefunktionen in der Amazon RDS-Konsole möglicherweise nicht verwenden. Nach dem 30. November 2025 unterstützt nur der erweiterte Modus von Database Insights Ausführungspläne und On-Demand-Analysen.
Mit CloudWatch Database Insights können Sie die Datenbanklast für Ihre Datenbankflotte überwachen und die Leistung in großem Umfang analysieren und Fehler beheben. Weitere Informationen zu Database Insights finden Sie unterÜberwachung von Amazon RDS mit CloudWatch Database Insights. Preisinformationen finden Sie unter CloudWatch Amazon-Preise.

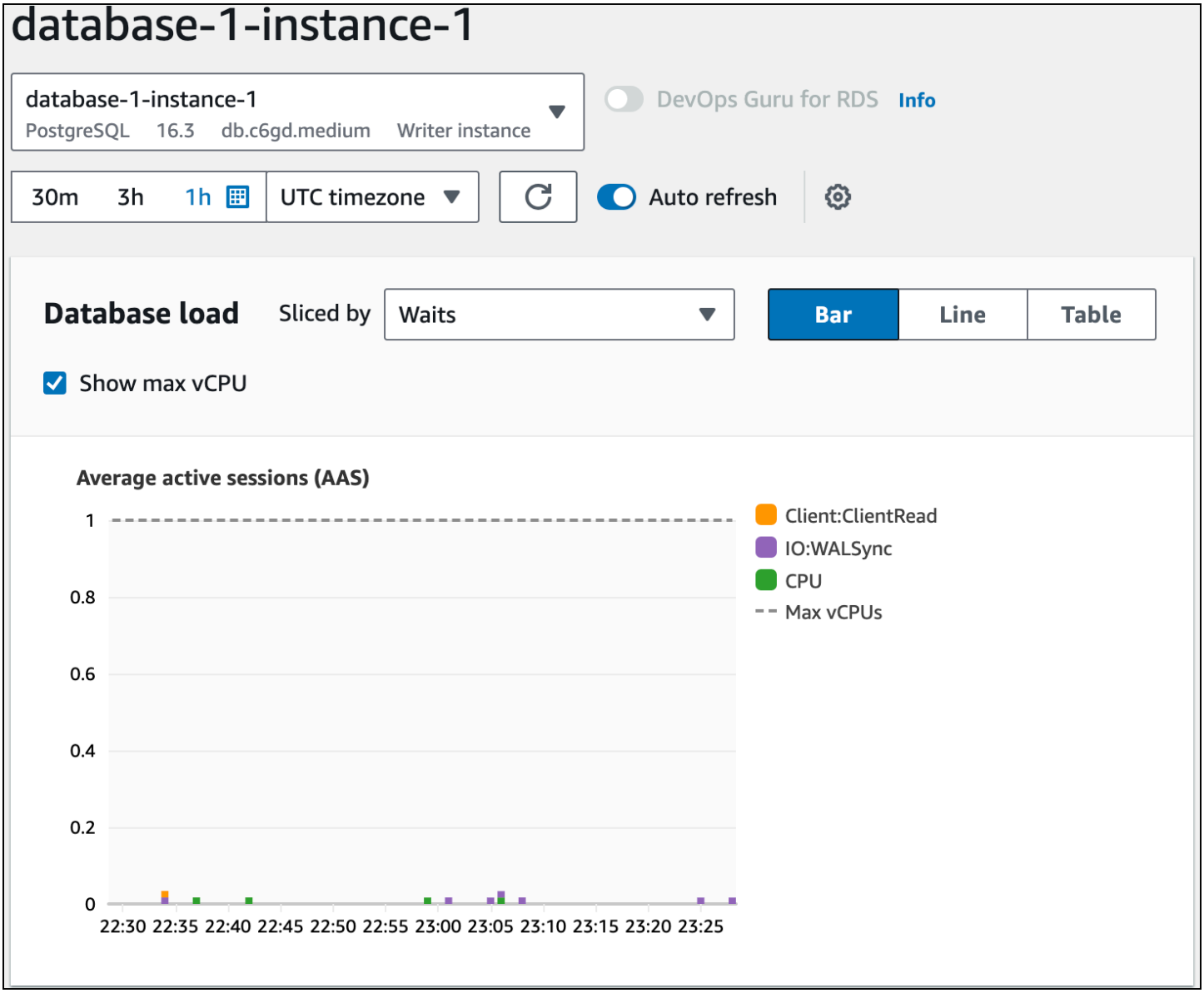
Das Dashboard von Performance Insights enthält Informationen zur Datenbank-Performance, die Sie bei der Analyse und Behebung von Performance-Problemen unterstützen. Auf der Dashboard-Hauptseite finden Sie Informationen zur Datenbanklast. Sie können DB-Lasten nach Dimensionen wie Warteereignissen oder SQL „aufteilen“.

Verwenden der Performance Insights-Dashboard-Komponenten

- Überblick über Performance Insights
- Zugriff auf das Performance-Insights-Dashboard
- Analysieren der DB-Last nach Warteereignissen
- Analysieren der Datenbankleistung für einen bestimmten Zeitraum
- Analysieren von Abfragen mit der SQL Registerkarte „Top“ in Performance Insights
- Analyse der höchsten PDB Oracle-Auslastung
- Analysieren von Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS

Überblick über Performance Insights

Das Dashboard ist die einfachste Möglichkeit, mit Performance Insights zu interagieren. Das folgende Beispiel zeigt das Dashboard für eine SQL Postgre-DB-Instance.

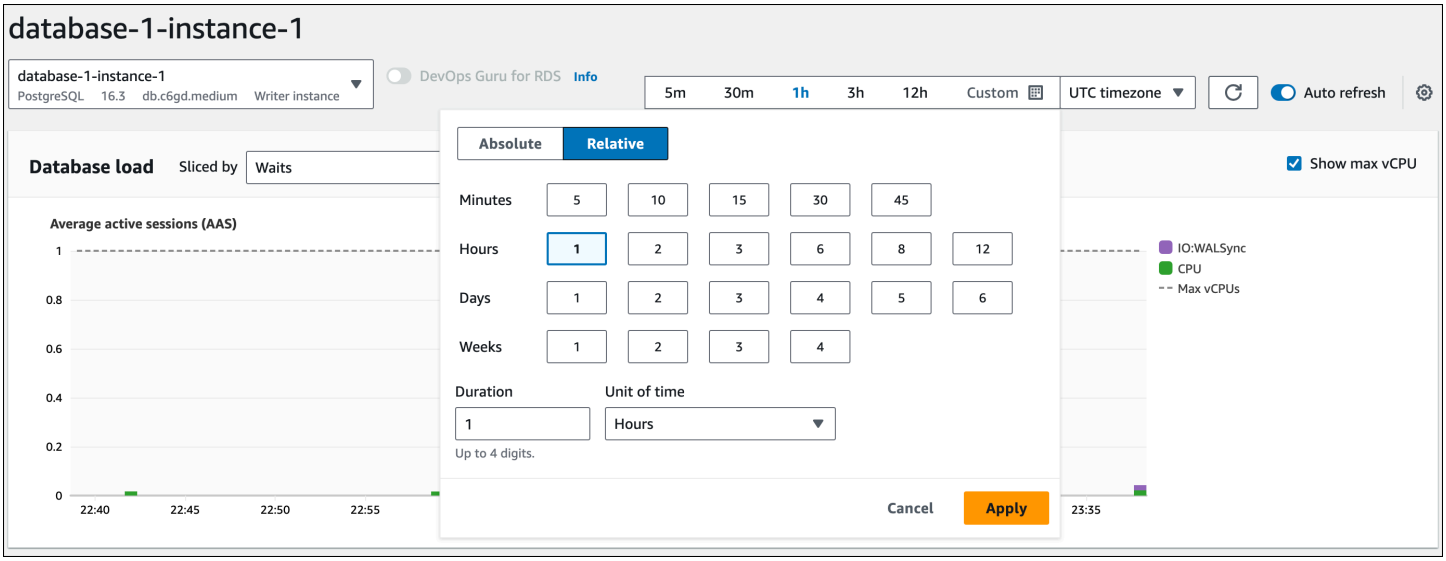


Themen

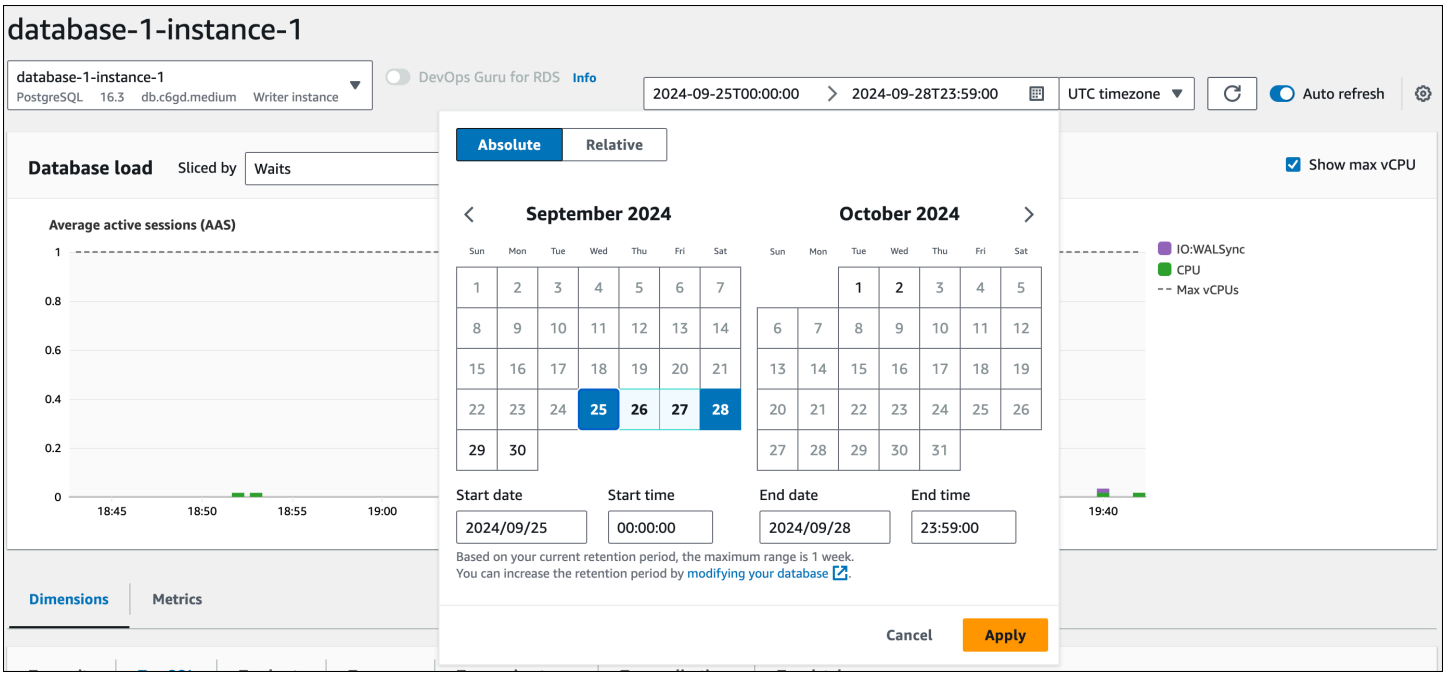
- Zeitraum-Filter
- Zählermetriken-Diagramm
- Datenbank-Ladediagramm
- Dimensionen pro Tabelle

Zeitraum-Filter

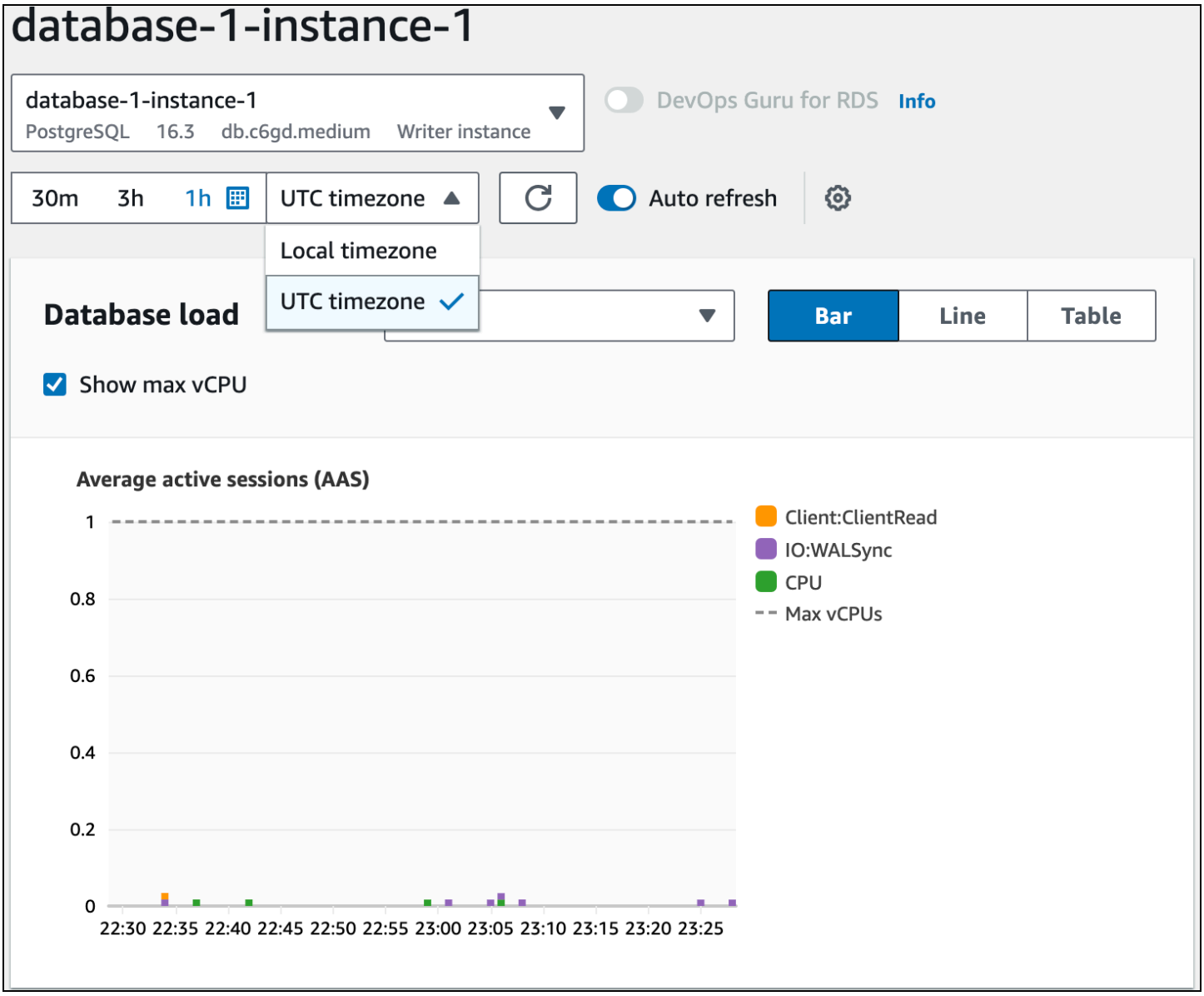
Standardmäßig zeigt das Dashboard von Performance Insights die DB-Last der letzten Stunde an. Sie können diesen Bereich so einstellen, dass er 5 Minuten oder bis zu 2 Jahre lang ist. Sie können auch einen benutzerdefinierten relativen Bereich auswählen.



Sie können einen absoluten Bereich mit einem Anfangs- und Enddatum und einer Uhrzeit auswählen. Das folgende Beispiel zeigt den Zeitraum, der am 25.09.24 um Mitternacht beginnt und am 28.09.24 um 23:59 Uhr endet.



Standardmäßig ist die Zeitzone für das Performance Insights Insights-Dashboard Coordinated Universal Time (UTC). Sie können auch die lokale Zeitzone wählen.

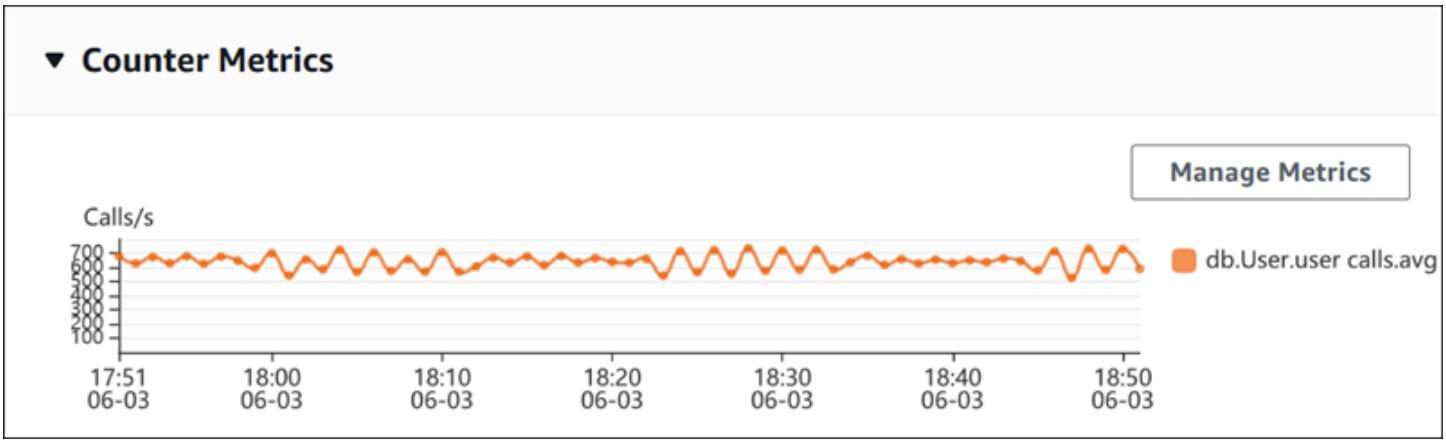


Zählermetriken-Diagramm

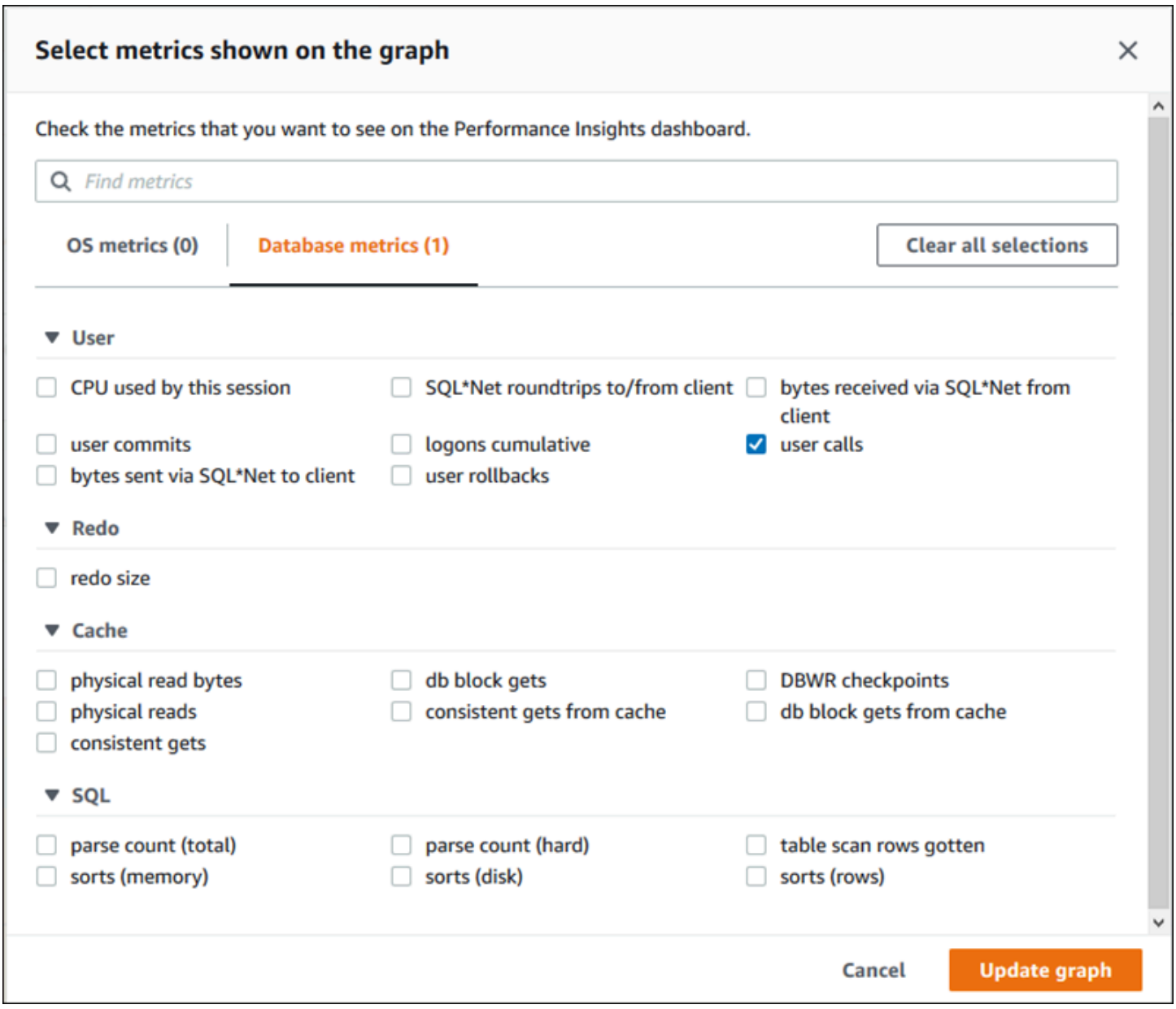
Mithilfe von Zählermetriken können Sie das Performance Insights-Dashboard anpassen und bis zu 10 weitere Diagramme aufnehmen. Diese Diagramme enthalten eine Auswahl von Dutzenden von Betriebssystem- und Datenbank-Performance-Metriken. Diese Informationen können mit der Datenbanklast korreliert werden, um Performance-Probleme zu identifizieren und zu analysieren.

Das Counter Metrics (Zählermetriken)-Diagramm enthält Daten zu Leistungsindikatoren. Die Standardmetriken hängen von der DB-Engine ab.

- My SQL und MariaDB — `db.SQL.Innodb_rows_read.avg`
- Oracle – `db.User.user calls.avg`
- Microsoft SQL Server — `db.Databases.Active Transactions(_Total).avg`
- Postfach — SQL `db.Transactions.xact_commit.avg`



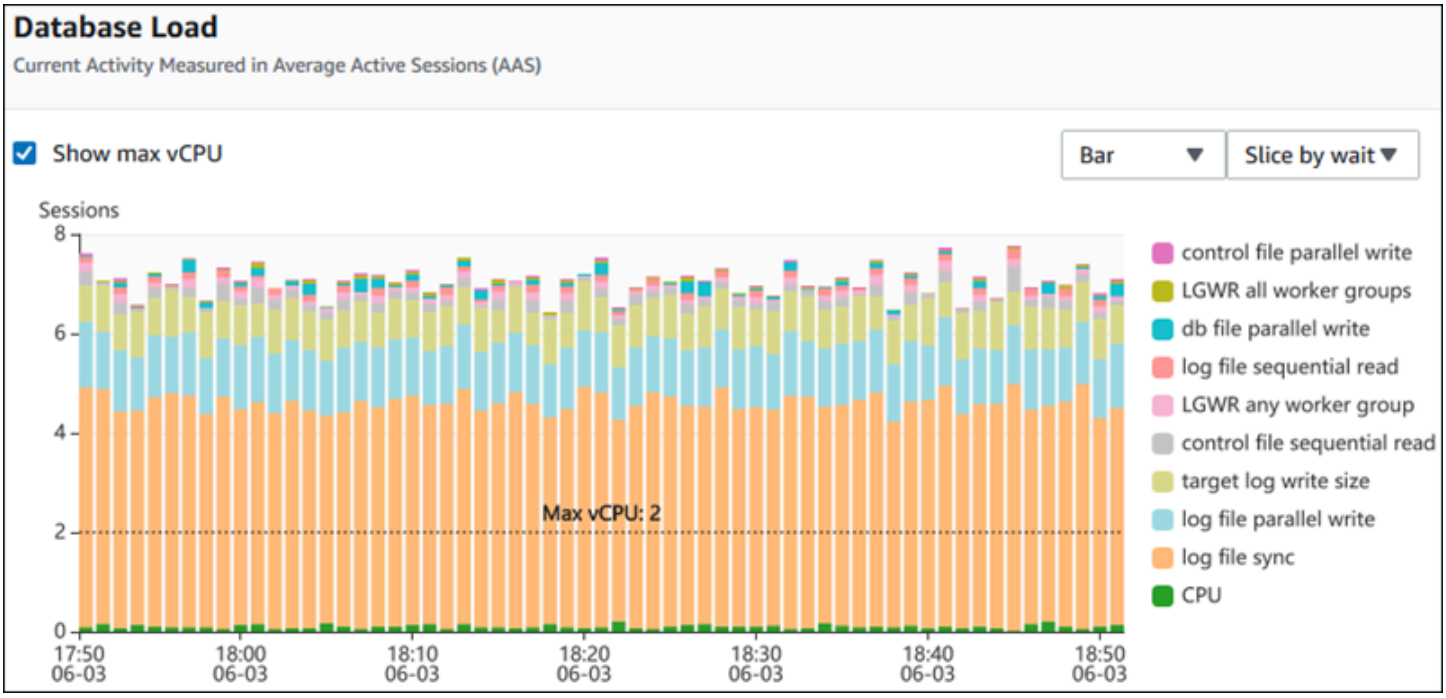
Ändern Sie die Leistungsindikatoren, indem Sie Metriken verwalten wählen. Sie können mehrere Betriebssystem-Metriken oder Datenbank-Metriken, auswählen, wie im folgenden Screenshot veranschaulicht. Um Details für jede Metrik anzuzeigen, bewegen Sie den Mauszeiger über den Metriknamen.



Beschreibungen der Zählermetriken, die Sie für jede DB-Engine hinzufügen können, finden Sie unter Performance-Insights-Zählermetriken.

Datenbank-Ladediagramm

Das Diagramm zur Datenbankauslastung zeigt, wie die Datenbankaktivität im Vergleich zur Kapazität der DB-Instance abschneidet, wie sie in der CPU Zeile Max v dargestellt wird. Standardmäßig stellt das gestapelte Liniendiagramm die DB-Last als durchschnittliche aktive Sitzungen pro Zeiteinheit dar. Die DB-Last wird nach Wartestatus aufgeteilt (gruppiert).



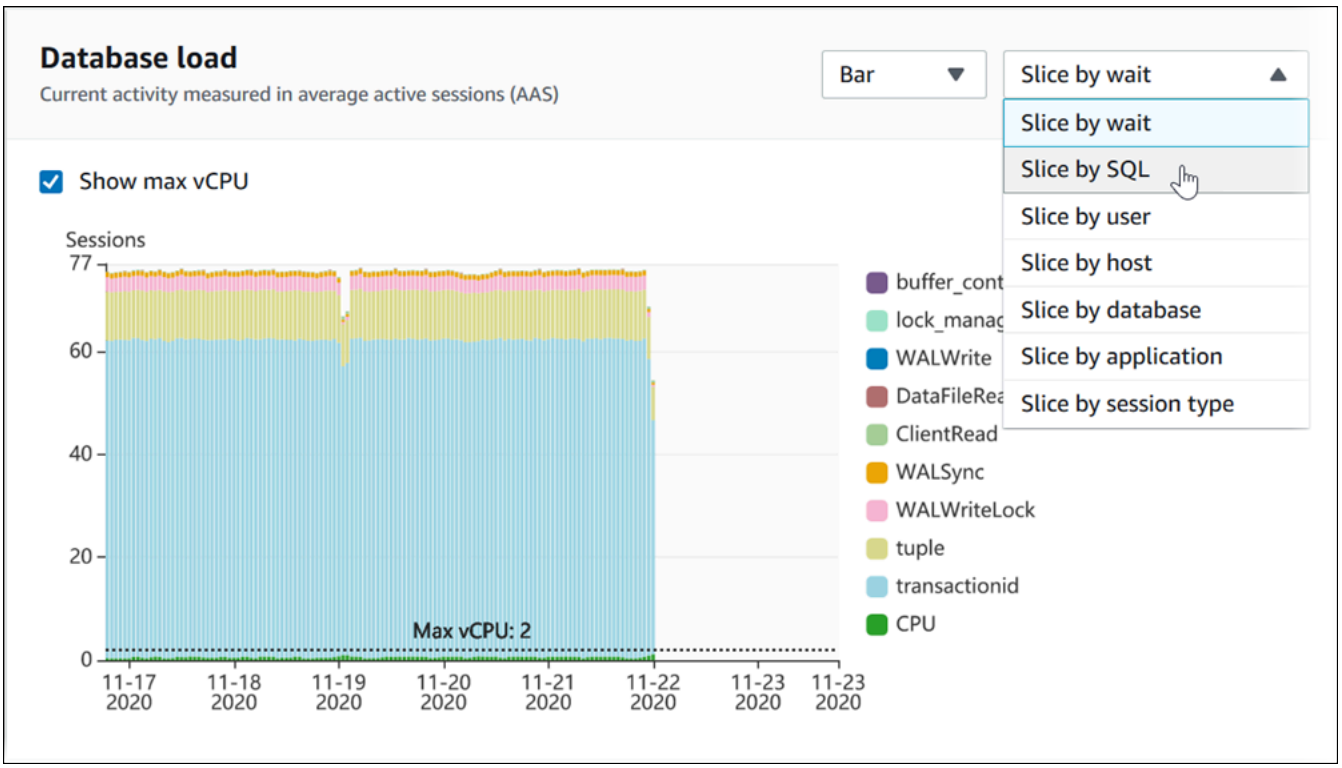
DB-Last aufgeteilt nach Dimensionen

Sie können die Last als aktive Sitzungen anzeigen, die nach unterstützten Dimensionen gruppiert sind. Die folgende Tabelle zeigt, welche Dimensionen für die verschiedenen Engines unterstützt werden.

Dimension	Oracle	SQL-Server	Postgret SQL	Mein SQL
Host	Ja	Ja	Ja	Ja
SQL	Ja	Ja	Ja	Ja
Benutzer	Ja	Ja	Ja	Ja
Waits (Warteereignis)	Ja	Ja	Ja	Ja
Plans (Pläne)	Ja	Nein	Nein	Nein
Anwendung	Nein	Nein	Ja	Nein
Datenbank	Nein	Nein	Ja	Ja

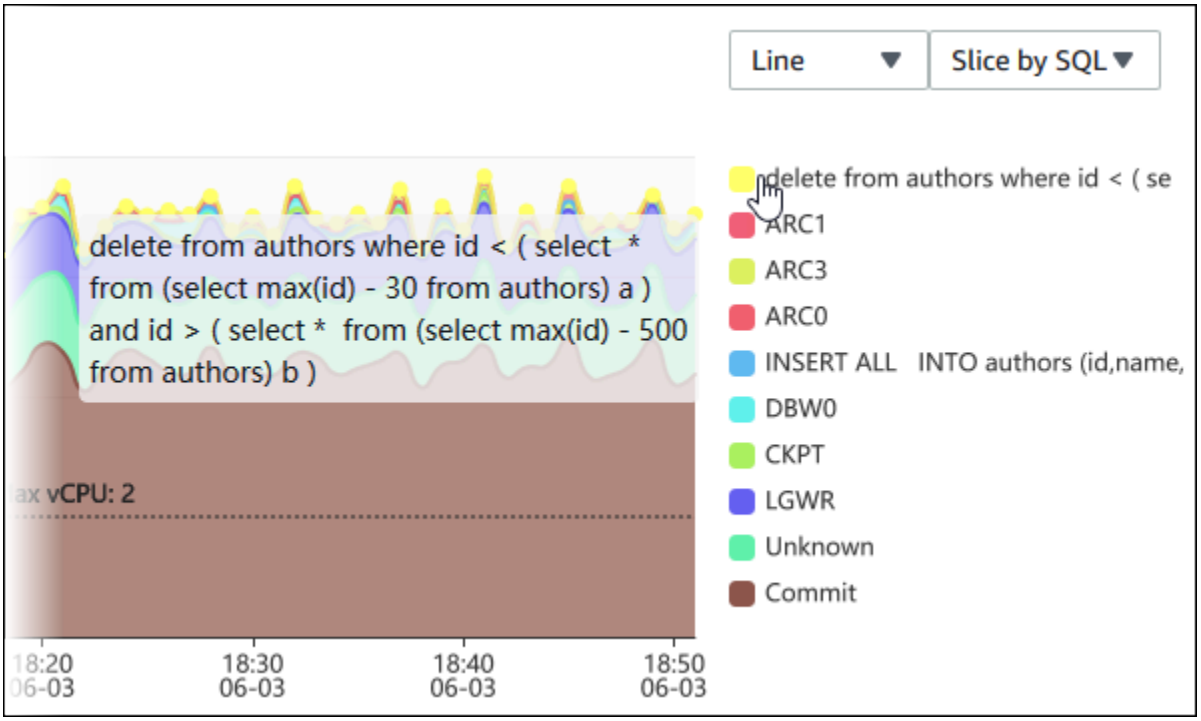
Dimension	Oracle	SQL-Server	Postgret SQL	Mein SQL
Session type (Sitzungstyp)	Nein	Nein	Ja	Nein

Die folgende Abbildung zeigt die Abmessungen für eine SQL Postgre-DB-Instance.

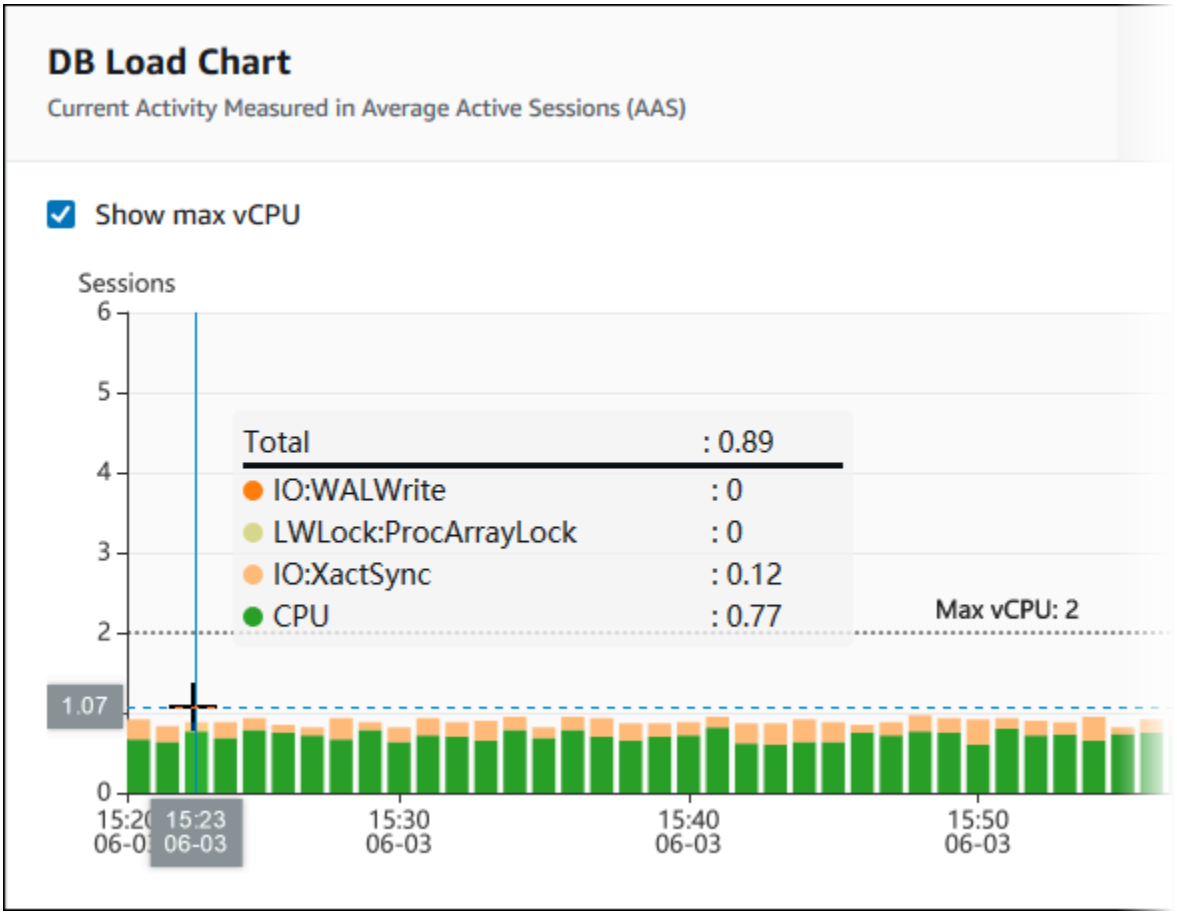


DB-Ladedetails für ein Dimensionselement

Um Details zu einem DB-Lastelement innerhalb einer Dimension anzuzeigen, bewegen Sie den Mauszeiger über den Elementnamen. Die folgende Abbildung zeigt Details für eine SQL Anweisung.

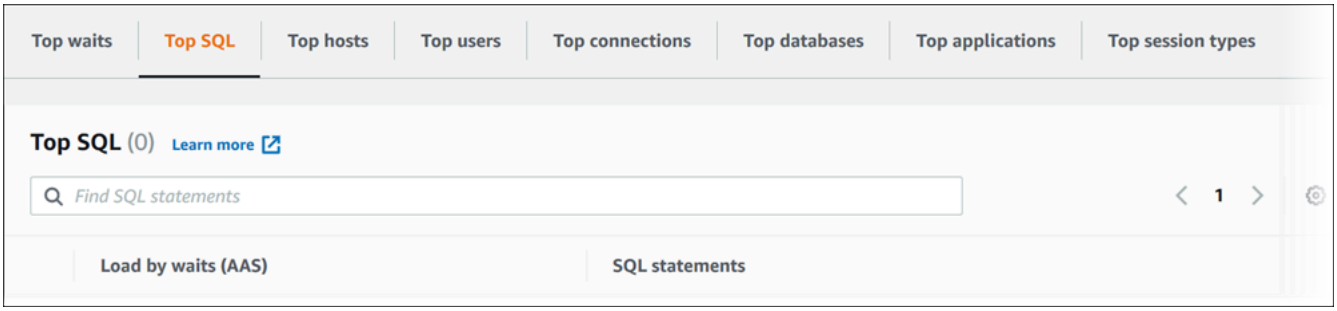


Um Details zu einem Element für den ausgewählten Zeitraum in der Legende anzuzeigen, bewegen Sie den Mauszeiger über dieses Element.



Dimensionen pro Tabelle

Die Tabelle mit den oberen Abmessungen schneidet die DB-Ladung um verschiedene Dimensionen auf. Eine Dimension ist eine Kategorie oder „Aufteilung“ für verschiedene Merkmale der DB-Last. Wenn die Dimension istSQL, SQL zeigt Top die SQL Anweisungen, die am meisten zur Datenbanklast beitragen.



Wählen Sie eine der folgenden Dimensionsregisterkarten.

Tab	Beschreibung	Unterstützte Engines
Oben SQL	Die SQL Anweisungen, die derzeit ausgeführt werden	Alle
Top waits (Top-Warteereignis)	Das Ereignis, auf das das Datenbank-Backend wartet	Alle
Top hosts (Top-Hosts)	Der Hostname des verbunden en Clients	Alle
Top users (Top-Benutzer)	Der bei der Datenbank angemeldete Benutzer	Alle
Top databases (Top-Date nbanken)	Der Name der Datenbank, mit der der Client verbunden ist	Nur PostgreSQL, MySQL, MariaDB und Server SQL
Top applications (Top-Anwe ndungen)	Der Name der Anwendung , die mit der Datenbank verbunden ist	Postgre SQL und SQL Server
Top session types (Top-Sitz ungstypen)	Der Typ der aktuellen Sitzung	Postgre SQL

Informationen zum Analysieren von Abfragen mithilfe der SQL Registerkarte „Oben“ finden Sie unterÜberblick über den SQL Tab „Top“.

Zugriff auf das Performance-Insights-Dashboard

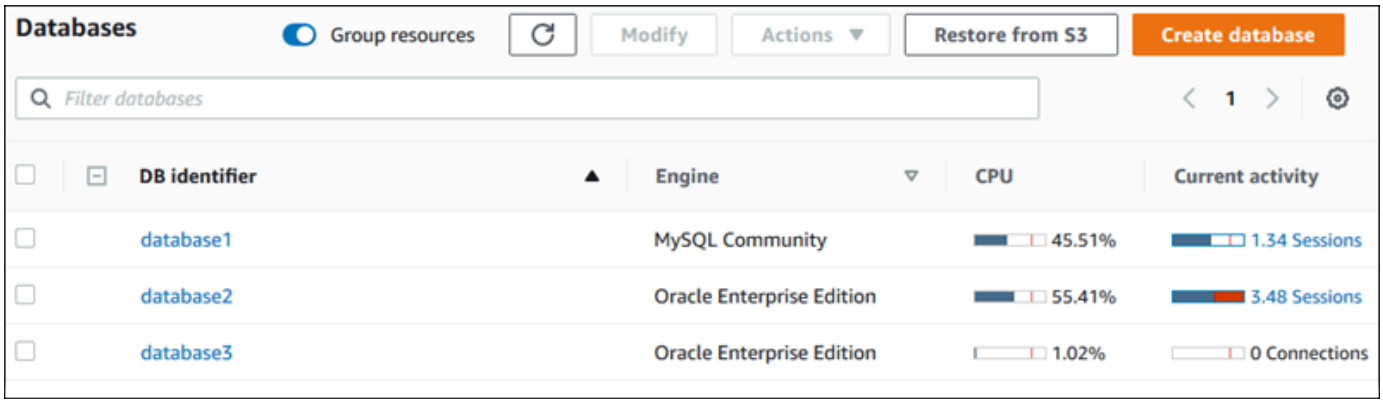
Amazon RDS bietet im Performance Insights-Dashboard eine konsolidierte Ansicht von Performance Insights und CloudWatch Metriken.

Gehen Sie wie folgt vor, um auf das Performance-Insights-Dashboard zuzugreifen.

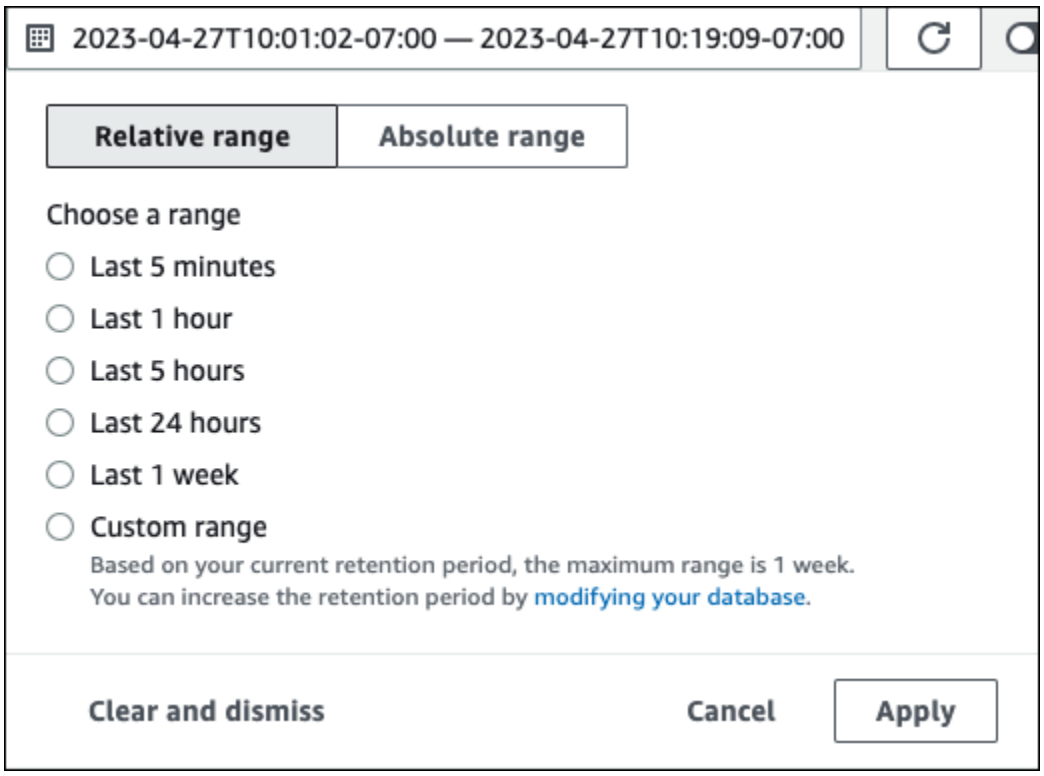
So zeigen Sie das Performance Insights Insights-Dashboard in der AWS Management Console an

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.

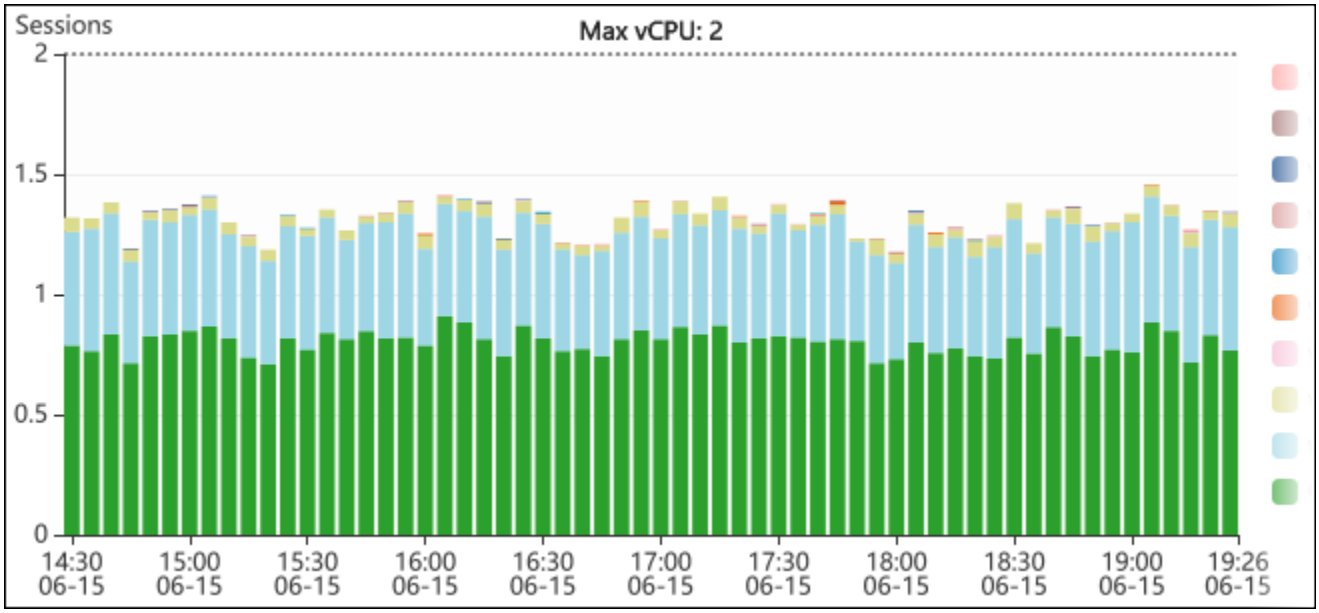
Für DB-Instances mit aktiviertem Performance Insights können Sie auch auf das Performance Insights Insights-Dashboard zugreifen, indem Sie in der Liste der DB-Instances den Eintrag Sessions auswählen. Unter Aktuelle Aktivität zeigt das Element Sitzungen die Datenbanklast von durchschnittlichen, aktiven Sitzungen der letzten fünf Minuten an. Der Balken zeigt die Last grafisch an. Ist der Balken leer, wird die DB-Instance nicht verwendet. Wenn die Last ansteigt, wird der Balken blau ausgefüllt. Wenn die Last die Anzahl von virtual CPUs (vCPUs) in der DB-Instance-Klasse überschreitet, färbt sich der Balken rot, was auf einen potenziellen Engpass hinweist.



4. (Optional) Wählen Sie das Datum oder den Zeitraum oben rechts aus und geben Sie ein anderes relatives oder absolutes Zeitintervall an. Sie können jetzt einen Zeitraum angeben und einen Bericht zur Datenbankleistungsanalyse erstellen. Der Bericht enthält die identifizierten Einblicke und Empfehlungen. Weitere Informationen finden Sie unter Erstellen eines Leistungsanalyseberichts in Performance Insights.

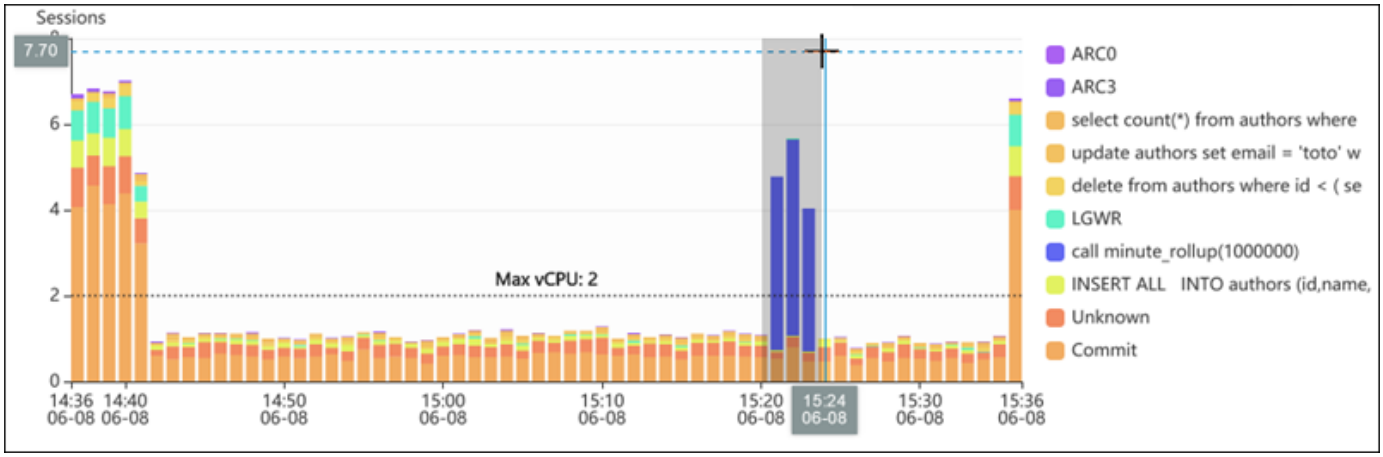


Im folgenden Screenshot beträgt das DB-Last-Intervall 5 Stunden.

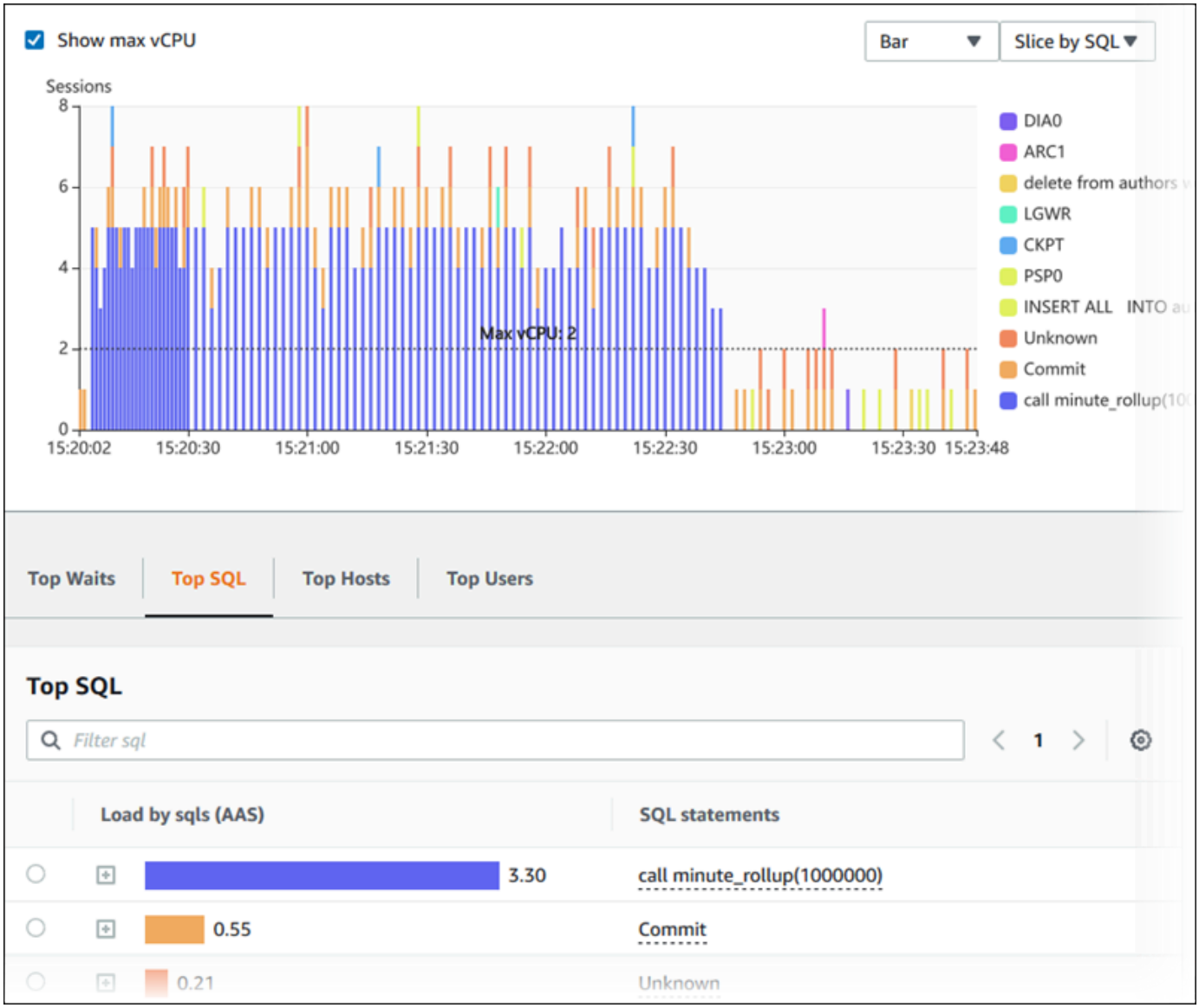


5. (Optional) Um einen Teil des DB-Lastdiagramms zu vergrößern, wählen Sie die Startzeit und ziehen sie mit der Maus an das Ende des gewünschten Zeitraums.

   Der ausgewählte Bereich wird im DB-Lastdiagramm hervorgehoben.



Wenn Sie die Maustaste loslassen, vergrößert das DB-Lastdiagramm die ausgewählte AWS Region, und die Tabelle mit den wichtigsten Abmessungen wird neu berechnet.



6. (Optional) Um Ihre Daten automatisch zu aktualisieren, aktivieren Sie Automatische Aktualisierung.



Das Performance-Insights-Dashboard wird automatisch mit neuen Daten aktualisiert. Die Aktualisierungsrate hängt von der Menge der angezeigten Daten ab:

- 5 Minuten wird alle 10 Sekunden aktualisiert.
- 1 Stunde wird alle 5 Minuten aktualisiert.
- 5 Stunden wird alle 5 Minuten aktualisiert.
- 24 Stunden wird alle 30 Minuten aktualisiert.

- 1 Woche wird jeden Tag aktualisiert.
- 1 Monat wird jeden Tag aktualisiert.

Analysieren der DB-Last nach Warteereignissen

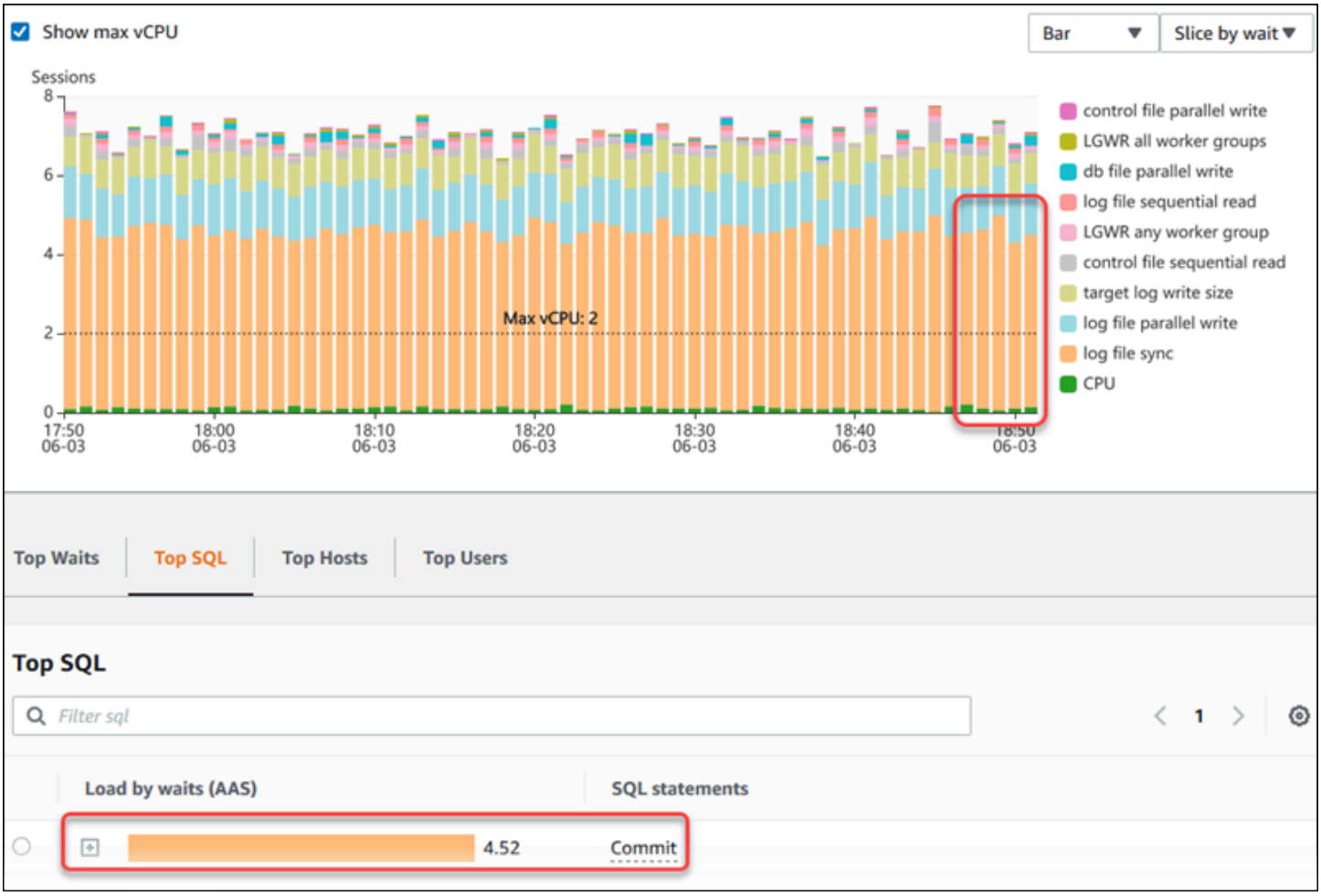
Wenn das Diagramm der durchschnittlich aktiven Sitzungen einen Engpass anzeigt, können Sie herausfinden, woher die Last kommt. Betrachten Sie dazu die Tabelle mit den Hauptlastelementen unterhalb des Datenbanklast-Diagramms Wählen Sie ein bestimmtes Element aus, z. B. eine SQL Abfrage oder einen Benutzer, um dieses Element genauer zu untersuchen und Details zu diesem Element anzuzeigen.

DB-Load, gruppiert nach Waits und SQL Top-Abfragen, ist die Standardansicht des Performance Insights Insights-Dashboards. Diese Kombination bietet typischerweise den besten Einblick in Performance-Probleme. DB-Last gruppiert nach Wartezeiten zeigt an, ob Ressourcen- oder Parallelitätsengpässe in der Datenbank vorhanden sind. In diesem Fall wird SQLauf der Registerkarte der Tabelle mit den am häufigsten geladenen Elementen angezeigt, welche Abfragen für diese Auslastung verantwortlich sind.

Ihr typischer Workflow für die Diagnose von Performance-Problemen ist folgendermaßen:

1. Sehen Sie sich das Diagramm zur Datenbanklast an und prüfen Sie, ob es Vorfälle gibt, bei denen die Datenbanklast die CPUHöchstgrenze überschreitet.
2. Wenn ja, schauen Sie sich das Diagramm der durchschnittlich aktiven Sitzungen an und identifizieren Sie, welcher Wartezustand oder welche Zustände primär dafür verantwortlich sind.
3. Identifizieren Sie die Digest-Abfragen, die die Last verursacht haben, indem Sie herausfinden, welche der Abfragen auf der SQLRegisterkarte in der Tabelle mit den wichtigsten Ladeelementen am meisten zu diesen Wartezuständen beitragen. Sie finden sie in der Spalte DB Load by Wait (DB-Last nach Wartezuständen).
4. Wählen Sie auf der SQLRegisterkarte eine dieser Digest-Abfragen aus, um sie zu erweitern und die untergeordneten Abfragen zu sehen, aus denen sie besteht.

Beispielsweise wird im folgenden Dashboard die Protokolldateisynchronisierung für den größten Teil der DB-Last berücksichtigt. Die Wartezeit LGWRaller Arbeitergruppen ist ebenfalls hoch. Das obere SQL Diagramm zeigt, was die Ursache für die Wartezeiten bei der Protokolldateisynchronisierung ist: häufige `COMMIT` Anweisungen. In diesem Fall wird durch eine weniger häufige Übergabe mit Commit die DB-Last reduziert.



Analysieren der Datenbankleistung für einen bestimmten Zeitraum

Analysieren Sie die Datenbankleistung mit On-Demand-Analysen, indem Sie einen Leistungsanalysebericht für einen bestimmten Zeitraum erstellen. Sehen Sie sich Leistungsanalyseberichte an, um Leistungsprobleme wie Ressourcenengpässe oder Änderungen an einer Abfrage in Ihrer DB-Instance zu finden. Im Performance-Insights-Dashboard können Sie einen Zeitraum auswählen und einen Leistungsanalysebericht erstellen. Sie können dem Bericht auch ein oder mehrere Tags hinzufügen.

Um diese Funktion nutzen zu können, müssen Sie die Aufbewahrungsfrist für die kostenpflichtige Stufe verwenden. Weitere Informationen finden Sie unter Preisgestaltung und Datenspeicherung für Performance Insights

Der Bericht kann auf der Registerkarte Leistungsanalyseberichte – neu ausgewählt und angezeigt werden. Der Bericht enthält die Erkenntnisse, zugehörigen Metriken und Empfehlungen zur

Lösung des Leistungsproblems. Der Bericht kann für die Dauer des Aufbewahrungszeitraums von Performance Insights eingesehen werden.

Der Bericht wird gelöscht, wenn die Startzeit des Berichtsanalysezeitraums außerhalb des Aufbewahrungszeitraums liegt. Sie können den Bericht auch vor Ablauf der Aufbewahrungsfrist löschen.

Sie müssen Performance Insights aktivieren, um die Leistungsprobleme zu erkennen und den Analysebericht für Ihre DB-Instance zu erstellen. Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten.

Informationen zur Unterstützung dieser Funktion nach Region, DB-Engine und Instance-Klasse finden Sie unter DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen.

In den folgenden Abschnitten können Sie einen Leistungsanalysebericht erstellen, anzeigen, hinzufügen und löschen.

Themen

- Erstellen eines Leistungsanalyseberichts in Performance Insights
- Einen Leistungsanalysebericht in Performance Insights anzeigen
- Hinzufügen von Tags zu einem Leistungsanalysebericht in Performance Insights
- Löschen eines Leistungsanalyseberichts in Performance Insights

Erstellen eines Leistungsanalyseberichts in Performance Insights

Im Performance-Insights-Dashboard können Sie einen Leistungsanalysebericht für einen bestimmten Zeitraum erstellen. Sie können einen Zeitraum auswählen und dem Analysebericht ein oder mehrere Tags hinzufügen.

Der Analysezeitraum kann zwischen 5 Minuten und 6 Tagen liegen. Vor dem Start der Analyse müssen mindestens 24 Stunden an Leistungsdaten vorliegen.

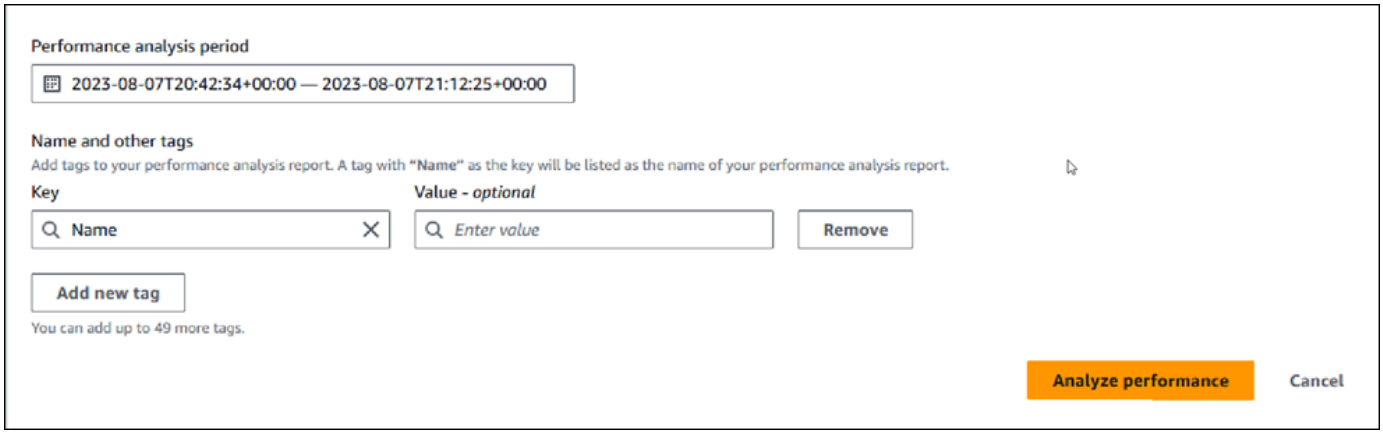
Informationen zur Unterstützung dieser Funktion nach Region, DB-Engine und Instance-Klasse finden Sie unter DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen.

So erstellen Sie einen Leistungsanalysebericht für einen bestimmten Zeitraum

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.
4. Wählen Sie im Performance Insights-Dashboard den Abschnitt Leistung beim Laden der Datenbank analysieren aus.

   Die Felder zum Festlegen des Zeitraums und zum Hinzufügen eines oder mehrerer Tags zum Leistungsanalysebericht werden angezeigt.

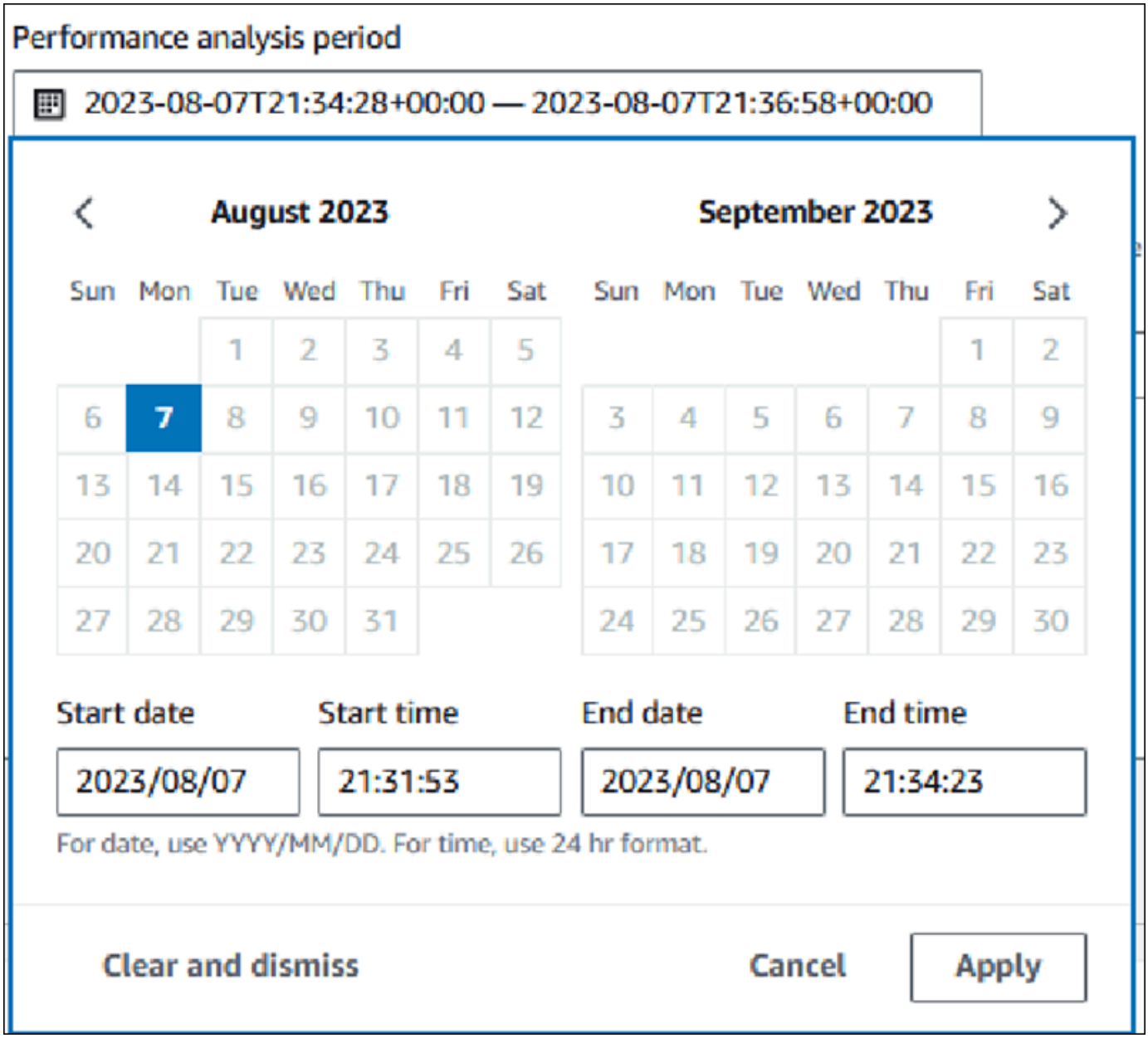


5. Wählen Sie den Zeitraum aus. Wenn Sie oben rechts einen Zeitraum im Abschnitt Relativer Bereich oder Absoluter Bereich festlegen, können Sie nur Datum und Uhrzeit des Analyseberichts innerhalb dieses Zeitraums eingeben oder auswählen. Wenn Sie den Analysezeitraum außerhalb dieses Zeitraums auswählen, wird eine Fehlermeldung angezeigt.

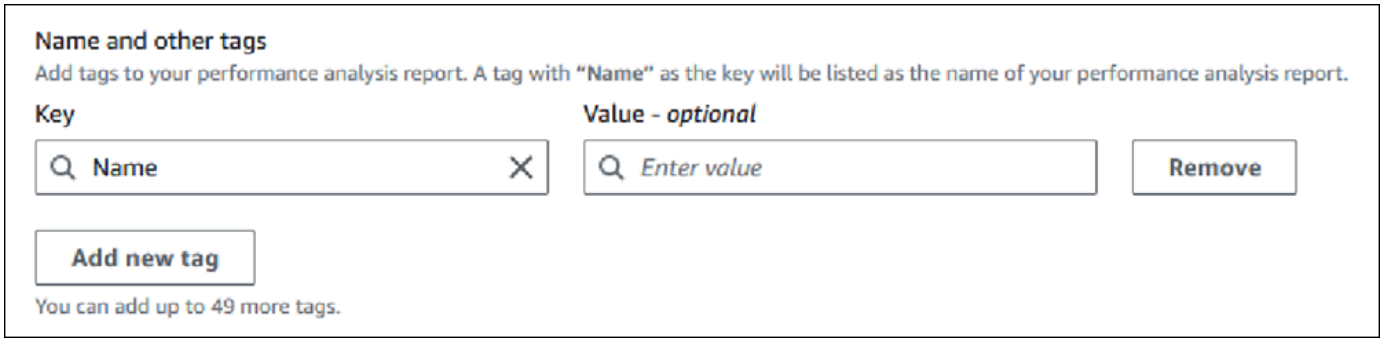
   Führen Sie einen der folgenden Schritte aus, um den Zeitraum festzulegen:

   - Drücken und ziehen Sie einen der Schieberegler im DB-Lastdiagramm.

     Das Feld Zeitraum der Leistungsanalyse zeigt den ausgewählten Zeitraum an und das DB-Lastdiagramm hebt den ausgewählten Zeitraum hervor.
   - Wählen Sie Startdatum, Startzeit, Enddatum und Endzeit im Feld Zeitraum der Leistungsanalyse aus.



6. (Optional) Geben Sie Schlüssel und Wertoptional ein, um ein Tag für den Bericht hinzuzufügen.



7. Wählen Sie Leistung analysieren aus.

   In einem Banner wird eine Meldung angezeigt, die angibt, ob die Berichtserstellung erfolgreich war oder fehlgeschlagen ist. Die Nachricht enthält auch den Link zum Anzeigen des Berichts.

   Das folgende Beispiel zeigt das Banner mit der Meldung, dass der Bericht erfolgreich erstellt wurde.



   Der Bericht kann auf der Registerkarte Leistungsanalyseberichte – neu angezeigt werden.

Mit der AWS CLI können Sie einen Leistungsanalysebericht erstellen. Ein Beispiel zur Erstellung eines Berichts mithilfe von AWS CLI finden Sie unterErstellen eines Leistungsanalyseberichts für einen bestimmten Zeitraum.

Einen Leistungsanalysebericht in Performance Insights anzeigen

Die Registerkarte Leistungsanalysebericht – neu listet alle Berichte auf, die für die DB-Instance erstellt wurden. Für jeden Bericht wird Folgendes angezeigt:

- ID: eindeutige Kennung des Berichts.
- Name: der Tag-Schlüssel, der dem Bericht hinzugefügt wurde.
- Erstellungszeit des Berichts: Uhrzeit, zu der Sie den Bericht erstellt haben.
- Startzeit der Analyse: Startzeit der Analyse im Bericht.
- Endzeit der Analyse: Endzeit der Analyse im Bericht.

So zeigen Sie einen Leistungsanalysebericht an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus, für die Sie den Analysebericht anzeigen möchten.
4. Scrollen Sie nach unten und wählen Sie Leistungsanalyseberichte — neuer Tab im Performance Insights Insights-Dashboard.

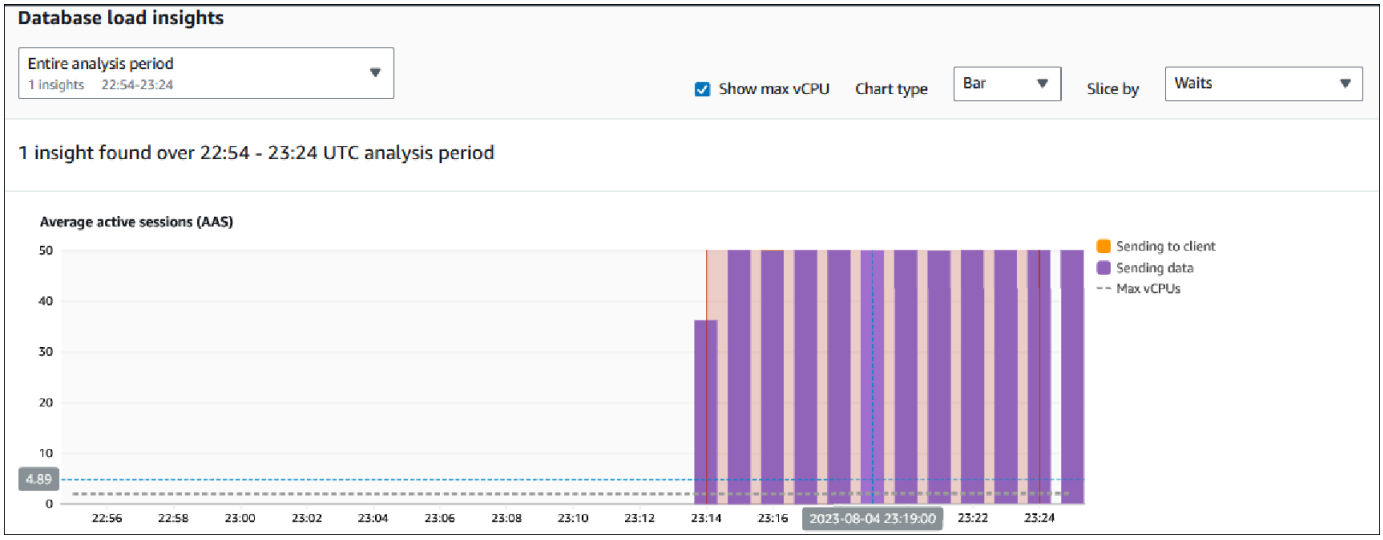
   Es werden alle Analyseberichte für die verschiedenen Zeiträume angezeigt.

5. Wählen Sie die ID des Berichts aus, den Sie ansehen möchten.

   Das DB-Lastdiagramm zeigt standardmäßig den gesamten Analysezeitraum an, wenn mehr als ein Einblick identifiziert wurde. Wenn der Bericht einen Einblick identifiziert hat, zeigt das DB-Lastdiagramm den Einblick standardmäßig an.

   Das Dashboard listet außerdem die Tags für den Bericht im Abschnitt Tags auf.

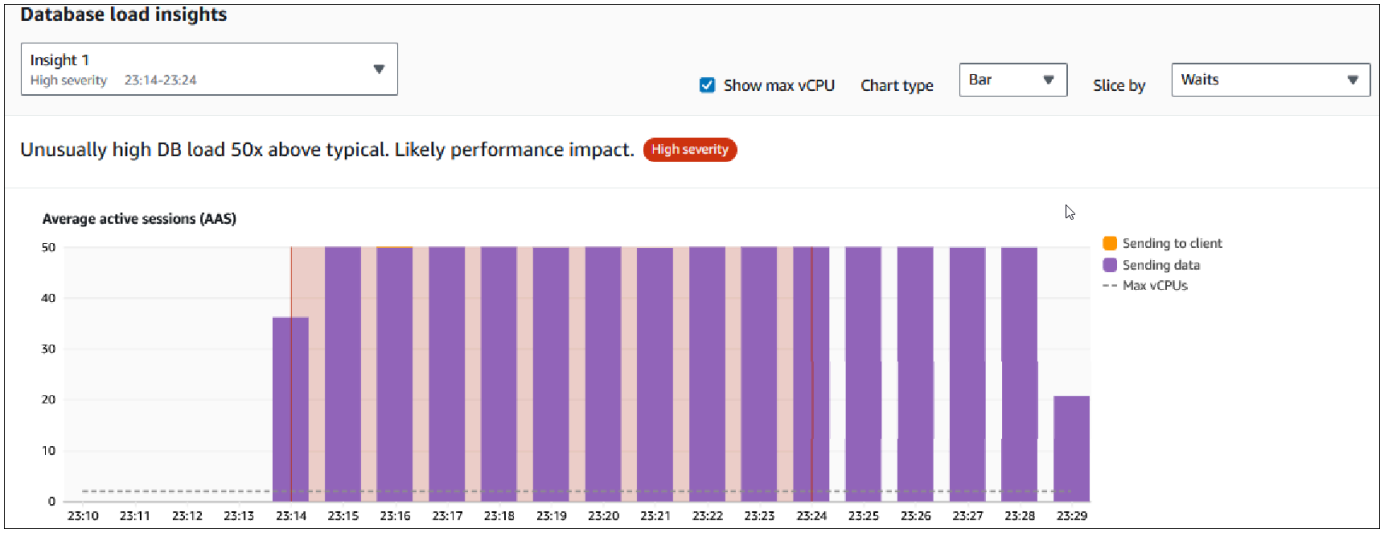
   Das folgende Beispiel zeigt den gesamten Analysezeitraum für den Bericht.

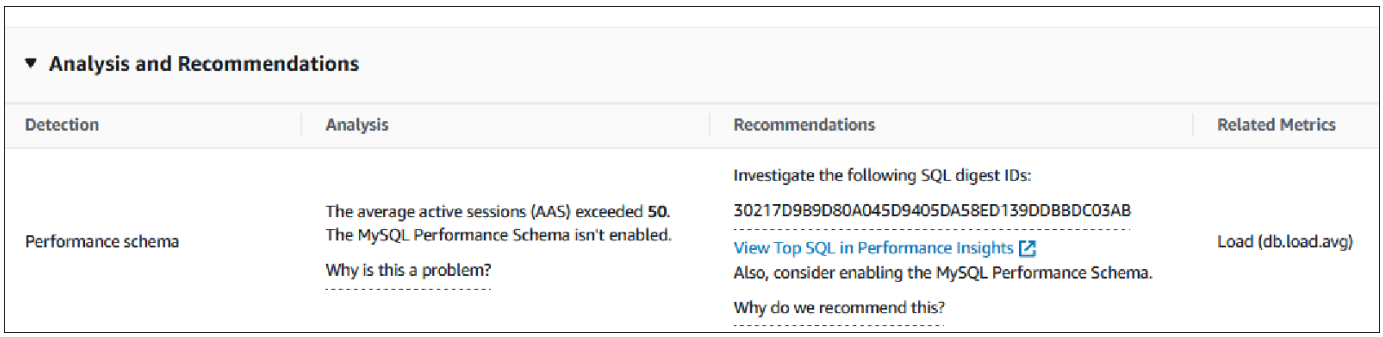


6. Wählen Sie die Einsicht in der Liste Einblicke in die Datenbanklast aus, die Sie anzeigen möchten, wenn im Bericht mehr als ein Einblick identifiziert wird.

   Das Dashboard zeigt die Einblickmeldung, wobei im DB-Lastdiagramm der Zeitraum des Einblicks, die Analyse und Empfehlungen hervorgehoben werden sowie die Liste der Berichts-Tags enthalten ist.

   Das folgende Beispiel zeigt den DB-Lasteinblick im Bericht.





Hinzufügen von Tags zu einem Leistungsanalysebericht in Performance Insights

Sie können ein Tag hinzufügen, wenn Sie einen Bericht erstellen oder ansehen. Sie können bis zu 50 Tags für einen Bericht hinzufügen.

Sie benötigen Berechtigungen, um Tags hinzuzufügen. Weitere Informationen zu Zugriffsrichtlinien für Performance Insights finden Sie unter Konfigurieren von Zugriffsrichtlinien für Performance Insights.

Informationen zum Hinzufügen eines oder mehrerer Tags bei der Erstellung eines Berichts finden Sie in Schritt 6 des Verfahrens Erstellen eines Leistungsanalyseberichts in Performance Insights.

So fügen Sie beim Anzeigen eines Berichts ein oder mehrere Tags hinzu

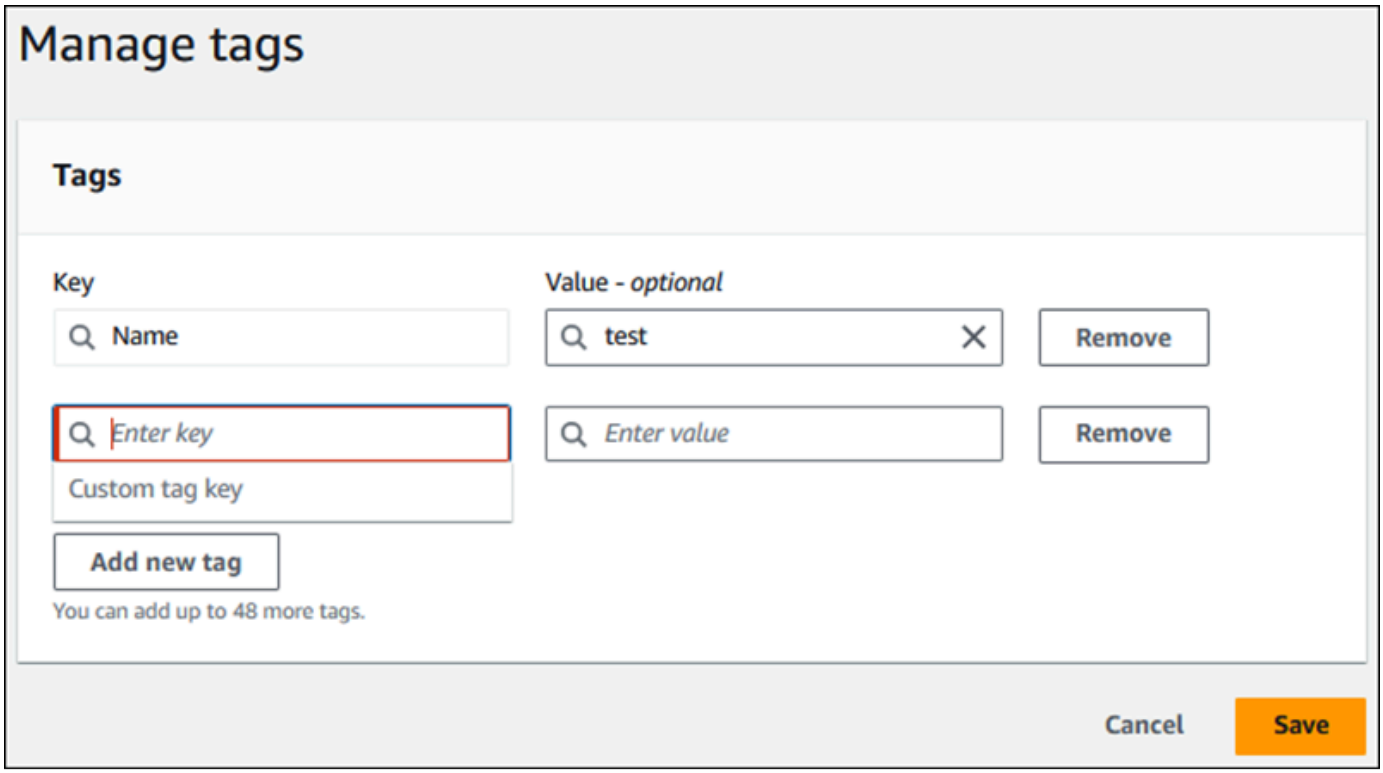
1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.

Das Performance-Insights-Dashboard wird für die DB-Instance angezeigt.

4. Scrollen Sie nach unten und wählen Sie die Registerkarte Leistungsanalyseberichte – neu aus.
5. Wählen Sie den Bericht aus, für den Sie die Tags hinzufügen möchten.

   Das Dashboard zeigt den Bericht an.
6. Scrollen Sie nach unten zu Tags und wählen Sie Tags verwalten aus.
7. Wählen Sie Neues Tag hinzufügen aus.
8. Geben Sie Schlüssel und Wert – optional ein und wählen Sie Neues Tag hinzufügen aus.

   Das folgende Beispiel bietet die Option, ein neues Tag für den ausgewählten Bericht hinzuzufügen.



   Ein neues Tag wird für den Bericht erstellt.

   Im Dashboard wird die Liste der Tags für den Bericht im Abschnitt Tags aufgelistet. Wenn Sie ein Tag aus dem Bericht entfernen möchten, wählen Sie Entfernen neben dem Tag aus.

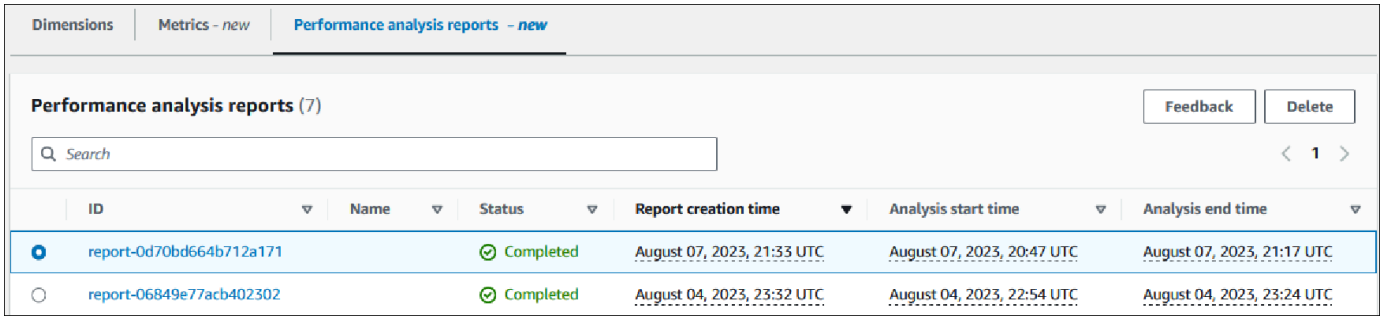
Löschen eines Leistungsanalyseberichts in Performance Insights

Sie können einen Bericht aus der Liste der Berichte, die auf der Registerkarte Leistungsanalyseberichte angezeigt werden, oder beim Anzeigen eines Berichts löschen.

So löschen Sie einen Bericht

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine DB-Instance aus.

   Das Performance-Insights-Dashboard wird für die DB-Instance angezeigt.
4. Scrollen Sie nach unten und wählen Sie die Registerkarte Leistungsanalyseberichte – neu aus.
5. Wählen Sie den Bericht aus, den Sie löschen möchten, und klicken Sie oben rechts auf Löschen.



   Ein Bestätigungsfenster wird angezeigt. Der Bericht wird gelöscht, nachdem Sie „Bestätigen“ ausgewählt haben.
6. (Optional) Wählen Sie die ID des Berichts, den Sie löschen möchten.

   Wählen Sie oben rechts auf der Seite Löschen aus.

   Ein Bestätigungsfenster wird angezeigt. Der Bericht wird gelöscht, nachdem Sie „Bestätigen“ ausgewählt haben.

Analysieren von Abfragen mit der SQL Registerkarte „Top“ in Performance Insights

Im Amazon RDS Performance Insights Insights-Dashboard finden Sie Informationen zu laufenden und aktuellen Abfragen auf der SQL Registerkarte Top in der Tabelle Top-Dimensionen. Sie können diese Informationen verwenden, um Ihre Abfragen zu optimieren.

Themen

- Überblick über den SQL Tab „Top“
- Zugriff auf mehr SQL-Text im Performance-Insights-Dashboard
- SQLStatistiken im Performance Insights Insights-Dashboard anzeigen

Überblick über den SQL Tab „Top“

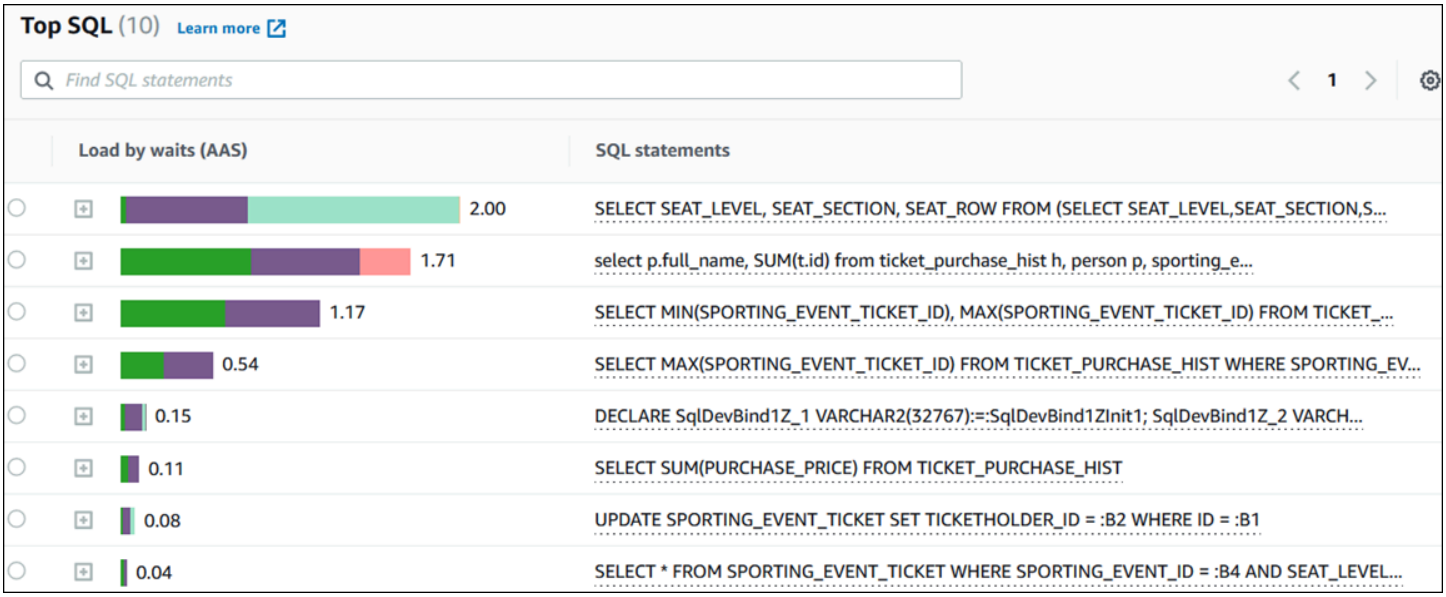
Standardmäßig werden auf der SQL Registerkarte „Top“ die 25 Abfragen angezeigt, die am meisten zur DB-Auslastung beitragen. Um Ihre Abfragen zu optimieren, können Sie Informationen wie den Abfragetext und SQL Statistiken analysieren. Sie können auch die Statistiken auswählen, die auf der SQL Registerkarte „Oben“ angezeigt werden sollen.

Themen

- SQLText
- SQL-Statistiken
- Nach Wartezeiten laden () AAS
- Anzeigen der Informationen SQL
- Wählen Sie Statistikeinstellungen

SQLText

Standardmäßig enthält jede Zeile in der SQL Tabelle Top 500 Byte Text für jede Anweisung.

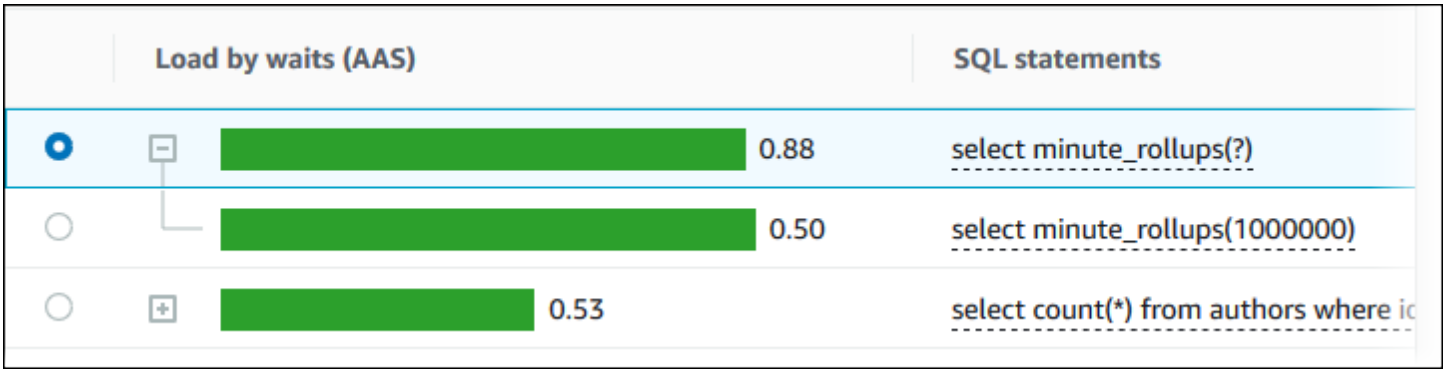


Informationen dazu, wie Sie mehr als die standardmäßigen 500 Byte SQL Text anzeigen können, finden Sie unterZugriff auf mehr SQL-Text im Performance-Insights-Dashboard.

Ein SQLDigest besteht aus mehreren tatsächlichen Abfragen, die sich strukturell ähneln, aber unterschiedliche Literalwerte haben können. Der Digest ersetzt fest codierte Werte durch ein Fragezeichen. Zum Beispiel könnte `SELECT * FROM emp WHERE lname= ?` ein Digest sein. Dieser Digest kann die folgenden untergeordneten Abfragen enthalten:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Um die wörtlichen SQL Aussagen in einem Digest anzuzeigen, wählen Sie die Abfrage und dann das Pluszeichen (+) aus. Im folgenden Beispiel ist die ausgewählte Abfrage ein Digest.



> **Note**
>
> In einer SQL Zusammenfassung werden ähnliche SQL Aussagen gruppiert, sensible Informationen werden jedoch nicht geschwärzt.

Performance Insights kann SQL Oracle-Text als Unbekannt anzeigen. Der Text hat diesen Status in folgenden Situationen:

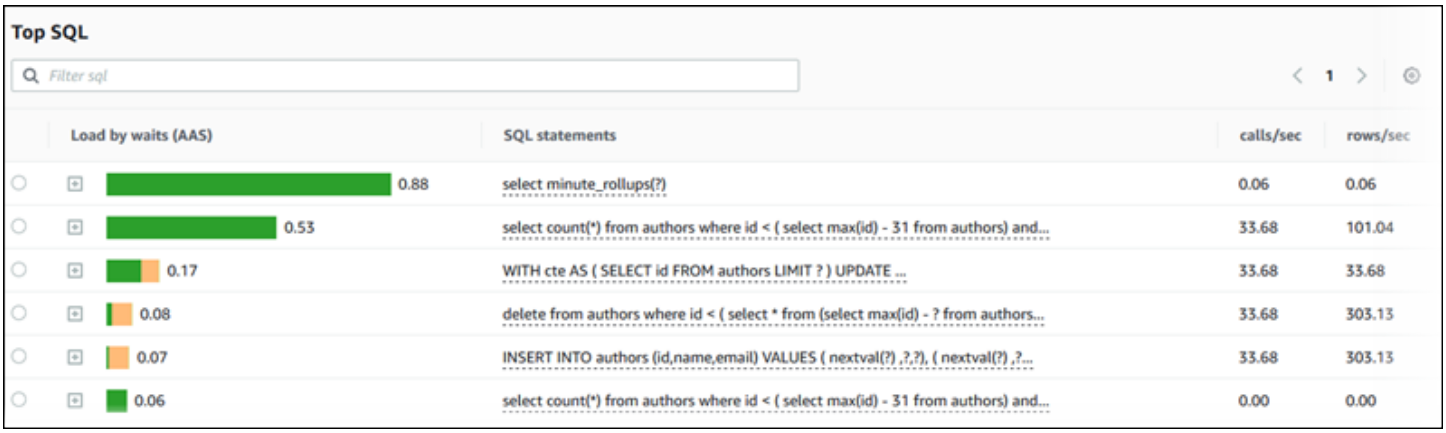
- Ein anderer Oracle-Datenbankbenutzer als SYS ist aktiv, wird aber derzeit nicht ausgeführtSQL. Wenn beispielsweise eine parallel Abfrage abgeschlossen wird, wartet der Abfragekoordinator darauf, dass Hilfsprozesse ihre Sitzungsstatistiken senden. Für die Dauer der Wartezeit wird der Abfragetext als Unknown (Unbekannt) angezeigt.

- RDSFür eine Oracle-Instance auf Standard Edition 2 begrenzt Oracle Resource Manager die Anzahl der parallel Threads. Der entsprechende Hintergrundprozess bewirkt, dass der Abfragetext als Unknown (Unbekannt) angezeigt wird.

SQL-Statistiken

SQLStatistiken sind leistungsbezogene Metriken zu SQL Abfragen. Performance Insights könnte beispielsweise Ausführungen pro Sekunde oder pro Sekunde verarbeitete Zeilen anzeigen. Performance Insights erfasst Statistiken nur für die häufigsten Abfragen. In der Regel entsprechen diese den Top-Abfragen nach Last, die im Performance Insights-Dashboard angezeigt werden.

Jede Zeile in der SQL Tabelle Top enthält relevante Statistiken für die SQL Anweisung oder den Digest, wie im folgenden Beispiel gezeigt.



Performance Insights kann Berichte `0.00` und `-` (unbekannt) für SQL Statistiken erstellen. Diese Situation tritt unter den folgenden Bedingungen auf:

- Es ist nur eine Stichprobe vorhanden. Performance Insights berechnet beispielsweise die Änderungsraten für SQLPostgre-Abfragen auf der Grundlage mehrerer Stichproben aus der `pg_stat_statements` Ansicht. Wenn eine Workload für kurze Zeit ausgeführt wird, erfasst Performance Insights möglicherweise nur eine Stichprobe, was bedeutet, dass keine Änderungsrate berechnet werden kann. Der unbekannte Wert wird durch einen Bindestrich (`-`) dargestellt.
- Zwei Stichproben haben die gleichen Werte. Performance Insights kann keine Änderungsrate berechnen, da keine Änderung stattgefunden hat, weshalb die Rate als `0.00` gemeldet wird.
- Einer SQLPostgre-Anweisung fehlt ein gültiger Bezeichner. Postgre SQL erstellt erst nach dem Parsen und Analysieren einen Bezeichner für eine Anweisung. Somit kann eine Anweisung in den SQL internen In-Memory-Strukturen von Postgre ohne Bezeichner existieren. Da Performance

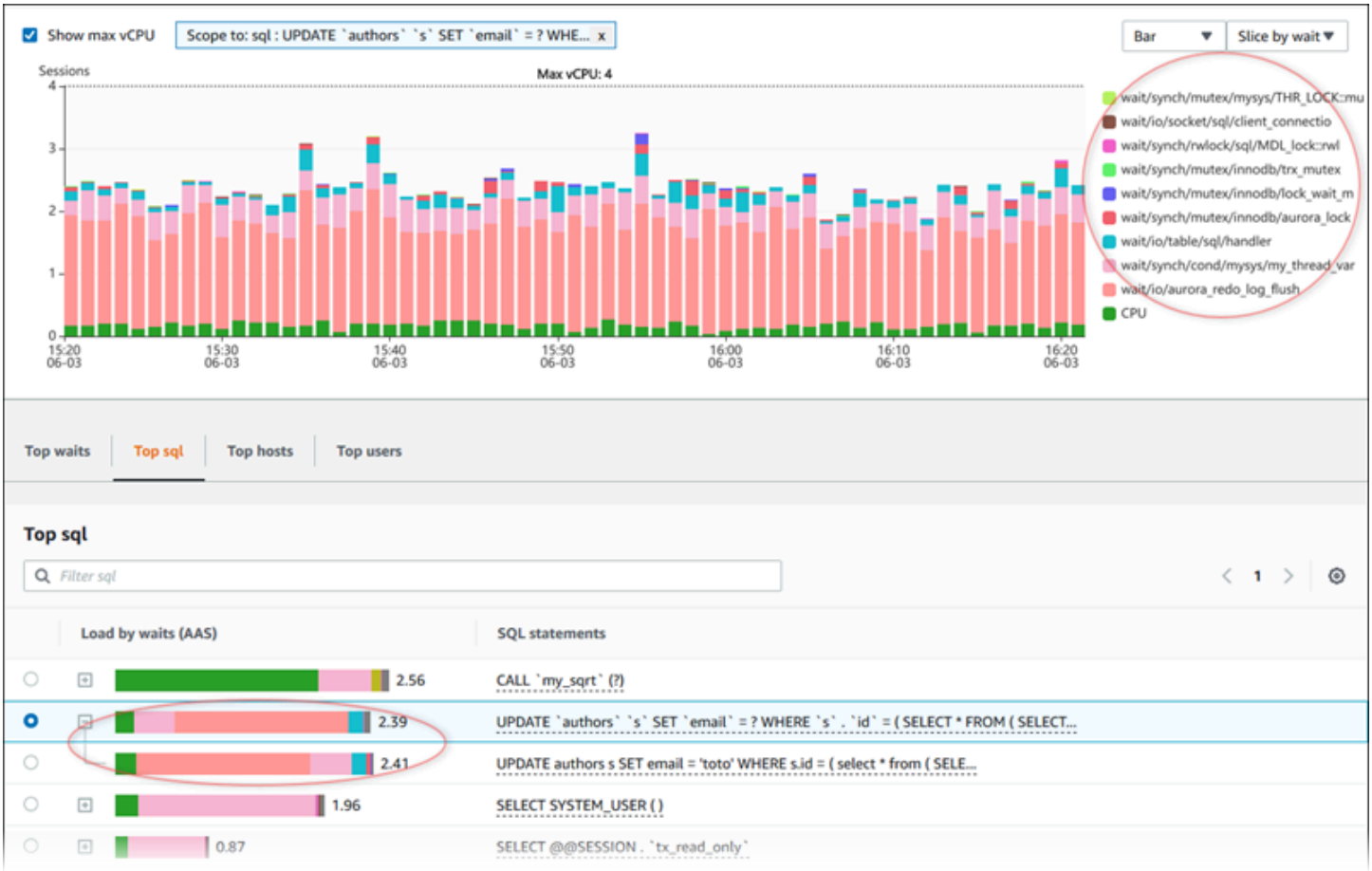
Insights einmal pro Sekunde interne In-Memory-Strukturen erfasst, werden Abfragen mit niedriger Latenz möglicherweise nur für eine einzige Stichprobe angezeigt. Wenn die Abfrage-ID für dieses Beispiel nicht verfügbar ist, kann Performance Insights diese Anweisung nicht mit den entsprechenden Statistiken verknüpfen. Der unbekannte Wert wird durch einen Bindestrich (-) dargestellt.

Eine Beschreibung der SQL Statistiken für die Amazon RDS finden Sie unterSQL-Statistiken für Performance Insights.

Nach Wartezeiten laden () AAS

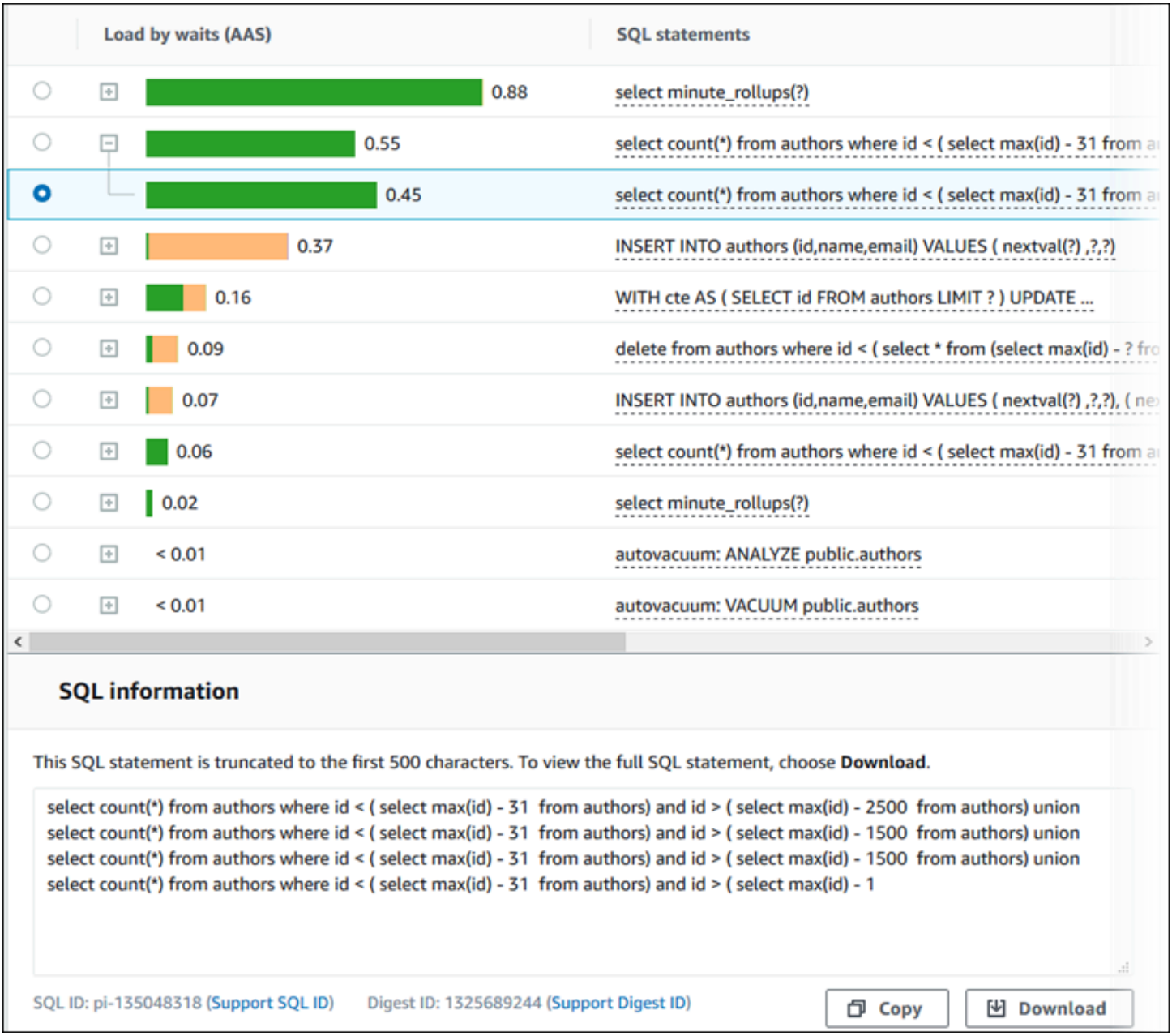
In Top SQL zeigt die Spalte Load by waits (AAS) den Prozentsatz der Datenbanklast, der jedem Top-Load-Element zugeordnet ist. In dieser Spalte wird die Last für dieses Element nach der aktuell im DB-Last-Diagramm ausgewählten Gruppierung wiedergegeben. Weitere Hinweise zum Durchschnitt der aktiven Sitzungen (AAS) finden Sie unterDurchschnittliche aktive Sitzungen.

Beispielsweise können Sie das DB-Last-Diagramm nach Wartezuständen gruppieren. Sie untersuchen SQL Abfragen in der Tabelle mit den Top-Load-Elementen. In diesem Fall ist der Balken DB Load by Waits (DB-Last nach Wartezuständen) so groß, segmentiert und farbcodiert, dass angezeigt wird, zu wieviel Prozent diese Abfrage zum betreffenden Wartezustand beiträgt. Es zeigt zudem auf, welche Wartezustände sich auf die ausgewählte Abfrage auswirken.



Anzeigen der Informationen SQL

In der SQL Tabelle Top können Sie eine Anweisung öffnen, um die zugehörigen Informationen einzusehen. Die Informationen werden im unteren Bereich angezeigt.

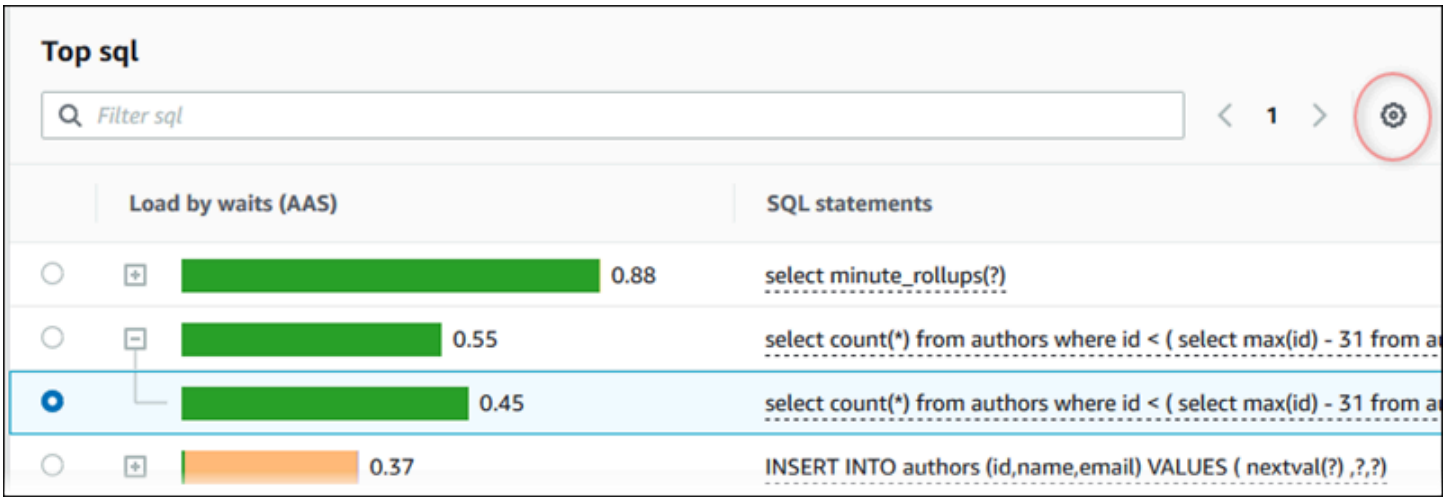


Die folgenden Typen von Bezeichnern (IDs), die Anweisungen zugeordnet SQL sind:

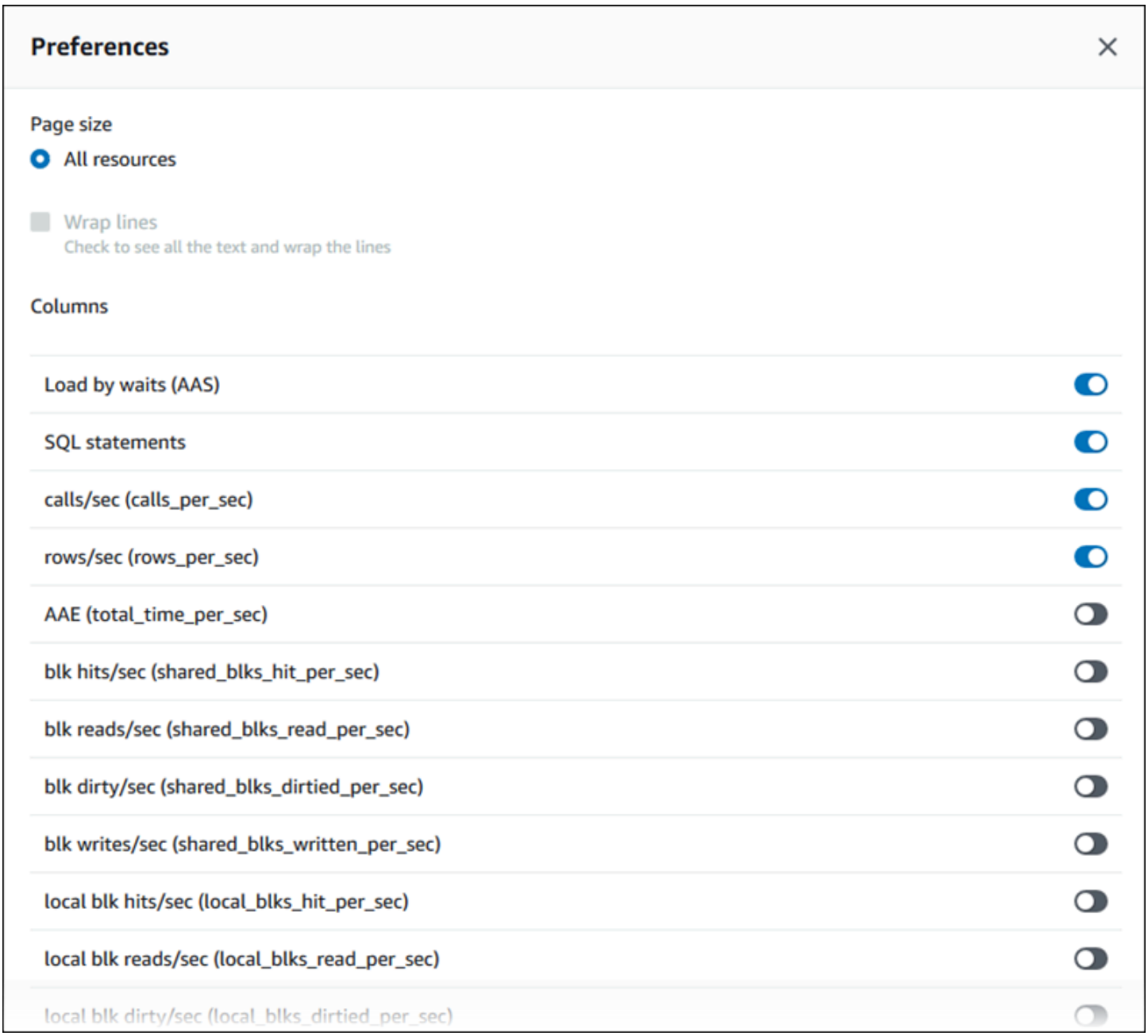
- SQLSupport-ID — Ein Hashwert der SQL ID. Dieser Wert dient nur zum Verweisen auf eine SQL ID, wenn Sie mit AWS Support zusammenarbeiten. AWS Der Support hat keinen Zugriff auf Ihre aktuellen Daten SQL IDs und Ihren SQL Text.
- Support-Digest-ID: ein Hash-Wert der Digest-ID Dieser Wert dient nur zum Verweisen auf eine Digest-ID, wenn Sie mit AWS Support zusammenarbeiten. AWS Der Support hat keinen Zugriff auf Ihre aktuelle Zusammenfassung IDs und Ihren SQL Text.

Wählen Sie Statistikeinstellungen

Sie können die im oberen SQL Tab angezeigten Statistiken steuern, indem Sie das Einstellungen-Symbol wählen.



Durch das Auswählen des Symbols Präferenzen wird das Fenster Präferenzen geöffnet. Der folgende Screenshot ist ein Beispiel für das Fenster Preferences (Präferenzen).



Um die Statistiken zu aktivieren, die auf der oberen SQL Registerkarte angezeigt werden sollen, scrollen Sie mit der Maus zum unteren Rand des Fensters und wählen Sie dann Weiter.

Weitere Informationen zu Statistiken pro Sekunde oder pro Anruf für die Amazon RDS finden Sie im Abschnitt Engine-spezifische SQL Statistiken unter SQL-Statistiken für Performance Insights

Zugriff auf mehr SQL-Text im Performance-Insights-Dashboard

Standardmäßig zeigt jede Zeile in der Tabelle Haupt-SQL für jede SQL-Anwendung 500 Byte SQL-Text an.



Wenn eine SQL-Anweisung 500 Byte überschreitet, können Sie mehr Text im SQL-Text-Abschnitt unterhalb der Haupt-SQL-Tabelle sehen. In diesem Fall beträgt die maximale Länge für den in SQL-Text angezeigten Text 4 KB. Dieses Limit wird von der Konsole eingeführt und unterliegt den von der Datenbank-Engine festgelegten Grenzwerten. Zum Speichern des in SQL-Text gezeigten Texts wählen Sie Herunterladen.

Themen

- Beschränkungen der Textgröße für Amazon RDS-Engines
- Einstellung des SQL Textlimits für Amazon RDS für SQL
- SQLText im Performance Insights Insights-Dashboard anzeigen und herunterladen

Beschränkungen der Textgröße für Amazon RDS-Engines

Beim Herunterladen von SQL-Text bestimmt die Datenbank-Engine dessen maximale Länge. Sie können SQL-Text bis zu den folgenden Grenzwerten pro Engine herunterladen.

DB-Engine	Maximale Länge des heruntergeladenen Textes
Amazon RDS für MySQL und MariaDB	Die Länge ist auf 4.096 Byte festgelegt, wenn das Performance-Schema aktiviert ist. Wenn das Leistungsschema nicht aktiviert ist, ist die Länge auf 65.535 Byte festgelegt.
Amazon RDS for Microsoft SQL Server	4,096 Zeichen
Amazon RDS für Oracle	1 000 Byte

Der SQL-Text-Abschnitt der Performance Insights-Konsole zeigt den maximalen Wert an, den die Engine zurückgibt. Wenn MySQL beispielsweise höchstens 1 KB an Performance Insights zurückgibt, kann es nur 1 KB sammeln und anzeigen, auch wenn die ursprüngliche Abfrage größer ist. Wenn Sie also die Abfrage in SQL-Text anzeigen oder herunterladen, gibt Performance Insights die gleiche Anzahl von Bytes zurück.

Wenn Sie die API AWS CLI oder verwenden, hat Performance Insights nicht das von der Konsole erzwungene Limit von 4 KB. `DescribeDimensionKeys`und `GetResourceMetrics` gibt höchstens 500 Byte zurück.

> **Note**
>
> `GetDimensionKeyDetails`gibt die vollständige Abfrage zurück, aber die Größe unterliegt dem Engine-Limit.

Einstellung des SQL Textlimits für Amazon RDS für SQL

Amazon RDS for Postgre SQL verarbeitet Text anders. Sie können die Textgrößenbeschränkung mit dem DB-Instance-Parameter `track_activity_query_size` festlegen. Dieser Parameter hat folgende Merkmale:

Standardtextgröße

Auf Amazon RDS for Postgre SQL Version 9.6 ist die Standardeinstellung für den `track_activity_query_size` Parameter 1.024 Byte. RDSBei Amazon for Postgre SQL Version 10 oder höher ist der Standardwert 4.096 Byte.

Maximale Textgröße

Das Limit für Amazon `track_activity_query_size` liegt bei 102.400 Byte RDS für Postgre SQL Version 12 und niedriger. Das Maximum beträgt 1 MB für Version 13 und höher.

Wenn die Engine 1 MB an Performance Insights zurückgibt, zeigt die Konsole nur die ersten 4 KB an. Wenn Sie die Abfrage herunterladen, erhalten Sie die gesamten 1 MB. In diesem Fall geben das Anzeigen und Herunterladen eine unterschiedliche Anzahl von Bytes zurück. Weitere Informationen zum `track_activity_query_size` DB-Instance-Parameter finden Sie unter Runtime Statistics in der Postgre-Dokumentation. SQL

Um die SQL Textgröße zu erhöhen, erhöhen Sie das Limit. `track_activity_query_size` Um den Parameter zu ändern, ändern Sie die Parametereinstellung in der Parametergruppe, die der Amazon RDS for Postgre SQL zugeordnet ist.

Ändern Sie die Einstellung wie folgt, wenn die Instance die Standardparametergruppe verwendet:

1. Erstellen Sie eine neue DB-Instance-Parametergruppe für die entsprechende DB-Engine und DB-Engine-Version.
2. Stellen Sie den Parameter in der neuen Parametergruppe ein.
3. Ordnen Sie die neue Parametergruppe der DB-Instance zu.

Informationen über das Einstellen eines DB-Instance-Parameters finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

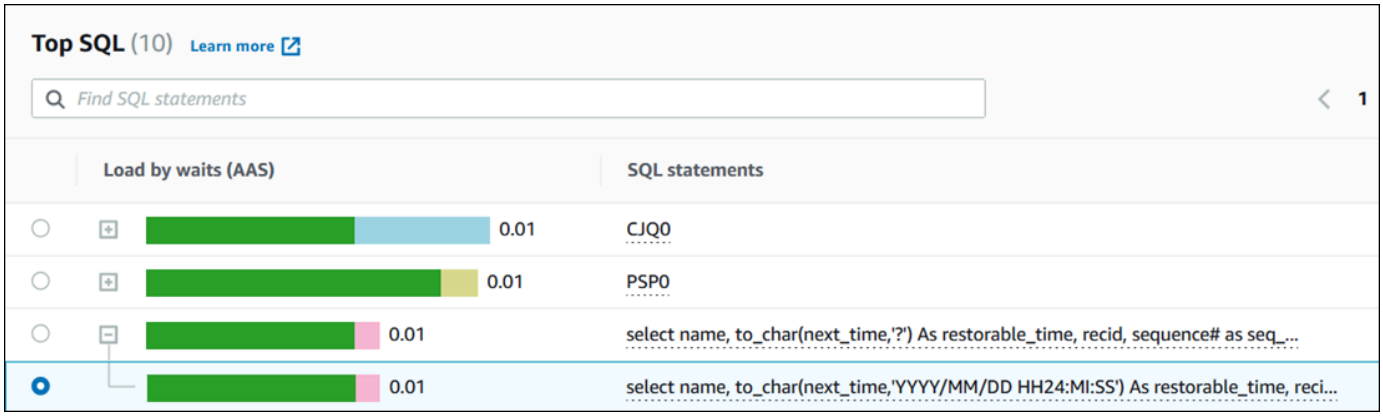
SQLText im Performance Insights Insights-Dashboard anzeigen und herunterladen

Im Performance Insights Insights-Dashboard können Sie SQL Text anzeigen oder herunterladen.

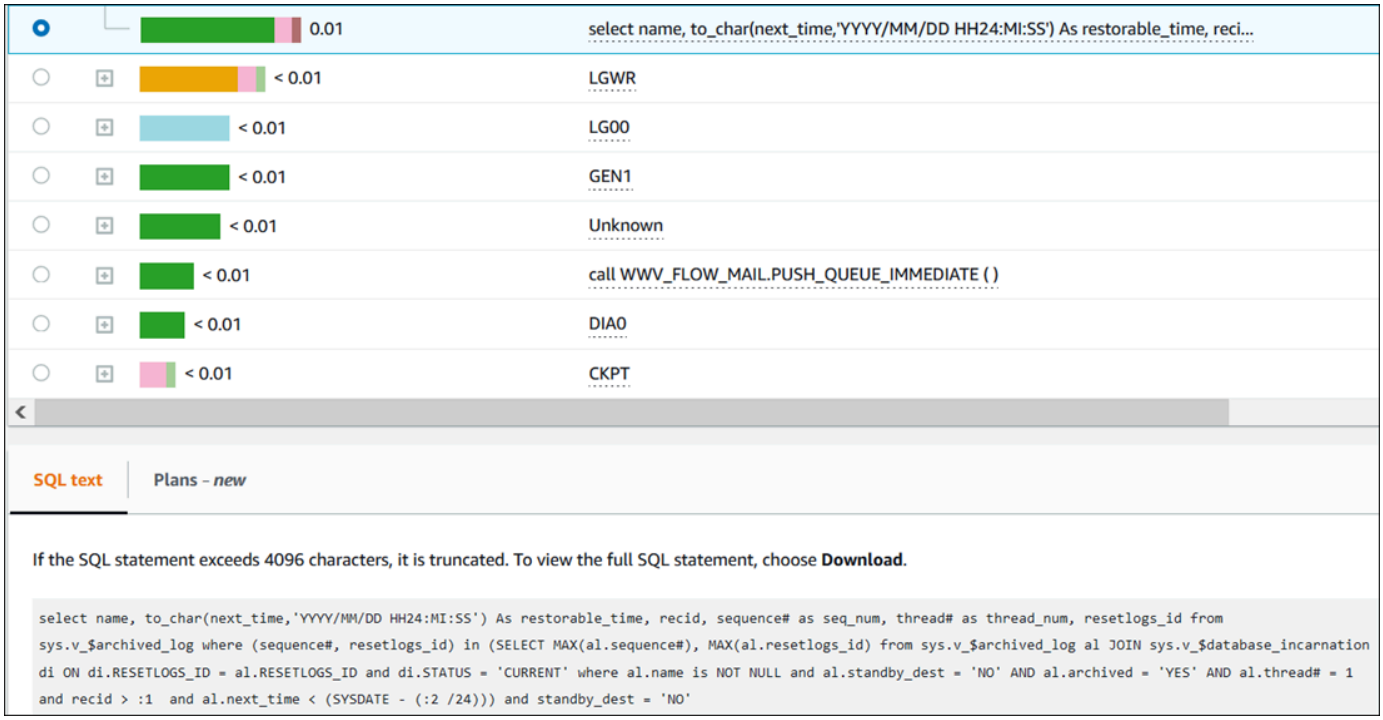
Um mehr SQL Text im Performance Insights Insights-Dashboard anzuzeigen

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Performance-Insights aus.
3. Wählen Sie eine DB-Instance aus.
4. Scrollen Sie im Performance Insights Insights-Dashboard nach unten zum SQL Tab „Top“.
5. Klicken Sie auf das Pluszeichen, um einen SQL Digest zu erweitern, und wählen Sie eine der untergeordneten Abfragen des Digests aus.

   SQLAnweisungen mit Text, der größer als 500 Byte ist, sehen der folgenden Abbildung ähnlich.



6. Scrollen Sie nach unten zur Registerkarte SQLText.



Das Performance Insights Insights-Dashboard kann bis zu 4.096 Byte für jede SQL Aussage anzeigen.

7. (Optional) Wählen Sie Kopieren, um die angezeigte SQL Anweisung zu kopieren, oder wählen Sie Herunterladen, um die SQL Anweisung herunterzuladen und den SQL Text bis zum DB-Engine-Limit anzuzeigen.

> **Note**
>
> Um die SQL Erklärung zu kopieren oder herunterzuladen, deaktivieren Sie Popup-Blocker.

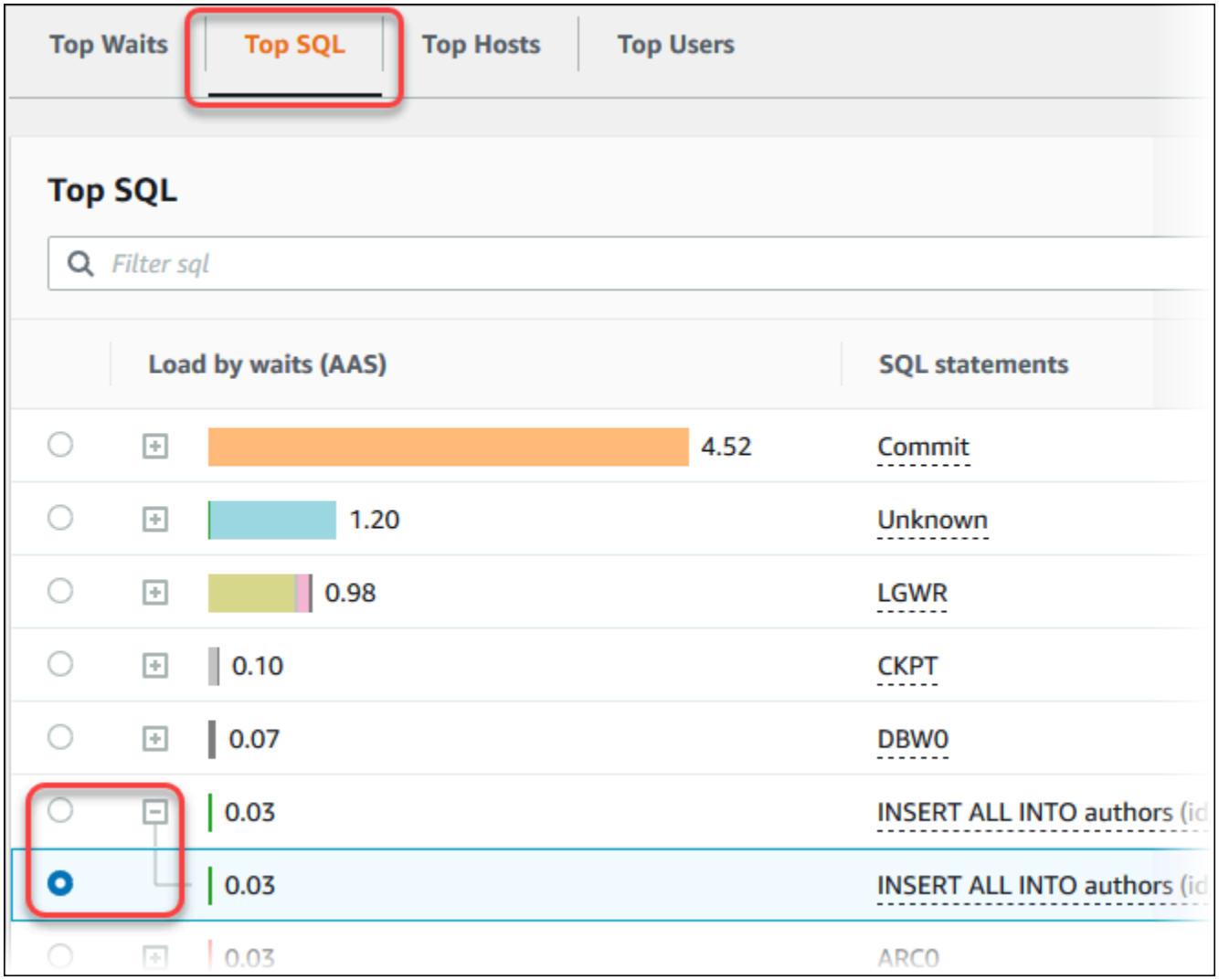
SQLStatistiken im Performance Insights Insights-Dashboard anzeigen

Im Performance Insights Insights-Dashboard sind SQL Statistiken auf der oberen SQL Registerkarte des Datenbank-Ladediagramms verfügbar.

Sehen Sie sich SQL-Statistiken wie folgt an:

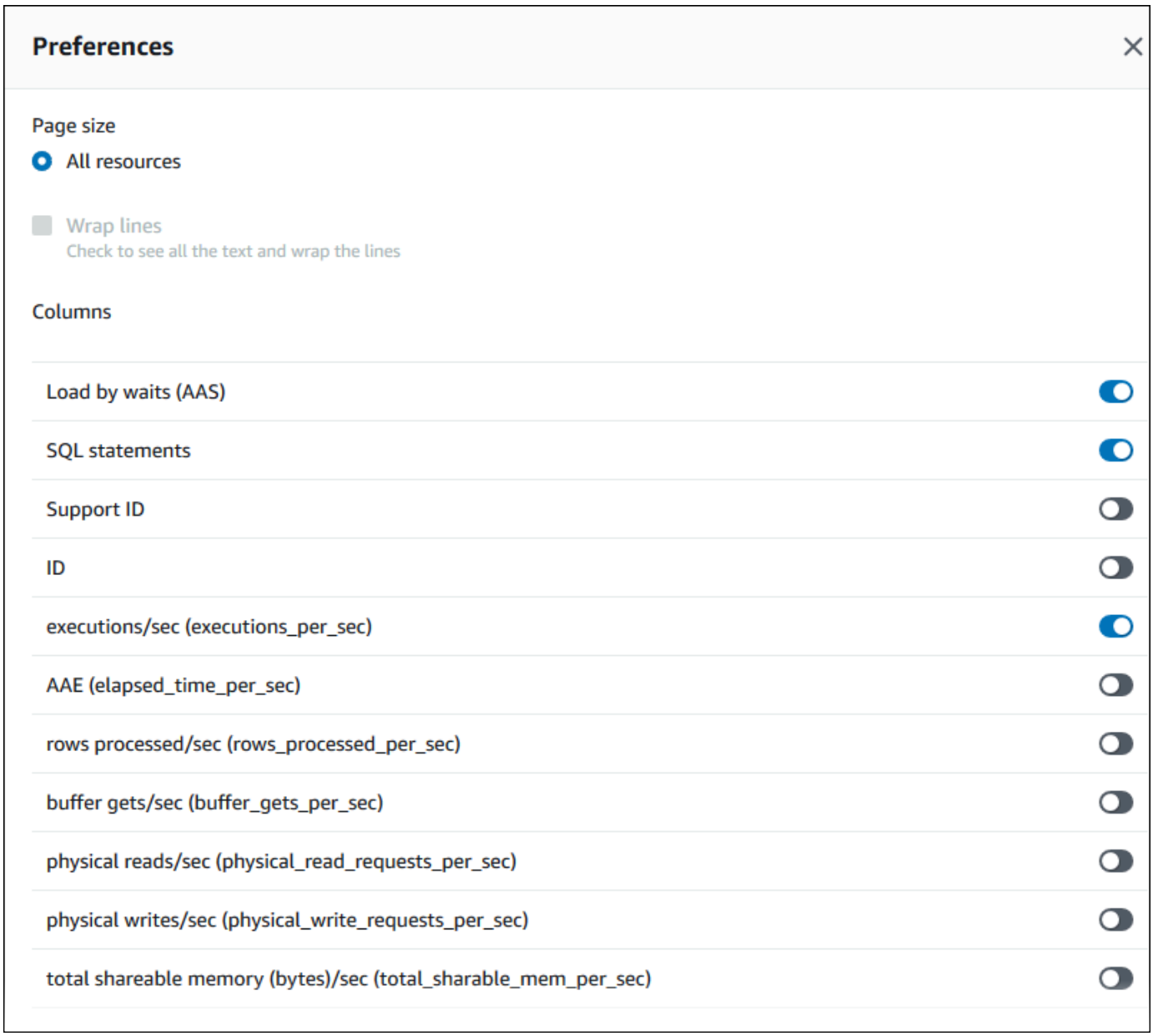
1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.

3. Wählen Sie oben auf der Seite die Datenbank aus, deren SQL Statistiken Sie sehen möchten.
4. Scrollen Sie zum Ende der Seite und wählen Sie die SQL Registerkarte Top.
5. Wählen Sie einen individuellen Kontoauszug oder eine Übersichtsabfrage.

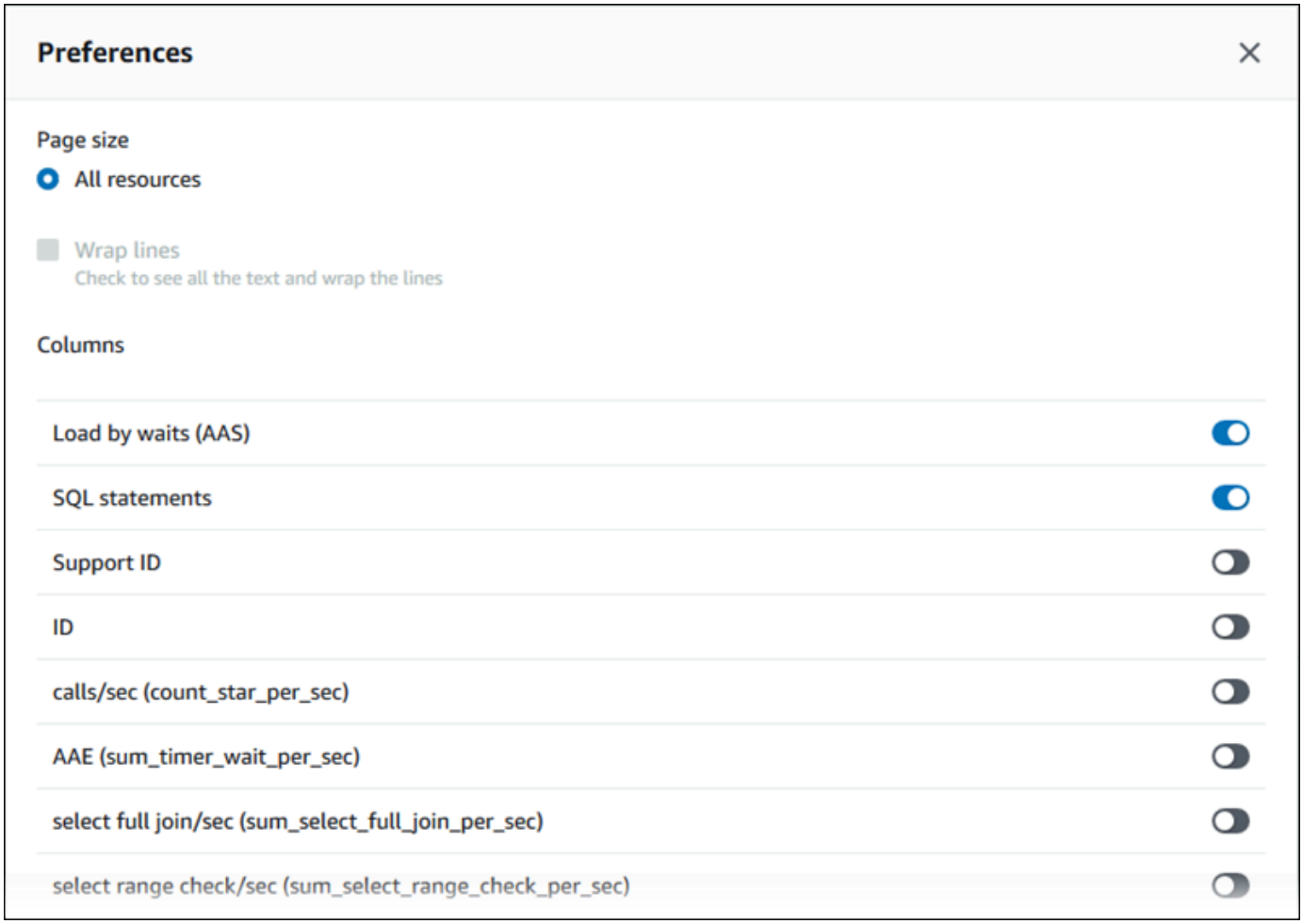


6. Sie können die Statistiken auswählen, die angezeigt werden sollen, indem Sie oben rechts im Diagramm das Zahnradsymbol auswählen. Eine Beschreibung der SQL Statistiken für die Amazon RDS finden Sie unterSQL-Statistiken für Performance Insights.

   Das folgende Beispiel zeigt die Einstellungen für Oracle-DB-Instances.



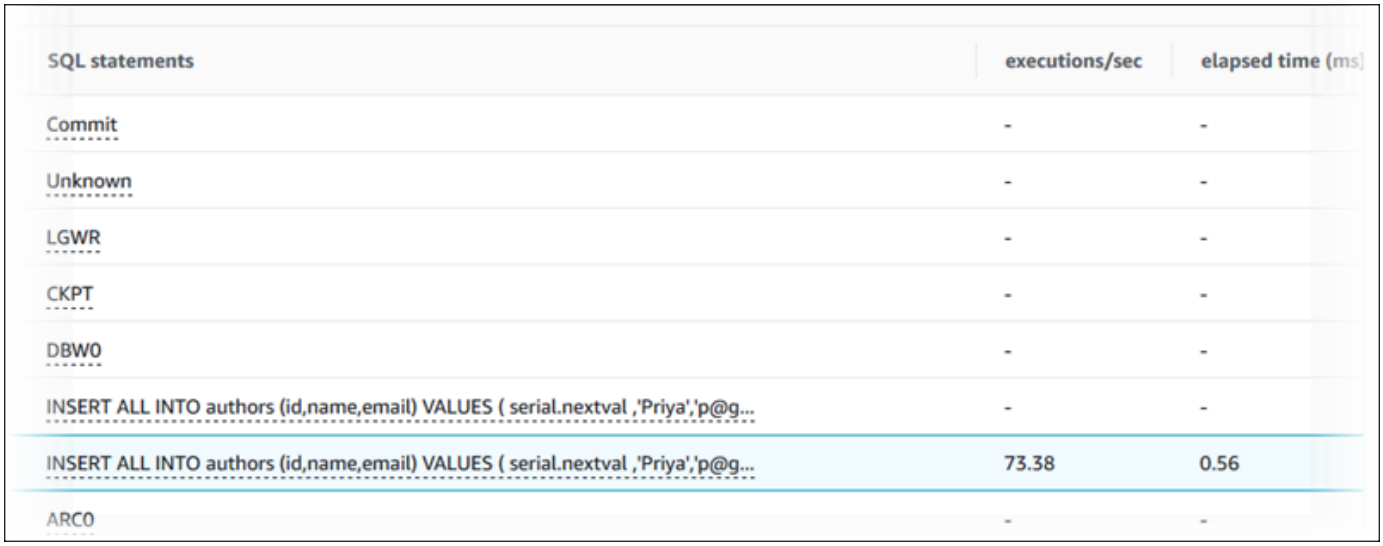
Das folgende Beispiel zeigt die Einstellungen für MariaDB- und My SQL Aurora My SQL .



7. Wählen Sie „Save“ (Speichern) aus, um Ihre Einstellungen zu speichern.

   Die SQLTop-Tabelle wird aktualisiert.

   Das folgende Beispiel zeigt Statistiken für eine SQL Oracle-Abfrage.



Analyse der höchsten PDB Oracle-Auslastung

Bei der Analyse der Auslastung einer Oracle-Container-Datenbank (CDB) möchten Sie vielleicht herausfinden, welche austauschbaren Datenbanken (PDBs) am meisten zur DB-Auslastung beitragen. Möglicherweise möchten Sie auch die Leistung von Personen vergleichenPDBs, die ähnliche Abfragen ausführen, um die Leistung zu optimieren. Weitere Informationen zu Oracle finden CDBs Sie unterRDS für Oracle-Datenbankarchitektur.

Im Amazon RDS Performance Insights Insights-Dashboard finden Sie Informationen zu austauschbaren Datenbanken (PDBs) unter der PDB Registerkarte Top auf der Registerkarte Dimensionen.

Informationen zur Unterstützung dieser Funktion für die Region, die DB-Engine und die Instance-Klasse finden Sie unter. DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen

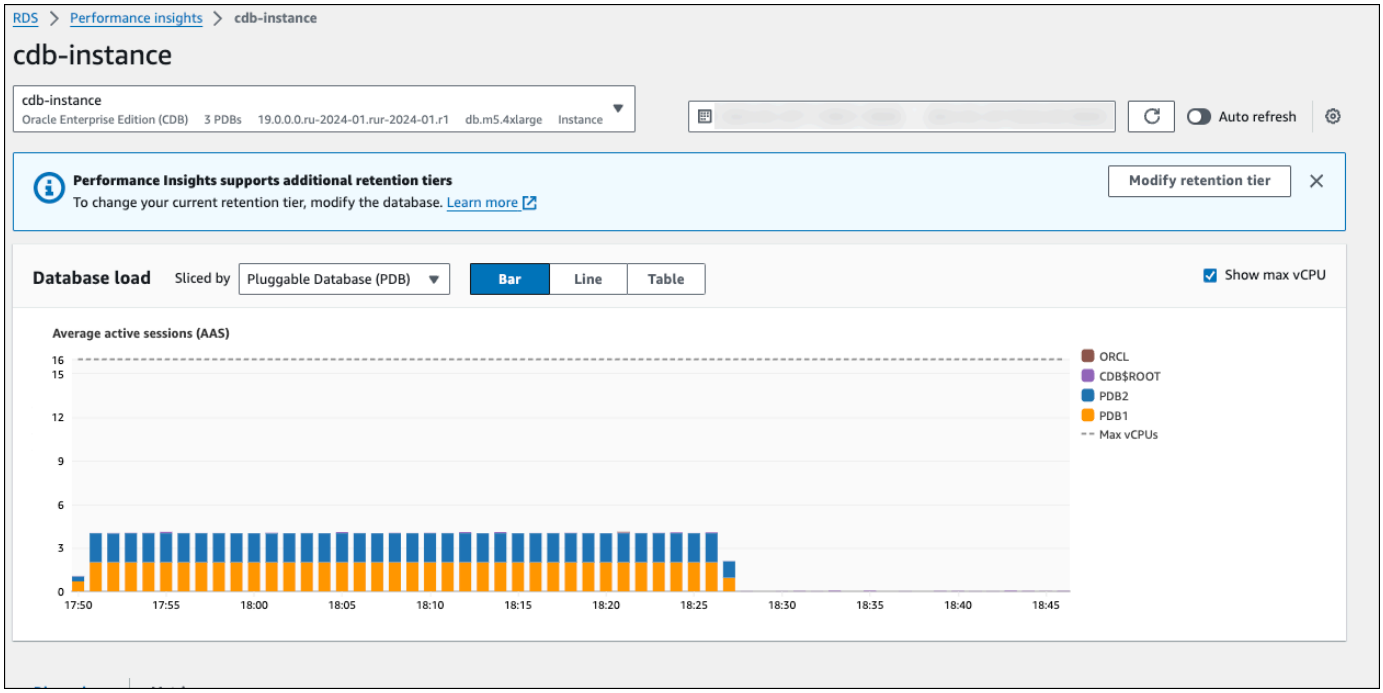
So analysieren Sie die PDB Top-Auslastung in einem Oracle CDB

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Performance Insights aus.
3. Wählen Sie eine CDB Oracle-Instanz aus.

   Das Performance-Insights-Dashboard wird für die DB-Instance angezeigt.

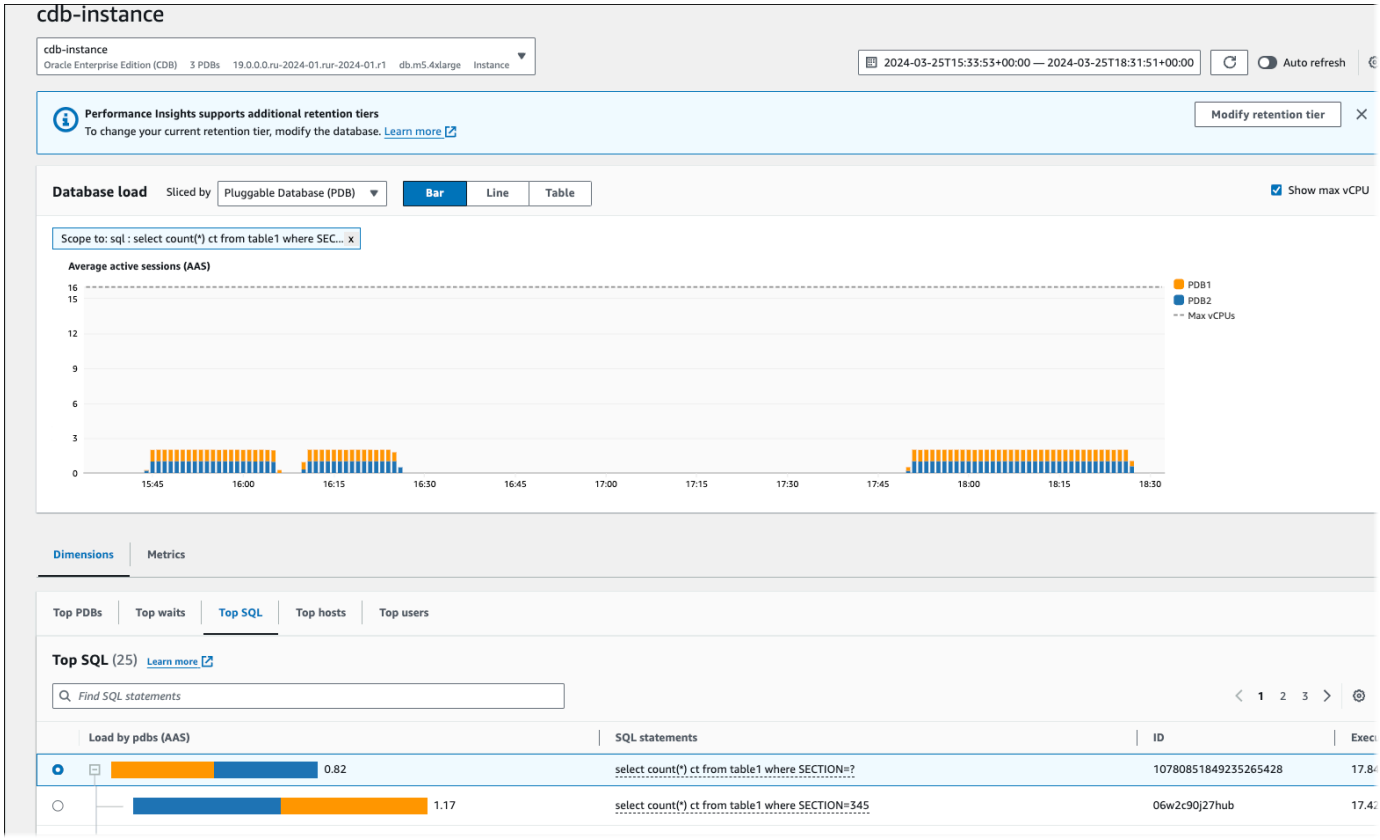
4. Wählen Sie im Abschnitt Database load (DB load) die Option Pluggable database (PDB) neben Slice by aus.

   Das Diagramm „Durchschnittliche Anzahl aktiver Sitzungen“ zeigt die Sitzungen PDB mit der höchsten Auslastung. Die PDB Identifikatoren werden rechts neben den farbcodierten Quadraten angezeigt. Jeder Bezeichner identifiziert eindeutig einen. PDB

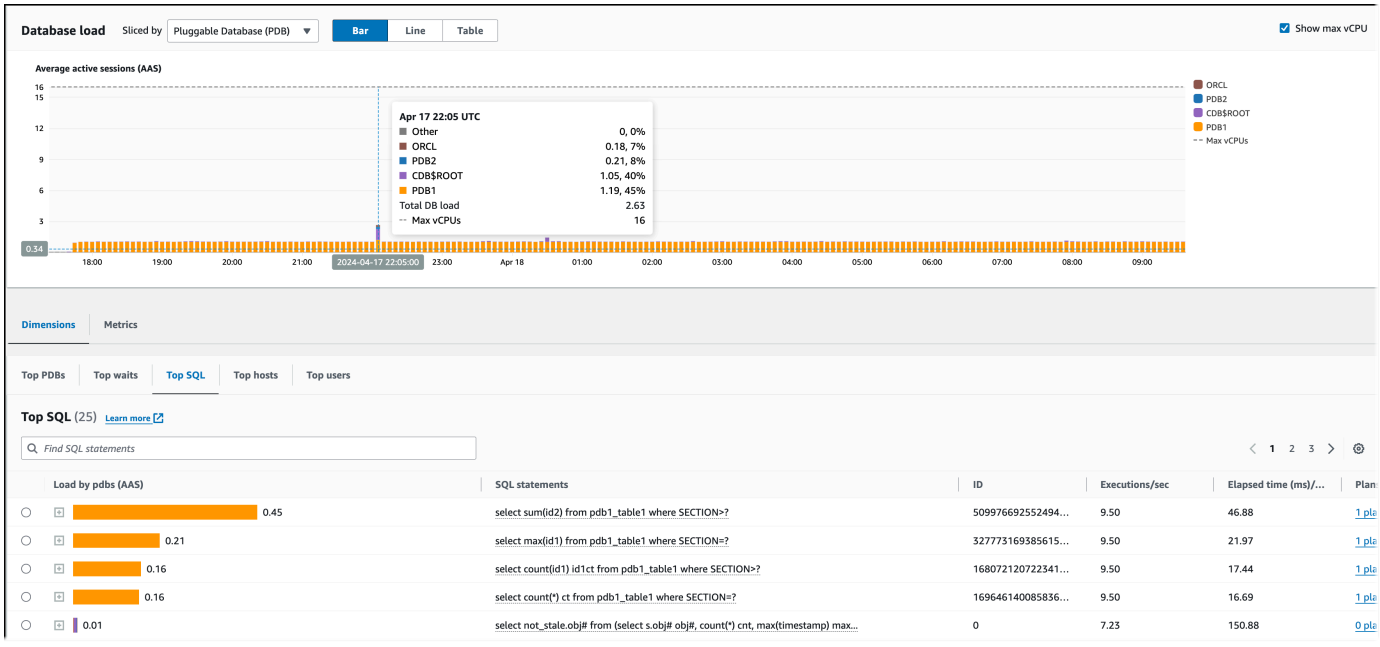


5. Scrollen Sie nach unten zur SQL Registerkarte „Top“.

   Im folgenden Beispiel sehen Sie dieselbe SQL Abfrage und die Last, die sie auf mehrere auslöstPDBs.



Im folgenden Beispiel verarbeitet eine einzelne PDB eine höhere Last als eine andere PDBs in derCDB.



Weitere Informationen zu Oracle CDBs finden Sie unter CDBsund PDBs.

Analysieren von Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS

Im Amazon RDS Performance Insights Insights-Dashboard finden Sie Informationen zu Ausführungsplänen für Oracle- und SQL Server-DB-Instances. Anhand dieser Informationen können Sie herausfinden, welche Pläne am meisten zur DB-Auslastung beitragen.

Um zu analysieren Oracle or SQL Server Ausführungspläne finden Sie in den folgenden Themen.

Ausführungspläne analysieren

- Analysieren von Oracle-Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS
- Analysieren von SQL Serverausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS

Überblick über die Analyse von Ausführungsplänen für Amazon RDS

Sie können das Amazon RDS Performance Insights Insights-Dashboard verwenden, um zu erfahren, welche Pläne am meisten zur DB-Auslastung für Oracle- und SQL Server-DB-Instances beitragen.

Beispielsweise könnten die wichtigsten SQL Kontoauszüge zu einem bestimmten Zeitpunkt die in der folgenden Tabelle aufgeführten Pläne verwenden.

Oben SQL	Plan
SELECTSUM(amount_sold) FROM sales WHERE prod_id = 10	Plan A
SELECTSUM(amount_sold) sales prod_id = 521 FROM WHERE	Plan B
SELECTSUM(s_total) Verkaufsregion = 10 FROM WHERE	Plan A
SELECT* FROM emp WHERE emp_id = 1000	Plan C
SELECTSUM(amount_sold) sales prod_id = 72 FROM WHERE	Plan A

Mit der Planfunktion von Performance Insights können Sie Folgendes tun:

- Finden Sie heraus, welche Pläne von den häufigsten Abfragen verwendet werden. SQL

  Sie könnten beispielsweise herausfinden, dass der Großteil der DB-Last durch Abfragen generiert wird, die Plan A und Plan B verwenden, und nur ein kleiner Prozentsatz Plan C verwendet.
- verschiedene Pläne für dieselbe Abfrage vergleichen

  Im vorhergehenden Beispiel sind drei Abfragen mit Ausnahme der Produkt-ID identisch. Zwei Abfragen verwenden Plan A, aber eine Abfrage verwendet Plan B. Um den Unterschied zwischen den beiden Plänen zu erkennen, können Sie Performance Insights verwenden.
- herausfinden, wann eine Abfrage auf einen neuen Plan umgeschaltet hat

  Sie könnten sehen, dass eine Abfrage Plan A verwendet hat und dann zu einem bestimmten Zeitpunkt zu Plan B gewechselt ist. Gab es zu diesem Zeitpunkt eine Änderung in der Datenbank? Wenn beispielsweise eine Tabelle leer ist, kann der Optimierer einen vollständigen Tabellenscan auswählen. Wenn die Tabelle mit einer Million Zeilen geladen wird, wechselt der Optimierer möglicherweise zu einem Indexbereichs-Scan.
- einen Drilldown zu den einzelnen Schritten eines Plans mit den höchsten Kosten durchführen

  Beispielsweise könnte die Abfrage für eine lang andauernde Abfrage eine fehlende Join-Bedingung in einem Equi-Join anzeigen. Diese fehlende Bedingung erzwingt ein kartesisches Join, das alle Zeilen von zwei Tabellen verbindet.

Sie können die oben genannten Aufgaben mithilfe der Planerfassungsfunktion von Performance Insights ausführen. So wie Sie Abfragen nach Warteereignissen und Top aufteilen könnenSQL, können Sie sie auch nach der Plandimension segmentieren.

Analysieren von Oracle-Ausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS

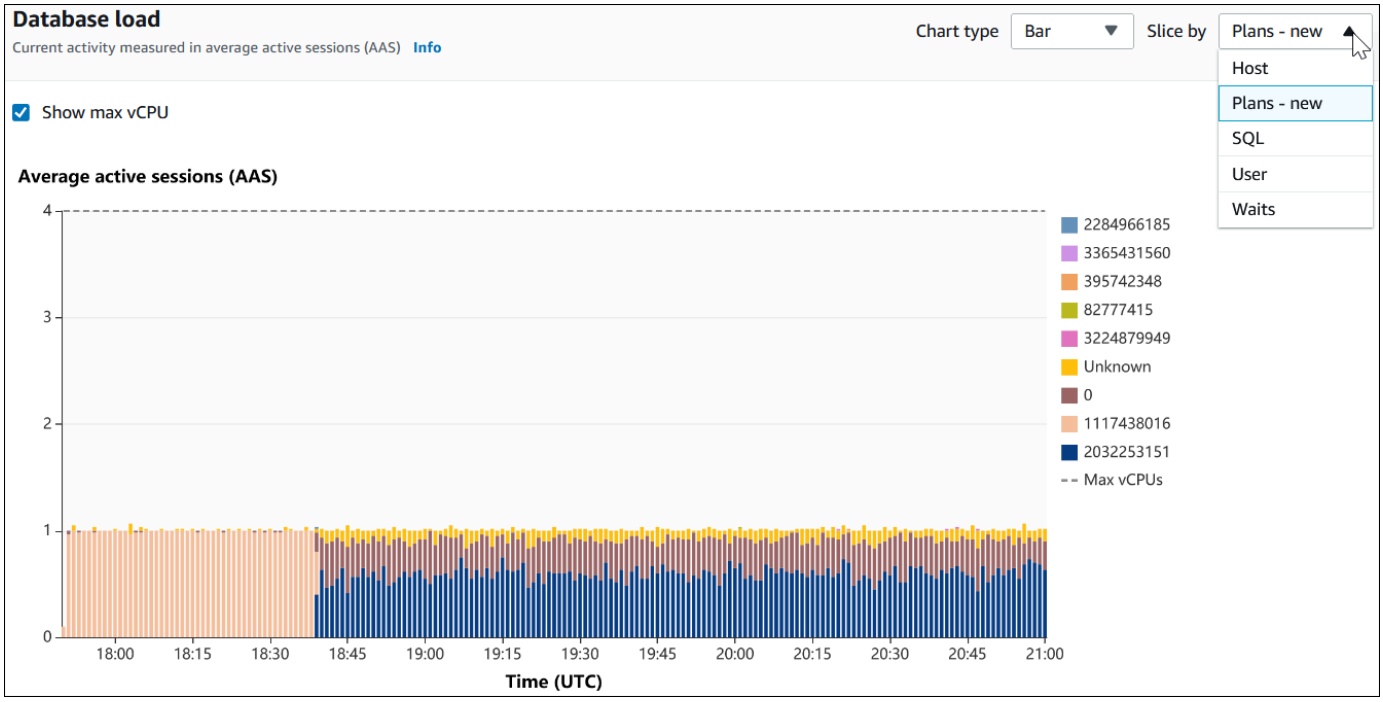
Wenn Sie die DB-Last in einer Oracle-Datenbank analysieren, möchten Sie möglicherweise wissen, welche Pläne am meisten zur DB-Last beitragen. Mithilfe der Planerfassungsfunktion von Performance Insights können Sie ermitteln, welche Pläne am meisten zur DB-Auslastung beitragen.

Oracle-Ausführungspläne über die Konsole analysieren

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Performance-Insights aus.

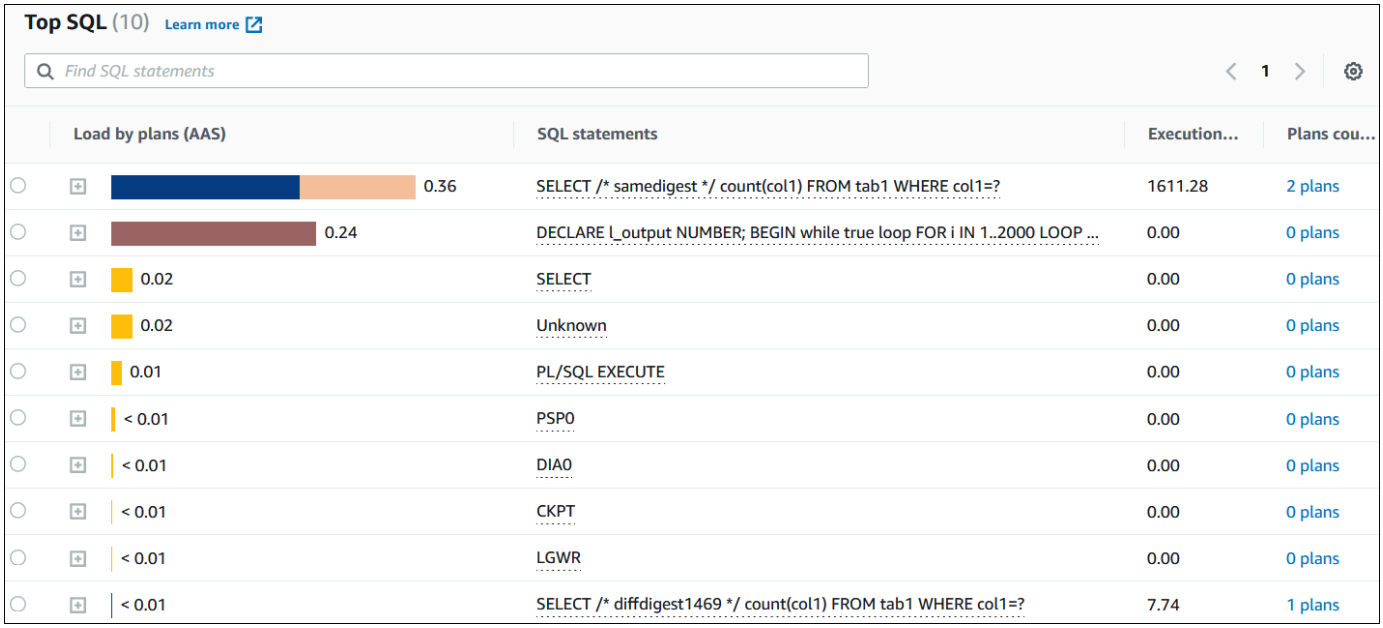
3. Wählen Sie eine Oracle-DB-Instance aus. Das Performance Insights-Dashboard wird für diese DB-Instance angezeigt.
4. Wählen Sie im Bereich Database load (DB load) (Datenbanklast (DB-Last)) neben Slice by (Aufteilen nach) die Option Plans (Pläne) aus.

   Das Diagramm „Durchschnittliche Anzahl aktiver Sitzungen“ zeigt die Tarife, die für Ihre wichtigsten SQL Kontoauszüge verwendet wurden. Die Plan-Hash-Werte erscheinen rechts neben den farbcodierten Quadraten. Jeder Hash-Wert identifiziert eindeutig einen Plan.



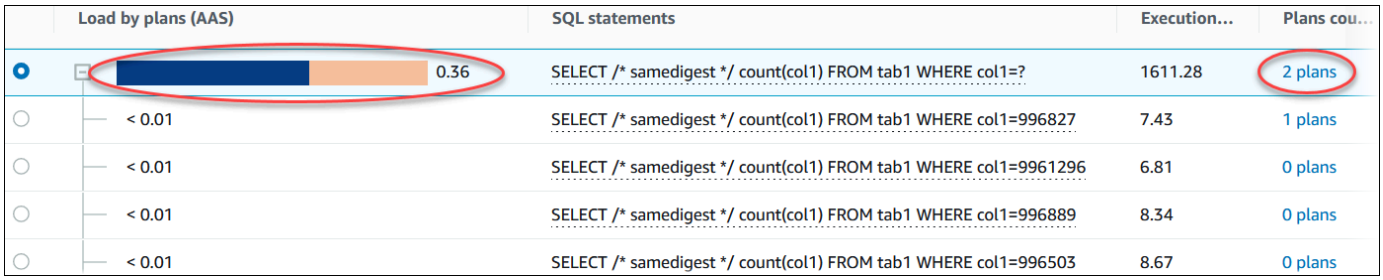
5. Scrollen Sie nach unten zum SQL Tab „Top“.

   Im folgenden Beispiel hat der SQL Top-Digest zwei Pläne. Am Fragezeichen in der Anweisung erkennen Sie, dass es sich um ein Digest handelt.



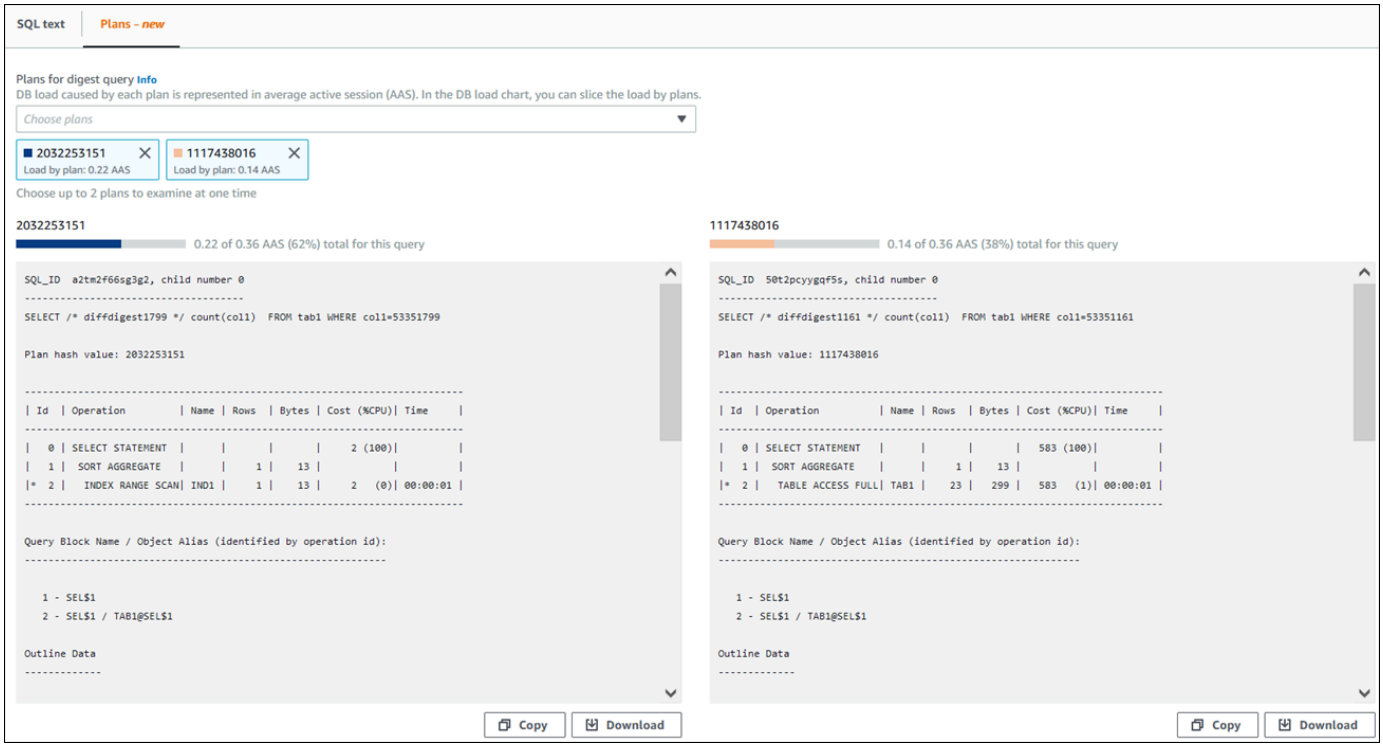
6. Wählen Sie den Digest aus, um ihn auf seine Komponentenanweisungen zu erweitern.

   Im folgenden Beispiel ist die SELECT-Anweisung eine Digest-Abfrage. Die Komponentenabfragen im Digest verwenden zwei verschiedene Pläne. Die Farben der Pläne entsprechen dem Datenbanklastdiagramm. Die Gesamtzahl der Pläne im Digest ist in der zweiten Spalte gezeigt.

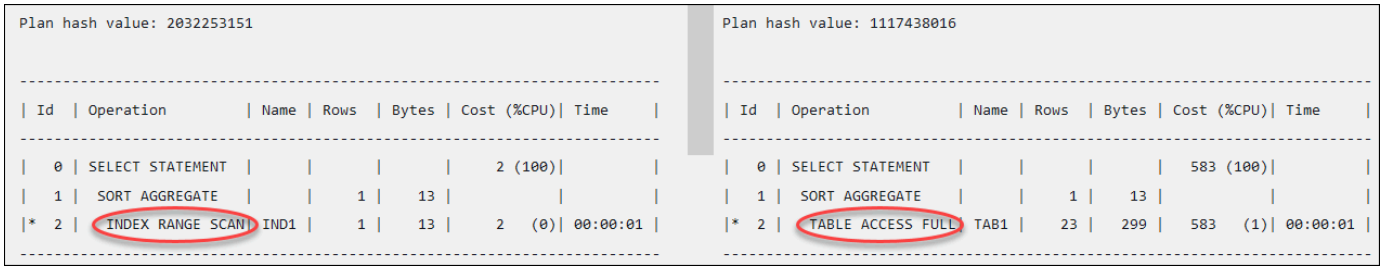


7. Blättern Sie nach unten und wählen Sie aus der Liste Plans for digest query (Pläne für Digest-Abfrage) zwei Pläne zum Vergleich aus.

   Sie können jeweils einen oder zwei Pläne für eine Abfrage anzeigen. Der folgende Screenshot vergleicht die beiden Pläne im Digest, mit Hash 2032253151 und Hash 1117438016. Im folgenden Beispiel verwenden 62 % der durchschnittlichen aktiven Sitzungen, die diese Digest-Abfrage ausführen, den Plan auf der linken Seite, während 38 % den Plan auf der rechten Seite verwenden.



In diesem Beispiel unterscheiden sich die Pläne in einem wichtigen Punkt. Schritt 2 in Plan 2032253151 verwendet einen Index-Scan, während Plan 1117438016 einen vollständigen Tabellenscan verwendet. Bei einer Tabelle mit einer großen Zeilenzahl ist eine Abfrage einer einzelnen Zeile mit einem Index-Scan fast immer schneller.



8. (Optional) Wählen Sie Copy (Kopieren) aus, um den Plan in die Zwischenablage zu kopieren, oder Download (Herunterladen), um den Plan auf der Festplatte zu speichern.

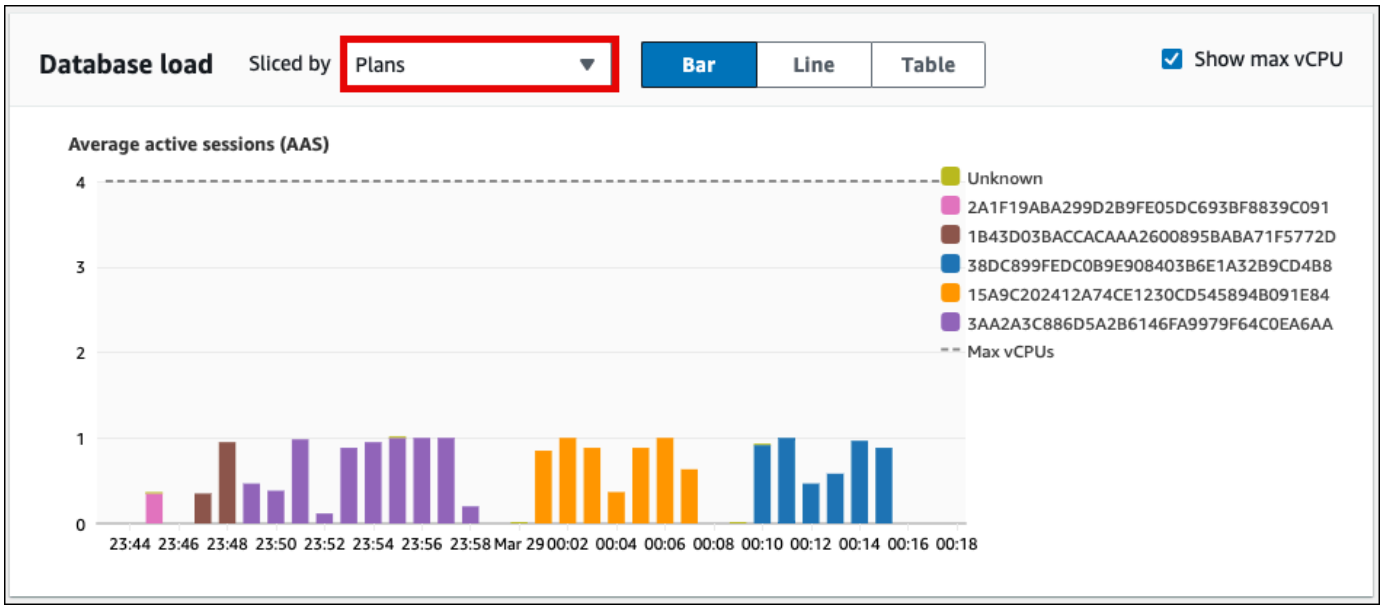
Analysieren von SQL Serverausführungsplänen mithilfe des Performance Insights Insights-Dashboards für Amazon RDS

Wenn Sie die DB-Auslastung einer SQL Serverdatenbank analysieren, möchten Sie vielleicht wissen, welche Pläne am meisten zur DB-Auslastung beitragen. Mithilfe der Planerfassungsfunktion von Performance Insights können Sie ermitteln, welche Pläne am meisten zur DB-Auslastung beitragen.

Um SQL Serverausführungspläne mithilfe der Konsole zu analysieren

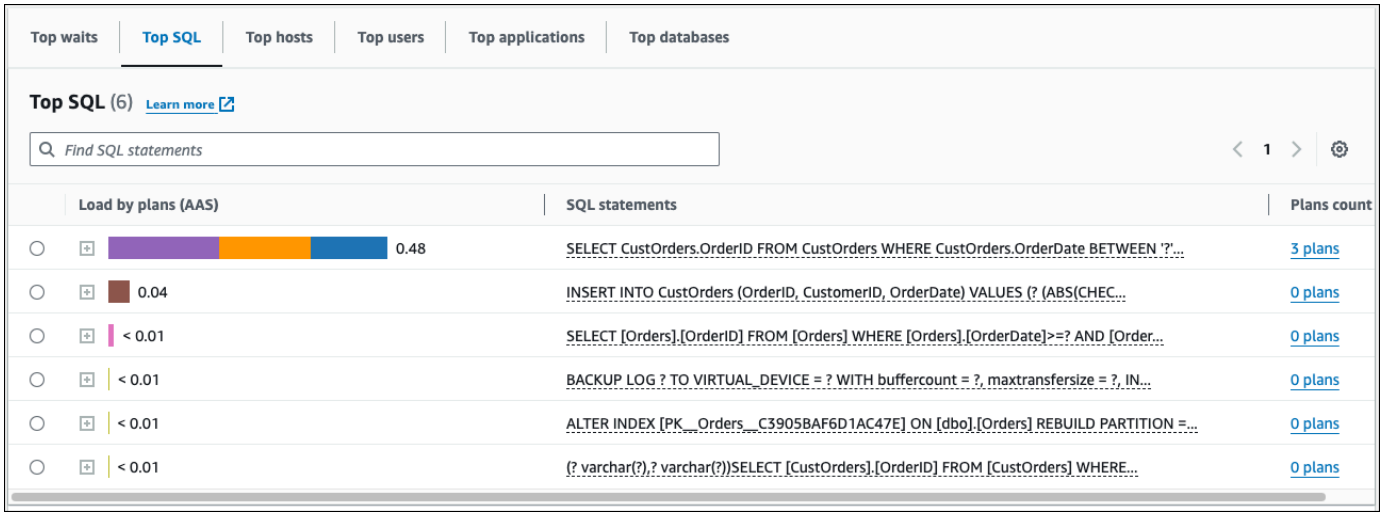
1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Performance-Insights aus.
3. Wählen Sie eine SQL Server-DB-Instance aus. Das Performance Insights-Dashboard wird für diese DB-Instance angezeigt.
4. Wählen Sie im Bereich Database load (DB load) (Datenbanklast (DB-Last)) neben Slice by (Aufteilen nach) die Option Plans (Pläne) aus.

   Das Diagramm „Durchschnittliche Anzahl aktiver Sitzungen“ zeigt die Tarife, die für Ihre wichtigsten SQL Kontoauszüge verwendet wurden. Die Plan-Hash-Werte erscheinen rechts neben den farbcodierten Quadraten. Jeder Hash-Wert identifiziert eindeutig einen Plan.



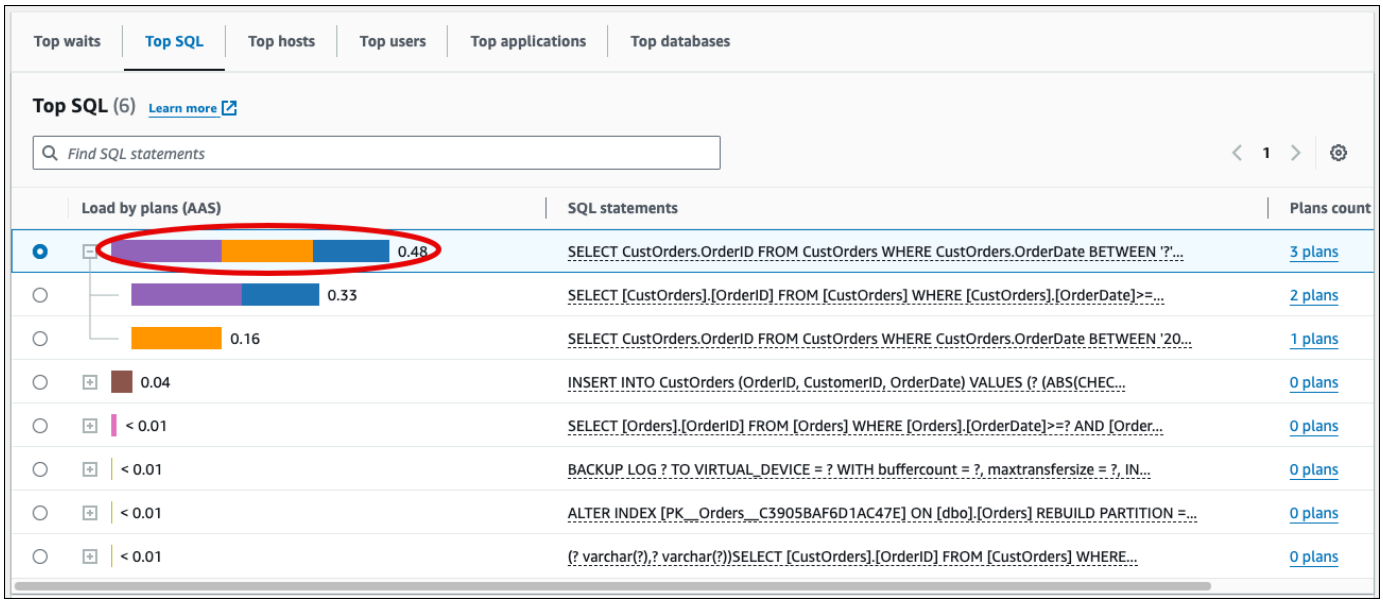
5. Scrollen Sie nach unten zum SQL Tab „Top“.

   Im folgenden Beispiel hat der SQL Top-Digest drei Pläne. Das Vorhandensein eines Fragezeichens in der SQL Aussage weist darauf hin, dass es sich bei der Aussage um eine Zusammenfassung handelt. Um die vollständige SQL Aussage anzuzeigen, wählen Sie einen Wert in der Spalte SQLKontoauszüge aus.



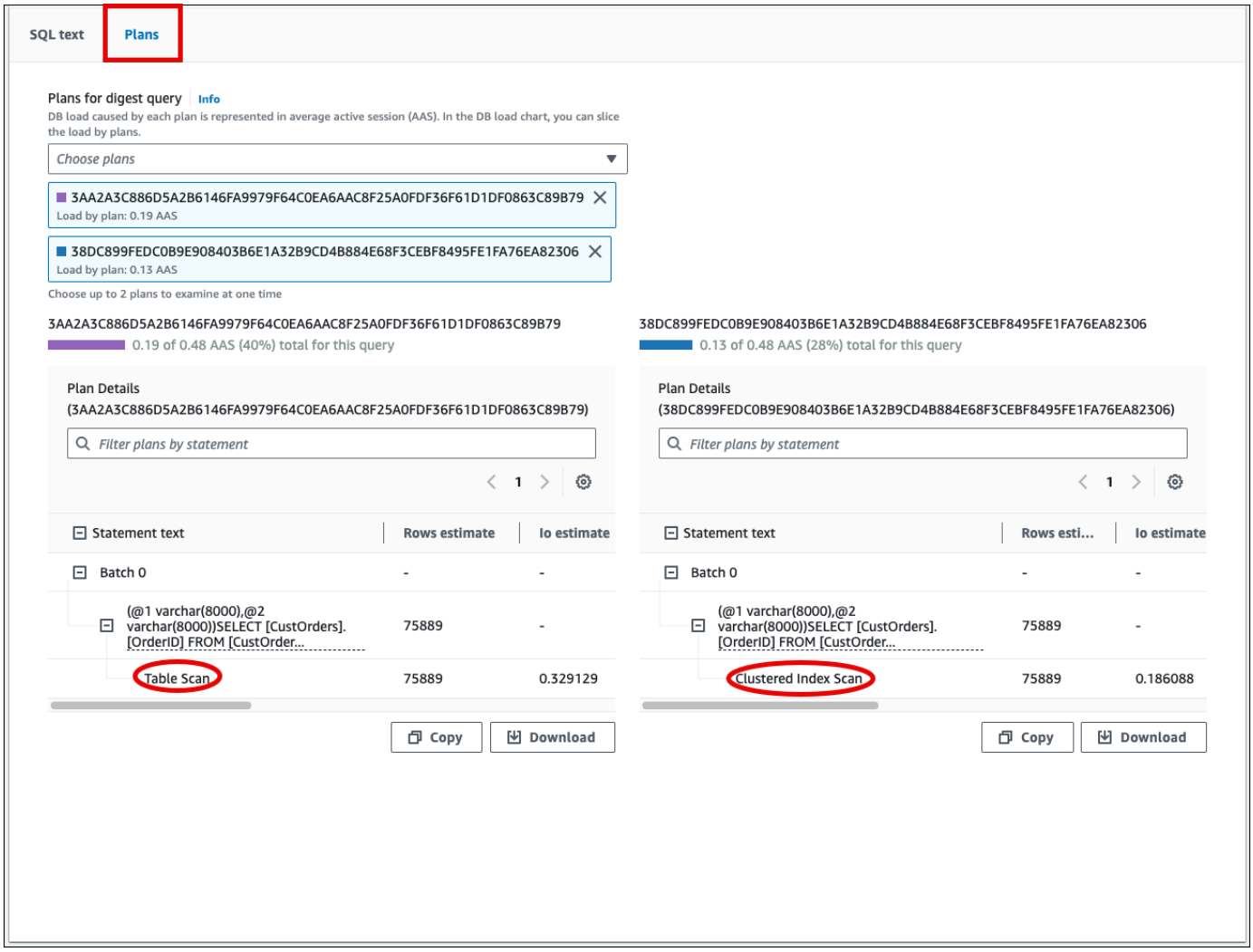
6. Wählen Sie den Digest aus, um ihn auf seine Komponentenanweisungen zu erweitern.

   Im folgenden Beispiel ist die SELECT-Anweisung eine Digest-Abfrage. Die Komponentenabfragen im Digest verwenden drei verschiedene Ausführungspläne. Die den Plänen zugewiesenen Farben entsprechen dem Ladediagramm der Datenbank.



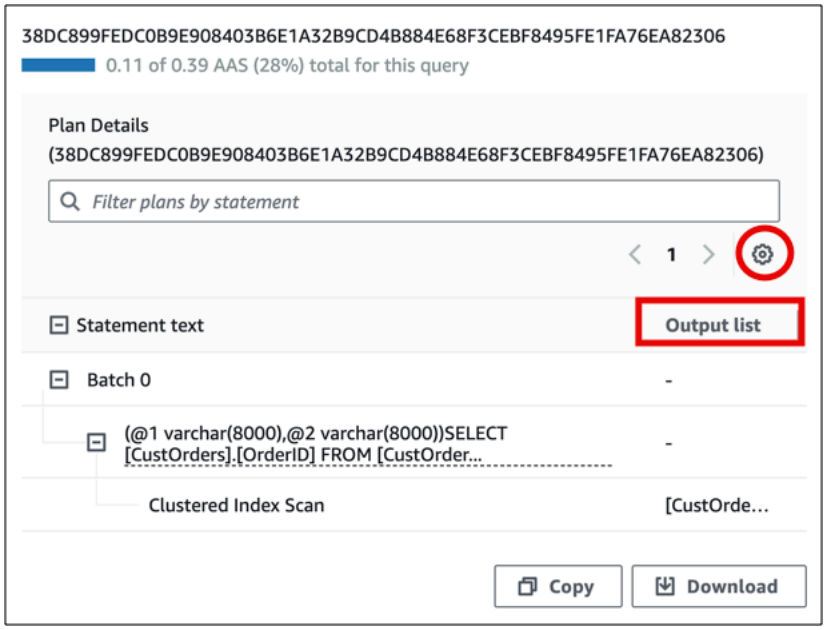
7. Blättern Sie nach unten und wählen Sie aus der Liste Plans for digest query (Pläne für Digest-Abfrage) zwei Pläne zum Vergleich aus.

   Sie können jeweils einen oder zwei Pläne für eine Abfrage anzeigen. Der folgende Screenshot vergleicht zwei Pläne im Digest. Im folgenden Beispiel verwenden 40% der durchschnittlichen aktiven Sitzungen, in denen diese Digest-Abfrage ausgeführt wird, den Plan auf der linken Seite, wohingegen 28% den Plan auf der rechten Seite verwenden.



Im vorherigen Beispiel unterscheiden sich die Pläne in einem wichtigen Punkt. Schritt 2 im Plan auf der linken Seite verwendet einen Tabellenscan, wohingegen der Plan auf der rechten Seite einen Clustered-Index-Scan verwendet. Bei einer Tabelle mit einer großen Anzahl von Zeilen ist eine Abfrage, die eine einzelne Zeile abruft, bei einem gruppierten Indexscan fast immer schneller.

8. (Optional) Wählen Sie das Einstellungssymbol in der Tabelle Plandetails, um die Sichtbarkeit und Reihenfolge der Spalten anzupassen. Der folgende Screenshot zeigt die Tabelle mit den Plandetails mit der Spalte Ausgabeliste als zweite Spalte.



9. (Optional) Wählen Sie Copy (Kopieren) aus, um den Plan in die Zwischenablage zu kopieren, oder Download (Herunterladen), um den Plan auf der Festplatte zu speichern.

> **Note**
>
> Performance Insights zeigt geschätzte Ausführungspläne anhand einer hierarchischen Baumtabelle an. Die Tabelle enthält die Teilausführungsinformationen für jede Anweisung. Weitere Informationen zu den Spalten in der Tabelle mit den Plandetails finden Sie ALL in der SQL Serverdokumentation unter SETSHOWPLAN_. Um die vollständigen Ausführungsinformationen für einen voraussichtlichen Ausführungsplan anzuzeigen, wählen Sie Herunterladen, um den Plan herunterzuladen, und laden Sie ihn dann in SQL Server Management Studio hoch. Weitere Informationen zum Anzeigen eines geschätzten Ausführungsplans mit SQL Server Management Studio finden Sie in der SQL Serverdokumentation unter Anzeige eines geschätzten Ausführungsplans.

Proaktive Empfehlungen von Performance Insights anzeigen

Amazon RDS Performance Insights überwacht bestimmte Kennzahlen und erstellt automatisch Schwellenwerte, indem analysiert wird, welche Werte für eine bestimmte Ressource potenziell problematisch sein könnten. Wenn die neuen Metrikwerte über einen bestimmten Zeitraum einen vordefinierten Schwellenwert überschreiten, generiert Performance Insights eine proaktive Empfehlung. Diese Empfehlung trägt dazu bei, future Auswirkungen auf die Datenbankleistung zu

verhindern. Um diese proaktiven Empfehlungen zu erhalten, müssen Sie Performance Insights mit einer Aufbewahrungsfrist für ein kostenpflichtiges Kontingent aktivieren.

Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten. Informationen zu Preisen und Datenspeicherung für Performance Insights finden Sie unterPreisgestaltung und Datenspeicherung für Performance Insights.

Informationen zu den Regionen, DB-Engines und Instance-Klassen, die für die proaktiven Empfehlungen unterstützt werden, finden Sie unterDB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen.

Die detaillierte Analyse und die empfohlenen Untersuchungen der proaktiven Empfehlungen finden Sie auf der Seite mit den Empfehlungsdetails.

Weitere Informationen zu Empfehlungen finden Sie unterEmpfehlungen von Amazon RDS.

Um die detaillierte Analyse einer proaktiven Empfehlung einzusehen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:

   - Wählen Sie Empfehlungen aus.

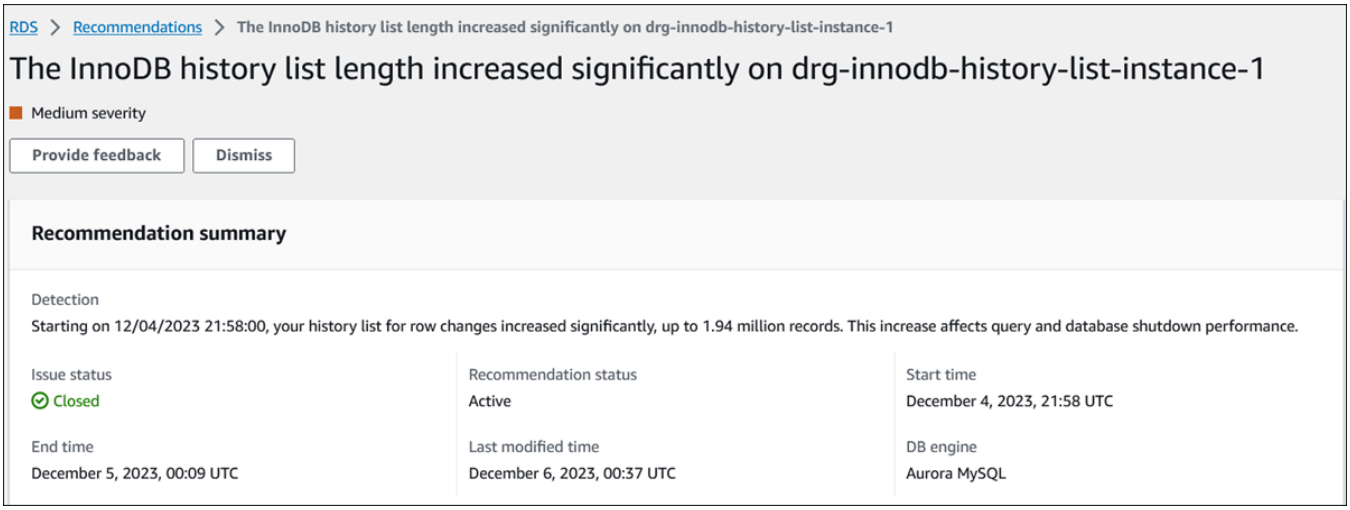
     Auf der Seite „Empfehlungen“ wird eine nach Schweregrad sortierte Liste mit Empfehlungen für alle Ressourcen in Ihrem Konto angezeigt.
   - Wählen Sie Datenbanken und dann auf der Datenbankseite Empfehlungen für eine Ressource aus.

     Auf der Registerkarte Empfehlungen werden die Empfehlungen und ihre Details für die ausgewählte Ressource angezeigt.
3. Suchen Sie nach einer proaktiven Empfehlung und wählen Sie Details anzeigen aus.

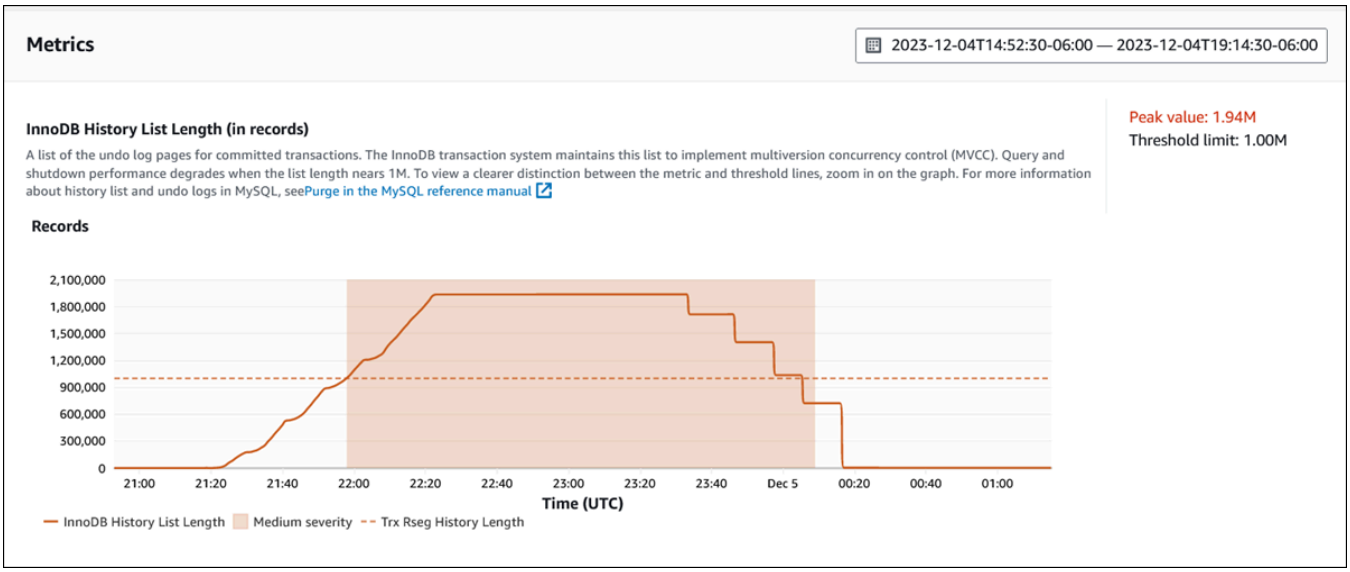
   Die Seite mit den Empfehlungsdetails wird angezeigt. Der Titel enthält den Namen der betroffenen Ressource mit dem erkannten Problem und dem Schweregrad.

   Auf der Seite mit den Empfehlungsdetails finden Sie die folgenden Komponenten:

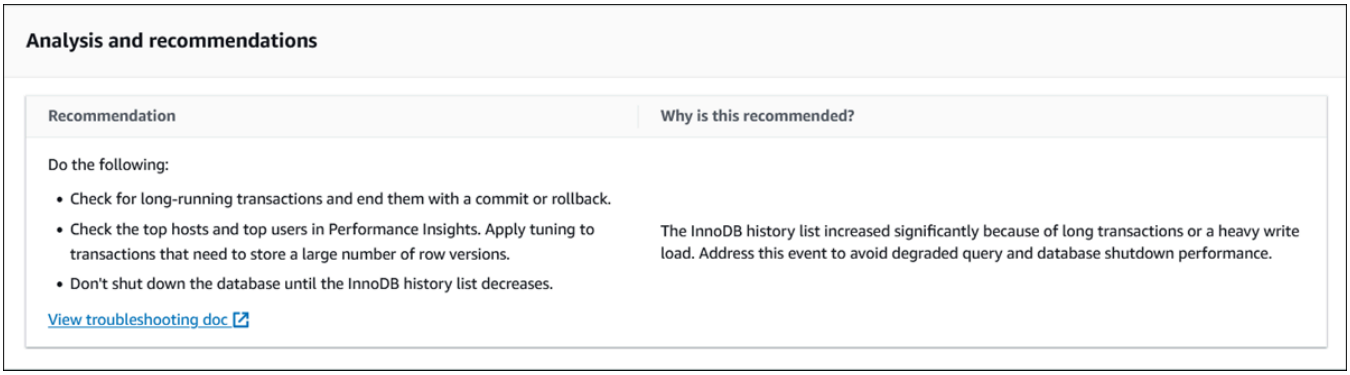
- Zusammenfassung der Empfehlungen — Das erkannte Problem, Empfehlung und Problemstatus, Start- und Endzeit des Problems, Änderungszeit der Empfehlung und Engine-Typ.



- Metriken — Die Grafiken des erkannten Problems. Jedes Diagramm zeigt einen Schwellenwert, der durch das Ausgangsverhalten der Ressource und die Daten der Metrik, die ab der Startzeit des Problems gemeldet wurden, bestimmt wird.



- Analyse und Empfehlungen — Die Empfehlung und der Grund für die vorgeschlagene Empfehlung.



Sie können die Ursache des Problems überprüfen und dann die empfohlenen Maßnahmen zur Behebung des Problems ergreifen oder oben rechts auf Ablehnen klicken, um die Empfehlung zu verwerfen.

Metriken mit Performance Insights API für Amazon abrufen RDS

Wenn Performance Insights aktiviert ist, API bietet das Einblick in die Instanzleistung. Amazon CloudWatch Logs bietet die maßgebliche Quelle für verkaufte Monitoring-Metriken für AWS Services.

Performance Insights bietet eine domänenspezifische Ansicht der Datenbanklast, gemessen als durchschnittliche aktive Sitzungen ()AAS. Diese Metrik erscheint API Verbrauchern als zweidimensionaler Zeitreihendatensatz. Die Zeitdimension der Daten stellt die Datenbanklastdaten für jeden Zeitpunkt im abgefragten Zeitraum bereit. Für jeden Zeitpunkt wird die Gesamtlast bezogen auf die angeforderten Dimensionen zerlegt, z. B. `SQL`, `Wait-event`, `User` oder `Host`, gemessen zum betreffenden Zeitpunkt.

Amazon RDS Performance Insights überwacht den Amazon DB-Instance, sodass Sie die Datenbankleistung analysieren und Fehler beheben können. Eine Möglichkeit zum Anzeigen von Performance Insights-Daten bietet die AWS Management Console. Performance Insights bietet auch eine öffentliche API Ansicht, sodass Sie Ihre eigenen Daten abfragen können. Sie können das verwendenAPI, um Folgendes zu tun:

- Auslagern von Daten in eine Datenbank
- Hinzufügen von Performance Insights-Daten zu bestehenden Überwachungs-Dashboards
- Entwickeln von Überwachungstools

Um Performance Insights zu verwendenAPI, aktivieren Sie Performance Insights auf einer Ihrer Amazon RDS DB-Instances. Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten. Weitere Informationen zu Performance Insights API finden Sie in der Amazon RDS Performance Insights API Insights-Referenz.

Performance Insights API bietet die folgenden Operationen.

Performance-Insights-Aktion	AWS CLI Befehl	Beschreibung
CreatePerformanceAnalysisReport	aws pi create-performance-analysis-report	Erstellt einen Leistungsanalysebericht für die DB-Instance für einen bestimmten Zeitraum. Das Ergebnis lautet `AnalysisReportId` . Dies ist der eindeutige Bezeichner des Berichts.
DeletePerformanceAnalysisReport	aws pi delete-performance-analysis-report	Löscht einen Leistungsanalysebericht.
DescribeDimensionKeys	aws pi describe-dimension-keys	Ruft die Schlüssel der Top N-Dimension für eine Metrik für einen bestimmten Zeitraum ab.
GetDimensionKeyDetails	aws pi get-dimension-key-details	Ruft die Attribute der angegebenen Dimensionsgruppe für eine DB-Instance oder Datenquelle ab. Wenn Sie beispielsweise eine SQL ID angeben und die Dimensionsdetails verfügbar sind, wird der vollständige Text der Dimension `GetDimensionKeyDetails` abgerufen, die dieser

Performance-Insights-Aktion	AWS CLI Befehl	Beschreibung
		ID `db.sql.statement` zugeordnet ist. Dieser Vorgang ist nützlich, weil er `GetResourceMetrics` das Abrufen umfangrei cher SQL Anweisungstexte `DescribeDimensionK eys` nicht unterstützt.
`GetPerformanceAnal ysisReport`	`aws pi get-perf ormance-analysis-r eport`	Ruft den Bericht einschlie ßlich der Erkenntnisse für den Bericht ab. Das Ergebnis umfasst den Berichtsstatus, die Berichts-ID, Details zum Berichtszeitpunkt, Erkenntni sse und Empfehlungen.
`GetResourceMetadata`	`aws pi get-re source-metadata`	Rufen Sie die Metadaten für verschiedene Funktionen ab. Die Metadaten könnten beispielsweise darauf hindeuten, dass eine Funktion für eine bestimmte DB-Instan ce ein- oder ausgeschaltet ist.
`GetResourceMetrics`	`aws pi get-res ource-metrics`	Ruft Performance Insights-Metriken für eine Reihe von Datenquellen über einen Zeitraum ab. Sie können spezifische Dimension sgruppen und Dimensionen bereitstellen und Aggregati on und Filterkriterien für jede Gruppe bereitstellen.

Performance-Insights-Aktion	AWS CLI Befehl	Beschreibung
ListAvailableResourceDimensions	aws pi list-available-resource-dimensions	Rufen Sie die Dimensionen ab, die für jeden angegebenen Metriktyp für eine bestimmte Instance abgefragt werden können.
ListAvailableResourceMetrics	aws pi list-available-resource-metrics	Rufen Sie alle verfügbaren Metriken der angegebenen Metriktypen ab, die für eine bestimmte DB-Instance abgefragt werden können.
ListPerformanceAnalysisReports	aws pi list-performance-analysis-reports	Ruft alle Analyseberichte ab, die für die DB-Instance verfügbar sind. Die Berichte werden auf der Grundlage der Startzeit jedes Berichts aufgelistet.
ListTagsForResource	aws pi list-tags-for-resource	Listet alle Metadaten-Tags auf, die der Ressource hinzugefügt wurden. Die Liste enthält den Namen und den Wert des Tags.
TagResource	aws pi tag-resource	Fügt der RDS Amazon-Ressource Metadaten-Tags hinzu. Das Tag enthält einen Namen und einen Wert.
UntagResource	aws pi untag-resource	Entfernt die Metadaten-Tags von der Ressource.

Weitere Informationen zum Abrufen von Zeitreihenmetriken und AWS CLI Beispiele für Performance Insights finden Sie in den folgenden Themen.

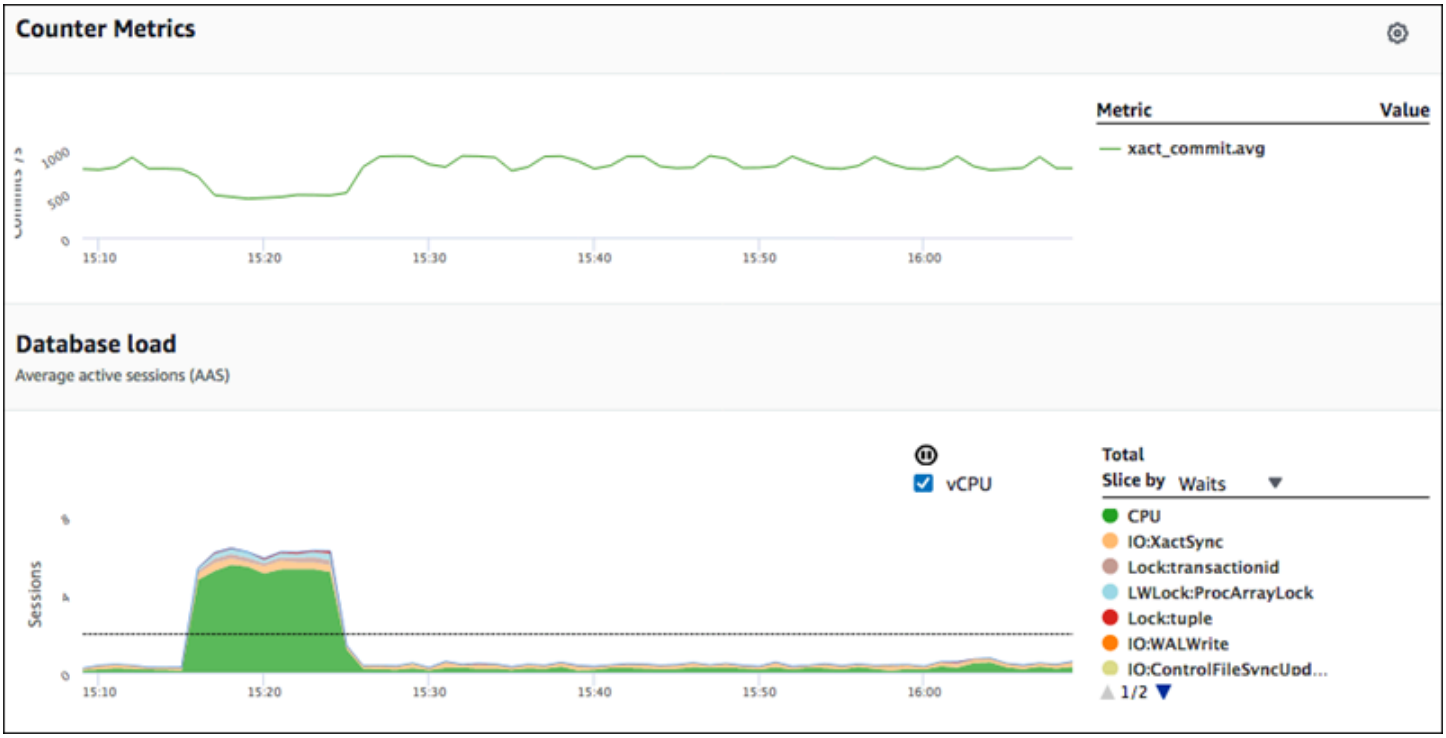
Themen

- Abrufen von Zeitreihenmetriken für Performance Insights
- AWS CLI Beispiele für Performance Insights

Abrufen von Zeitreihenmetriken für Performance Insights

Mit der `GetResourceMetrics`-Operation werden ein oder mehrere Zeitreihenmetriken aus den Performance Insights-Daten abgerufen. Für `GetResourceMetrics` ist eine Metrik und ein Zeitraum erforderlich, damit eine Antwort mit einer Liste von Datenpunkten zurückgegeben wird.

Dies ist beispielsweise der Fall, AWS Management Console `GetResourceMetrics` um das Diagramm „Zählermetriken“ und das Diagramm zum Laden der Datenbank zu füllen, wie in der folgenden Abbildung dargestellt.



Alle von zurückgegebenen Metriken`GetResourceMetrics` sind Standard-Zeitreihenmetriken, mit Ausnahme von`db.load`. Diese Metrik wird im Diagramm Database Load (Datenbanklast) angezeigt. Die`db.load` Metrik unterscheidet sich von den anderen Zeitreihenmetriken, da Sie sie in Unterkomponenten aufteilen können, die als Dimensionen bezeichnet werden. In der vorherigen Abbildung wird `db.load` unterteilt und nach Wartezuständen gruppiert, aus denen `db.load` besteht.

> **Note**
>
> `GetResourceMetrics` kann auch die `db.sampleload`-Metrik zurückgeben, aber die `db.load`-Metrik ist in den meisten Fällen angemessen.

Informationen zu den Zählermetriken, die von `GetResourceMetrics` zurückgegeben werden, finden Sie unter Performance-Insights-Zählermetriken.

Die folgenden Berechnungen werden für die Metriken unterstützt:

- Durchschnitt – Der durchschnittliche Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.avg` an.
- Minimum – Der minimale Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.min` an.
- Maximum – Der maximale Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.max` an.
- Summe – Die Summe der Metrikwerte über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.sum` an.
- Beispielanzahl – Die Anzahl, wie oft die Metrik über einen bestimmten Zeitraum erfasst wurde. Fügen Sie dem Metriknamen `.sample_count` an.

Nehmen wir an, dass eine Metrik beispielsweise 300 Sekunden (5 Minuten) lang erfasst wird und dass die Metrik einmal pro Minute erfasst wird. Die Werte für jede Minute sind 1, 2, 3, 4 und 5. In diesem Fall werden die folgenden Berechnungen zurückgegeben:

- Durchschnitt – 3
- Minimum – 1
- Maximum – 5
- Summe – 15
- Beispielanzahl – 5

Hinweise zur Verwendung des `get-resource-metrics` AWS CLI Befehls finden Sie unter `get-resource-metrics`.

Geben Sie für die `--metric-queries`-Option eine oder mehrere Abfragen an, um die entsprechenden Ergebnisse zu erhalten. Jede Abfrage besteht aus einem obligatorischen `Metric`- sowie optionalen `GroupBy`- und `Filter`-Parametern. Es folgt ein Beispiel für eine Spezifikation der `--metric-queries`-Option.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

AWS CLI Beispiele für Performance Insights

In den folgenden Abschnitten erfahren Sie mehr über das AWS Command Line Interface (AWS CLI) für Performance Insights und AWS CLI Anwendungsbeispiele.

Themen

- Integrierte Hilfe AWS CLI für Performance Insights
- Abrufen von Zählermetriken
- Abrufen des DB-Lastdurchschnitts für Top-Warteereignisse
- Der durchschnittliche DB-Load-Durchschnitt für Top wird abgerufen SQL
- Der DB-Lastdurchschnitt wird abgerufen, gefiltert nach SQL
- Der vollständige Text einer SQL Anweisung wird abgerufen
- Erstellen eines Leistungsanalyseberichts für einen bestimmten Zeitraum
- Abrufen eines Leistungsanalyseberichts
- Auflisten aller Leistungsanalyseberichte für die DB-Instance
- Löschen eines Leistungsanalyseberichts
- Hinzufügen eines Tags zu einem Leistungsanalysebericht
- Auflisten aller Tags für einen Leistungsanalysebericht
- Löschen der Tags eines Leistungsanalyseberichts

Integrierte Hilfe AWS CLI für Performance Insights

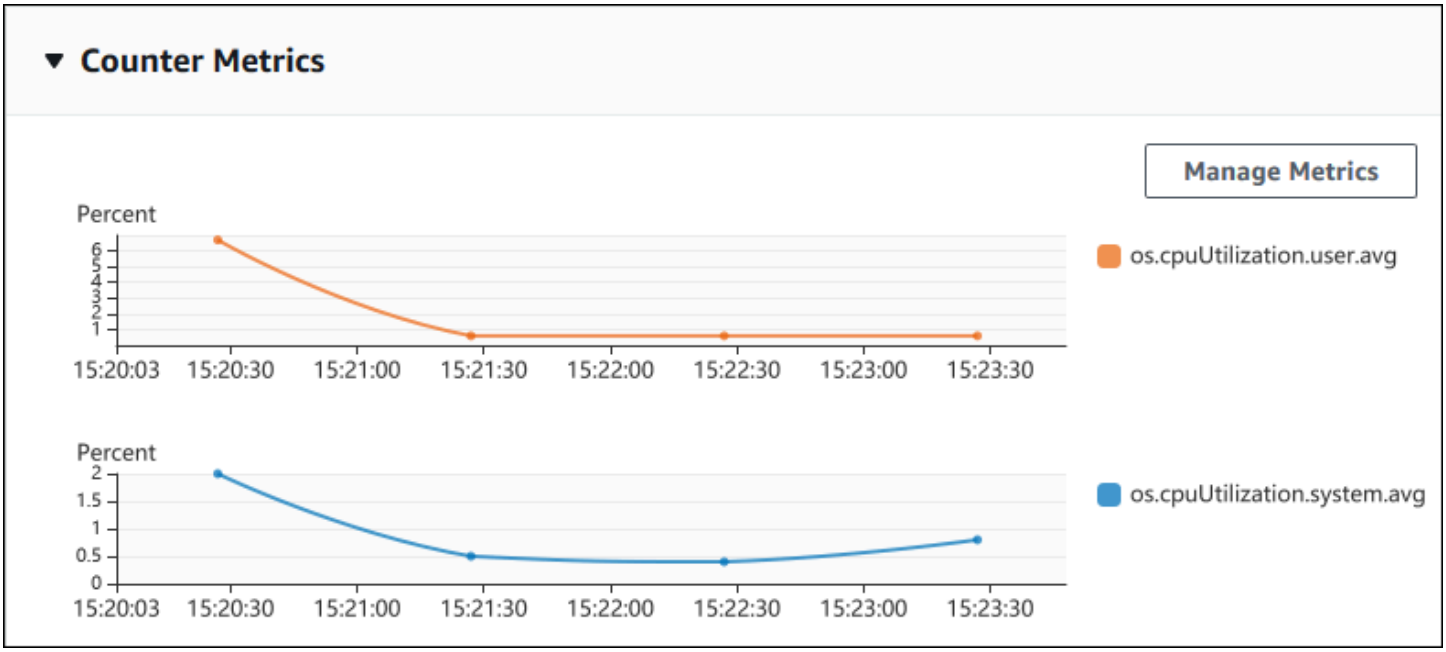
Sie können Performance Insights-Daten über die anzeige AWS CLI. Sie können die Hilfe zu den AWS CLI Befehlen für Performance Insights anzeigen, indem Sie in der Befehlszeile Folgendes eingeben.

```
aws pi help
```

Falls Sie das nicht AWS CLI installiert haben, finden Sie unter Installation von AWS CLI im AWS CLI Benutzerhandbuch weitere Informationen zur Installation.

Abrufen von Zählermetriken

Der folgende Screenshot zeigt zwei Zählermetriken-Diagramme in der AWS Management Console.



Das folgende Beispiel zeigt, wie dieselben Daten erfasst werden, die er zur Generierung der beiden Zählermetrikdiagramme AWS Management Console verwendet.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
```

```
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Sie können einen Befehl besser lesbar gestalten, indem Sie eine Datei für die Option `--metrics-query` angeben. Im folgenden Beispiel wird eine Datei namens query.json für die Option verwendet. Die Datei enthält Folgendes.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Führen Sie den folgenden Befehl aus, um die Datei zu verwenden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws pi get-resource-metrics ^
```

```
    --service-type RDS ^
    --identifier db-ID ^
    --start-time 2018-10-30T00:00:00Z ^
    --end-time   2018-10-30T01:00:00Z ^
    --period-in-seconds 60 ^
    --metric-queries file://query.json
```

Das vorige Beispiel gibt die folgenden Werte für die Optionen an:

- `--service-type`— RDS für Amazon RDS
- `--identifier` – Die Ressource-ID für die DB-Instance
- `--start-time`und `--end-time` — Die ISO 8601 `DateTime` Werte für den abzufragenden Zeitraum, mit mehreren unterstützten Formaten

Der Abfragezeitraum beträgt eine Stunde:

- `--period-in-seconds` – `60` für eine Abfrage pro Minute
- `--metric-queries` – Ein Array mit zwei Abfragen, jeweils für nur eine Metrik.

  Der Metrikname verwendet Punkte, um die Metrik in eine sinnvolle Kategorie einzustufen, wobei das letzte Element eine Funktion ist. Im Beispiel lautet die Funktion `avg` für jede Abfrage. Wie bei Amazon CloudWatch sind die unterstützten Funktionen `min`,`max`,`total`, und`avg`.

Die Antwort sieht in etwa so aus:

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                "Metric": "os.cpuUtilization.user.avg" //Metric1
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 4.0
```

```
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 10.0
                }
                //... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        },
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg" //Metric2
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 12.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 13.5
                },
                //... 60 datapoints for the os.cpuUtilization.idle.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

Die Antwort enthält Werte für `Identifier`, `AlignedStartTime` und `AlignedEndTime`. Bei einem `--period-in-seconds`-Wert von `60` wurden Start- und Endzeiten auf die Minute ausgerichtet. Wenn der `--period-in-seconds`-Wert `3600` lautet, werden Start- und Endzeiten auf die Stunde ausgerichtet.

Die `MetricList` in der Antwort enthält eine Reihe von Einträgen, und zwar jeweils mit einem `Key`- und einem `DataPoints`-Eintrag. Jeder `DataPoint` verfügt über einen `Timestamp` und einen `Value`. Jede `Datapoints`-Liste enthält 60 Datenpunkte, da die Abfragen eine Stunde lang jede Minute Daten abfragen, und zwar mit den Werten `Timestamp1/Minute1`, `Timestamp2/Minute2` usw. bis `Timestamp60/Minute60`.

Da sich die Abfrage auf zwei verschiedene Zählermetriken bezieht, enthält die -Antwort zwei Element `MetricList`.

Abrufen des DB-Lastdurchschnitts für Top-Warteereignisse

Das folgende Beispiel ist dieselbe Abfrage, die AWS Management Console verwendet wird, um ein gestapeltes Flächenliniendiagramm zu generieren. Mit diesem Beispiel wird der `db.load.avg`-Wert für die letzte Stunde abgerufen, wobei die Last auf die sieben Top-Warteereignisse aufgeteilt ist. Der Befehl ist mit dem Befehl unter identisc Abrufen von Zählermetriken. Die Datei query.json enthält hingegen Folgendes.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
    }
]
```

Führen Sie den folgenden Befehl aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

Das Beispiel gibt die Metrik `db.load.avg` und eine `GroupBy`-Sortierung der sieben Top-Warteereignisse an. Einzelheiten zu gültigen Werten für dieses Beispiel finden Sie DimensionGroupin der Performance Insights API Insights-Referenz.

Die Antwort sieht in etwa so aus:

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.5166666666666667
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.38333333333333336
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 0.26666666666666666
                }
                //... 60 datapoints for the total db.load.avg key
            ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.name": "CPU",
                    "db.wait_event.type": "CPU"
                }
            },
```

```
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.35
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.15
                },
                //... 60 datapoints for the CPU key
            ]
        },
        //... In total we have 8 key/datapoints entries, 1) total, 2-8) Top Wait Events
    ] //end of MetricList
} //end of response
```

In dieser Antwort gibt es acht Einträge in der `MetricList`. Davon bezieht sich ein Eintrag auf den `db.load.avg`-Gesamtwert und sieben Einträge jeweils auf den `db.load.avg`-Wert, der auf eines der sieben Top-Warteereignisse aufgeteilt ist. Im Gegensatz zum ersten Beispiel, bei dem eine Gruppierungsdimension vorlag, muss für jede Gruppierung der Metrik ein Schlüssel vorliegen. Für jede Metrik kann nicht nur ein Schlüssel vorhanden sein, wie im Anwendungsfall der Basiszählermetrik.

Der durchschnittliche DB-Load-Durchschnitt für Top wird abgerufen SQL

Das folgende Beispiel `db.wait_events` gruppiert nach den zehn wichtigsten SQL Aussagen. Es gibt zwei verschiedene Gruppen für SQL Aussagen:

- `db.sql`— Die vollständige SQL Erklärung, wie `select * from customers where customer_id = 123`
- `db.sql_tokenized`— Die tokenisierte SQL Aussage, wie `select * from customers where customer_id = ?`

Bei der Analyse der Datenbankleistung kann es nützlich sein, SQL Anweisungen, die sich nur durch ihre Parameter unterscheiden, als ein Logikelement zu betrachten. In diesem Fall können Sie `db.sql_tokenized` beim Abfragen verwenden. Vor allem, wenn Sie Pläne erklären möchten, ist es manchmal sinnvoller, vollständige SQL Anweisungen mit Parametern und die Gruppierung nach `db.sql` Abfragen zu untersuchen. Es besteht eine Eltern-Kind-Beziehung zwischen tokenized und fullSQL, wobei mehrere vollständige SQL (untergeordnete) Elemente unter demselben tokenisierten Element (übergeordnet) gruppiert werden. SQL

Der Befehl in diesem Beispiel ähnelt dem Befehl unter Abrufen des DB-Lastdurchschnitts für Top-Warteereignisse. Die Datei query.json enthält hingegen Folgendes.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }
    }
]
```

Im folgenden Beispiel wird verwende `db.sql_tokenized`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-29T00:00:00Z \
   --end-time   2018-10-30T00:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-29T00:00:00Z ^
   --end-time   2018-10-30T00:00:00Z  ^
   --period-in-seconds 3600 ^
   --metric-queries file://query.json
```

In diesem Beispiel werden Abfragen über 24 Stunden mit einer Stunde abgefragt period-in-seconds.

Das Beispiel gibt die Metrik `db.load.avg` und eine `GroupBy`-Sortierung der sieben Top-Warteereignisse an. Einzelheiten zu gültigen Werten für dieses Beispiel finden Sie DimensionGroupin der Performance Insights API Insights-Referenz.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": 1540771200.0,
```

```
    "AlignedEndTime": 1540857600.0,
    "Identifier": "db-XXX",

    "MetricList": [ //11 entries in the MetricList
        {
            "Key": { //First key is total
                "Metric": "db.load.avg"
            }
            "DataPoints": [ //Each DataPoints list has 24 per-hour Timestamps and a
 value
                {
                    "Value": 1.6964980544747081,
                    "Timestamp": 1540774800.0
                },
                //... 24 datapoints
            ]
        },
        {
            "Key": { //Next key is the top tokenized SQL
                "Dimensions": {
                    "db.sql_tokenized.statement": "INSERT INTO authors (id,name,email)
 VALUES\n( nextval(?)  ,?,?)",
                    "db.sql_tokenized.db_id": "pi-2372568224",
                    "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE"
                },
                "Metric": "db.load.avg"
            },
            "DataPoints": [ //... 24 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized SQL, 1 total key
    ] //End of MetricList
} //End of response
```

Diese Antwort enthält 11 Einträge `MetricList` (insgesamt 1, 10 topisierte EinträgeSQL), wobei jeder Eintrag 24 Einträge pro Stunde hat. `DataPoints`

Bei tokenized SQL gibt es drei Einträge in jeder Dimensionsliste:

- `db.sql_tokenized.statement`— Die tokenisierte AussageSQL.
- `db.sql_tokenized.db_id` — Entweder die native Datenbank-ID, mit der auf die verwiesen wirdSQL, oder eine synthetische ID, die Performance Insights für Sie generiert, falls die native

Datenbank-ID nicht verfügbar ist. In diesem Beispiel wird die synthetische ID `pi-2372568224` zurückgegeben.

- `db.sql_tokenized.id` – Die ID der Abfrage innerhalb von Performance-Insights.

  In der AWS Management Console wird diese ID als Support-ID bezeichnet. Es trägt diesen Namen, weil es sich bei der ID um Daten handelt, die der AWS Support untersuchen kann, um Ihnen bei der Behebung eines Problems mit Ihrer Datenbank zu helfen. AWS nimmt die Sicherheit und den Datenschutz Ihrer Daten sehr ernst und fast alle Daten werden mit Ihrem AWS KMS Schlüssel verschlüsselt gespeichert. Daher AWS kann niemand im Inneren diese Daten einsehen. Im vorherigen Beispiel wird sowohl `tokenized.statement` als auch `tokenized.db_id` verschlüsselt gespeichert. Wenn Sie ein Problem mit Ihrer Datenbank haben, kann Ihnen der AWS Support unter Angabe der Support-ID weiterhelfen.

Beim Abfragen empfiehlt es sich ggf., eine `Group` in `GroupBy` anzugeben. Für eine präzisere Kontrolle der Daten, die zurückgegeben werden, sollten Sie allerdings die Dimensionsliste angeben. Wenn z. B. lediglich eine `db.sql_tokenized.statement` erforderlich ist, kann der query.json-Datei ein `Dimensions`-Attribut hinzugefügt werden.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.sql_tokenized",
            "Dimensions":["db.sql_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Der DB-Lastdurchschnitt wird abgerufen, gefiltert nach SQL



Die vorherige Abbildung zeigt, dass eine bestimmte Abfrage ausgewählt ist, und das Stapelflächendiagramm der durchschnittlichen aktiven Top-Sitzungen ist auf diese Abfrage beschränkt. Obwohl sich die Abfrage nach wie vor auf die sieben Top-Gesamtwarteereignisse bezieht, wird der Wert der Antwort gefiltert. Durch das Filtern werden nur die Sitzungen berücksichtigt, die mit dem entsprechenden Filter übereinstimmen.

Die entsprechende API Abfrage in diesem Beispiel ähnelt dem Befehl inDer durchschnittliche DB-Load-Durchschnitt für Top wird abgerufen SQL. Die Datei query.json enthält hingegen Folgendes.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 5  },
        "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
]
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws pi get-resource-metrics \
```

```
    --service-type RDS \
    --identifier db-ID \
    --start-time 2018-10-30T00:00:00Z \
    --end-time   2018-10-30T01:00:00Z \
    --period-in-seconds 60 \
    --metric-queries file://query.json
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws pi get-resource-metrics ^
    --service-type RDS ^
    --identifier db-ID ^
    --start-time 2018-10-30T00:00:00Z ^
    --end-time   2018-10-30T01:00:00Z ^
    --period-in-seconds 60 ^
    --metric-queries file://query.json
```

Die Antwort sieht in etwa so aus:

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1556215200.0,
    "MetricList": [
        {
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.4878117913832196
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.192823803967328
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
```

```
                    "db.wait_event.name": "wait/io/aurora_redo_log_flush"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.1360544217687074
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.058051341890315
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/table/sql/handler"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.16241496598639457
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05163360560093349
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
                }
            },
            "DataPoints": [
                {
```

```
                    "Timestamp": 1556218800.0,
                    "Value": 0.11479591836734694
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.013127187864644107
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "CPU",
                    "db.wait_event.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.05215419501133787
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05805134189031505
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/lock_wait_mutex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.017573696145124718
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.002333722287047841
```

```
                }
            ]
        }
    ],
    "AlignedEndTime": 1556222400.0
} //end of response
```

In dieser Antwort werden alle Werte nach dem Beitrag von tokenized gefiltert, der in der Datei query.json SQL AKIAIOSFODNN7EXAMPLE angegeben ist. Die Schlüssel haben möglicherweise auch eine andere Reihenfolge als eine Abfrage ohne Filter, da sich die fünf wichtigsten Warteereignisse auf die gefilterten Ereignisse ausgewirkt haben. SQL

Der vollständige Text einer SQL Anweisung wird abgerufen

Im folgenden Beispiel wird der vollständige Text einer SQL Anweisung für eine DB-Instance abgerufen. `db-10BCD2EFGHIJ3KL4M5NO6PQRS5` `--group` ist `db.sql` und `--group-identifier` ist `db.sql.id`. In diesem Beispiel *`my-sql-id`* steht dies für eine SQL ID, die durch Aufrufen `pi get-resource-metrics` von oder abgerufen wurde. `pi describe-dimension-keys`

Führen Sie den folgenden Befehl aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws pi get-dimension-key-details \
   --service-type RDS \
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
   --group db.sql \
   --group-identifier my-sql-id \
   --requested-dimensions statement
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws pi get-dimension-key-details ^
   --service-type RDS ^
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 ^
   --group db.sql ^
   --group-identifier my-sql-id ^
   --requested-dimensions statement
```

In diesem Beispiel sind die Dimensionsdetails verfügbar. Somit ruft Performance Insights den vollständigen Text der SQL Aussage ab, ohne ihn zu kürzen.

```
{
    "Dimensions":[
    {
        "Value": "SELECT e.last_name, d.department_name FROM employees e, departments d
 WHERE e.department_id=d.department_id",
        "Dimension": "db.sql.statement",
        "Status": "AVAILABLE"
    },
    ...
    ]
}
```

Erstellen eines Leistungsanalyseberichts für einen bestimmten Zeitraum

Im folgenden Beispiel wird ein Leistungsanalysebericht mit der Startzeit `1682969503` und der Endzeit `1682979503` für die`db-loadtest-0`-Datenbank erstellt.

```
aws pi create-performance-analysis-report \
        --service-type RDS \
        --identifier db-loadtest-0 \
        --start-time 1682969503 \
        --end-time 1682979503 \
        --region us-west-2
```

Die Antwort ist der eindeutige Bezeichner `report-0234d3ed98e28fb17` für den Bericht.

```
{
   "AnalysisReportId": "report-0234d3ed98e28fb17"
}
```

Abrufen eines Leistungsanalyseberichts

Im folgenden Beispiel werden die Details des Analyseberichts für den Bericht `report-0d99cc91c4422ee61` abgerufen.

```
aws pi get-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--region us-west-2
```

Die Antwort enthält den Berichtsstatus, die ID, Zeitdetails und Einblicke.

```
        {
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-loadtest-0",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
           ... (Condensed for space)
        ]
    }
}
```

Auflisten aller Leistungsanalyseberichte für die DB-Instance

Das folgende Beispiel listet alle verfügbaren Leistungsanalyseberichte für die `db-loadtest-0`-Datenbank auf.

```
aws pi list-performance-analysis-reports \
--service-type RDS \
--identifier db-loadtest-0 \
--region us-west-2
```

In der Antwort werden alle Berichte mit der Berichts-ID, dem Status und den Details zum Zeitraum aufgeführt.

```
{
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreationTime": 1680587087.139,
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
```

```
            "EndTime": 1681491137.914,
            "CreationTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreationTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
            "EndTime": 1682979503.0,
            "CreationTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
        }
    ]
}
```

Löschen eines Leistungsanalyseberichts

Im folgenden Beispiel wird der Analysebericht für die `db-loadtest-0`-Datenbank gelöscht.

```
aws pi delete-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--region us-west-2
```

Hinzufügen eines Tags zu einem Leistungsanalysebericht

Im folgenden Beispiel wird ein Tag mit dem Schlüssel `name` und dem Wert `test-tag` zum Bericht `report-01ad15f9b88bcbd56` hinzugefügt.

```
aws pi tag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tags Key=name,Value=test-tag \
```

```
--region us-west-2
```

Auflisten aller Tags für einen Leistungsanalysebericht

Das folgende Beispiel listet alle Tags für den Bericht `report-01ad15f9b88bcbd56` auf.

```
aws pi list-tags-for-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--region us-west-2
```

In der Antwort werden der Wert und der Schlüssel für alle dem Bericht hinzugefügten Tags aufgeführt:

```
{
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

Löschen der Tags eines Leistungsanalyseberichts

Im folgenden Beispiel wird das Tag `name` aus dem Bericht `report-01ad15f9b88bcbd56` gelöscht.

```
aws pi untag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tag-keys name \
--region us-west-2
```

Nachdem das Tag gelöscht wurde, wird dieses Tag beim Aufrufen von `list-tags-for-resource` API nicht aufgeführt.

Protokollieren von Performance Insights Insights-Aufrufen mit AWS CloudTrail

Performance Insights wird mit einem Dienst ausgeführt AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in Performance Insights ausgeführt wurden. CloudTrailerfasst alle API Aufrufe von Performance Insights als Ereignisse. Diese Erfassung umfasst Aufrufe von der RDS Amazon-Konsole und von Codeaufrufen an die Performance Insights API Insights-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Performance Insights. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von CloudTrail gesammelten Daten können Sie bestimmte Informationen ermitteln. Diese Informationen umfassen die Anforderung an Performance Insights, die IP-Adresse, von der die Anforderung gesendet wurde, den Initiator sowie den Zeitpunkt der Anforderung. Sie enthalten auch weitere Angaben.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Arbeiten mit Performance Insights Insights-Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn in Performance Insights eine CloudTrail Aktivität auftritt, wird diese Aktivität zusammen mit anderen AWS Dienstereignissen in der CloudTrail Konsole im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Performance Insights, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen AWS Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail -Benutzerhandbuch:

- Übersicht zum Erstellen eines Trails

- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von SNS Amazon-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle Performance Insights Insights-Operationen werden von der Performance Insights API Insights-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe der `GetResourceMetrics` Operationen `DescribeDimensionKeys` und Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder IAM-Benutzeranmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie im CloudTrail userIdentityElement.

Performance Insights-Protokolldateieinträge

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar. Jedes Ereignis enthält Informationen über den angeforderten Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrailProtokolldateien sind kein geordneter Stack-Trace der öffentlichen API Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der den `GetResourceMetrics` Vorgang demonstriert.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
```

```
          "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2019-12-18T19:28:46Z",
    "eventSource": "pi.amazonaws.com",
    "eventName": "GetResourceMetrics",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.67",
    "userAgent": "aws-cli/1.16.240 Python/3.7.4 Darwin/18.7.0 botocore/1.12.230",
    "requestParameters": {
        "identifier": "db-YTDU5J5V66X7CXSCVDFD2V3SZM",
        "metricQueries": [
            {
                "metric": "os.cpuUtilization.user.avg"
            },
            {
                "metric": "os.cpuUtilization.idle.avg"
            }
        ],
        "startTime": "Dec 18, 2019 5:28:46 PM",
        "periodInSeconds": 60,
        "endTime": "Dec 18, 2019 7:28:46 PM",
        "serviceType": "RDS"
    },
    "responseElements": null,
    "requestID": "9ffbe15c-96b5-4fe6-bed9-9fccff1a0525",
    "eventID": "08908de0-2431-4e2e-ba7b-f5424f908433",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


Performance Insights API und VPC Schnittstellenendpunkte ()AWS PrivateLink

Sie können AWS PrivateLink es verwenden, um eine private Verbindung zwischen Ihnen VPC und Amazon RDS Performance Insights herzustellen. Sie können auf Performance Insights zugreifen, als ob es in Ihrem wäreVPC, ohne ein Internet-Gateway, ein NAT Gerät, eine Verbindung oder AWS Direct Connect eine VPN Verbindung verwenden zu müssen. Instances in Ihrem VPC System benötigen keine öffentlichen IP-Adressen, um auf Performance Insights zuzugreifen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Dabei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Einstiegspunkt für Traffic dienen, der für Performance Insights bestimmt ist.

Weitere Informationen finden Sie AWS PrivateLink im Handbuch unter Access AWS-Services through.AWS PrivateLink

Überlegungen zu Performance Insights

Bevor Sie einen Schnittstellenendpunkt für Performance Insights einrichten, lesen Sie die Überlegungen im AWS PrivateLink Handbuch.

Performance Insights unterstützt das Aufrufen all seiner API Aktionen über den Schnittstellenendpunkt.

Standardmäßig ist der vollständige Zugriff auf Performance Insights über den Schnittstellenendpunkt zulässig. Um den Datenverkehr zu Performance Insights über den Schnittstellenendpunkt zu steuern, ordnen Sie den Endpunkt-Netzwerkschnittstellen eine Sicherheitsgruppe zu.

Verfügbarkeit

Performance Insights unterstützt API derzeit VPC Endgeräte AWS-Regionen , die Performance Insights unterstützen. Informationen zur Verfügbarkeit von Performance Insights finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.

Erstellen Sie einen Schnittstellenendpunkt für Performance Insights

Sie können einen Schnittstellenendpunkt für Performance Insights entweder mit der VPC Amazon-Konsole oder mit AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für Performance Insights mit dem folgenden Dienstnamen:

Wenn Sie privat DNS für den Schnittstellenendpunkt aktivieren, können Sie API Anfragen an Performance Insights unter Verwendung DNS des regionalen Standardnamens stellen. Beispiel, `pi.us-east-1.amazonaws.com`.

Eine VPC Endpunktrichtlinie für Performance Insights erstellen API

Eine Endpunktrichtlinie ist eine IAM Ressource, die Sie an einen Schnittstellenendpunkt anhängen können. Die standardmäßige Endpunktrichtlinie ermöglicht den vollen Zugriff auf Performance Insights über den Schnittstellenendpunkt. Um den Zugriff auf Performance Insights von Ihrem aus zu kontrollierenVPC, fügen Sie dem Schnittstellenendpunkt eine benutzerdefinierte Endpunktrichtlinie hinzu.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Principals, die Aktionen ausführen können (AWS-Konten, IAM Benutzer und IAM Rollen).
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuern des Zugriffs auf Services mit Endpunktrichtlinien im AWS PrivateLink -Leitfaden.

Beispiel: VPC Endpunktrichtlinie für Performance Insights Insights-Aktionen

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Endpunktrichtlinie. Wenn Sie diese Richtlinie an Ihren Schnittstellenendpunkt anhängen, gewährt sie allen Principals auf allen Ressourcen Zugriff auf die aufgelisteten Performance Insights Insights-Aktionen.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "rds:CreatePerformanceAnalysisReport",
            "rds:DeletePerformanceAnalysisReport",
            "rds:GetPerformanceAnalysisReport"
         ],
         "Resource":"*"
      }
   ]
}
```

Beispiel: VPC Endpunktrichtlinie, die jeglichen Zugriff von einem bestimmten Konto aus verweigert AWS

Die folgende VPC Endpunktrichtlinie verweigert dem AWS Konto 123456789012 jeglichen Zugriff auf Ressourcen, die den Endpunkt verwenden. Die Richtlinie erlaubt alle Aktionen von anderen Konten.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
}
```

IP-Adressierung für Performance Insights

IP-Adressen ermöglichen es Ressourcen in Ihrem VPC System, miteinander und mit Ressourcen über das Internet zu kommunizieren. Performance Insights unterstützt beide IPv4 und IPv6 Adressierungsprotokolle. Standardmäßig VPC verwenden Performance Insights und Amazon das IPv4 Adressierungsprotokoll. Sie können dieses Standardverhalten nicht deaktivieren. Achten Sie beim Erstellen eines daraufVPC, einen IPv4 CIDR Block (einen Bereich von privaten IPv4 Adressen) anzugeben.

Sie können optional Ihren VPC und Ihren Subnetzen einen IPv6 CIDR Block zuweisen und IPv6 Adressen aus diesem Block RDS Ressourcen in Ihrem Subnetz zuweisen. Die Support des IPv6 Protokolls erweitert die Anzahl der unterstützten IP-Adressen. Durch die Verwendung des IPv6 Protokolls stellen Sie sicher, dass Sie über ausreichend verfügbare Adressen für das future Wachstum des Internets verfügen. Neue und bestehende RDS Ressourcen können verwendet werden IPv4 und IPv6 Adressen in IhremVPC. Das Konfigurieren, Sichern und Übersetzen des Netzwerkverkehrs zwischen den beiden Protokollen, die in verschiedenen Teilen einer Anwendung verwendet werden, können den Betriebsaufwand erhöhen. Sie können das IPv6 Protokoll für RDS Amazon-Ressourcen standardisieren, um Ihre Netzwerkkonfiguration zu vereinfachen. Weitere

Informationen zu Service-Endpunkten und Kontingenten finden Sie unter Amazon Relational Database Service Endpoints and Quotas.

Weitere Informationen zur Amazon RDS finden Sie unter Amazon RDS IP-Adressierung.

Analysieren von Leistungsanomalien mit Amazon DevOps Guru für Amazon RDS

Amazon DevOps Guru ist ein vollständig verwalteter Betriebsservice, der Entwicklern und Betreibern hilft, die Leistung und Verfügbarkeit ihrer Anwendungen zu verbessern. DevOpsGuru überträgt Ihnen die Aufgaben im Zusammenhang mit der Identifizierung betrieblicher Probleme, sodass Sie schnell Empfehlungen zur Verbesserung Ihrer Anwendung umsetzen können. Weitere Informationen finden Sie unter Was ist Amazon DevOps Guru? im Amazon DevOps Guru-Benutzerhandbuch.

DevOpsGuru erkennt, analysiert und gibt Empfehlungen für bestehende Betriebsprobleme für alle Amazon RDS DB-Engines. DevOpsGuru for RDS erweitert diese Funktion, indem es maschinelles Lernen auf Performance Insights Insights-Metriken für for SQL Postgre-Datenbanken anwendet. Diese Überwachungsfunktionen ermöglichen es DevOps Guru fürRDS, Leistungsengpässe zu erkennen und zu diagnostizieren und spezifische Korrekturmaßnahmen zu empfehlen. DevOpsGuru for RDS kann auch problematische Bedingungen in Ihrer für SQL Postgre-Datenbank erkennen, bevor sie auftreten.

Sie können sich diese Empfehlungen jetzt in der RDS Konsole ansehen. Weitere Informationen finden Sie unter Empfehlungen von Amazon RDS.

Das folgende Video gibt einen Überblick über DevOps Guru forRDS.

Einen tiefen Einblick in dieses Thema finden Sie unter Amazon DevOps Guru RDS unter der Haube.

Themen

- Vorteile von DevOps Guru für RDS
- Wie RDS funktioniert DevOps Guru for
- Guru einrichten für DevOps RDS

Vorteile von DevOps Guru für RDS

Wenn Sie für for SQL Postgre-Datenbank verantwortlich sind, wissen Sie möglicherweise nicht, dass ein Ereignis oder eine Regression stattfindet, die sich auf diese Datenbank auswirkt. Wenn Sie von dem Problem erfahren, wissen Sie möglicherweise nicht, warum es auftritt und was Sie dagegen tun können. Anstatt sich an einen Datenbankadministrator (DBA) zu wenden, um Hilfe zu erhalten oder sich auf Tools von Drittanbietern zu verlassen, können Sie den Empfehlungen von DevOps Guru für folgen. RDS

Die detaillierte Analyse von DevOps Guru bietet Ihnen folgende Vorteile fürRDS:

Schnelle Diagnose

DevOpsGuru for überwacht und analysiert RDS kontinuierlich die Datenbanktelemetrie. Performance Insights, Enhanced Monitoring und Amazon CloudWatch sammeln Telemetriedaten für Ihre . DevOpsGuru for RDS verwendet statistische Techniken und Techniken des maschinellen Lernens, um diese Daten zu analysieren und Anomalien zu erkennen. Weitere Informationen zu Telemetriedaten finden Sie unter Überwachen der DB-Auslastung mit Performance Insights auf Amazon RDS und Überwachen von Betriebssystemmetriken mit erweiterter Überwachung im RDSAmazon-Benutzerhandbuch.

Schnelle Auflösung

Jede Anomalie identifiziert das Leistungsproblem und schlägt Möglichkeiten für Untersuchungen oder Korrekturmaßnahmen vor. DevOpsGuru for RDS könnte Ihnen beispielsweise empfehlen, bestimmte Warteereignisse zu untersuchen. Oder es empfiehlt sich, Ihre Anwendungspooleinstellungen zu optimieren, um die Anzahl der Datenbankverbindungen zu begrenzen. Basierend auf diesen Empfehlungen können Sie Leistungsprobleme schneller beheben als durch eine manuelle Fehlerbehebung.

Proaktive Einblicke

DevOpsGuru for RDS verwendet Metriken aus Ihren Ressourcen, um potenziell problematisches Verhalten zu erkennen, bevor es zu einem größeren Problem wird. Es kann beispielsweise erkennen, wann Ihre Datenbank eine zunehmende Anzahl von temporären Tabellen auf der Festplatte verwendet, was negative Auswirkungen auf die Leistung zur Folge haben kann. DevOpsGuru gibt Ihnen dann Empfehlungen, mit denen Sie Probleme lösen können, bevor sie zu größeren Problemen werden.

Fundierte Kenntnisse der Amazon-Ingenieure und Machine Learning

Um Leistungsprobleme zu erkennen und Ihnen bei der Behebung von Engpässen zu helfen, RDS setzt DevOps Guru for auf maschinelles Lernen (ML) und fortschrittliche mathematische Formeln. Die Datenbankingenieure von Amazon haben an der Entwicklung von The DevOps Guru for RDS Findings mitgewirkt, in dem viele Jahre der Verwaltung von Hunderttausenden von Datenbanken zusammengefasst sind. Durch die Nutzung dieses kollektiven Wissens RDS kann DevOps Guru for Ihnen bewährte Verfahren beibringen.

Wie RDS funktioniert DevOps Guru for

DevOpsGuru for RDS sammelt Daten über Ihre for SQL Postgre-Datenbanken von Amazon RDS Performance Insights. Die wichtigste Metrik ist`DBLoad`. DevOpsGuru for RDS nutzt die Performance Insights Insights-Metriken, analysiert sie mit maschinellem Lernen und veröffentlicht die Erkenntnisse im Dashboard.

Ein Einblick ist eine Sammlung verwandter Anomalien, die von Guru entdeckt wurden. DevOps

In DevOps Guru for ist eine Anomalie ein MusterRDS, das von dem abweicht, was als normale Leistung für Ihre for SQL Postgre-Datenbank angesehen wird.

Proaktive Einblicke

Ein proaktiver Einblick informiert Sie über problematisches Verhalten, bevor es auftritt. Es enthält Anomalien mit Empfehlungen und zugehörigen Kennzahlen, die Ihnen helfen sollen, Probleme in Ihren for SQL Postgre-Datenbanken zu beheben, bevor sie zu größeren Problemen werden. Diese Erkenntnisse werden im Guru-Dashboard veröffentlicht. DevOps

DevOpsGuru könnte beispielsweise feststellen, dass Ihre SQL Datenbank RDS für Postgre viele temporäre Tabellen auf der Festplatte erstellt. Wenn dieser Trend nicht angegangen wird, kann er zu Leistungsproblemen führen. Jeder proaktive Einblick umfasst Empfehlungen für korrigierendes Verhalten und Links zu relevanten Themen in Optimiert RDSfür Postgre SQL mit proaktiven Erkenntnissen von Amazon DevOps Guru. Weitere Informationen finden Sie unter Working with Insights in DevOps Guru im Amazon DevOps Guru-Benutzerhandbuch.

Reaktive Einblicke

Ein reaktiver Einblick identifiziert anomales Verhalten, sobald es auftritt. Wenn DevOps Guru for Leistungsprobleme in Ihren for SQL Postgre-DB-Instances RDS feststellt, veröffentlicht Guru for einen reaktiven Einblick im DevOps Guru-Dashboard. Weitere Informationen finden Sie unter Working with Insights in DevOps Guru im Amazon DevOps Guru-Benutzerhandbuch.

Kausale Anomalien

Eine kausale Anomalie ist eine Anomalie der obersten Ebene innerhalb eines Einblicks. Das Laden der Datenbank (DB-Load) ist die ursächliche Anomalie für DevOps Guru for. RDS

Eine Anomalie misst die Leistungseinbußen durch Zuweisen eines Schweregrads von Hoch, Mittel oder Niedrig. Weitere Informationen finden Sie unter Wichtige Konzepte für DevOps Guru RDS im Amazon DevOps Guru-Benutzerhandbuch.

Wenn DevOps Guru eine aktuelle Anomalie in Ihrer DB-Instance feststellt, werden Sie auf der Datenbankseite der RDS Konsole benachrichtigt. Die Konsole warnt Sie auch bei Anomalien, die in den letzten 24 Stunden aufgetreten sind. Um von der RDS Konsole aus zur Seite mit den Anomalien zu gelangen, wählen Sie den Link in der Warnmeldung. Die RDS Konsole warnt Sie auch auf der Seite für Ihren für SQL Postgre-DB-Instance.

Kontextbezogene Anomalien

Eine kontextbezogene Anomalie ist ein Befund innerhalb der Datenbanklast (DB-Last), der zu einem reaktiven Einblick gehört. Jede kontextuelle Anomalie beschreibt ein bestimmtes SQL Leistungsproblem von for Postgre, das untersucht werden muss. DevOpsGuru for RDS könnte Ihnen beispielsweise empfehlen, eine Erhöhung der CPU Kapazität in Betracht zu ziehen oder Warteereignisse zu untersuchen, die zur DB-Auslastung beitragen.

> **Important**
>
> Wir empfehlen Ihnen alle Änderungen in einer Test-Instance zu prüfen, bevor Sie eine produktive Instance ändern. Auf diese Weise verstehen Sie die Auswirkungen der Änderung.

Weitere Informationen finden Sie unter Analysieren von Anomalien bei Amazon RDS im Amazon DevOps Guru-Benutzerhandbuch.

Guru einrichten für DevOps RDS

Gehen Sie wie folgt vorRDS, damit DevOps Guru for a for SQL Postgre-Datenbank veröffentlichen kann.

Themen

- Konfiguration von IAM Zugriffsrichtlinien für DevOps Guru für RDS
- Performance Insights für Ihre for SQL Postgre-DB-Instances aktivieren
- DevOpsGuru einschalten und die Ressourcenabdeckung angeben

Konfiguration von IAM Zugriffsrichtlinien für DevOps Guru für RDS

Um Benachrichtigungen von DevOps Guru in der RDS Konsole zu sehen, müssen Sie AWS Identity and Access Management (IAM) Für den Benutzer oder die Rolle muss eine der folgenden Richtlinien gelten:

- Das Tool AWS verwaltete Richtlinie `AmazonDevOpsGuruConsoleFullAccess`
- Das Tool AWS verwaltete Richtlinie `AmazonDevOpsGuruConsoleReadOnlyAccess` und eine der folgenden Richtlinien:
  - Das Tool AWS verwaltete Richtlinie `AmazonRDSFullAccess`
  - Eine vom Kunden verwaltete Richtlinie, die `pi:GetResourceMetrics` und `pi:DescribeDimensionKeys` einschließt

Weitere Informationen finden Sie unter Konfigurieren von Zugriffsrichtlinien für Performance Insights.

Performance Insights für Ihre for SQL Postgre-DB-Instances aktivieren

DevOpsGuru for RDS verlässt sich bei seinen Daten auf Performance Insights. Ohne Performance Insights veröffentlicht DevOps Guru Anomalien, beinhaltet aber keine detaillierten Analysen und Empfehlungen.

Wenn Sie eine RDS für SQL Postgre-DB-Instance erstellen oder ändern, können Sie Performance Insights aktivieren. Weitere Informationen finden Sie unter Performance Insights für Amazon RDS ein- und ausschalten.

DevOpsGuru einschalten und die Ressourcenabdeckung angeben

Sie können DevOps Guru aktivieren, damit es Ihre for SQL Postgre-Datenbanken auf eine der folgenden Arten überwacht.

Themen

- DevOpsGuru in der Konsole RDS einschalten
- Hinzufügen von for SQL Postgre-Ressourcen in der DevOps Guru-Konsole
- Hinzufügen von for SQL Postgre-Ressourcen mit AWS CloudFormation

DevOpsGuru in der Konsole RDS einschalten

Sie können in der RDS Amazon-Konsole mehrere Wege wählen, um DevOps Guru zu aktivieren.

Themen

- DevOpsGuru einschalten, wenn Sie eine for Postgre-Datenbank erstellen SQL
- DevOpsGuru über das Benachrichtigungsbanner einschalten

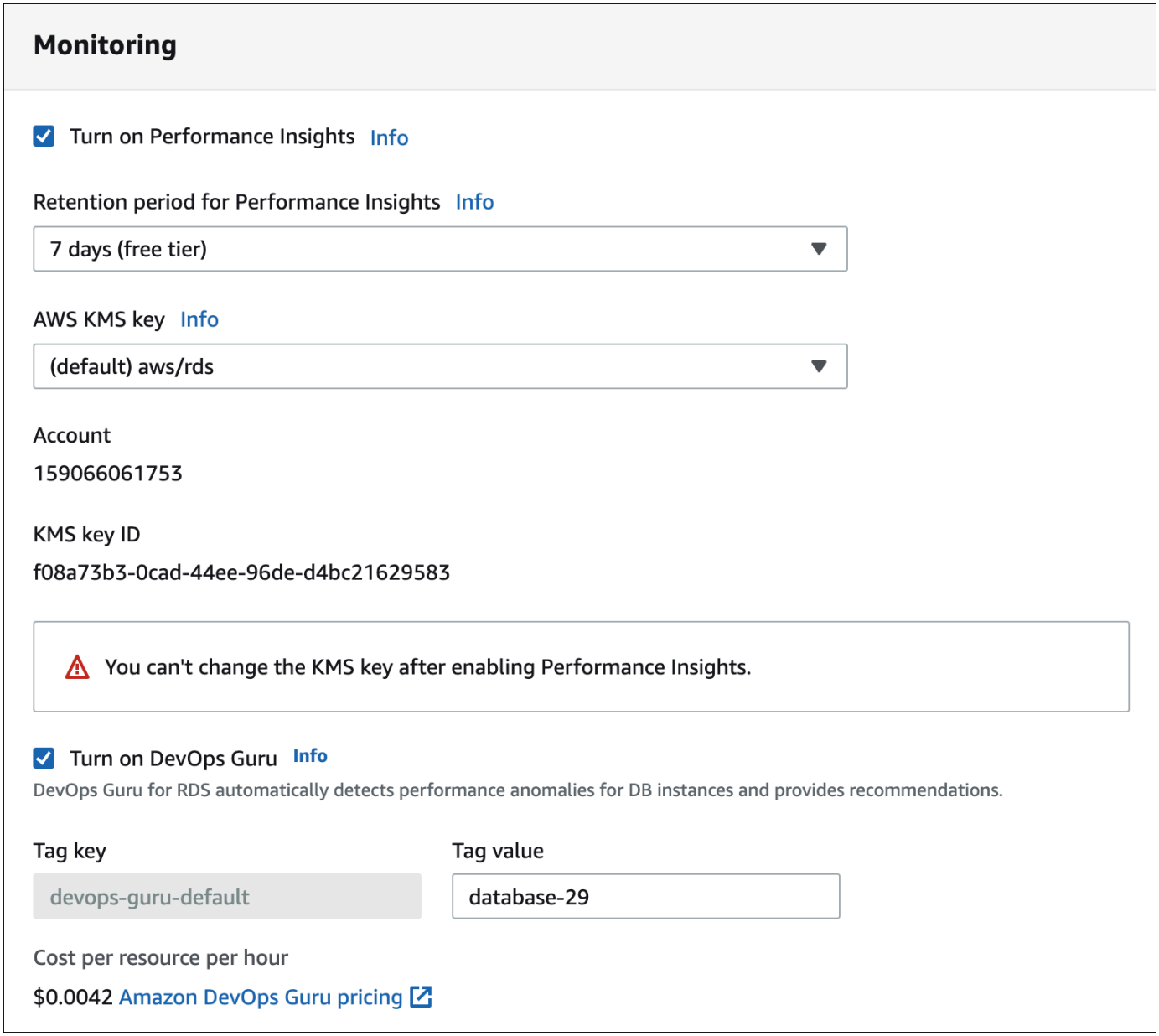
- Du reagierst auf einen Berechtigungsfehler, wenn du DevOps Guru einschaltest

DevOpsGuru einschalten, wenn Sie eine for Postgre-Datenbank erstellen SQL

Der Erstellungs-Workflow umfasst eine Einstellung, mit der die DevOps Guru-Abdeckung für Ihre Datenbank aktiviert wird. Diese Einstellung ist standardmäßig aktiviert, wenn Sie die Vorlage Production (Produktion) auswählen.

Um DevOps Guru zu aktivieren, wenn Sie eine for Postgre-Datenbank erstellen SQL

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie die Schritte unter Erstellen einer DB-Instance bis auf den Schritt aus, in dem Sie Überwachungseinstellungen wählen.
3. Wählen Sie unter Monitoring (Überwachung) die Option Turn on Performance Insights (Performance Insights aktivieren) aus. Damit DevOps Guru for RDS eine detaillierte Analyse von Leistungsanomalien durchführen kann, muss Performance Insights aktiviert sein.
4. Wähle Guru einschalten. DevOps



5. Erstellen Sie ein Tag für Ihre Datenbank, damit DevOps Guru es überwachen kann. Gehen Sie wie folgt vor:

   - Geben Sie im Textfeld für Tag key (Tag-Schlüssel) einen Namen ein, der mit **`Devops-Guru-`** beginnt.
   - Geben Sie im Textfeld für Tag value (Tag-Wert) einen beliebigen Wert ein. Wenn Sie beispielsweise den Namen Ihrer for SQL Postgre-Datenbank eingeben, können Sie ihn auch **`rds-database-1`** als Tag-Wert eingeben.

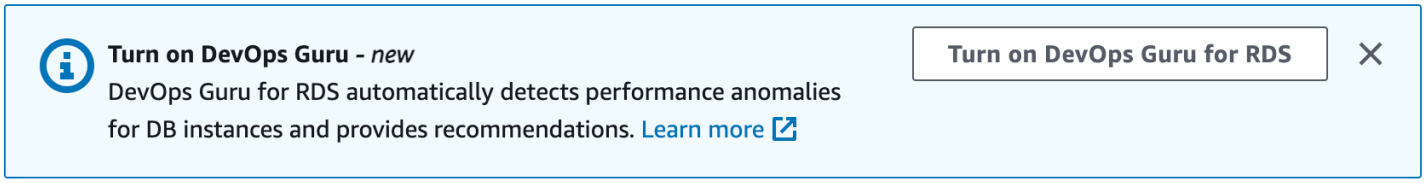
Weitere Informationen zu Tags finden Sie unter „Verwenden Sie Tags, um Ressourcen in Ihren DevOps Guru-Anwendungen zu identifizieren“ im Amazon DevOps Guru-Benutzerhandbuch.

6. Führen Sie die verbleibenden Schritte unter Erstellen einer DB-Instance aus.

DevOpsGuru über das Benachrichtigungsbanner einschalten

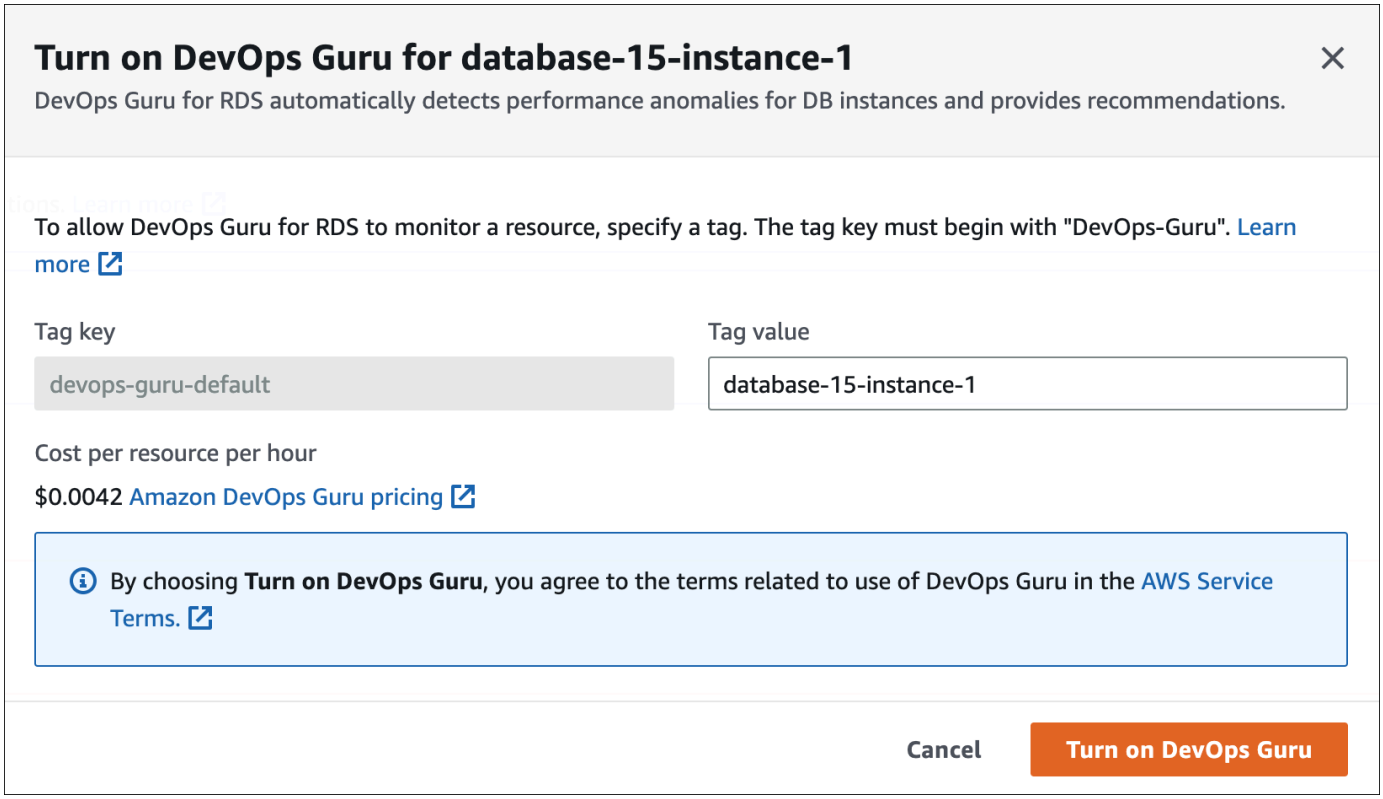
Wenn Ihre Ressourcen nicht von DevOps Guru abgedeckt werden, RDS benachrichtigt Sie Amazon mit einem Banner an den folgenden Stellen:

- Auf der Registerkarte Monitoring (Überwachung) einer DB-Cluster-Instance
- Im Performance-Insights-Dashboard



So aktivieren Sie DevOps Guru für Ihre for Postgre-Datenbank SQL

1. Wählen Sie im Banner die Option DevOpsGuru einschalten für RDS.
2. Geben Sie einen Schlüsselnamen und einen Wert für das Tag ein. Weitere Informationen zu Tags finden Sie unter „Verwenden Sie Tags, um Ressourcen in Ihren DevOps Guru-Anwendungen zu identifizieren“ im Amazon DevOps Guru-Benutzerhandbuch.



3. Wähle DevOpsGuru einschalten.

Du reagierst auf einen Berechtigungsfehler, wenn du DevOps Guru einschaltest

Wenn Sie DevOps Guru beim Erstellen einer Datenbank von der RDS Konsole aus aktivieren, wird RDS möglicherweise das folgende Banner mit fehlenden Berechtigungen angezeigt.

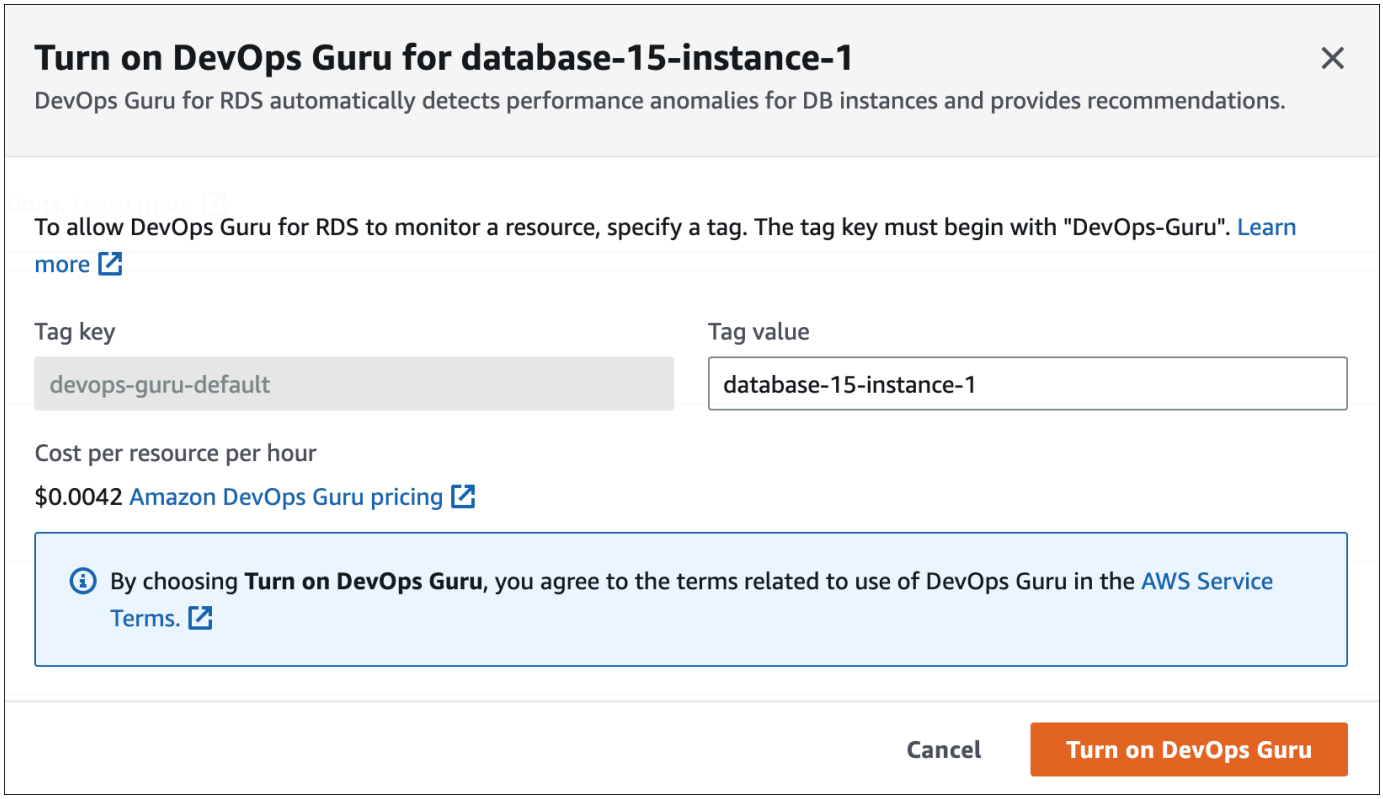


So reagieren Sie auf einen Berechtigungsfehler

1. Erteilen Sie Ihrem IAM Benutzer oder Ihrer Rolle die vom Benutzer verwaltete RolleAmazonDevOpsGuruConsoleFullAccess. Weitere Informationen finden Sie unter Konfiguration von IAM Zugriffsrichtlinien für DevOps Guru für RDS.
2. Öffnen Sie die RDS Konsole.
3. Wählen Sie im Navigationsbereich Performance-Insights aus.
4. Wählen Sie eine DB-Instance in dem Cluster aus, den Sie soeben erstellt haben.
5. Wähle den Schalter, für den DevOps Guru eingeschaltet werden sollRDS.

DevOps Guru for RDS

6. Wählen Sie einen Tag-Wert aus. Weitere Informationen finden Sie unter „Verwenden Sie Tags, um Ressourcen in Ihren DevOps Guru-Anwendungen zu identifizieren“ im Amazon DevOps Guru-Benutzerhandbuch.



7. Wählen Sie DevOpsGuru einschalten.

Hinzufügen von for SQL Postgre-Ressourcen in der DevOps Guru-Konsole

Sie können die Abdeckung Ihrer DevOps Guru-Ressourcen auf der DevOps Guru-Konsole angeben. Folgen Sie dem unter Spezifizieren Sie den Umfang Ihrer DevOps Guru-Ressourcen im Amazon DevOps Guru-Benutzerhandbuch beschriebenen Schritt. Wählen Sie eine der folgenden Optionen aus, wenn Sie Ihre analysierten Ressourcen bearbeiten:

- Wählen Sie Alle Kontoressourcen, um alle unterstützten Ressourcen, einschließlich der for SQL Postgre-Datenbanken, in Ihrem zu analysieren AWS-Konto und Region.
- Wählen Sie CloudFormation Stacks aus, um die for SQL Postgre-Datenbanken zu analysieren, die sich in Stacks Ihrer Wahl befinden. Weitere Informationen finden Sie unter Verwenden AWS CloudFormation Stapel zur Identifizierung von Ressourcen in Ihren DevOps Guru-Anwendungen im Amazon DevOps Guru-Benutzerhandbuch.

- Wählen Sie Tags, um die for SQL Postgre-Datenbanken zu analysieren, die Sie markiert haben. Weitere Informationen finden Sie unter Verwenden von Tags zur Identifizierung von Ressourcen in Ihren DevOps Guru-Anwendungen im Amazon DevOps Guru-Benutzerhandbuch.

Weitere Informationen finden Sie unter Enable DevOps Guru im Amazon DevOps Guru-Benutzerhandbuch.

Hinzufügen von for SQL Postgre-Ressourcen mit AWS CloudFormation

Sie können Tags verwenden, um Ihren CloudFormation Vorlagen die Berichterstattung über Ihre for SQL Postgre-Ressourcen hinzuzufügen. Das folgende Verfahren setzt voraus, dass Sie eine CloudFormation Vorlage sowohl für Ihre for SQLPostgre-DB-Instance als auch RDS für Ihren DevOps Guru-Stack haben.

So geben Sie eine for SQL Postgre-DB-Instance mithilfe eines Tags an CloudFormation

1. Definieren Sie in der CloudFormation Vorlage für Ihre DB-Instance ein Tag mit einem Schlüssel/Wert-Paar.

   Im folgenden Beispiel wird der Wert `my-db-instance1` `Devops-guru-cfn-default` RDS für eine SQL For-Postgre-DB-Instance zugewiesen.

   ```
   MyDBInstance1:
     Type: "AWS::RDS::DBInstance"
     Properties:
       DBInstanceIdentifier: my-db-instance1
       Tags:
         - Key: Devops-guru-cfn-default
           Value: devopsguru-my-db-instance1
   ```

2. Geben Sie in der CloudFormation Vorlage für Ihren DevOps Guru-Stack dasselbe Tag in Ihrem Ressourcensammlungsfilter an.

   Im folgenden Beispiel wird DevOps Guru so konfiguriert, dass die Ressource mit dem Tag-Wert abgedeckt wird. `my-db-instance1`

   ```
   DevOpsGuruResourceCollection:
     Type: AWS::DevOpsGuru::ResourceCollection
     Properties:
       ResourceCollectionFilter:
         Tags:
   ```

```
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "devopsguru-my-db-instance1"
```

Das folgende Beispiel deckt alle Ressourcen innerhalb der Anwendungsgrenze `Devops-guru-cfn-default` ab.

```
DevOpsGuruResourceCollection:
  Type: AWS::DevOpsGuru::ResourceCollection
  Properties:
    ResourceCollectionFilter:
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "*"
```

Weitere Informationen finden Sie unter AWS:::DevOpsGuru: ResourceCollection und: AWS:::RDS: DBInstance in der AWS CloudFormation Benutzerleitfaden.

Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung)

Mit „Enhanced Monitoring“ (Erweiterte Überwachung) können Sie das Betriebssystem Ihrer DB-Instance in Echtzeit überwachen. Wenn Sie sehen möchten, wie verschiedene Prozesse oder Threads die CPU verwenden, sind Enhanced Monitoring-Metriken nützlich.

Themen

- Überblick über „Enhanced Monitoring“ (Erweiterte Überwachung)
- Einrichten und Aktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung)
- Betriebssystemmetriken in der RDS Konsole anzeigen
- Betriebssystemmetriken mithilfe von CloudWatch Protokollen anzeigen

Überblick über „Enhanced Monitoring“ (Erweiterte Überwachung)

Amazon RDS stellt in Echtzeit Metriken für das Betriebssystem bereit, auf dem Ihre DB-Instance ausgeführt wird. Sie können alle Systemmetriken und Prozessinformationen für Ihre RDS-DB-Instances in der Konsole anzeigen. Sie können verwalten, welche Metriken Sie für jede Instance überwachen möchten, und das Dashboard entsprechend Ihren Anforderungen anpassen. Beschreibungen der Metriken für Enhanced Monitoring finden Sie unter Betriebssystemmetriken in „Erweiterte Überwachung“.

RDS übermittelt die Metriken von Enhanced Monitoring an Ihr Amazon CloudWatch Logs-Konto. Sie können in CloudWatch CloudWatch Logs Metrikfilter erstellen und die Grafiken auf dem CloudWatch Dashboard anzeigen. Sie können die JSON-Ausgabe von Enhanced Monitoring aus CloudWatch Logs in einem Überwachungssystem Ihrer Wahl verwenden. Weitere Informationen finden Sie unter Enhanced Monitoring in Amazon RDS FAQs.

Themen

- Verfügbarkeit von „Enhanced Monitoring“ (Erweiterte Überwachung)
- Unterschiede zwischen CloudWatch und Enhanced Monitoring-Metriken
- Aufbewahrung von Enhanced Monitoring-Metriken
- Kosten für „Enhanced Monitoring“ (Erweiterte Überwachung)

Verfügbarkeit von „Enhanced Monitoring“ (Erweiterte Überwachung)

„Enhanced Monitoring“ (Erweiterte Überwachung) ist für die folgenden Datenbank-Engines verfügbar:

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Unterschiede zwischen CloudWatch und Enhanced Monitoring-Metriken

Ein Hypervisor erstellt virtuelle Maschinen und führt sie aus (VMs). Mithilfe eines Hypervisors kann eine Instanz mehrere Gäste unterstützen, VMs indem sie Arbeitsspeicher und CPU virtuell gemeinsam nutzt. CloudWatch sammelt Metriken zur CPU-Auslastung vom Hypervisor für eine DB-Instance. Im Gegensatz dazu sammelt „Enhanced Monitoring“ (Erweiterte Überwachung) seine Metriken von einem Agenten auf der DB-Instance.

Möglicherweise stellen Sie Unterschiede zwischen den Messungen CloudWatch und Enhanced Monitoring fest, da die Hypervisor-Ebene nur wenig Arbeit leistet. Die Unterschiede können größer sein, wenn Ihre DB-Instances kleinere Instance-Klassen verwenden. In diesem Szenario werden wahrscheinlich mehr virtuelle Maschinen (VMs) von der Hypervisor-Ebene auf einer einzigen physischen Instanz verwaltet.

Beschreibungen der Metriken für Enhanced Monitoring finden Sie unter Betriebssystemmetriken in „Erweiterte Überwachung“. Weitere Informationen zu CloudWatch Metriken finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Aufbewahrung von Enhanced Monitoring-Metriken

Standardmäßig werden die Metriken von Enhanced Monitoring 30 Tage lang in den CloudWatch Protokollen gespeichert. Dieser Aufbewahrungszeitraum unterscheidet sich von typischen CloudWatch Kennzahlen.

Um zu ändern, wie lange die Metriken in den CloudWatch Protokollen gespeichert werden, ändern Sie die Aufbewahrungsdauer für die `RDSOSMetrics` Protokollgruppe in der CloudWatch Konsole.

Weitere Informationen finden Sie unter Ändern der Aufbewahrung von Protokolldaten in CloudWatch Protokollen im Amazon CloudWatch Logs-Benutzerhandbuch.

Kosten für „Enhanced Monitoring“ (Erweiterte Überwachung)

Metriken zur erweiterten Überwachung werden in den CloudWatch Protokollen und nicht in CloudWatch Metriken gespeichert. Die Kosten für „Enhanced Monitoring“ (Erweiterte Überwachung) hängen von folgenden Faktoren ab:

- Enhanced Monitoring wird Ihnen nur in Rechnung gestellt, wenn Sie das von Amazon CloudWatch Logs bereitgestellte Datenübertragungs- und Speichervolumen überschreiten. Die Gebühren basieren auf den Gebühren für die Übertragung und Speicherung von CloudWatch Logs.
- Die Menge der für eine RDS-Instance übertragenen Informationen ist direkt proportional zur definierten Granularität für „Enhanced Monitoring“ (Erweiterte Überwachung). Ein kürzeres Überwachungsintervall führt zu häufigeren Berichten über Betriebssystem-Metriken und erhöht Ihre Überwachungskosten. Um Kosten zu verwalten, legen Sie unterschiedliche Granularitäten für verschiedene Instances in Ihren Konten fest.
- Nutzungskosten für „Enhanced Monitoring“ (Erweiterte Überwachung) werden auf jede DB-Instance angewendet, für die Enhanced Monitoring (Erweiterte Überwachung) aktiviert ist. Die Überwachung einer großen Zahl von DB-Instances ist kostspieliger als die Überwachung von nur wenigen Instances.
- DB-Instances, die eine rechenintensivere Workload unterstützen, müssen mehr Aktivitäten von Betriebssystemprozessen melden und bedeuten höhere Kosten für „Enhanced Monitoring“ (Erweiterte Überwachung).

Weitere Informationen zur Preisgestaltung finden Sie unter CloudWatch Amazon-Preise.

Einrichten und Aktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung)

Um „Enhanced Monitoring“ (Erweiterte Überwachung) zu verwenden, müssen Sie eine IAM-Rolle erstellen und dann „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren.

Themen

- So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung)
- Aktivieren und Deaktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung)

- Schutz vor dem Confused-Deputy-Problem

So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung)

Für Enhanced Monitoring ist die Erlaubnis erforderlich, in Ihrem Namen handeln zu dürfen, um Betriebssystem-Metrikinformationen an CloudWatch Logs zu senden. Sie gewähren Enhanced Monitoring-Berechtigungen mithilfe einer AWS Identity and Access Management (IAM-) Rolle. Sie können diese Rolle entweder erstellen, wenn Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren oder vorher erstellen.

Themen

- Erstellen der IAM-Rolle, wenn Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren
- Erstellen der IAM-Rolle, bevor Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren

Erstellen der IAM-Rolle, wenn Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren

Wenn Sie „Enhanced Monitoring“ (Erweiterte Überwachung) in der RDS-Konsole aktivieren, kann Amazon RDS die erforderliche IAM-Rolle für Sie erstellen. Der Name der Rolle lautet `rds-monitoring-role`. RDS verwendet diese Rolle für die angegebene DB-Instance, das Lesereplikat oder den Multi-AZ-DB-Cluster.

So erstellen Sie die IAM-Rolle beim Aktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung)

1. Führen Sie die Schritte unter au Aktivieren und Deaktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung).
2. Setzen Sie die Überwachungsrolle in dem Schritt, in dem Sie eine Rolle auswählen auf Standard.

Erstellen der IAM-Rolle, bevor Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren

Sie können die erforderliche Rolle erstellen, bevor Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren. Wenn Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren, geben Sie den Namen Ihrer neuen Rolle an. Sie müssen diese erforderliche Rolle erstellen, wenn Sie „Enhanced Monitoring“ (Erweiterte Überwachung) mithilfe der AWS CLI oder RDS API aktivieren.

Der Benutzer, der „Enhanced Monitoring“ (Erweiterte Überwachung) aktiviert, muss über die `PassRole`-Berechtigung verfügen. Weitere Informationen finden Sie unter Beispiel 2 unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS Dienst im IAM-Benutzerhandbuch.

So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung) in Amazon RDS

1. Öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie die Registerkarte AWS -Service und RDS in der Liste der Services aus.
5. Wählen Sie RDS – Enhanced Monitoring (RDS – erweiterte Überwachung) und Next (Weiter) aus.
6. Vergewissern Sie sich, dass in den Berechtigungsrichtlinien Amazon angezeigt wird RDSEnhancedMonitoringRole, und wählen Sie dann Weiter aus.
7. Geben Sie unter Role name (Rollenname) einen Namen für Ihre Rolle ein. Geben Sie z. B. ei **`emaccess`**.

   Die vertrauenswürdige Entität für Ihre Rolle ist der AWS Service monitoring.rds.amazonaws.com.
8. Wählen Sie Rolle erstellen aus.

Aktivieren und Deaktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung)

Sie können Enhanced Monitoring mithilfe der, oder RDS-API verwalten. AWS Management Console AWS CLI Sie können für jede unterschiedliche Granularitäten für die Erfassung von Metriken festlegen.

Konsole

Sie können „Enhanced Monitoring“·(Erweiterte·Überwachung) aktivieren, wenn Sie eine DB-Instance, einen Multi-AZ-DB-Cluster, oder ein Lesereplikat erstellen, oder wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster ändern. Wenn Sie eine DB-Instance ändern, um Enhanced Monitoring zu aktivieren, müssen Sie Ihre DB-Instance nicht neu starten, damit die Änderung wirksam wird.

Sie können „Enhanced Monitoring“·(Erweiterte·Überwachung) in der RDS-Konsole aktivieren, wenn Sie eine der folgenden Aktionen auf der Seite Databases (Datenbanken) ausführen:

- Erstellen einer DB-Instance oder eines Multi-AZ-DB-Clusters – Wählen Sie Create database (Datenbank erstellen) aus.
- Erstellen eines Lesereplikats – Wählen Sie Actions (Aktionen) und dann Create Read Replica (Lesereplikat erstellen) aus.

- Ändern Sie eine DB-Instance, einen oder einen Multi-AZ-DB-Cluster — Wählen Sie Modify.

„Enhanced Monitoring“ (Erweiterte Überwachung) in der RDS-Konsole aktivieren/deaktivieren

1. Scrollen Sie zu Additional Configuration (Zusätzliche Konfiguration).
2. Wählen Sie unter Monitoring die Option Enable Enhanced Monitoring für Ihre DB-Instance oder Ihre Read Replica aus. Deaktivieren Sie die Option zur Deaktivierung von Enhanced Monitoring .
3. Setzen Sie die Eigenschaft Monitoring Role auf die IAM-Rolle, die Sie erstellt haben, damit Amazon RDS für Sie mit Amazon CloudWatch Logs kommunizieren kann, oder wählen Sie Standard, damit RDS eine Rolle für Sie erstellt. `rds-monitoring-role`
4. Stellen Sie die Eigenschaft Granularität auf das Intervall in Sekunden zwischen den Punkten ein, an denen Metriken für Ihre DB-Instance erfasst werden. Die Eigenschaft Granularität kann auf einen der folgenden Werte eingestellt werden: `1`, `5`, `10`, `15`, `30` oder `60`.

   Die schnellste Aktualisierung der RDS-Konsole erfolgt alle 5 Sekunden. Wenn Sie die Granularität in der RDS-Konsole auf 1 Sekunde einstellen, sehen Sie die aktualisierten Metriken dennoch nur alle 5 Sekunden. Mithilfe von Logs können Sie Metrik-Updates von einer Sekunde abrufen. CloudWatch

AWS CLI

Um Enhanced Monitoring mit den AWS CLI folgenden Befehlen zu aktivieren, setzen Sie die `--monitoring-interval` Option auf einen anderen Wert als `0` und setzen Sie die `--monitoring-role-arn` Option auf die Rolle, in So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung) der Sie sie erstellt haben.

- create-db-instance
- create-db-instance-read-replikat
- modify-db-instance
- create-db-cluster(Multi-AZ-DB-Cluster)
- modify-db-cluster(Multi-AZ-DB-Cluster)

Die Option `--monitoring-interval` gibt das Intervall in Sekunden zwischen den Punkten an, an denen Enhanced Monitoring-Metriken erfasst werden. Gültige Werte für die Option sind `0`, `1`, `5`, `10`, `15`, `30` und `60`.

Um Enhanced Monitoring mit dem zu deaktivieren AWS CLI, setzen Sie die `--monitoring-interval` Option `0` in diesen Befehlen auf.

Example

Im folgenden Beispiel wird „Enhanced Monitoring“·(Erweiterte·Überwachung) für eine DB-Instance aktiviert:

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Example

Im folgenden Beispiel wird „Enhanced Monitoring“·(Erweiterte·Überwachung) für ein Multi-AZ-DB-Cluster aktiviert:

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

RDS-API

Um „Enhanced Monitoring“ (Erweiterte Überwachung) mithilfe der RDS API zu aktivieren, setzen Sie den Parameter `MonitoringInterval` auf einen anderen Wert als `0` und legen Sie den Parameter `MonitoringRoleArn` auf die Rolle fest, die Sie in So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung) erstellt haben. Legen Sie diese Parameter in den folgenden Aktionen fest:

- CreateDBInstance
- CreateDBInstanceReadReplica
- Modifizieren SieDBInstance
- Erstellen DBCluster (Multi-AZ-DB-Cluster)
- Ändern DBCluster (Multi-AZ-DB-Cluster)

Der Parameter `MonitoringInterval` gibt das Intervall in Sekunden zwischen den Punkten an, an denen Enhanced Monitoring-Metriken erfasst werden. Gültige Werte sind: `0`, `1`, `5`, `10`, `15`, `30` und `60`.

Um „Enhanced Monitoring“ (Erweiterte Überwachung) mit Hilfe der RDS API zu deaktivieren, setzen Sie `MonitoringInterval` auf `0`.

Schutz vor dem Confused-Deputy-Problem

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben. Weitere Informationen finden Sie unter Confused-Deputy-Problem.

Um die Berechtigungen einzuschränken, die Amazon RDS einem anderen Service für eine Ressource gewährt, empfehlen wir die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in einer Vertrauensrichtlinie für Ihre Enhanced-Monitoring-Rolle. Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen diese dieselbe Konto-ID verwenden.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontextschlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Setzen Sie für Amazon RDS `aws:SourceArn` auf `arn:aws:rds:`*`Region`*`:`*`my-account-id`*`:db:`*`dbname`*.

Im folgenden Beispiel werden die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in einer Vertrauensrichtlinie verwendet, um das Confused-Deputy-Problem zu verhindern.

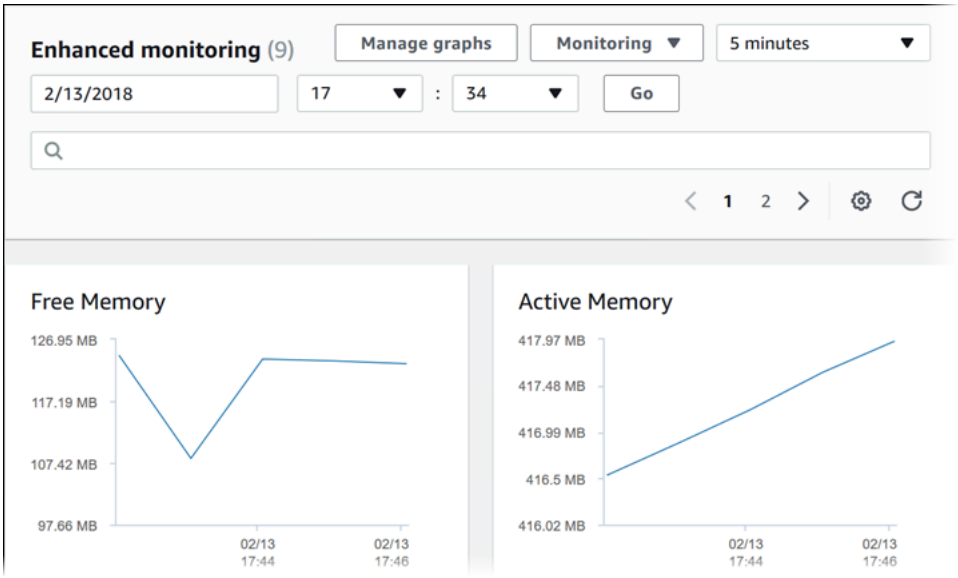
JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitoring.rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:rds:Region:my-account-id:db:dbname"
        },
        "StringEquals": {
          "aws:SourceAccount": "my-account-id"
        }
      }
    }
  ]
}
```

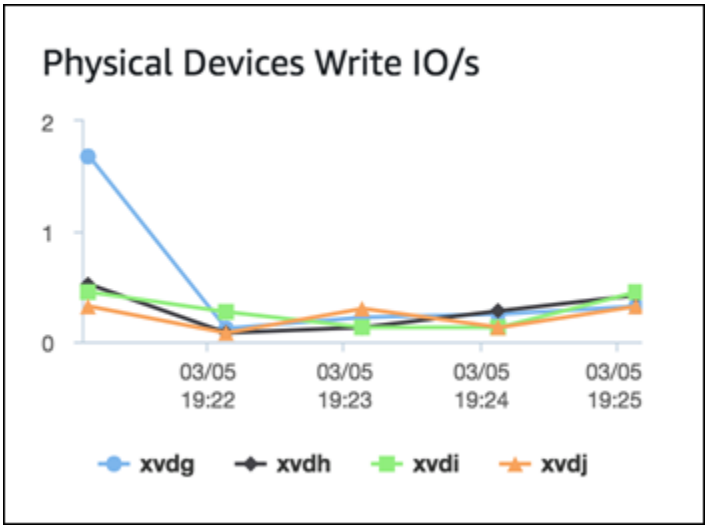
Betriebssystemmetriken in der RDS Konsole anzeigen

Sie können die von Enhanced Monitoring gemeldeten Betriebssystemmetriken in der RDS Konsole einsehen, indem Sie Enhanced Monitoring for Monitoring wählen.

Das folgende Beispiel zeigt die Seite „Enhanced Monitoring“ an. Beschreibungen der Metriken für Enhanced Monitoring finden Sie unter Betriebssystemmetriken in „Erweiterte Überwachung“.



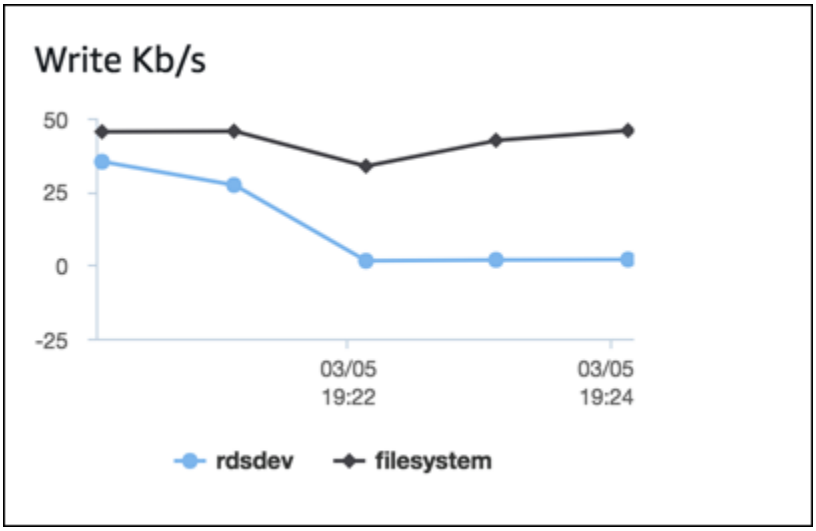
Einige DB-Instances verwenden für das Datenspeichervolume der DB-Instance mehr als einen Datenträger. Bei solchen DB-Instances zeigen die Diagramme unter Physical Devices (Physische Geräte) für jeden einzelnen der Datenträger Metriken an. Beispielsweise zeigt das folgende Diagramm Metriken für vier Datenträger an.



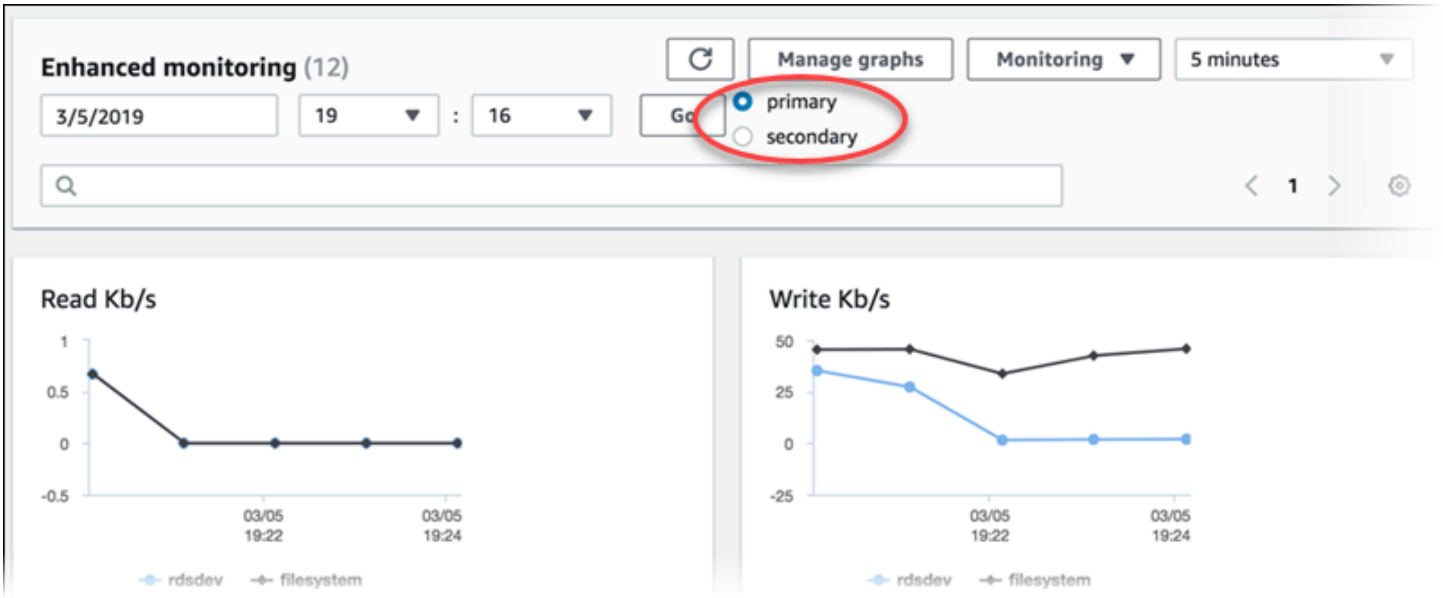
> **Note**
>
> Derzeit sind Diagramme für physische Geräte nicht für Microsoft SQL Server-DB-Instances verfügbar.

Wenn Sie aggregierte Diagramme für Disk I/O (Datenträger-I/O) und File system (Dateisystem) anzeigen, bezieht sich das Gerät rdsdev auf das `/rdsdbdata`-Dateisystem, in dem alle

Datenbankdateien und Protokolle gespeichert sind. Das Gerät unter filesystem (Dateisystem) bezieht sich auf das /-Dateisystem (auch als Stammverzeichnis oder Root bezeichnet), in dem auf das Betriebssystem bezogene Dateien gespeichert sind.



Wenn die DB-Instance eine Multi-AZ-Bereitstellung ist, können Sie die Betriebssystemmetriken für die primäre DB-Instance und ihre Multi-AZ-Standby-Replica anzeigen. Wählen Sie in der Ansicht Enhanced monitoring die Option primary (Primär) aus, um die Betriebssystemmetriken für die primäre DB-Instance anzuzeigen, oder wählen Sie secondary (Sekundär), um die OS-Metriken für die Standby-Replica anzuzeigen.

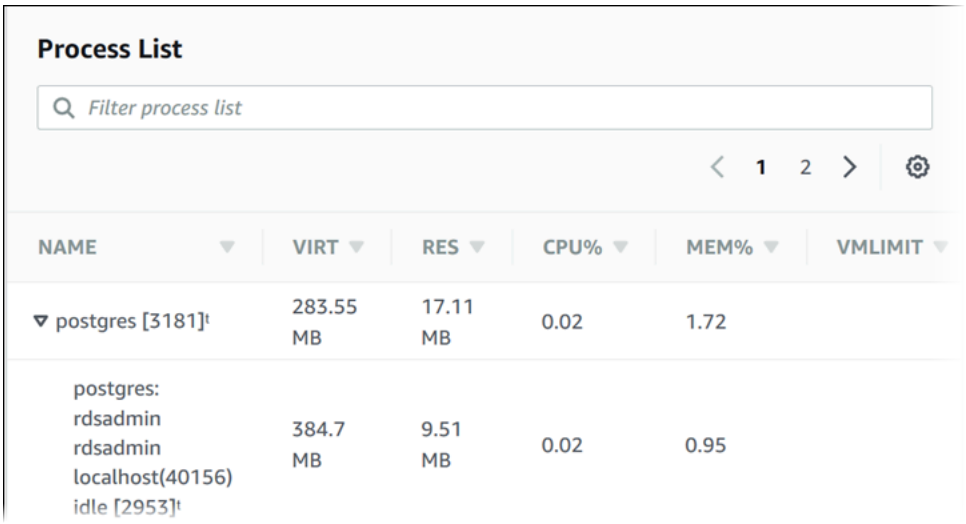


Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

> **Note**
>
> Derzeit wird die Ansicht von Betriebssystemmetriken für eine Multi-AZ-Standby-Replica für MariaDB- DB-Instances nicht unterstützt.

Wenn Sie Details für die auf Ihrer DB-Instance ausgeführten Prozesse ansehen möchten, wählen Sie die Betriebssystem-Prozessliste für Überwachung aus.

Die Ansicht Process List (Prozessliste) wird nachstehend angezeigt.



Die Enhanced Monitoring-Metriken, die in der Ansicht Process list (Prozessliste) gezeigt werden, sind wie folgt organisiert:

- RDSuntergeordnete Prozesse — Zeigt eine Zusammenfassung der RDS Prozesse an, die die DB-Instance unterstützen, zum Beispiel für My SQL DB-Instances. Prozess-Threads erscheinen unter dem übergeordneten Prozess. Prozess-Threads zeigen nur die CPU Auslastung an, da andere Metriken für alle Threads des Prozesses identisch sind. Die Konsole zeigt maximal 100 Prozesse und Threads. Die Ergebnisse sind eine Kombination der Prozesse und Threads mit dem höchsten CPU und dem höchsten Speicherverbrauch. Wenn über 50 Prozesse und über 50 Threads vorhanden sind, zeigt die Konsole für jede Kategorie die 50 mit dem höchsten Verbrauch. Anhand dieser Anzeige können Sie feststellen, welche Prozesse die größte Auswirkung auf die Performance haben.
- RDSProzesse — Zeigt eine Zusammenfassung der vom RDS Management-Agenten verwendeten Ressourcen, der Diagnosen, der Überwachungsprozesse und anderer AWS Prozesse an, die zur Unterstützung von RDS DB-Instances erforderlich sind.

- OS processes (Betriebssystemprozesse): Zeigt eine Übersicht über Kernel- und Systemprozesse, die im Allgemeinen einen geringen Einfluss auf die Performance haben.

Die aufgelisteten Elemente für jeden Prozess sind:

- VIRT— Zeigt die virtuelle Größe des Prozesses an.
- RES— Zeigt den tatsächlichen physischen Speicher an, der vom Prozess verwendet wird.
- CPU% — Zeigt den Prozentsatz der CPU Gesamtbandbreite an, der vom Prozess genutzt wird.
- MEM% — Zeigt den Prozentsatz des gesamten Speichers an, der vom Prozess genutzt wird.

Die in der RDS Konsole angezeigten Überwachungsdaten werden aus Amazon CloudWatch Logs abgerufen. Sie können die Metriken für eine DB-Instance auch als Protokollstream aus CloudWatch Logs abrufen. Weitere Informationen finden Sie unter Betriebssystemmetriken mithilfe von CloudWatch Protokollen anzeigen.

Metriken für „Enhanced Monitoring“ (Erweiterte Überwachung) werden in folgenden Situationen nicht geliefert:

- Failover der DB-Instance.
- Ändern der Instance-Klasse für die DB-Instance (Skalierung der Datenverarbeitung).

Metriken für „Enhanced Monitoring“ (Erweiterte Überwachung) werden während eines Neustarts einer DB-Instance zurückgegeben, da nur die Datenbank-Engine neu gestartet wird. Metriken für das Betriebssystem werden weiterhin mitgeteilt.

Betriebssystemmetriken mithilfe von CloudWatch Protokollen anzeigen

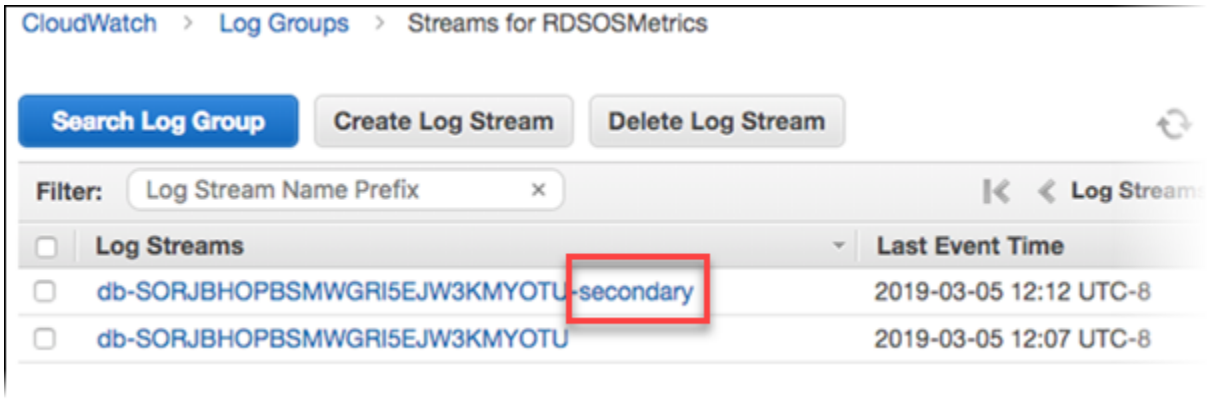
Nachdem Sie Enhanced Monitoring für Ihre DB-Instance oder Ihren aktiviert haben, können Sie die entsprechenden Metriken mithilfe von CloudWatch Logs anzeigen, wobei jeder Log-Stream eine einzelne DB-Instance oder einen einzelnen DB-Cluster darstellt, der überwacht wird. Die Protokoll-Stream-Kennung ist die Ressourcenkennung (`DbiResourceId`) für die DB-Instance oder das DB-Cluster.

So zeigen Sie Enhanced Monitoring-Protokolldaten an

1. Öffnen Sie die CloudWatch Konsole unter. https://console.aws.amazon.com/cloudwatch/

2. Wählen Sie bei Bedarf die aus, in der AWS-Region sich Ihre DB-Instance oder Ihr befindet. Weitere Informationen finden Sie unter Regionen und Endpunkte in der Allgemeinen Amazon Web Services-Referenz.
3. Wählen Sie im Navigationsbereich Protokolle.
4. Wählen Sie RDSOSMetricsaus der Liste der Protokollgruppen aus.

   In einer Multi-AZ-DB-Instance-Bereitstellung sind Protokolldateien mit an ihrem Namen angehängtem `-secondary` für die Multi-AZ-Standby-Replica bestimmt.



5. Wählen Sie aus der Liste den Protokoll-Stream, den Sie anzeigen möchten.

Metrik-Referenz für Amazon RDS

In dieser Referenz finden Sie Beschreibungen der Amazon RDS Amazon für Amazon CloudWatch, Performance Insights und Enhanced Monitoring.

Themen

- CloudWatch Amazon-Metriken für Amazon RDS
- CloudWatch Amazon-Abmessungen für Amazon RDS
- CloudWatch Amazon-Metriken für Amazon RDS Performance Insights
- Performance-Insights-Zählermetriken
- SQL-Statistiken für Performance Insights
- Betriebssystemmetriken in „Erweiterte Überwachung“

CloudWatch Amazon-Metriken für Amazon RDS

CloudWatch Amazon-Metriken bieten Einblicke in die Leistung und den Zustand von Amazon RDS-Instances und -Clustern, sodass Sie das Systemverhalten überwachen und datengestützte Entscheidungen treffen können. Diese Metriken helfen dabei, die Ressourcennutzung, die Datenbankaktivität und die betriebliche Effizienz nachzuverfolgen und bieten Einblick in die Leistung Ihrer Instances.

In dieser Referenz werden die spezifischen Metriken beschrieben, die für Amazon RDS verfügbar sind, und es wird erklärt, wie sie interpretiert und verwendet werden können, um die Datenbankleistung zu optimieren, Probleme zu beheben und eine hohe Verfügbarkeit sicherzustellen.

Amazon RDS veröffentlicht Metriken für Amazon CloudWatch in den `AWS/Usage` Namespaces `AWS/RDS` und.

Themen

- Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS
- CloudWatch Amazon-Nutzungsmetriken für Amazon RDS

Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS

Der `AWS/RDS` Namespace in Amazon CloudWatch umfasst die folgenden Metriken auf Instanzebene.

> **Note**
>
> Die Amazon RDS-Konsole zeigt möglicherweise Metriken in Einheiten an, die sich von den an Amazon gesendeten Einheiten unterscheiden CloudWatch. Beispielsweise kann die Amazon RDS-Konsole eine Metrik in Megabyte (MB) anzeigen, während die Metrik CloudWatch in Byte an Amazon gesendet wird.

Metrik	Beschreibung	Gilt für	Einheiten
`BinLogDiskUsage`	Die Menge des von Binärprotokollen belegten Speicherplatzes. Wenn automatische Backups für MySQL- und MariaDB-Instances aktiviert sind, einschließlich Lesereplikaten, werden Binärlogs erstellt.	MariaDB MySQL	Bytes
`BurstBalance`	Der Prozentsatz der verfügbaren I/O Burst-Bucket-Credits für Allzweck-SSD (GP2).	Alle	Prozent
`CheckpointLag`	Die seit dem letzten Checkpoint vergangene Zeit.		Sekunden
`ConnectionAttempts`	Die Anzahl der Versuche, eine Verbindung mit einer Instance herzustellen, unabhängig davon, ob erfolgreich oder nicht.	MySQL	Anzahl
`CPUUtilization`	Prozentsatz der CPU-Auslastung.	Alle	Prozent
`CPUCreditUsage`	Die Anzahl der von der Instance für die CPU-Nutzung verbrauchten CPU-Guthaben. Ein CPU-Guthaben entspricht einer vCPU, die eine Minute lang bei 100-prozentiger Auslastung läuft, oder einer gleichwertigen Kombinati		Guthaben (vCPU-Minuten)

Metrik	Beschreibung	Gilt für	Einheiten
	on aus vCPUs, Auslastung und Zeit. Beispielsweise können Sie eine vCPU zwei Minuten lang mit einer Auslastung von 50 Prozent oder zwei vCPU zwei Minuten lang mit einer Auslastung von 25 Prozent CPUs betreiben. Diese Metrik gilt nur für `db.t2`, `db.t3`, und `db.t4g` Instances. **Note** Wir empfehlen, die T-DB-Instance-Klassen nur für Entwicklungs- und Testserver oder andere Nicht-Produktionsserver zu verwenden. Weitere Informationen zu den T-Instance-Klassen finden Sie unter DB-Instance-Klassenarten Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar. Wenn Sie ein größeres Intervall als 5 Minuten angeben, verwenden Sie die Statistik `Sum` anstelle der Statistik `Average`.		

Metrik	Beschreibung	Gilt für	Einheiten
`CPUCredit Balance`	Die Anzahl verdienter CPU-Guthaben, die eine Instance angesammelt hat, seit sie gestartet wurde. Für T2 Standard beinhaltet `CPUCreditBalance` auch die Anzahl der angesammelten Startguthaben. Guthaben werden auf dem Guthaben-Konto angesammelt, nachdem sie verdient wurden, und davon entfernt, wenn sie verbraucht werden. Der Guthaben-Kontostand hat ein maximales Limit, das anhand der Instance-Größe bestimmt wird. Nachdem das Limit erreicht ist, verfallen alle neu verdienten Guthabenp unkte. Im Fall von T2 Standard werden Startguthaben nicht für das Limit berücksichtigt. Die Guthaben in `CPUCreditBalance` sind verfügbar, um die Leistung der Instance über die Baseline ihrer CPU-Nutzung hinaus zu steigern. Wenn eine Instance ausgeführt wird, verfallen Guthaben im `CPUCredit Balance` nicht. Wenn die Instance beendet wird, bleibt der `CPUCredit Balance` nicht erhalten, und alle angesammelten Guthaben gehen verloren. Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar.		Guthaben (vCPU-Minuten)

Metrik	Beschreibung	Gilt für	Einheiten
	Diese Metrik gilt nur für `db.t2` `db.t3` ,- und `db.t4g` Instances. **Note** Wir empfehlen, die T-DB-Instance-Klassen nur für Entwicklungs- und Testserver oder andere Nicht-Produktionsserver zu verwenden. Weitere Informationen zu den T-Instance-Klassen finden Sie unter DB-Instance-Klassenarten Startguthaben funktionieren in Amazon RDS genauso wie in Amazon EC2. Weitere Informationen dazu finden Sie unter Startguthaben im Amazon-Elastic-Compute-Cloud-Benutzerhandbuch für Linux-Instances.		

Metrik	Beschreibung	Gilt für	Einheiten
`CPUSurplusCreditBalance`	Die Anzahl überzähliger Guthaben, die von einer Unlimited-Instance verbraucht wurden, wenn ihr `CPUCreditBalance` -Wert null ist. Der `CPUSurplusCreditBalance` -Wert wird durch erworbene CPU-Guthaben abgezahlt. Wenn die Anzahl überzähliger Guthaben die Höchstzahl der Guthaben überschreitet, die die Instance in einem 24-Stunden-Zeitraum verdienen kann, fallen für die verbrauchten überzähligen Guthaben zusätzliche Gebühren an. Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar.	Alle	Guthaben (vCPU-Minuten)

Metrik	Beschreibung	Gilt für	Einheiten
`CPUSurplusCreditsCharged`	Die Anzahl verbrauchter überzähliger Guthaben, die nicht durch verdiente CPU-Guthaben zurückgezahlt wurden, und für die deshalb eine zusätzliche Gebühr anfällt. Verbrauchte überzählige Guthaben werden in Rechnung gestellt, wenn einer der folgenden Fälle auftritt: • Die ausgegebenen überzähligen Guthaben überschreiten die maximale Anzahl an Guthaben, die die Instance in einem 24-Stunden-Zeitraum verdienen kann. Über das Maximum hinaus ausgegebene überzählige Guthaben werden am Ende der Stunde abgerechnet. • Die Instance wird angehalten oder beendet. • Die Instance wird von `unlimited` in `standard` geändert. Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar.	Alle	Guthaben (vCPU-Minuten)

Metrik	Beschreibung	Gilt für	Einheiten
`DatabaseConnections`	Die Anzahl der Clientnetzwerkverbindungen zur Datenbank-Instance. Die Anzahl der Datenbanksitzungen kann höher als der Metrikwert sein, da der Metrikwert Folgendes nicht enthält: • Sitzungen, die keine Netzwerkverbindung mehr haben, die die Datenbank jedoch nicht bereinigt hat • Sitzungen, die von der Datenbank-Engine für eigene Zwecke erstellt wurden • Sitzungen, die durch die Funktionen zur parallelen Ausführung der Datenbank-Engine erstellt wurden • Vom Auftrags-Scheduler der Datenbank-Engine erstellte Sitzungen • Amazon-RDS-Verbindungen	Alle	Anzahl
`DiskQueueDepth`	Die Anzahl der ausstehenden I/Os (Lese-/Schreibanforderungen), die auf den Zugriff auf die Festplatte warten.	Alle	Anzahl
`DiskQueueDepthLogVolume`	Die Anzahl der ausstehenden I/Os (Lese-/Schreibanforderungen), die auf den Zugriff auf die Festplatte des Log-Volumes warten.	DB-Instances mit aktiviertem dediziertem Log-Volume	Anzahl

Metrik	Beschreibung	Gilt für	Einheiten
`EBSByteBalance%`	Der Prozentsatz der Durchsatz-Guthaben, die im Burst-Bucket Ihrer RDS-Datenbank verbleiben. Diese Metrik ist nur für die grundlegende Überwachung verfügbar. Der Metrikwert basiert auf dem Durchsatz aller Volumes, einschließlich des Root-Volumes, und nicht nur auf den Volumes, die Datenbankdateien enthalten. Die Instance-Größen, die diese Metrik unterstützen, finden Sie unter den mit einem Sternchen (*) markierten Instance-Größen in der standardmäßig optimierten EBS-Tabelle im EC2 Amazon-Benutzerhandbuch. Die `Sum`-Statistik ist für diese Metrik nicht anwendbar.	Alle	Prozent

Metrik	Beschreibung	Gilt für	Einheiten
`EBSIOBalance%`	Der Prozentsatz der verbleibenden I/O Credits im Burst-Bucket Ihrer RDS-Datenbank. Diese Metrik ist nur für die grundlegende Überwachung verfügbar. Der Metrikwert basiert auf den IOPS aller Volumes, einschließlich des Root-Volumes, und nicht nur auf den Volumes, die Datenbankdateien enthalten. Die Instance-Größen, die diese Metrik unterstützen, finden Sie unter Amazon EBS-optimierte Instance-Typen im EC2 Amazon-Benutzerhandbuch. Die `Sum` Statistik gilt nicht für diese Metrik. Diese Metrik unterscheidet sich von `BurstBalance` . Informationen zur Verwendung dieser Metrik finden Sie unter Verbessern der Anwendung sleistung und Senkung der Kosten mit Amazon EBS-optimierten Instance-Burst-Funktionen.	Alle	Prozent
`FailedSQLServerAgentJobsCount`	Die Anzahl der fehlgeschlagenen Microsoft SQL Server Agent-Aufträge in der letzten Minute.	Microsoft SQL Server	Anzahl pro Minute
`FreeableMemory`	Verfügbarer Arbeitsspeicher. Für MariaDB-, MySQL-, Oracle- und PostgreSQL-DB-Instances meldet diese Metrik als Wert des Feldes `MemAvailable` `/proc/meminfo` .	Alle	Bytes

Metrik	Beschreibung	Gilt für	Einheiten
`FreeLocalStorage`	Die Menge des verfügbaren lokalen Speicherplatzes. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Bytes
`FreeLocalStoragePercent`	Der Prozentsatz des verfügbaren lokalen Speicherplatzes. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Prozent
`FreeStorageSpace`	Verfügbarer Speicherplatz	Alle	Bytes

Metrik	Beschreibung	Gilt für	Einheiten
`FreeStorageSpaceLogVolume`	Der verfügbare Speicherplatz auf dem Protokoll-Volume	DB-Instances mit aktiviertem dediziertem Log-Volume	Bytes
`IamDbAuthConnectionRequests`	Die Anzahl der Verbindungsanfragen mit IAM-Authentifizierung für die DB-Instance.	Alle	Anzahl
`MaximumUsedTransactionIDs`	Die maximale Anzahl von Transaktionen IDs , die verwendet wurden.	PostgreSQL	Anzahl
`NetworkReceiveThroughput`	Eingehender Netzwerkverkehr (Receive) auf der DB-Instance, einschließlich Kundendatenbankverkehr und Amazon RDS-Datenverkehr, der zur Überwachung und Replikation verwendet wird.	Alle	Bytes pro Sekunde
`NetworkTransmitThroughput`	Ausgehender Netzwerkverkehr (Transmit) auf der DB-Instance, einschließlich Kundendatenbankverkehr und Amazon RDS-Datenverkehr, der zur Überwachung und Replikation verwendet wird.	Alle	Bytes pro Sekunde
`OldestLogicalReplicationSlotLag`	Die verzögerte Größe des Amazon RDS schreibt eine Transaktion in der Quelldatenbank fest und gibt den Zeitpunkt an, zu dem RDS die Transaktion in der Replikatdatenbank anwendet.	PostgreSQL	Bytes

Metrik	Beschreibung	Gilt für	Einheiten
`OldestReplicationSlotLag`	Der Verzögerungsgrößenwert des Replikats, das in Bezug auf die empfangenen Write-Ahead-Log-Daten (WAL) die höchste Verzögerung aufweist.	PostgreSQL	Bytes
`ReadIOPS`	Die durchschnittliche Anzahl von I/O Festplatten-Lesevorgängen pro Sekunde.	Alle	Anzahl pro Sekunde
`ReadIOPSLocalStorage`	Die durchschnittliche Anzahl von I/O Festplatten-Lesevorgängen pro Sekunde in den lokalen Speicher. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Anzahl pro Sekunde
`ReadIOPSLogVolume`	Die durchschnittliche Anzahl von I/O Festplatten-Lesevorgängen pro Sekunde für das Log-Volume.	DB-Instances mit aktiviertem dediziertem Log-Volume	Anzahl pro Sekunde
`ReadLatency`	Die durchschnittliche Zeit, die pro I/O Festplattenvorgang benötigt wird.	Alle	Sekunden

Metrik	Beschreibung	Gilt für	Einheiten
`ReadLatencyLocalStorage`	Die durchschnittliche Zeit, die pro I/O Festplattenvorgang für den lokalen Speicher benötigt wird. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Sekunden
`ReadLatencyLogVolume`	Die durchschnittliche Zeit, die pro Festplattenvorgang für das Protokoll volume benötigt wird. I/O	DB-Instances mit aktiviertem dediziertem Log-Volume	Sekunden
`ReadThroughput`	Die durchschnittliche Anzahl Byte, die pro Sekunde vom Datenträger gelesen werden.	Alle	Bytes pro Sekunde

Metrik	Beschreibung	Gilt für	Einheiten
`ReadThroughputLocalStorage`	Die durchschnittliche Anzahl von Bytes, die pro Sekunde von der Festplatte für den lokalen Speicher gelesen werden. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Bytes pro Sekunde
`ReadThroughputLogVolume`	Die durchschnittliche Anzahl von Bytes, die für das Protokoll-Volume pro Sekunde von der Festplatte gelesen werden	DB-Instances mit aktiviertem dediziertem Log-Volume	Bytes pro Sekunde
`ReplicaLag`	Bei Lesereplikat-Konfigurationen die Zeitspanne, die eine Lesereplikat-DB-Instance hinter der Quell-DB-Instance zurückbleibt. Gilt für MariaDB-, Microsoft-SQL-Server-, MySQL-, Oracle-, PostgreSQL-Lesereplikate. Bei Multi-AZ-DB-Clustern die Zeitdifferenz zwischen der letzten Transaktion auf der Writer-DB-Instance und der zuletzt angewendeten Transaktion auf einer Reader-DB-Instance.		Sekunden

Metrik	Beschreibung	Gilt für	Einheiten
`ReplicationChannelLag`	Bei Replikatkonfigurationen mit mehreren Quellen die Zeitspanne, um die ein bestimmter Kanal auf dem Multiquellen-Replikat der Quell-DB-Instance hinterherhinkt. Weitere Informationen finden Sie unter the section called “Überwachung von Replikationskanälen mit mehreren Quellen”.	MySQL	Sekunden
`ReplicationSlotDiskUsage`	Der Festplattenspeicher, der von Replikationsslotdateien verwendet wird.	PostgreSQL	Bytes
`SwapUsage`	Menge des für die DB-Instance verwendeten Auslagerungsbereichs.	MariaDB MySQL Oracle PostgreSQL	Bytes
`TempDbAvailableDataSpace`	Die Menge des verfügbaren Datenspeichers auf der tempdb und das Volume, auf dem sich tempdb befindet. Verwenden Sie diese Metrik, um die Verfügbarkeit von tempdb-Datenspeicher zu überwachen und die Kapazität entsprechend zu planen. Niedrige Werte können auf die Notwendigkeit hinweisen, den Speicherplatz zu erhöhen oder Abfragen zu optimieren, die tempdb stark nutzen.	SQL Server	Bytes

Metrik	Beschreibung	Gilt für	Einheiten
`TempDbAvailableLogSpace`	Die Menge des verfügbaren Protokollspeichers in der tempdb und das Volume, auf dem sich tempdb befindet. Verwenden Sie diese Metrik, um die Verfügbarkeit des Tempdb-Protokollspeichers zu überwachen und zu verhindern, dass das Transaktionsprotokoll vollständig ist. Kritisch für Workloads mit großen Transaktionen oder hoher Parallelität, die zu erheblicher Protokollaktivität führen.	SQL Server	Bytes
`TempDbDataFileUsage`	Der Prozentsatz der in der tempdb verwendeten Datendateien. Diese Metrik berücksichtigt nicht das potenzielle Dateiwachstum. Verwenden Sie diese Metrik, um die Nutzung von Tempdb-Datendateien zu überwachen und potenzielle Leistungsengpässe zu identifizieren. Hohe Werte können darauf hinweisen, dass Abfragen optimiert werden müssen, die große temporäre Objekte erstellen oder die Größe von tempdb erhöhen.	SQL Server	Prozent

Metrik	Beschreibung	Gilt für	Einheiten
`TempDbLogFileUsage`	Der Prozentsatz der in der tempdb verwendeten Protokolldateien. Diese Metrik berücksichtigt nicht das potenzielle Dateiwachstum. Verwenden Sie diese Metrik, um die Auslastung der tempdb-Protokolldateien zu überwachen und Leistungsprobleme zu vermeiden. Hohe Werte können auf lang andauernde Transaktionen oder übermäßige Protokollierungsaktivitäten hinweisen, die sich auf die Gesamtleistung der Datenbank auswirken könnten.	SQL Server	Prozent
`TransactionLogsDiskUsage`	Der von den Transaktionsprotokollen verwendete Festplattenspeicher.	PostgreSQL	Bytes
`TransactionLogsGeneration`	Die Größe der pro Sekunde generierten Transaktionsprotokolle.	PostgreSQL	Bytes pro Sekunde
`WriteIOPS`	Die durchschnittliche Anzahl von I/O Schreibvorgängen auf der Festplatte pro Sekunde.	Alle	Anzahl pro Sekunde

Metrik	Beschreibung	Gilt für	Einheiten
`WriteIOPS LocalStorage`	Die durchschnittliche Anzahl von I/O Schreibvorgängen auf der Festplatte pro Sekunde im lokalen Speicher. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Anzahl pro Sekunde
`WriteIOPS LogVolume`	Die durchschnittliche Anzahl von I/O Schreibvorgängen auf der Festplatte pro Sekunde für das Log-Volume.	DB-Instances mit aktiviertem dediziertem Log-Volume	Anzahl pro Sekunde
`WriteLatency`	Die durchschnittliche Zeit, die pro I/O Festplattenvorgang benötigt wird.	Alle	Sekunden

Metrik	Beschreibung	Gilt für	Einheiten
`WriteLatencyLocalStorage`	Die durchschnittliche Zeit, die pro I/O Festplattenvorgang im lokalen Speicher benötigt wird. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Sekunden
`WriteLatencyLogVolume`	Die durchschnittliche Zeit, die pro Festplattenvorgang für das Protokoll volume benötigt wird. I/O	DB-Instances mit aktiviertem dediziertem Log-Volume	Sekunden
`WriteThroughput`	Die durchschnittliche Anzahl von Bytes, die pro Sekunde auf den Datenträger geschrieben werden.	Alle	Bytes pro Sekunde
`WriteThroughputLogVolume`	Die durchschnittliche Anzahl von Bytes, die für das Protokoll-Volume pro Sekunde auf die Festplatte geschrieben werden	DB-Instances mit aktiviertem dediziertem Log-Volume	Bytes pro Sekunde

Metrik	Beschreibung	Gilt für	Einheiten
`WriteThroughputLocalStorage`	Die durchschnittliche Anzahl von Bytes, die pro Sekunde für den lokalen Speicher auf die Festplatte geschrieben werden. Diese Metrik gilt nur für DB-Instance-Klassen mit NVMe SSD-Instance-Speicher-Volumes. Informationen zu EC2 Amazon-Instances mit NVMe SSD-Instance-Speicher-Volumes finden Sie unter Instance-Speicher-Volumes. Die entsprechenden RDS-DB-Instance-Klassen haben dieselben Instance-Speicher-Volumes. Die DB-Instance-Klassen db.m6gd und db.r6gd verfügen beispielsweise über SSD-Instance-Speicher-Volumes. NVMe		Bytes pro Sekunde

CloudWatch Amazon-Nutzungsmetriken für Amazon RDS

Der `AWS/Usage` Namespace in Amazon CloudWatch enthält Nutzungsmetriken auf Kontoebene für Ihre Amazon RDS-Servicekontingenten. CloudWatch sammelt automatisch Nutzungsmetriken für alle. AWS-Regionen

Weitere Informationen finden Sie unter CloudWatch Nutzungsmetriken im CloudWatch Amazon-Benutzerhandbuch. Weitere Informationen zu Kontingenten finden Sie unter Kontingente und Beschränkungen für Amazon RDS und Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Metrik	Beschreibung	Einheiten *
`AllocatedStorage`	Der Gesamtspeicher für alle DB-Instances. In der Summe sind keine temporären Migrations-Instances enthalten.	Gigabytes

Metrik	Beschreibung	Einheiten *
AuthorizationsPerDBSecurityGroup	Die Anzahl der Eingangsregeln pro DB-Sicherheitsgruppe in Ihrer AWS-Konto. Der verwendete Wert ist die höchste Anzahl von Eingangsregeln in einer DB-Sicherheitsgruppe im Konto. Andere DB-Sicherheitsgruppen im Konto haben möglicherweise eine geringere Anzahl von Eingangsregeln.	Anzahl
CustomEndpointsPerDBCluster	Die Anzahl der benutzerdefinierten Endpunkte pro DB-Cluster in Ihrem. AWS-Konto Der verwendete Wert ist die höchste Anzahl von benutzerdefinierten Endpunkten in einem DB-Cluster im Konto. Andere DB-Cluster im Konto haben möglicherweise eine geringere Anzahl von benutzerdefinierten Endpunkten.	Anzahl
CustomEngineVersions	Die Anzahl der benutzerdefinierten Engine-Versionen (CEVs) für Amazon RDS Custom in Ihrem AWS-Konto.	Anzahl
DBClusterParameterGroups	Die Anzahl der DB-Cluster-Parametergruppen in Ihrem AWS-Konto. In der Anzahl sind keine Standardparametergruppen enthalten.	Anzahl
DBClusterRoles	Die Anzahl der zugehörigen Rollen AWS Identity and Access Management (IAM) pro DB-Cluster in Ihrem AWS-Konto. Der verwendete Wert ist die höchste Anzahl zugeordneter IAM-Rollen für einen DB-Cluster im Konto. Andere DB-Cluster im Konto haben möglicherweise eine geringere Anzahl zugeordneter IAM-Rollen.	Anzahl
DBClusters	Die Anzahl der Amazon-Aurora-DB-Cluster in Ihrem AWS-Konto.	Anzahl

Metrik	Beschreibung	Einheiten *
DBInstanceRoles	Die Anzahl der zugehörigen Rollen AWS Identity and Access Management (IAM) pro DB-Instance in Ihrer. AWS-Konto Der verwendete Wert ist die höchste Anzahl zugeordneter IAM-Rollen für eine DB-Instance im Konto. Andere DB-Instances im Konto haben möglicherweise eine geringere Anzahl zugeordneter IAM-Rollen.	Anzahl
DBInstances	Die Anzahl der DB-Instances in Ihrem AWS-Konto.	Anzahl
DBParameterGroups	Die Anzahl der DB-Parametergruppen in Ihrem AWS-Konto. In der Anzahl sind keine Standard-DB-Parametergruppen enthalten.	Anzahl
DBSecurityGroups	Die Anzahl der Sicherheitsgruppen in Ihrem AWS-Konto. In der Anzahl sind die Standardsicherheitsgruppe und die Standard-VPC-Sicherheitsgruppe nicht enthalten.	Anzahl
DBSubnetGroups	Die Anzahl der DB-Subnetzgruppen in Ihrem AWS-Konto. In der Anzahl ist die Standardsubnetzgruppe nicht enthalten.	Anzahl
EventSubscriptions	Die Anzahl der Abonnements für Ereignisbenachrichtigungen in Ihrem AWS-Konto.	Anzahl
Integrations	Die Anzahl der Zero-ETL-Integrationen mit Amazon Redshift in Ihrem. AWS-Konto	Anzahl
ManualClusterSnapshots	Die Anzahl der manuell erstellten DB-Cluster-Snapshots in Ihrem AWS-Konto. In der Anzahl sind ungültige Snapshots nicht enthalten.	Anzahl
ManualSnapshots	Die Anzahl der manuell erstellten DB-Snapshots in Ihrem AWS-Konto. In der Anzahl sind ungültige Snapshots nicht enthalten.	Anzahl

Metrik	Beschreibung	Einheiten *
`OptionGroups`	Die Anzahl der Optionsgruppen in Ihrem AWS-Konto . In der Anzahl sind die Standardoptionsgruppen nicht enthalten.	Anzahl
`Proxies`	Die Anzahl der RDS-Proxys in Ihrem Konto. AWS	Anzahl
`ReadReplicasPerMaster`	Die Anzahl der Read Replicas pro DB-Instance in Ihrem Konto. Der verwendete Wert ist die höchste Anzahl von Read Replicas für eine DB-Instance im Konto. Andere DB-Instances im Konto haben möglicherweise eine geringere Anzahl von Read Replicas.	Anzahl
`ReservedDBInstances`	Die Anzahl der reservierten DB-Instances in Ihrem AWS-Konto. In der Anzahl sind außer Betrieb genommene oder abgelehnte Instances nicht enthalten.	Anzahl
`SubnetsPerDBSubnetGroup`	Die Anzahl der Subnetze pro DB-Subnetzgruppe in Ihrer. AWS-Konto Die höchste Anzahl von Subnetzen für eine DB-Subnetzgruppe im Konto. Andere DB-Subnetzgruppen im Konto haben möglicherweise eine geringere Anzahl von Subnetzen.	Anzahl

> **Note**
>
> Amazon RDS veröffentlicht keine Einheiten für Nutzungsmetriken CloudWatch. Die Einheiten werden nur in der Dokumentation angezeigt.

CloudWatch Amazon-Abmessungen für Amazon RDS

Sie können Amazon RDS filtern, indem Sie eine beliebige Dimension in der folgenden Tabelle verwenden.

Dimension	Filtert die angeforderten Daten für . . .
`DBInstanceIdentifier`	Eine angegebene DB-Instance.
`DatabaseClass`	Alle Instances in einer Datenbankklasse. Sie können beispielsweise Metriken für alle Instances der Datenbankklasse zusammenfasse `db.r5.large` .
`EngineName`	Nur der identifizierte Engine-Name. Sie können beispielsweise Metriken für alle Instances mit dem Engine-Namen zusammenfasse `postgres`.
`SourceRegion`	Nur die angegebene Region. Sie können beispielsweise Metriken für alle DB-Instances in der Region `us-east-1` zusammenfassen.

CloudWatch Amazon-Metriken für Amazon RDS Performance Insights

Performance Insights veröffentlicht automatisch einige Kennzahlen auf Amazon CloudWatch. Dieselben Daten können von Performance Insights abgefragt werden, aber wenn die Metriken vorhanden sind, ist es einfach, Alarme hinzuzufügen CloudWatch. CloudWatch Es macht es auch einfach, die Metriken zu bestehenden CloudWatch Dashboards hinzuzufügen.

Metrik	Beschreibung
DBLoad	Die Anzahl der aktiven Sitzungen für die Datenbank. In der Regel sind Sie an den Daten für die durchschnittliche Anzahl der aktiven Sitzungen interessiert. Diese Daten werden in Performance Insights als abgefrag `db.load.avg` .
DBLoadCPU	Anzahl aktiver Sitzungen mit dem Warteereignistyp CPU Diese Daten werden in Performance Insights als `db.load.avg` abgefragt, gefiltert durch den Warteereignistyp `CPU`.

Metrik	Beschreibung
DBLoadKeine CPU	Anzahl aktiver Sitzungen mit einem anderen Warteereignistyp als CPU
DBLoadRelativeToNumVCPUs	Das Verhältnis der Datenbanklast zur Anzahl der virtuellen CPUs Datenbanken.

> **Note**
>
> Diese Metriken werden CloudWatch nur veröffentlicht, wenn die DB-Instance ausgelastet ist.

Sie können diese Metriken mithilfe der CloudWatch Konsole AWS CLI, der oder der CloudWatch API untersuchen. Sie können auch andere Performance Insights Insights-Zählermetriken mithilfe einer speziellen mathematischen Metrikfunktion untersuchen. Weitere Informationen finden Sie unter Abfragen anderer Performance Insights Insights-Zählermetriken in CloudWatch.

Sie können beispielsweise die Statistiken für die `DBLoad` Metrik abrufen, indem Sie den get-metric-statisticsBefehl ausführen.

```
aws cloudwatch get-metric-statistics \
    --region us-west-2 \
    --namespace AWS/RDS \
    --metric-name DBLoad  \
    --period 60 \
    --statistics Average \
    --start-time 1532035185 \
    --end-time 1532036185 \
    --dimensions Name=DBInstanceIdentifier,Value=db-loadtest-0
```

Dieses Beispiel generiert eine Ausgabe wie die folgende.

```
{
  "Datapoints": [
  {
  "Timestamp": "2021-07-19T21:30:00Z",
  "Unit": "None",
  "Average": 2.1
```

```
},
{
"Timestamp": "2021-07-19T21:34:00Z",
"Unit": "None",
"Average": 1.7
},
{
"Timestamp": "2021-07-19T21:35:00Z",
"Unit": "None",
"Average": 2.8
},
{
"Timestamp": "2021-07-19T21:31:00Z",
"Unit": "None",
"Average": 1.5
},
{
"Timestamp": "2021-07-19T21:32:00Z",
"Unit": "None",
"Average": 1.8
},
{
"Timestamp": "2021-07-19T21:29:00Z",
"Unit": "None",
"Average": 3.0
},
{
"Timestamp": "2021-07-19T21:33:00Z",
"Unit": "None",
"Average": 2.4
}
],
"Label": "DBLoad"
}
```


Weitere Informationen zu CloudWatch finden Sie unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch.

Abfragen anderer Performance Insights Insights-Zählermetriken in CloudWatch

> **Note**
>
> Wenn Sie den erweiterten Modus von Database Insights aktivieren, veröffentlicht Amazon RDS Performance Insights Insights-Zählermetriken an Amazon CloudWatch. Bei Database Insights müssen Sie die mathematische Funktion für `DB_PERF_INSIGHTS` Metriken nicht verwenden. Sie können das CloudWatch Database Insights-Dashboard verwenden, um nach Performance Insights Insights-Zählermetriken zu suchen, abzufragen und Alarme einzustellen.

Sie können RDS Performance Insights Insights-Metriken von abfragen, Alarme ausgeben und Grafiken erstellen CloudWatch. Sie können auf Informationen zu Ihrer zugreifen, indem Sie die `DB_PERF_INSIGHTS` metrische mathematische Funktion für verwenden CloudWatch. Mit dieser Funktion können Sie die Performance Insights Insights-Metriken verwenden, an die nicht direkt berichtet wird CloudWatch , um eine neue Zeitreihe zu erstellen.

Sie können die neue Metric Math-Funktion verwenden, indem Sie im Bildschirm Metrik auswählen in der CloudWatch Konsole auf das Drop-down-Menü Mathematik hinzufügen klicken. Sie können damit Alarme und Grafiken zu Performance Insights-Metriken oder zu Kombinationen von CloudWatch Performance Insights Insights-Metriken erstellen, einschließlich hochauflösender Alarme für Metriken unter einer Minute. Sie können die Funktion auch programmgesteuert verwenden, indem Sie den Metric Math-Ausdruck in eine Anfrage aufnehmen. `get-metric-data` Weitere Informationen finden Sie unter Mathematische Syntax und Funktionen von Metriken und Erstellen eines Alarms für Performance Insights Insights-Zählermetriken aus einer AWS Datenbank.

Performance-Insights-Zählermetriken

Zählermetriken sind Betriebssystem- und Datenbank-Performance-Metriken im Performance-Insights-Dashboard. Um Leistungsprobleme zu identifizieren und zu analysieren, können Sie Zählermetriken mit der DB-Last korrelieren. Sie müssen eine Statistikfunktion an die Metrik anhängen, um die Metrikwerte zu erhalten. Die unterstützten Funktionen für die Metrik `os.memory.active` sind beispielsweise `.avg`, `.min`, `.max`,`.sum` und `.sample_count`.

Die Zählermetriken werden einmal pro Minute erfasst. Die Erfassung der Betriebssystemmetriken hängt davon ab, ob die erweiterte Überwachung aktiviert oder deaktiviert ist. Wenn die erweiterte Überwachung deaktiviert ist, werden die Betriebssystemmetriken einmal pro Minute erfasst. Ist die

erweiterte Überwachung aktiviert, werden die Betriebssystemmetriken für den ausgewählten Zeitraum erfasst. Weitere Informationen zum Aktivieren und Deaktivieren der erweiterten Überwachung finden Sie unter Aktivieren und Deaktivieren von „Enhanced Monitoring“ (Erweiterte Überwachung).

Themen

- Performance Insights-Betriebssystemzähler
- Performance Insights-Zähler für Amazon RDS for MariaDB und MySQL
- Performance Insights-Zähler für Amazon RDS for Microsoft SQL Server
- Performance Insights-Zähler für Amazon RDS for Oracle
- Performance Insights-Zähler für Amazon RDS for PostgreSQL

Performance Insights-Betriebssystemzähler

Die folgenden Betriebssystemzähler, denen `os` vorangestellt ist, sind bei Performance Insights für alle RDS-Engines mit Ausnahme von RDS für SQL-Server verfügbar.

Sie können die `ListAvailableResourceMetrics`-API für die Liste der verfügbaren Zählermetriken für Ihre DB-Instance verwenden. Weitere Informationen finden Sie ListAvailableResourceMetricsim Amazon RDS Performance Insights API-Referenzhandbuch.

Zähler	Typ	Einheit	Metrik	Beschreibung
Aktiv	Arbeitsspeicher	Kilobytes	os.memory .active	Umfang des zugewiesenen Arbeitsspeichers in Kilobyte.
Puffer	Arbeitsspeicher	Kilobytes	os.memory .buffers	Die Speicherm enge, die für die Pufferung von I/O Anfragen vor dem Schreiben auf das Speicherg erät verwendet wird, in Kilobyte.

Zähler	Typ	Einheit	Metrik	Beschreibung
Cached	Arbeitsspeicher	Kilobytes	os.memory .cached	Die Größe des verwendeten Arbeitsspeichers für das Caching von Dateisyst em-basierter I/O in Kilobyte.
DB Cache	Arbeitsspeicher	Bytes	os.memory .db.cache	Die Größe des Arbeitssp eichers, die vom Datenbank prozess einschlie ßlich tmpfs (shmem) für den Seiten-Cache verwendet wird, in Byte.
DB Resident Set Size	Arbeitsspeicher	Bytes	os.memory.db. residentSetSize	Die Größe des Arbeitssp eichers, die vom Datenbank prozess ohne tmpfs (shmem) für den anonymen Cache und den Swap-Cache verwendet wird, in Byte.

Zähler	Typ	Einheit	Metrik	Beschreibung
DB Swap	Arbeitsspeicher	Bytes	os.memory .db.swap	Die Größe des Arbeitssp eichers, die vom Datenbank prozess für Swap verwendet wird, in Byte.
Dirty	Arbeitsspeicher	Kilobytes	os.memory.dirty	Menge an Memory-Pages im RAM, die geändert, aber noch nicht in den entsprech enden Datenbloc k im Speicher geschrieben wurden, in Kilobyte.
Kostenfrei	Arbeitsspeicher	Kilobytes	os.memory.free	Umfang des nicht zugewiese nen Arbeitssp eichers in Kilobyte.
Huge Pages frei	Arbeitsspeicher	Seiten	Betriebss ystemspeicher. hugePagesFree	Die Anzahl von freien "Huge Pages". "Huge Pages" sind eine Funktion des Linux-Kernel.

Zähler	Typ	Einheit	Metrik	Beschreibung
Huge Pages reserviert	Arbeitsspeicher	Seiten	Betriebssystemspeicher.hugePagesRsvd	Die Anzahl von gebundenen "Huge Pages".
Größe Huge Pages	Arbeitsspeicher	Kilobytes	Betriebssystemspeicher.hugePagesSize	Die Größe für jede Huge Pages-Einheit, in Kilobyte.
Überschuss Huge Pages	Arbeitsspeicher	Seiten	Betriebssystemspeicher.hugePagesSurp	Die Anzahl der verfügbaren überzähligen Huge Pages.
Huge Pages insgesamt	Arbeitsspeicher	Seiten	Betriebssystemspeicher.hugePagesTotal	Die Gesamtzahl der Huge Pages.
Inaktiv	Arbeitsspeicher	Kilobytes	os.memory.inactive	Umfang der am seltensten verwendeten Memory-Pages in Kilobyte.
Mapped	Arbeitsspeicher	Kilobytes	os.memory.mapped	Die Gesamtmenge der Dateisysteminhalte, die Arbeitsspeicher in einem Prozess-Adressraum zugeordnet ist, in Kilobyte.

Zähler	Typ	Einheit	Metrik	Beschreibung
Out of Memory Kill Count	Arbeitsspeicher	Tötet	Betriebssystemspeicher.outOfMemoryKillCount	Die Anzahl der OOM-Kills im letzten Erfassungsintervall.
Seitentabellen	Arbeitsspeicher	Kilobytes	os.memory.pageTables	Umfang des von Page-Tabellen verwendeten Arbeitsspeichers in Kilobyte.
Slab	Arbeitsspeicher	Kilobytes	os.memory.slab	Umfang der wiederverwendbaren Kernel-Datenstrukturen in Kilobyte.
Gesamt	Arbeitsspeicher	Kilobytes	os.memory.total	Gesamtumfang des Arbeitsspeichers in Kilobyte.
Writeback	Arbeitsspeicher	Kilobytes	os.memory.writeback	Gesamtumfang an modifizierten Speicherbereichen im RAM, die noch in den Sicherungsspeicher geschrieben werden, in Kilobyte.

Zähler	Typ	Einheit	Metrik	Beschreibung
Guest	CPU-Auslastung	Prozentsatz	os.cpuUtilization.guest	Der prozentuale Anteil der von Gastprogrammen verwendeten CPU.
Inaktiv	CPU-Auslastung	Prozentsatz	os.cpuUtilization.idle	Der prozentuale Anteil der CPU, der sich im Leerlauf befindet.
Irq	CPU-Auslastung	Prozentsatz	os.cpuUtilization.irq	Der prozentuale Anteil der von Software-Interrupts verwendeten CPU.
Nice	CPU-Auslastung	Prozentsatz	os.cpuUtilization.nice	Der prozentuale Anteil der von Programmen mit niedrigster Priorität verwendeten CPU.
Steal	CPU-Auslastung	Prozentsatz	os.cpuUtilization.steal	Der prozentuale Anteil der von virtuellen Maschinen verwendeten CPU.

Zähler	Typ	Einheit	Metrik	Beschreibung
System (System)	CPU-Auslastung	Prozentsatz	os.cpuUtilization.system	Der prozentuale Anteil der vom Kernel verwendeten CPU.
Gesamt	CPU-Auslastung	Prozentsatz	os.cpuUtilization.total	Der prozentuale Anteil der insgesamt verwendeten CPU. Diese Angabe enthält den Nice-Wert.
Benutzer	CPU-Auslastung	Prozentsatz	os.cpuUtilization.user	Der prozentuale Anteil der von Benutzerprogrammen verwendeten CPU.
Wait	CPU-Auslastung	Prozentsatz	os.cpuUtilization.wait	Der Prozentsatz der CPUs, die beim Warten auf den I/O Zugriff nicht genutzt wurden.
Lesen Sie IOs PS	Festplatten-IO	Anforderungen pro Sekunde	Betriebssystem.Diskio.<devicename>... PS IOs lesen	Die Anzahl der Lesevorgänge pro Sekunde.

Zähler	Typ	Einheit	Metrik	Beschreibung
Schreiben Sie IOs PS	Festplatten-IO	Anforderungen pro Sekunde	Betriebssystem.Diskio.<devicename>.schreibe IOs PS	Die Anzahl der Schreibvorgänge pro Sekunde.
Avg Queue Len	Festplatten-IO	Anforderungen	OS Diskio.<devicename>.avgQueueLen	Die Anzahl der Anfragen, die in der Warteschlange des I/O Geräts warten.
Avg Req Sz	Festplatten-IO	Anforderungen	OS Diskio.<devicename>.avgReqSz	Die Anzahl der Anfragen, die in der Warteschlange des I/O Geräts warten.
Await	Festplatten-IO	Millisekunden	os.diskIO.<devicename>.await	Die erforderliche Anzahl an Millisekunden für die Antwort auf Anfragen, einschließlich Warteschlangen- und Servicedauer.
Lesen Sie PS IOs	Festplatten-IO	Anforderungen	Betriebssystem.Diskio.<devicename>... PS IOs lesen	Die Anzahl der Lesevorgänge pro Sekunde.

Zähler	Typ	Einheit	Metrik	Beschreibung
Read KB	Festplatten-IO	Kilobytes	os.diskIO.<devicename>.readKb	Gesamtzahl der gelesenen Kilobyte.
Read KB PS	Festplatten-IO	Kilobyte pro Sekunde	os.diskIO.<devicename>.readKbPS	Die Anzahl der pro Sekunde gelesenen Kilobyte.
Rrqm PS	Festplatten-IO	Anforderungen pro Sekunde	os.diskIO.<devicename>.rrqmPS	Die Anzahl der zusammengeführten Leseanfragen in der Warteschlange pro Sekunde.
TPS	Festplatten-IO	Transaktionen pro Sekunde	os.diskIO.<devicename>.tps	Die Anzahl der I/O Transaktionen pro Sekunde.
Util	Festplatten-IO	Prozentsatz	os.diskIO.<devicename>.util	Der Prozentsatz der CPU-Zeit, während der Anfragen ausgegeben wurden.
Write KB	Festplatten-IO	Kilobytes	os.diskIO.<devicename>.writeKb	Gesamtzahl der geschriebenen Kilobyte.

Zähler	Typ	Einheit	Metrik	Beschreibung
Write KB PS	Festplatten-IO	Kilobyte pro Sekunde	os.diskIO .<devicename>.writeKbPS	Die Anzahl der pro Sekunde geschriebenen Kilobyte.
Wrqm PS	Festplatten-IO	Anforderungen pro Sekunde	os.diskIO .<devicename>.wrqmPS	Die Anzahl der zusammengeführten Schreibanfragen in der Warteschlange pro Sekunde.
Blocked	Aufgaben	Aufgaben	os.tasks.blocked	Die Anzahl von blockierten Aufgaben.
In Ausführung	Aufgaben	Aufgaben	os.tasks.running	Die Anzahl von laufenden Aufgaben.
Sleeping	Aufgaben	Aufgaben	os.tasks.sleeping	Die Anzahl von Aufgaben im Ruhezustand.
Angehalten	Aufgaben	Aufgaben	os.tasks.stopped	Die Anzahl von angehaltenen Aufgaben.
Gesamt	Aufgaben	Aufgaben	os.tasks.total	Die Gesamtanzahl der Aufgaben.

Zähler	Typ	Einheit	Metrik	Beschreibung
Zombie	Aufgaben	Aufgaben	os.tasks.zombie	Die Anzahl der untergeordneten Aufgaben, die unter einer aktiven übergeordneten Aufgabe inaktiv sind.
One	Durchschnittliche Auslastung Minute	Prozesse	os. loadAvera geMinute. eins	Die Anzahl der Prozesse, die während der letzten Minute CPU-Zeit angefordert haben.
Fifteen	Durchschnittliche Auslastung Minute	Prozesse	os. loadAvera geMinute. fünfzehn	Die Anzahl der Prozesse, die während der letzten 15 Minuten CPU-Zeit angefordert haben.
Five	Durchschnittliche Auslastung Minute	Prozesse	os. loadAvera geMinute. fünf	Die Anzahl der Prozesse, die während der letzten 5 Minuten CPU-Zeit angefordert haben.

Zähler	Typ	Einheit	Metrik	Beschreibung
Cached	Auslagerung	Kilobytes	os.swap.cached	Umfang des Swap-Arbeitsspeichers, der als Cache-Speicher verwendet wird, in Kilobyte.
Kostenfrei	Auslagerung	Kilobytes	os.swap.free	Die Menge des freien Swap-Arbeitsspeichers in Kilobyte.
In	Auslagerung	Kilobytes	os.swap.in	Die Menge des von der Festplatte ausgelagerten Speichers in Kilobyte.
Out	Auslagerung	Kilobytes	os.swap.out	Die Menge des auf die Festplatte ausgelagerten Speichers in Kilobyte.
Gesamt	Auslagerung	Kilobytes	os.swap.total	Die Gesamtmenge des verfügbaren Swap-Arbeitsspeichers in Kilobyte.
Max Files	Dateisystem	Dateien	os.fileSys.maxFiles	Die maximale Anzahl an Dateien, die für das Dateisystem erstellt werden können.

Zähler	Typ	Einheit	Metrik	Beschreibung
Used Files	Dateisystem	Dateien	os.fileSys.usedFiles	Die Anzahl der Dateien im Dateisystem.
Used File Percent	Dateisystem	Dateien	Betriebssystemdatei Sys.usedFilePercent	Der Prozentsatz von verfügbaren Dateien, die in Gebrauch sind.
Used Percent	Dateisystem	Prozentsatz	os.fileSys.usedPercent	Der prozentuale Anteil des verwendeten Speicherplatzes für das Dateisystem.
Used	Dateisystem	Kilobytes	os.fileSys.used	Der durch Dateien belegte Speicherplatz im Dateisystem in Kilobyte.
Gesamt	Dateisystem	Kilobytes	os.fileSys.total	Die Gesamtmenge des für das Dateisystem verfügbaren Speicherplatzes in Kilobyte.
Rx	Netzwerk	Bytes pro Sekunde	os.network.rx	Die Anzahl der pro Sekunde empfangenen Byte.

Zähler	Typ	Einheit	Metrik	Beschreibung
Tx	Netzwerk	Bytes pro Sekunde	os.network.tx	Die Anzahl der pro Sekunde hochgeladenen Byte.
Acu Utilization	Allgemeines	Prozentsatz	os.genera l.acuUtilization	Der Anteil der aktuellen Kapazität an der maximal konfigurierten Kapazität in Prozent.
Max Configured Acu	Allgemeines	ACUs	os. allgemein . maxConfig uredAcu	Die vom Benutzer konfigurierte maximale Kapazität in Aurora-Ka pazitätseinheiten (ACUs).
Min Configured Acu	Allgemeines	ACUs	os.allgemein. minConfig uredAcu	Die vom Benutzer konfigurierte Mindestka pazität, in ACUs.
Num VCPUs	Allgemeines	v CPUs	os.general.num VCPUs	Die Anzahl der virtuellen CPUs (vCPUs) für die DB-Instance.

Zähler	Typ	Einheit	Metrik	Beschreibung
Serverles s Database Capacity	Allgemeines	ACUs	os.genera l. serverles sDatabase Capacity	Die aktuelle Kapazität der Instanz, in ACUs.

Performance Insights-Zähler für Amazon RDS for MariaDB und MySQL

Die folgenden Datenbankzähler sind bei Performance Insights für Amazon RDS for MariaDB und MySQL verfügbar.

Themen

- Native Zähler für RDS for MariaDB und RDS for MySQL
- Nicht-native Zähler für Amazon RDS for MariaDB und MySQL

Native Zähler für RDS for MariaDB und RDS for MySQL

Native Metriken werden von der Datenbank-Engine und nicht von Amazon RDS definiert. Definitionen dieser systemeigenen Metriken finden Sie in der MySQL-Dokumentation unter Serverstatusvariablen (für 8.0) und Serverstatusvariablen (für 8.4).

Zähler	Typ	Einheit	Metrik
Com_analyze	SQL	Abfragen pro Sekunde	db.SQL.Com_analyze
Com_optimize	SQL	Abfragen pro Sekunde	db.SQL.Com_optimize
Com_select	SQL	Abfragen pro Sekunde	db.SQL.Com_select
Verbindungen	SQL	Die Anzahl der Verbindun gsversuche	db.Users.Connections

Zähler	Typ	Einheit	Metrik
		pro Minute (erfolgreich oder nicht) mit dem MySQL-Server	
Innodb_rows_deleted	SQL	Zeilen pro Sekunde	db.SQL.Innodb_rows_deleted
Innodb_rows_inserted	SQL	Zeilen pro Sekunde	db.SQL.Innodb_rows_inserted
Innodb_rows_read	SQL	Zeilen pro Sekunde	db.SQL.Innodb_rows_read
Innodb_rows_updated	SQL	Zeilen pro Sekunde	db.SQL.Innodb_rows_updated
Select_full_join	SQL	Abfragen pro Sekunde	db.SQL.Select_full_join
Select_full_range_join	SQL	Abfragen pro Sekunde	db.SQL.Select_full_range_join
Bereich auswählen	SQL	Abfragen pro Sekunde	db.SQL.Select_range
Select_range_check	SQL	Abfragen pro Sekunde	db.SQL.Select_range_check
Select_scan	SQL	Abfragen pro Sekunde	db.SQL.Select_scan
Slow_queries	SQL	Abfragen pro Sekunde	db.SQL.Slow_queries

Zähler	Typ	Einheit	Metrik
Sort_merge_passes	SQL	Abfragen pro Sekunde	db.SQL.Sort_merge_passes
Sort_range	SQL	Abfragen pro Sekunde	db.SQL.Sort_range
Sort_rows	SQL	Abfragen pro Sekunde	db.SQL.Sort_rows
Sort_scan	SQL	Abfragen pro Sekunde	db.SQL.Sort_scan
Fragen	SQL	Abfragen pro Sekunde	db.SQL.Questions
Innodb_row_lock_time	Sperren	Millisekunden (durchschnittlich)	db.Locks.Innodb_row_lock_time
Table_locks_immediate	Sperren	Anforderungen pro Sekunde	db.Locks.Table_locks_immediate
Table_locks_waited	Sperren	Anforderungen pro Sekunde	db.Locks.Table_locks_waited
Aborted_clients	Benutzer	Verbindungen	db.Users.Aborted_clients
Aborted_connects	Benutzer	Verbindungen	db.Users.Aborted_connects
max_connections	Benutzer	Verbindungen	db.User.max_connections

Zähler	Typ	Einheit	Metrik
Threads_created	Benutzer	Verbindungen	db.Users.Threads_created
Threads_running	Benutzer	Verbindungen	db.Users.Threads_running
Innodb_data_writes	I/O	Operationen pro Sekunde	db.IO.Innodb_data_writes
Innodb_dblwr_writes	I/O	Operationen pro Sekunde	db.IO.Innodb_dblwr_writes
Innodb_log_write_requests	I/O	Operationen pro Sekunde	db.IO.Innodb_log_write_requests
Innodb_log_writes	I/O	Operationen pro Sekunde	db.IO.Innodb_log_writes
Innodb_pages_written	I/O	Seiten pro Sekunde	db.IO.Innodb_pages_written
Created_tmp_disk_tables	Temporäre Dateien	Tabellen pro Sekunde	db.Temp.Created_tmp_disk_tables
Created_tmp_tables	Temporäre Dateien	Tabellen pro Sekunde	db.Temp.Created_tmp_tables
Innodb_buffer_pool_pages_data	Cache	Seiten	db.Cache.Innodb_buffer_pool_pages_data
Innodb_buffer_pool_pages_total	Cache	Seiten	db.Cache.Innodb_buffer_pool_pages_total
Innodb_buffer_pool_read_requests	Cache	Seiten pro Sekunde	db.Cache.Innodb_buffer_pool_read_requests
Innodb_buffer_pool_reads	Cache	Seiten pro Sekunde	db.Cache.Innodb_buffer_pool_reads

Zähler	Typ	Einheit	Metrik
Opened_tables	Cache	Tabellen	db.Cache.Opened_tables
Opened_table_definitions	Cache	Tabellen	db.Cache.Opened_table_definitions
Qcache_hits	Cache	Abfragen	db.Cache.Qcache_hits

Nicht-native Zähler für Amazon RDS for MariaDB und MySQL

Nicht-native Zähler-Metriken sind durch Amazon RDS definierte Zähler. Eine nicht-native Metrik kann eine Metrik sein, die Sie mit einer bestimmten Abfrage erhalten. Eine nicht-native Metrik kann auch eine abgeleitete Metrik sein, bei der zwei oder mehrere native Zähler bei Berechnungen von Verhältnissen, Trefferraten oder Latenzen verwendet werden.

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
innodb_buffer_pool_hits	Cache	Liest	db.Cache.innoDB_buffer_pool_hits	Die Anzahl der Lesevorgänge, die InnoDB aus dem Pufferpool verarbeiten konnte.	`innodb_buffer_pool_read_requests - innodb_buffer_pool_reads`
innodb_buffer_pool_hit_rate	Cache	Prozentsatz	db.Cache.innoDB_buffer_pool_hit_rate	Der Prozentsatz der Lesevorgänge, den InnoDB aus dem	`100 * innodb_buffer_pool_read_requests / (innodb_buffer_pool_read_re`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
				Pufferpool verarbeiten konnte.	`quests + innodb_buffer_pool_reads)`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
innodb_buffer_pool_usage	Cache	Prozentsatz	db.Cache.innoDB_buffer_pool_usage	Die Prozentsatz des InnoDB-Pufferpools, der Daten (Seiten) enthält. **Note** Bei komprirten Tabelle kann dieser Wert variiere. Weiter Informonen finden Sie in den Informonen zu Innodffer_	`Innodb_buffer_pool_pages_data / Innodb_buffer_pool_pages_total * 100.0`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
				_pageta und Innodffer__pagetal in Servertusvarilen (für 8.0) und Servertusvarilen (für 8.4) in der MySQl Dokumentan.	

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
query_cache_hit_rate	Cache	Prozentsatz	db.Cache.query_cache_hit_rate	Trefferrate des MySQL-Ergebnissatz-Caches (Abfrage-Cache).	`Qcache_hits / (QCache_hits + Com_select) * 100`
innodb_datafile_writes_to_disk	I/O	Schreibt	db.IO.innoDB_datafile_writes_to_disk	Die Anzahl der InnoDB-Datendatei-Schreibvorgänge auf der Festplatte, ausschließlich doppelter Schreibvorgänge und Schreibvorgänge zur Wiederholungsprotokollierung.	`Innodb_data_writes - Innodb_log_writes - Innodb_dblwr_writes`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
innodb_rows_changed	SQL	Rows	db.SQL.innodb_rows_changed	Die Gesamtzahl von InnoDB-Zeilenvorgängen.	`db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated`
active_transactions	Transaktionen	Transaktionen	db.Transactions.active_transactions	Die Gesamtzahl aktiver Transaktionen.	`SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA.INNODB_TRX`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
trx_rseg_history_len	Transaktionen	Keine	db.Transactions.trx_rseg_history_len	Eine Liste der Undo-Protokollseiten für übernommene Transaktionen, die vom InnoDB-Transaktionssystem verwaltet wird, um die Parallelitätskontrolle für mehrere Versionen zu implementieren. Weitere Informationen zu Undo-Log-Datensätzen finden Sie unter InnoDB Multi-Ver	`SELECT COUNT AS trx_rseg_history_len FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='trx_rseg_history_len'`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
				sioning (für 8.0) und InnoDB Multi-Versioning (für 8.4) in der MySQL-Dokumentation.	
innodb_deadlocks	Sperren	Sperren	db.Locks.innodb_deadlocks	Die Gesamtzahl von Deadlocks.	`SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_deadlocks'`
innodb_lock_timeouts	Sperren	Sperren	db.Locks.innodb_lock_timeouts	Die Gesamtanzahl der Sperren, die das Zeitlimit überschritten haben.	`SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_timeouts'`

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
innodb_row_lock_waits	Sperren	Sperren	db.Locks.innodb_row_lock_waits	Die Gesamtanzahl von Zeilensperren, die zu einer Wartezeit geführt haben.	`SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'`

Performance Insights-Zähler für Amazon RDS for Microsoft SQL Server

Die folgenden Datenbankzähler sind mit Performance Insights für RDS for Microsoft SQL Server verfügbar.

Native Zähler für RDS for Microsoft SQL Server

Native Metriken werden von der Datenbank-Engine und nicht von Amazon RDS definiert. Sie finden die Definitionen für diese nativen Metriken in Verwenden von SQL Server-Objekten in der Microsoft SQL Server-Dokumentation.

Zähler	Typ	Einheit	Metrik
Weitergeleitete Datensätze	Zugriffsmethoden	Datensätze pro Sekunde	db.Access Methods.Forwarded Records
Seiten-Splits	Zugriffsmethoden	Splits pro Sekunde	db.Access Methods.Page Splits
Puffer-Cache-Trefferrate	Puffer-Manager	Verhältnis	db.Buffer Manager.Buffer cache hit ratio
Lebensdauererwartung der Seite	Puffer-Manager	Erwartung in Sekunden	db.Buffer Manager.Page life expectancy

Zähler	Typ	Einheit	Metrik
Seiten-Nachschlagevorgänge	Puffer-Manager	Nachschlagevorgänge pro Sekunde	db.Buffer Manager.Page lookups
Seiten-Lesevorgänge	Puffer-Manager	Lesevorgänge pro Sekunde	db.Buffer Manager.Page reads
Seiten-Schreibvorgänge	Puffer-Manager	Schreibvorgänge pro Sekunde	db.Buffer Manager.Page writes
Aktive Transaktionen.	Datenbanken	Transaktionen	db.Databases.Active Transactions (_Total)
Protokoll – bereinigte Bytes	Datenbanken	Pro Sekunde bereinigte Bytes	db.Databases.Log Bytes Flushed (_Total)
Protokoll – Bereinigungswartevorgänge	Datenbanken	Wartevorgänge pro Sekunde	db.Databases.Log Flush Waits (_Total)
Protokollbereinigungen	Datenbanken	Bereinigungen pro Sekunde	db.Databases.Log Flushes (_Total)
Schreibtransaktionen	Datenbanken	Transaktionen pro Sekunde	db.Databases.Write Transactions (_Total)
Geblockte Prozesse	Allgemeine Statistiken	Geblockte Prozesse	db.General Statistics.Processes blocked
Benutzerverbindungen	Allgemeine Statistiken	Verbindungen	db.General Statistics.User Connections
Latch-Wartevorgänge	Latches	Wartevorgänge pro Sekunde	db.Latches.Latch Waits
Anzahl der Deadlocks	Sperren	Deadlocks pro Sekunde	db.Locks.Number of Deadlocks (_Total)

Zähler	Typ	Einheit	Metrik
Ausstehende Arbeitsspeicherzuteilungen	Arbeitsspeicher-Manager	Arbeitsspeicherzuteilungen	db.Memory Manager.Memory Grants Pending
Stapelanforderungen	SQL-Statistiken	Anforderungen pro Sekunde	db.SQL Statistics.Batch Requests
SQL-Kompilierungen	SQL-Statistiken	Kompilierungen pro Sekunde	db.SQL Statistics.SQL Compilations
SQL-Neukompilierungen	SQL-Statistiken	Neukompilierungen pro Sekunde	db.SQL Statistics.SQL Re-Compilations

Performance Insights-Zähler für Amazon RDS for Oracle

Die folgenden Datenbankzähler sind mit Performance Insights für RDS for Oracle verfügbar.

Native Zähler für RDS for Oracle

Native Metriken werden von der Datenbank-Engine und nicht von Amazon RDS definiert. Definitionen dieser nativen Metriken finden Sie unter Beschreibungen von Statistiken in der Oracle-Dokumentation.

> **Note**
>
> Für die Zählermetrik `CPU used by this session` wurde die Einheit von nativen Zentisekunden in aktive Sitzungen umgewandelt, um den Wert benutzerfreundlicher zu machen. Der „CPU send“-Wert im DB-Lastdiagramm repräsentiert die CPU-Nachfrage. Die Zählermetrik `CPU used by this session` steht für die von Oracle-Sitzungen in Anspruch genommene CPU-Auslastung. Sie können den „CPU send“-Wert mit der Zählermetrik `CPU used by this session` vergleichen. Wenn die CPU-Nachfrage die CPU-Auslastung übersteigt, warten die Sitzungen auf CPU-Zeit.

Zähler	Typ	Einheit	Metrik
Von dieser Sitzung verwendete CPU	Benutzer	Aktive Sitzungen	Von dieser Sitzung verwendete db.User.CPU
Der to/from SQL*Net-Roundtrips-Client	Benutzer	Roundtrips pro Sekunde	db.User.SQL*Net-Roundtrip-Client to/from
Vom Client über SQL*Net empfangene Byte	Benutzer	Bytes pro Sekunde	Vom Client über SQL*Net empfangene db.User.bytes
Benutzer-Commits	Benutzer	Commits pro Sekunde	db.User.user-Commits
Kumulative Anmeldungen	Benutzer	Anmeldungen pro Sekunde	Kumulative db.User.logons
Benutzeraufrufe	Benutzer	Aufrufe pro Sekunde	db.User.user-Aufrufe
Über SQL*Net an den Client gesendete Byte	Benutzer	Bytes pro Sekunde	Über SQL*Net an den Client gesendete db.User.bytes
Benutzer-Rollbacks	Benutzer	Rollbacks pro Sekunde	db.User.user-Rollbacks
Größe wiederholen	Wiederholen	Bytes pro Sekunde	db.Redo.redo-Größe
Parse-Zähler (gesamt)	SQL	Parse-Vorgänge pro Sekunde	db.SQL.parse-Zähler (gesamt)
Parse-Zähler (hart)	SQL	Parse-Vorgänge pro Sekunde	db.SQL.parse-Zähler (hart)
Erhaltene Tabellenscanzeilen	SQL	Zeilen pro Sekunde	Erhaltene db.SQL.table-Scanzeilen

Zähler	Typ	Einheit	Metrik
Sortiervorgänge (Speicher)	SQL	Sortiervorgänge pro Sekunde	db.SQL.sorts (Speicher)
Sortiervorgänge (Festplatte)	SQL	Sortiervorgänge pro Sekunde	db.SQL.sorts (Festplatte)
Sortiervorgänge (Zeilen)	SQL	Sortiervorgänge pro Sekunde	db.SQL.sorts (Zeilen)
Physischer Lesevorgang in Bytes	Cache	Bytes pro Sekunde	db.Cache.physical-Lesevorgang in Bytes
DB-Blockabrufe	Cache	Blöcke pro Sekunde	db.Cache.db-Blockabrufe
DBWR-Prüfpunkte	Cache	Prüfpunkte pro Minute	db.Cache.DBWR-Prüfpunkte
Physische Lesevorgänge	Cache	Lesevorgänge pro Sekunde	db.Cache.physical-Lesevorgänge
Konsistente Abrufe aus dem Cache	Cache	Abrufe pro Sekunde	db.Cache.consistent-Abrufe aus dem Cache
DB-Blockabrufe aus dem Cache	Cache	Abrufe pro Sekunde	db.Cache.db-Blockabrufe aus dem Cache
Konsistente Abrufe	Cache	Abrufe pro Sekunde	db.Cache.consistent-Abrufe

Performance Insights-Zähler für Amazon RDS for PostgreSQL

Die folgenden Datenbankzähler sind bei Performance Insights für Amazon RDS for PostgreSQL verfügbar.

Themen

- Native Zähler für Amazon RDS for PostgreSQL
- Nicht native Zähler für Amazon RDS for PostgreSQL

Native Zähler für Amazon RDS for PostgreSQL

Native Metriken werden von der Datenbank-Engine und nicht von Amazon RDS definiert. Definitionen dieser nativen Metriken finden Sie unter Anzeigen von Statistiken in der PostgreSQL-Dokumentation.

Zähler	Typ	Einheit	Metrik
blks_hit	Cache	Blöcke pro Sekunde	db.Cache.blks_hit
buffers_alloc	Cache	Blöcke pro Sekunde	db.Cache.buffers_alloc
buffers_checkpoint	Prüfpunkt	Blöcke pro Sekunde	db.Checkpoint.buffers_checkpoint
checkpoint_sync_time	Prüfpunkt	Millisekunden pro Prüfpunkt	db.Checkpoint.checkpoint_sync_time
checkpoint_write_time	Prüfpunkt	Millisekunden pro Prüfpunkt	db.Checkpoint.checkpoint_write_time
checkpoints_req	Prüfpunkt	Prüfpunkte pro Minute	db.Checkpoint.checkpoints_req
checkpoints_timed	Prüfpunkt	Prüfpunkte pro Minute	db.Checkpoint.checkpoints_timed
maxwritten_clean	Prüfpunkt	Bgwriter-Bereinigungsstopps pro Minute	db.Checkpoint.maxwritten_clean
Deadlocks	Gleichzeitigkeit	Deadlocks pro Minute	db.Concurrency.deadlocks
blk_read_time	I/O	Millisekunden	db.IO.blk_read_time

Zähler	Typ	Einheit	Metrik
blks_read	I/O	Blöcke pro Sekunde	db.IO.blks_read
buffers_backend	I/O	Blöcke pro Sekunde	db.IO.buffers_backend
buffers_backend_fsync	I/O	Blöcke pro Sekunde	db.IO.buffers_backend_fsync
buffers_clean	I/O	Blöcke pro Sekunde	db.IO.buffers_clean
tup_deleted	SQL	Tupel pro Sekunde	db.SQL.tup_deleted
tup_fetched	SQL	Tupel pro Sekunde	db.SQL.tup_fetched
tup_inserted	SQL	Tupel pro Sekunde	db.SQL.tup_inserted
tup_returned	SQL	Tupel pro Sekunde	db.SQL.tup_returned
tup_updated	SQL	Tupel pro Sekunde	db.SQL.tup_updated
temp_bytes	Temporäre Dateien	Bytes pro Sekunde	db.Temp.temp_bytes
temp_files	Temporäre Dateien	Dateien pro Minute	db.Temp.temp_files
xact_commit	Transaktionen	Commits pro Sekunde	db.Transactions.xact_commit
xact_rollback	Transaktionen	Rollbacks pro Sekunde	db.Transactions.xact_rollback
numbackends	Benutzer	Verbindungen	db.User.numbackends

Zähler	Typ	Einheit	Metrik
archived_count	Write-Ahead Log (WAL)	Dateien pro Minute	db.WAL.archived_count

Nicht native Zähler für Amazon RDS for PostgreSQL

Nicht-native Zähler-Metriken sind durch Amazon RDS definierte Zähler. Eine nicht-native Metrik kann eine Metrik sein, die Sie mit einer bestimmten Abfrage erhalten. Eine nicht-native Metrik kann auch eine abgeleitete Metrik sein, bei der zwei oder mehrere native Zähler bei Berechnungen von Verhältnissen, Trefferraten oder Latenzen verwendet werden.

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
checkpoint_sync_latency	Prüfpu	db.Checkpoint.checkpoint_sync_latency		Die Gesamtzeit, die auf den Teil der Prüfpunktverarbeitung aufgewendet wurde, bei dem Dateien auf der Festplatte synchronisiert werden.	`checkpoint_sync_time / (checkpoints_timed + checkpoints_req)`
checkpoint_write_latency	Prüfpu	db.Checkpoint.checkpoint_write_latency	Die Gesamtzeit, die auf den Teil der Prüfpunktverarbeitung aufgewendet wurde, bei dem Dateien auf die Festplatte geschrieben werden.	`checkpoint_write_time / (checkpoints_timed + checkpoints_req)`	

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
read_latency	I/O	db.IO.r d_later	Die Zeit, die für das Lesen von Datendateiblöcken durch Backends in dieser Instance aufgewendet wurde.	`blk_read_time / blks_read`	
idle_in_transaction_aborted_count	Status	Sitzun	db.state.idle_in_transaction_aborted_count	Die Anzahl der Sitzungen im Bundesstaat. `idle in transaction (aborted)`	Nicht zutreffend
idle_in_transaction_count	Status	Sitzun	db.state.idle_in_transaction_count	Die Anzahl der Sitzungen im Bundesstaat. `idle in transaction`	Nicht zutreffend
idle_in_transaction_max_time	Status	Sekun	db.state.idle_in_transaction_max_time	Die Dauer der am längsten laufenden Transaktion im Bundesstaat in Sekunden. `idle in transaction`	Nicht zutreffend
active_transactions	Transaktionen	Transaktionen	db.Transactions.active_transactions	Die Anzahl der aktiven Transaktionen.	Nicht zutreffend
blocked_transactions	Transaktionen	Transaktionen	db.Transactions.blocked_transactions	Die Anzahl der blockierten Transaktionen.	Nicht zutreffend

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
oldest_active_logical_replication_slot_xid_age	Transaktionen	db.transactions.oldest_active_logical_replication_slot_xid_a	Das Alter der ältesten Transaktion in einem aktiven logischen Replikationsslot. Weitere Informationen finden Sie unter Steckplatz für logische Replikation.	–	
oldest_inactive_logical_replication_slot_xid_age	Transaktionen	db.transactions.oldest_inactive_logical_replication_slot_xid_e	Das Alter der ältesten Transaktion in einem inaktiven logischen Replikationsslot. Weitere Informationen finden Sie unter Steckplatz für logische Replikation.	–	
oldest_prepared_transaction_xid_age	Transaktionen	db.transactions.DEST_ARED_SACTID_AG	Das Alter der ältesten vorbereiteten Transaktion. Weitere Informationen finden Sie unter Vorbereitete Transaktion.	–	

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
oldest_running_transaction_xid_age	Transaonen	db.transactions.DEST_ING_TACTIOD_AGE	Das Alter der ältesten laufenden Transaktion. Weitere Informationen finden Sie unter Aktive Aussage für die älteste laufende aktive Transaktion und Leerlauf in Transaktion für die älteste laufende Transaktion idle-in-transaction.	–	
oldest_hot_standby_feedback_xid_age	Transaonen	db.transactions.dest_hstandbeedbaid_age	Das Alter der ältesten laufenden Transaktion auf einer Read Replica mit aktivierter Option. `hot_standby_feedback` Weitere Informationen finden Sie unter Read Replicas.	–	
max_used_xact_ids	Transaonen	Transaonen	db.Transactions.max_used_xact_ids	Die Anzahl der Transaktionen, die nicht vakuiert wurden.	Nicht zutreffend

Zähler	Typ	Einheit	Metrik	Beschreibung	Definition
max_connections	Benutz	Verbindungen	db.User.max_connections	Die maximale Anzahl von Verbindungen, die für eine DB-Instance zulässig sind, wie im Parameter konfiguriert. `max_connections`	Nicht zutreffend
archive_failed_count	WAL	Dateien pro Minute	db.WAL.archive_failed_count	Die Anzahl der fehlgeschlagenen Versuche, WAL-Dateien zu archivieren, in Dateien pro Minute.	Nicht zutreffend

SQL-Statistiken für Performance Insights

SQL-Statistiken sind leistungsbezogene Metriken zu SQL-Abfragen, die von Performance Insights gesammelt werden. Performance Insights sammelt Statistiken für jede Sekunde, in der eine Abfrage ausgeführt wird, und für jeden SQL-Aufruf. Die SQL-Statistiken bilden den Durchschnitt für den ausgewählten Zeitraum ab.

Ein SQL-Digest setzt sich aus allen Abfragen zusammen, die ein bestimmtes Muster haben, aber nicht unbedingt dieselben Literalwerte haben. Der Digest ersetzt Literalwerte durch ein Fragezeichen. Beispiel, `SELECT * FROM emp WHERE lname= ?`. Dieser Digest kann die folgenden untergeordneten Abfragen enthalten:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Alle Engines unterstützen SQL-Statistiken für Digest-Abfragen.

Informationen zur Unterstützung dieser Funktion nach Region, DB-Engine und Instance-Klasse finden Sie unter DB-Engine-, Regions- und Instance-Klassenunterstützung von Amazon RDS für Performance-Insights-Funktionen.

Themen

- SQL-Statistiken für MariaDB und MySQL
- SQL-Statistiken für Amazon RDS for Oracle
- SQL-Statistiken für Amazon RDS for SQL Server
- SQL-Statistiken für RDS PostgreSQL

SQL-Statistiken für MariaDB und MySQL

MariaDB und MySQL sammelt SQL-Statistiken nur auf Digest-Ebene. Auf der Statement-Ebene werden keine Statistiken angezeigt.

Themen

- Digest-Statistiken für MariaDB und MySQL
- Statistiken für MariaDB und MySQL
- Statistiken für MariaDB und MySQL
-

Digest-Statistiken für MariaDB und MySQL

Performance Insights sammelt SQL-Digest-Statistiken aus der `events_statements_summary_by_digest`-Tabelle. Die `events_statements_summary_by_digest`-Tabelle wird von der Datenbank verwaltet.

Diese Tabelle verfügt nicht über eine Bereinigungsrichtlinie. Wenn die Tabelle voll ist, wird die AWS Management Console folgende Meldung angezeigt:

```
Performance Insights is unable to collect SQL Digest statistics on new queries because
 the table events_statements_summary_by_digest is full.
Please truncate events_statements_summary_by_digest table to clear the issue. Check the
 User Guide for more details.
```

In dieser Situation verfolgen MariaDB und MySQL nicht SQL-Abfragen. Um dieses Problem zu beheben, kürzt Performance Insights die Digest-Tabelle automatisch, wenn die folgenden Bedingungen beide erfüllt sind:

- Die Tabelle ist voll.
- Performance Insights verwaltet das Leistungsschema automatisch.

  Für die automatische Verwaltung muss der Parameter `performance_schema` auf den Wert `0` festgelegt werden und Source (Quelle) auf `user` eingestellt sein. Wenn Performance Insights die Leistung nicht automatisch verwaltet, finden Sie weitere Informationen unter Überblick über das Leistungsschema für Performance Insights auf , Amazon RDS for MariaDB oder MySQL.

Überprüfen Sie in der AWS CLI die Quelle eines Parameterwerts, indem Sie den describe-db-parametersBefehl ausführen.

Statistiken für MariaDB und MySQL

Die folgenden SQL-Statistiken stehen für MariaDB und MySQL-DB-Instances zur Verfügung.

Metrik	Einheit
db.sql_tokenized.stats.count_star_per_sec	Aufrufe pro Sekunde
db.sql_tokenized.stats.sum_timer_wait_per_sec	Durchschnittliche Latenz pro Sekunde (in ms)
db.sql_tokenized.stats.sum_select_full_join_per_sec	Vollständigen Join pro Sekunde auswählen
db.sql_tokenized.stats.sum_select_range_check_per_sec	Bereichsprüfung pro Sekunde auswählen
db.sql_tokenized.stats.sum_select_scan_per_sec	Scan pro Sekunde auswählen
db.sql_tokenized.stats.sum_sort_merge_passes_per_sec	Zusammenführungsdurchläufe pro Sekunde sortieren
db.sql_tokenized.stats.sum_sort_scan_per_sec	Scans pro Sekunde sortieren

Metrik	Einheit
db.sql_tokenized.stats.sum_sort_rang e_per_sec	Bereiche pro Sekunde sortieren
db.sql_tokenized.stats.sum_sort_rows_per_sec	Zeilen pro Sekunde sortieren
db.sql_tokenized.stats.sum_rows_affected_per_ sec	Betroffene Zeilen pro Sekunde
db.sql_tokenized.stats.sum_rows_exam ined_per_sec	Überprüfte Zeilen pro Sekunde
db.sql_tokenized.stats.sum_rows_sent_per_sec	Gesendete Zeilen pro Sekunde
db.sql_tokenized.stats.sum_created_tmp_disk_t ables_per_sec	Temporäre Datenträgertabellen pro Sekunde erstellt
db.sql_tokenized.stats.sum_created_tmp_tables _per_sec	Temporäre Tabellen pro Sekunde erstellt
db.sql_tokenized.stats.sum_lock_time_per_sec	Sperrzeit pro Sekunde (in ms)

Statistiken für MariaDB und MySQL

Die folgenden Metriken stellen Statistiken pro einzelnen Abruf für SQL-Anweisungen bereit.

Metrik	Einheit
db.sql_tokenized.stats.sum_timer_wait_per_call	Durchschnitt Latenz pro Aufruf (in ms)
db.sql_tokenized.stats.sum_select_full_join_p er_call	Vollständige Joins pro Aufruf auswählen
db.sql_tokenized.stats.sum_select_ra nge_check_per_call	Bereichsprüfung pro Aufruf auswählen
db.sql_tokenized.stats.sum_select_sc an_per_call	Scans pro Aufruf auswählen

Metrik	Einheit
db.sql_tokenized.stats.sum_sort_merge_passes_per_call	Zusammenführungsdurchläufe pro Aufruf sortieren
db.sql_tokenized.stats.sum_sort_scan_per_call	Scans pro Aufruf sortieren
db.sql_tokenized.stats.sum_sort_range_per_call	Bereiche pro Aufruf sortieren
db.sql_tokenized.stats.sum_sort_rows_per_call	Zeilen pro Aufruf sortieren
db.sql_tokenized.stats.sum_rows_affected_per_call	Betroffene Zeilen pro Aufruf
db.sql_tokenized.stats.sum_rows_examined_per_call	Überprüfte Zeilen pro Aufruf
db.sql_tokenized.stats.sum_rows_sent_per_call	Gesendete Zeilen pro Aufruf
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_call	Temporäre Datenträgertabellen pro Aufruf erstellt
db.sql_tokenized.stats.sum_created_tmp_tables_per_call	Temporäre Tabellen pro Aufruf erstellt
db.sql_tokenized.stats.sum_lock_time_per_call	Sperrzeit pro Aufruf (in ms)

Die folgenden SQL-Statistiken stehen für MariaDB und MySQL-DB-Instances zur Verfügung.

Metrik	Einheit
db.sql_tokenized.stats.count_star	Calls
db.sql_tokenized.stats.sum_timer_wait	Wartezeit (in ms)
db.sql_tokenized.stats.sum_select_full_join	Wählen Sie Vollständigen Beitritt
db.sql_tokenized.stats.sum_select_range_check	Wählen Sie Bereichsprüfungen aus

Metrik	Einheit
db.sql_tokenized.stats.sum_select_scan	Wählen Sie Scans aus
db.sql_tokenized.stats.sum_sort_merge_passes	Merge-Pässe sortieren
db.sql_tokenized.stats.sum_sort_scan	Scans sortieren
db.sql_tokenized.stats.sum_sort_range	Bereiche sortieren
db.sql_tokenized.stats.sum_sort_rows	Zeilen sortieren
db.sql_tokenized.stats.sum_rows_betroffen	Betroffene Zeilen
db.sql_tokenized.stats.sum_rows_examined	Untersuchte Zeilen
db.sql_tokenized.stats.sum_rows_sent	Gesendete Zeilen
db.sql_tokenized.stats.sum_created_tmp_disk_tables	Temporäre Festplattentabellen erstellt
db.sql_tokenized.stats.sum_created_tmp_tables	Temporäre Tabellen wurden erstellt
db.sql_tokenized.stats.sum_lock_time	Sperrzeit (in ms)

SQL-Statistiken für Amazon RDS for Oracle

Amazon RDS for Oracle sammelt SQL-Statistiken sowohl auf Anweisung- als auch auf Digest-Ebene. Auf Anweisungsebene repräsentiert die ID-Spalte den Wert von `V$SQL.SQL_ID`. Auf der Digest-Ebene zeigt die ID-Spalte den Wert von `V$SQL.FORCE_MATCHING_SIGNATURE`.

Wenn die ID `0` lautet, wird auf Digest-Ebene von Oracle Database festgestellt, dass diese Anweisung nicht für die Wiederverwendung geeignet ist. In diesem Fall könnten die untergeordneten SQL-Anweisungen zu verschiedenen Digests gehören. Die Anweisungen sind jedoch unter dem `digest_text` für die erste erfasste SQL-Anweisung gruppiert.

Themen

- Sekundenschnelle Statistiken für Oracle
- Statistiken pro Anruf für Oracle

- Primäre Statistiken für Oracle

Sekundenschnelle Statistiken für Oracle

Die folgenden Metriken stellen Statistiken pro Sekunde für eine Oracle-SQL-Abfrage bereit.

Metrik	Einheit
db.sql.stats.executions_per_sec	Anzahl der Ausführungen pro Sekunde
db.sql.stats.elapsed_time_per_sec	Durchschnitt der aktiven Ausführungen (Average active executions, AAE)
db.sql.stats.rows_processed_per_sec	Pro Sekunde verarbeitete Zeilen
db.sql.stats.buffer_gets_per_sec	Pufferabrufe pro Sekunde
db.sql.stats.physical_read_requests_per_sec	Physische Lesevorgänge pro Sekunde
db.sql.stats.physical_write_requests_per_sec	Physische Schreibvorgänge pro Sekunde
db.sql.stats.total_sharable_mem_per_sec	Gesamter freigabefähiger Arbeitsspeicher pro Sekunde (in Bytes)
db.sql.stats.cpu_time_per_sec	CPU-Zeit pro Sekunde (in ms)

Die folgenden Metriken bieten Statistiken pro Sekunde für eine Oracle SQL Digest-Abfrage.

Metrik	Einheit
db.sql_tokenized.stats.executions_per_sec	Anzahl der Ausführungen pro Sekunde
db.sql_tokenized.stats.elapsed_time_per_sec	Durchschnitt der aktiven Ausführungen (Average active executions, AAE)
db.sql_tokenized.stats.rows_processed_per_sec	Pro Sekunde verarbeitete Zeilen
db.sql_tokenized.stats.buffer_gets_per_sec	Pufferabrufe pro Sekunde

Metrik	Einheit
db.sql_tokenized.stats.physical_read_requests_per_sec	Physische Lesevorgänge pro Sekunde
db.sql_tokenized.stats.physical_write_requests_per_sec	Physische Schreibvorgänge pro Sekunde
db.sql_tokenized.stats.total_sharable_mem_per_sec	Gesamter freigabefähiger Arbeitsspeicher pro Sekunde (in Bytes)
db.sql_tokenized.stats.cpu_time_per_sec	CPU-Zeit pro Sekunde (in ms)

Statistiken pro Anruf für Oracle

Die folgenden Metriken stellen Statistiken pro einzelnen Abruf für SQL-Anweisungen bereit.

Metrik	Einheit
db.sql.stats.elapsed_time_per_exec	Pro Ausführung verstrichene Zeit (in ms)
db.sql.stats.rows_processed_per_exec	Pro Ausführung verarbeitete Zeilen
db.sql.stats.buffer_gets_per_exec	Pufferabrufe pro Ausführung
db.sql.stats.physical_read_requests_per_exec	Physische Lesevorgänge pro Ausführung
db.sql.stats.physical_write_requests_per_exec	Physische Schreibvorgänge pro Ausführung
db.sql.stats.total_sharable_mem_per_exec	Gesamter freigabefähiger Arbeitsspeicher pro Ausführung (in Bytes)
db.sql.stats.cpu_time_per_exec	CPU-Zeit pro Ausführung (in ms)

Die folgenden Metriken stellen Statistiken pro einzelnen Abruf für SQL-Anweisungen bereit.

Metrik	Einheit
db.sql_tokenized.stats.elapsed_time_per_exec	Pro Ausführung verstrichene Zeit (in ms)
db.sql_tokenized.stats.rows_processed_per_exec	Pro Ausführung verarbeitete Zeilen
db.sql_tokenized.stats.buffer_gets_per_exec	Pufferabrufe pro Ausführung
db.sql_tokenized.stats.physical_read_requests_per_exec	Physische Lesevorgänge pro Ausführung
db.sql_tokenized.stats.physical_write_requests_per_exec	Physische Schreibvorgänge pro Ausführung
db.sql_tokenized.stats.total_sharable_mem_per_exec	Gesamter freigabefähiger Arbeitsspeicher pro Ausführung (in Bytes)
db.sql_tokenized.stats.cpu_time_per_exec	CPU-Zeit pro Ausführung (in ms)

Primäre Statistiken für Oracle

Die folgenden Metriken stellen Primärstatistiken für eine Oracle-SQL-Abfrage bereit.

Metrik	Einheit
db.sql.stats.executions	Anzahl der Ausführungen
db.sql.stats.elapsed_time	Verstrichene Zeit (in ms)
db.sql.stats.rows_processed	Verarbeitete Zeilen
db.sql.stats.buffer_gets	Buffer erhält
db.sql.stats.physical_read_requests	Physische Lesevorgänge
db.sql.stats.physical_write_requests	Physische Schreibvorgänge

Metrik	Einheit
db.sql.stats.total_sharable_mem	Gesamter gemeinsam nutzbarer Speicher (in Byte)
db.sql.stats.cpu_time	CPU-Zeit (in ms)

Die folgenden Metriken stellen Primärstatistiken für eine Oracle SQL Digest-Abfrage bereit.

Metrik	Einheit
db.sql_tokenized.stats.executions	Anzahl der Ausführungen
db.sql_tokenized.stats.elapsed_time	Verstrichene Zeit (in ms)
db.sql_tokenized.stats.rows_processed	Verarbeitete Zeilen
db.sql_tokenized.stats.buffer_gets	Buffer erhält
db.sql_tokenized.stats.physical_read_requests	Physische Lesevorgänge
db.sql_tokenized.stats.physical_write_requests	Physische Schreibvorgänge
db.sql_tokenized.stats.total_sharable_mem	Gesamter gemeinsam nutzbarer Speicher (in Byte)
db.sql_tokenized.stats.cpu_time	CPU-Zeit (in ms)

SQL-Statistiken für Amazon RDS for SQL Server

Amazon RDS für SQL Server sammelt SQL-Statistiken sowohl auf Anweisung- als auch auf Digest-Ebene. Auf Anweisungsebene repräsentiert die ID-Spalte den Wert von `sql_handle`. Auf der Digest-Ebene zeigt die ID-Spalte den Wert von `query_hash`.

SQL Server gibt für einige Anweisungen NULL-Werte für `query_hash` zurück. Zum Beispiel ALTER INDEX, CHECKPOINT, UPDATE STATISTICS, COMMIT TRANSACTION, FETCH NEXT FROM Cursor und einige INSERT-Anweisungen, SELECT @<variable>, bedingte Anweisungen und

ausführbare gespeicherte Prozeduren. In diesem Fall wird der Wert „`sql_handle`“ als ID auf der Digest-Ebene für diese Anweisung angezeigt.

Themen

- Statistiken pro Sekunde für SQL Server
- Statistiken pro Aufruf für SQL Server
- Primäre Statistiken für SQL Server

Statistiken pro Sekunde für SQL Server

Die folgenden Metriken stellen Statistiken pro Sekunde für eine SQL-Server-SQL-Abfrage bereit.

Metrik	Einheit
db.sql.stats.execution_count_per_sec	Anzahl der Ausführungen pro Sekunde
db.sql.stats.total_elapsed_time_per_sec	Insgesamt verstrichene Zeit pro Sekunde
db.sql.stats.total_rows_per_sec	Gesamtzahl verarbeiteter Zeilen pro Sekunde
db.sql.stats.total_logical_reads_per_sec	Gesamtzahl logischer Lesevorgänge pro Sekunde
db.sql.stats.total_logical_writes_per_sec	Gesamtzahl logischer Schreibvorgänge pro Sekunde
db.sql.stats.total_physical_reads_per_sec	Gesamtzahl physischer Lesevorgänge pro Sekunde
db.sql.stats.total_worker_time_per_sec	CPU-Zeit insgesamt (in ms)

Die folgenden Metriken stellen Statistiken pro Sekunde für eine SQL-Server-SQL-Digest-Abfrage bereit.

Metrik	Einheit
db.sql_tokenized.stats.execution_count_per_sec	Anzahl der Ausführungen pro Sekunde
db.sql_tokenized.stats.total_elapsed_time_per_sec	Insgesamt verstrichene Zeit pro Sekunde
db.sql_tokenized.stats.total_rows_per_sec	Gesamtzahl verarbeiteter Zeilen pro Sekunde
db.sql_tokenized.stats.total_logical_reads_per_sec	Gesamtzahl logischer Lesevorgänge pro Sekunde
db.sql_tokenized.stats.total_logical_writes_per_sec	Gesamtzahl logischer Schreibvorgänge pro Sekunde
db.sql_tokenized.stats.total_physical_reads_per_sec	Gesamtzahl physischer Lesevorgänge pro Sekunde
db.sql_tokenized.stats.total_worker_time_per_sec	CPU-Zeit insgesamt (in ms)

Statistiken pro Aufruf für SQL Server

Die folgenden Metriken stellen Statistiken pro einzelnen Aufruf für SQL-Server-SQL-Anweisungen bereit.

Metrik	Einheit
db.sql.stats.total_elapsed_time_per_call	Insgesamt verstrichene Zeit pro Ausführung (in ms)
db.sql.stats.total_rows_per_call	Gesamtzahl der pro Ausführung verarbeiteten Zeilen
db.sql.stats.total_logical_reads_per_call	Gesamtzahl logischer Lesevorgänge pro Ausführung

Metrik	Einheit
db.sql.stats.total_logical_writes_per_call	Gesamtzahl logischer Schreibvorgänge pro Ausführung
db.sql.stats.total_physical_reads_per_call	Gesamtzahl physischer Lesevorgänge pro Ausführung
db.sql.stats.total_worker_time_per_call	CPU-Zeit insgesamt pro Ausführung (in ms)

Die folgenden Metriken stellen Statistiken pro Aufruf für eine SQL-Server-SQL-Digest-Abfrage bereit.

Metrik	Einheit
db.sql_tokenized.stats.total_elapsed_time_per_call	Insgesamt verstrichene Zeit pro Ausführung
db.sql_tokenized.stats.total_rows_per_call	Gesamtzahl der pro Ausführung verarbeiteten Zeilen
db.sql_tokenized.stats.total_logical_reads_per_call	Gesamtzahl logischer Lesevorgänge pro Ausführung
db.sql_tokenized.stats.total_logical_writes_per_call	Gesamtzahl logischer Schreibvorgänge pro Ausführung
db.sql_tokenized.stats.total_physical_reads_per_call	Gesamtzahl physischer Lesevorgänge pro Ausführung
db.sql_tokenized.stats.total_worker_time_per_call	CPU-Zeit insgesamt pro Ausführung (in ms)

Primäre Statistiken für SQL Server

Die folgenden Metriken stellen Primärstatistiken für eine SQL Server-SQL-Abfrage bereit.

Metrik	Einheit
db.sql.stats.execution_count	Anzahl der Ausführungen
db.sql.stats.total_elapsed_time	Insgesamt verstrichene Zeit (in ms)
db.sql.stats.total_rows	Gesamtzahl der verarbeiteten Zeilen
db.sql.stats.total_logical_reads	Gesamtzahl der logischen Lesevorgänge
db.sql.stats.total_logical_writes	Gesamtzahl der logischen Schreibvorgänge
db.sql.stats.total_physical_reads	Gesamtzahl der physischen Lesevorgänge
db.sql.stats.total_worker_time	CPU-Zeit insgesamt (in ms)

Die folgenden Metriken stellen Primärstatistiken für eine SQL Server-SQL-Digest-Abfrage bereit.

Metrik	Einheit
db.sql_tokenized.stats.execution_count	Anzahl der Ausführungen
db.sql_tokenized.stats.total_elapsed_time	Insgesamt verstrichene Zeit (in ms)
db.sql_tokenized.stats.total_rows	Gesamtzahl der verarbeiteten Zeilen
db.sql_tokenized.stats.total_logical_reads	Gesamtzahl der logischen Lesevorgänge
db.sql_tokenized.stats.total_logical_writes	Gesamtzahl der logischen Schreibvorgänge
db.sql_tokenized.stats.total_physical_reads	Gesamtzahl der physischen Lesevorgänge
db.sql_tokenized.stats.total_worker_time	CPU-Zeit insgesamt (in ms)

SQL-Statistiken für RDS PostgreSQL

Performance Insights sammelt für jeden SQL-Aufruf und für jede Sekunde, in der eine Abfrage ausgeführt wird, SQL-Statistiken. RDS für PostgreSQL sammelt SQL-Statistiken nur auf Digest-Ebene. Auf der Anweisungsebene werden keine Statistiken angezeigt.

Im Folgenden finden Sie Informationen zu Statistiken auf Digest-Ebene für RDS für PostgreSQL.

Themen

- Digest-Statistiken für RDS PostgreSQL
- Sekundengenaue Digest-Statistiken für RDS PostgreSQL
- Digest-Statistiken pro Aufruf für RDS PostgreSQL
- Primärstatistiken für RDS PostgreSQL

Digest-Statistiken für RDS PostgreSQL

Damit SQL-Digest-Statistiken angezeigt werden können, muss RDS PostgreSQL die Bibliothek `pg_stat_statements` laden. Für PostgreSQL-DB-Instances, die mit PostgreSQL 11 oder höher kompatibel sind, wird diese Bibliothek von der Datenbank standardmäßig geladen. Für PostgreSQL-DB-Instances, die mit PostgreSQL 10 oder früher kompatibel sind, aktivieren Sie diese Bibliothek manuell. Zur manuellen Aktivierung fügen Sie in der DB-Parametergruppe, die der DB-Instance zugeordnet ist, `pg_stat_statements` zu `shared_preload_libraries` hinzu. Starten Sie anschließend die DB-Instance neu. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

> **Note**
>
> Performance-Insights kann nur Statistiken für nicht abgeschnittene Abfragen in `pg_stat_activity` erfassen. Standardmäßig kürzen PostgreSQL-Datenbanken Abfragen, die länger als 1 024 Bytes sind. Um die Abfragegröße zu erhöhen, ändern Sie den Parameter `track_activity_query_size` in der DB-Parametergruppe, die mit Ihrer DB-Instance verknüpft ist. Wenn Sie diesen Parameter ändern, ist ein Neustart der DB-Instance erforderlich.

Sekundengenaue Digest-Statistiken für RDS PostgreSQL

Die folgenden SQL Digest-Statistiken sind für PostgreSQL DB-Instances verfügbar.

Metrik	Einheit
db.sql_tokenized.stats.calls_per_sec	Aufrufe pro Sekunde

Metrik	Einheit
db.sql_tokenized.stats.rows_per_sec	Zeilen pro Sekunde
db.sql_tokenized.stats.total_time_per_sec	Durchschnitt der aktiven Ausführungen (Average active executions, AAE) pro Sekunde
db.sql_tokenized.stats.shared_blks_hit_per_sec	Blocktreffer pro Sekunde
db.sql_tokenized.stats.shared_blks_read_per_sec	Blocklesevorgänge pro Sekunde
db.sql_tokenized.stats.shared_blks_dirtied_per_sec	Blöcke kontaminiert pro Sekunde
db.sql_tokenized.stats.shared_blks_written_per_sec	Blockschreibvorgänge pro Sekunde
db.sql_tokenized.stats.local_blks_hit_per_sec	Lokale Blocktreffer pro Sekunde
db.sql_tokenized.stats.local_blks_read_per_sec	Lokale Blocklesevorgänge pro Sekunde
db.sql_tokenized.stats.local_blks_dirtied_per_sec	Lokaler Block, der pro Sekunde verschmutzt wird
db.sql_tokenized.stats.local_blks_written_per_sec	Lokale Blockschreibvorgänge pro Sekunde
db.sql_tokenized.stats.temp_blks_written_per_sec	Temporäre Schreibvorgänge pro Sekunde
db.sql_tokenized.stats.temp_blks_read_per_sec	Temporäre Lesevorgänge pro Sekunde
db.sql_tokenized.stats.blk_read_time_per_sec	Durchschnitt gleichzeitige Lesevorgänge pro Sekunde
db.sql_tokenized.stats.blk_write_time_per_sec	Durchschnitt gleichzeitige Schreibvorgänge pro Sekunde

Digest-Statistiken pro Aufruf für RDS PostgreSQL

Die folgenden Metriken stellen Statistiken pro einzelnen Abruf für SQL-Anweisungen bereit.

Metrik	Einheit
db.sql_tokenized.stats.rows_per_call	Zeilen pro Aufruf
db.sql_tokenized.stats.avg_latency_per_call	Durchschnitt Latenz pro Aufruf (in ms)
db.sql_tokenized.stats.shared_blks_hit_per_call	Blocktreffer pro Aufruf
db.sql_tokenized.stats.shared_blks_read_per_call	Blocklesevorgänge pro Aufruf
db.sql_tokenized.stats.shared_blks_written_per_call	Blockschreibvorgänge pro Aufruf
db.sql_tokenized.stats.shared_blks_dirtied_per_call	Blöcke kontaminiert pro Aufruf
db.sql_tokenized.stats.local_blks_hit_per_call	Lokale Blocktreffer pro Aufruf
db.sql_tokenized.stats.local_blks_read_per_call	Lokale Blocklesevorgänge pro Aufruf
db.sql_tokenized.stats.local_blks_dirtied_per_call	Lokaler Block wird pro Anruf verschmutzt
db.sql_tokenized.stats.local_blks_written_per_call	Lokale Blockschreibvorgänge pro Aufruf
db.sql_tokenized.stats.temp_blks_written_per_call	Temporäre Blockschreibvorgänge pro Aufruf
db.sql_tokenized.stats.temp_blks_read_per_call	Temporäre Blocklesevorgänge pro Aufruf
db.sql_tokenized.stats.blk_read_time_per_call	Lesezeit pro Aufruf (in ms)
db.sql_tokenized.stats.blk_write_time_per_call	Schreibzeit pro Aufruf (in ms)

Primärstatistiken für RDS PostgreSQL

Die folgenden SQL-Statistiken sind für PostgreSQL-DB-Instances verfügbar.

Metrik	Einheit
db.sql_tokenized.stats.calls	Calls
db.sql_tokenized.stats.rows	Rows
db.sql_tokenized.stats.total_time	Gesamtzeit (in ms)
db.sql_tokenized.stats.shared_blks_hit	Treffer blockieren
db.sql_tokenized.stats.shared_blks_read	Lesevorgänge blockieren
db.sql_tokenized.stats.shared_blks_dirtied	Verschmutzte Blöcke
db.sql_tokenized.stats.shared_blks_written	Schreibvorgänge blockieren
db.sql_tokenized.stats.local_blks_hit	Lokale Block-Treffer
db.sql_tokenized.stats.local_blks_read	Lokale Blocklesevorgänge
db.sql_tokenized.stats.local_blks_dirtied	Lokale Blöcke sind verschmutzt
db.sql_tokenized.stats.local_blks_written	Lokale Blockschreibvorgänge
db.sql_tokenized.stats.temp_blks_written	Temporäre Schreibvorgänge
db.sql_tokenized.stats.temp_blks_read	Temporäre Lesevorgänge
db.sql_tokenized.stats.blk_read_time	Durchschnittliche Anzahl gleichzeitiger Lesevorgänge (in ms)
db.sql_tokenized.stats.blk_write_time	Durchschnittliche Anzahl gleichzeitiger Schreibvorgänge (in ms)

Weitere Informationen zu diesen Metriken finden Sie unter pg_stat_statements in der PostgreSQL-Dokumentation.

Betriebssystemmetriken in „Erweiterte Überwachung“

Amazon RDS stellt Metriken in Echtzeit für das Betriebssystem (OS) bereit, auf dem Ihr läuft. RDS die Metriken von Enhanced Monitoring an Ihr Amazon CloudWatch Logs-Konto. In den folgenden Tabellen sind die Betriebssystemmetriken aufgeführt, die mit Amazon CloudWatch Logs verfügbar sind.

Themen

- Betriebssystemmetriken für Db2, MariaDB, MySQL, Oracle und Postgre SQL
- Betriebssystemmetriken für Microsoft SQL Server

Betriebssystemmetriken für Db2, MariaDB, MySQL, Oracle und Postgre SQL

Gruppe	Metrik	Name der Konsole	Beschreibung
`General`	`engine`	Nicht zutreffend	Die Datenbank-Engine für die DB-Instance.
	`instanceID`	Nicht zutreffend	Die DB-Instance-Kennung.
	`instanceResourceID`	Nicht zutreffend	Eine unveränderliche Kennung für die DB-Instance, die für eine AWS Region eindeutig ist und auch als Log-Stream-ID verwendet wird.
	`numVCPUs`	Nicht zutreffend	Die Anzahl der virtuellen Dateien CPUs für die DB-Instance.
	`timestamp`	Nicht zutreffend	Die Uhrzeit, zu der die Metriken erfasst wurden.
	`uptime`	Nicht zutreffend	Die Zeitdauer, für die die DB-Instance aktiv war.
	`version`	Nicht zutreffend	Die Version des JSON Stream-Formats der Betriebssystemmetriken.

Gruppe	Metrik	Name der Konsole	Beschreibung
`cpuUtilization`	`guest`	CPUGast	Der Prozentsatz, CPU der von Gastprogrammen genutzt wird.
	`idle`	CPUIm Leerlauf	Der Prozentsatz CPU davon ist ungenutzt.
	`irq`	CPU IRQ	Der Prozentsatz, CPU der von Software-Interrupts genutzt wird.
	`nice`	CPUNett	Der Prozentsatz, der von CPU Programmen verwendet wird, die mit der niedrigsten Priorität ausgeführt werden.
	`steal`	CPUStehlen	Der Prozentsatz, CPU der von anderen virtuellen Maschinen genutzt wird.
	`system`	CPUSystem	Der Prozentsatz, CPU der vom Kernel verwendet wird.
	`total`	CPUInsgesamt	Der Gesamtprozentsatz CPU der verwendeten. Diese Angabe enthält den Wert `nice`.
	`user`	CPUBenutzer	Der Prozentsatz, CPU der von Benutzerprogrammen verwendet wird.
	`wait`	CPUWarte	Der Prozentsatz der CPU ungenutzten, während auf den I/O-Zugriff gewartet wurde.
`diskIO`	`avgQueueLen`	Durchschn. Warteschlangengröße	Die Anzahl der Anfragen, die in der Warteschlange des I/O-Geräts warten.
	`avgReqSz`	Durchschn. Anforderungsgröße	Die durchschnittliche Anfragegröße in Kilobyte.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`await`	Festplatten-I/O-Warten	Die erforderliche Anzahl an Millisekunden für die Antwort auf Anfragen, einschließlich Warteschlangen- und Servicedauer.
	`device`	Nicht zutreffend	Die Kennung des verwendeten Datenträgers.
	`readIOsPS`	Gelesene IO/s	Die Anzahl der Lesevorgänge pro Sekunde.
	`readKb`	Insgesamt gelesen	Gesamtzahl der gelesenen Kilobyte.
	`readKbPS`	Gelesene KB/s	Die Anzahl der pro Sekunde gelesenen Kilobyte.
	`readLatency`	Lese-Latenz	Die verstrichene Zeit zwischen dem Senden einer Lese-I/O-Anforderung und deren Abschluss in Millisekunden. Diese Metrik ist nur für Amazon Aurora verfügbar.
	`readThroughput`	Lesedurchsatz	Die Menge des Netzwerkdurchsatzes, der von Anforderungen an den DB-Cluster verwendet wird, in Bytes pro Sekunde. Diese Metrik ist nur für Amazon Aurora verfügbar.
	`rrqmPS`	Rrqms	Die Anzahl der zusammengeführten Leseanfragen in der Warteschlange pro Sekunde.
	`tps`	TPS	Die Anzahl der I/O-Transaktionen pro Sekunde.
	`util`	Datenträger-I/O-Util	Der prozentuale CPU Zeitraum, in dem Anfragen gestellt wurden.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`writeIOsPS`	Geschriebene IO/s	Die Anzahl der Schreibvorgänge pro Sekunde.
	`writeKb`	Insgesamt geschrieben	Gesamtzahl der geschriebenen Kilobyte.
	`writeKbPS`	Geschriebene KB/s	Die Anzahl der pro Sekunde geschriebenen Kilobyte.
	`writeLatency`	Schreib-Latenz	Die durchschnittliche verstrichene Zeit zwischen dem Senden einer Schreib-I/O-Anforderung und deren Abschluss in Millisekunden. Diese Metrik ist nur für Amazon Aurora verfügbar.
	`writeThroughput`	Schreibdurchsatz	Die Menge des Netzwerkdurchsatzes, der von Antworten vom DB-Cluster verwendet wird, in Bytes pro Sekunde. Diese Metrik ist nur für Amazon Aurora verfügbar.
	`wrqmPS`	Wrqms	Die Anzahl der zusammengeführten Schreibanfragen in der Warteschlange pro Sekunde.
`physicalDeviceIO`	`avgQueueLen`	Durchschn. Größe der Warteschlange für physische Geräte	Die Anzahl der Anfragen, die in der Warteschlange des I/O-Geräts warten.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`avgReqSz`	Durchschn. Anfragegröße für physische Geräte	Die durchschnittliche Anfragegröße in Kilobyte.
	`await`	Datenträger-I/O-Abwarten für physische Geräte	Die erforderliche Anzahl an Millisekunden für die Antwort auf Anfragen, einschließlich Warteschlangen- und Servicedauer.
	`device`	Nicht zutreffend	Die Kennung des verwendeten Datenträgers.
	`readIOsPS`	Gelesene IO/s für physische Geräte	Die Anzahl der Lesevorgänge pro Sekunde.
	`readKb`	Insgesamt gelesen für physische Geräte	Gesamtzahl der gelesenen Kilobyte.
	`readKbPS`	Gelesene Kb/s für physische Geräte	Die Anzahl der pro Sekunde gelesenen Kilobyte.
	`rrqmPS`	Rrqms für physische Geräte	Die Anzahl der zusammengeführten Leseanfragen in der Warteschlange pro Sekunde.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`tps`	Physische Geräte TPS	Die Anzahl der I/O-Transaktionen pro Sekunde.
	`util`	Datenträger-I/O-Util für physische Geräte	Der prozentuale CPU Zeitraum, in dem Anfragen gestellt wurden.
	`writeIOsPS`	Geschriebene IO/s für physische Geräte	Die Anzahl der Schreibvorgänge pro Sekunde.
	`writeKb`	Insgesamt geschrieben für physische Geräte	Gesamtzahl der geschriebenen Kilobyte.
	`writeKbPS`	Geschriebene KB/s für physische Geräte	Die Anzahl der pro Sekunde geschriebenen Kilobyte.
	`wrqmPS`	Wrqms für physische Geräte	Die Anzahl der zusammengeführten Schreibanfragen in der Warteschlange pro Sekunde.
`fileSys`	`maxFiles`	Max Inodes	Die maximale Anzahl an Dateien, die für das Dateisystem erstellt werden können.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`mountPoint`	Nicht zutreffend	Der Pfad zum Dateisystem.
	`name`	Nicht zutreffend	Der Name des Dateisystems.
	`total`	Dateisystem insgesamt	Die Gesamtmenge des für das Dateisystem verfügbaren Speicherplatzes in Kilobyte.
	`used`	Verwendetes Dateisystem	Der durch Dateien belegte Speicherplatz im Dateisystem in Kilobyte.
	`usedFilePercent`	Gebrauchte Inodes	Der Prozentsatz von verfügbaren Dateien, die in Gebrauch sind.
	`usedFiles`	Used%	Die Anzahl der Dateien im Dateisystem.
	`usedPercent`	Verwendetes Dateisystem	Der prozentuale Anteil des verwendeten Speicherplatzes für das Dateisystem.
`loadAverageMinute`	`fifteen`	Last Durchschn. 15 Min	Die Anzahl der Prozesse, die in den letzten 15 Minuten CPU Zeit angefordert haben.
	`five`	Laden Durchschn. 5 Min	Die Anzahl der Prozesse, die in den letzten 5 Minuten CPU Zeit angefordert haben.

Gruppe	Metrik	Name der Konsole	Beschreibung
	one	Laden Durchschn. 1 Min	Die Anzahl der Prozesse, die in der letzten Minute CPU Zeit angefordert haben.
memory	active	Aktiver Speicher	Umfang des zugewiesenen Arbeitsspeichers in Kilobyte.
	buffers	Gepufferter Speicher	Umfang des verwendeten Arbeitsspeichers für die Pufferung von I/O-Anfragen vor dem Schreiben auf das Speichergerät, in Kilobyte.
	cached	Zwischenspeicher	Umfang des verwendeten Arbeitsspeichers für Caching von Dateisystem-basierter I/O.
	dirty	Geänderter Speicher	Die Anzahl der SpeicherseitenRAM, die geändert, aber nicht in den entsprechenden Datenblock im Speicher geschrieben wurden, in Kilobyte.
	free	Freier Speicher	Umfang des nicht zugewiesenen Arbeitsspeichers in Kilobyte.
	hugePagesFree	Huge Pages frei	Die Anzahl von freien "Huge Pages". "Huge Pages" sind eine Funktion des Linux-Kernel.
	hugePagesRsvd	Huge Pages reserviert	Die Anzahl von gebundenen "Huge Pages".
	hugePagesSize	Größe Huge Pages	Die Größe für jede Huge Pages-Einheit, in Kilobyte.
	hugePagesSurp	Überschuss Huge Pages	Die Anzahl der verfügbaren überzähligen Huge Pages.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`hugePagesTotal`	Huge Pages insgesamt	Die Gesamtzahl der Huge Pages.
	`inactive`	Inaktiver Speicher	Umfang der am seltensten verwendeten Memory-Pages in Kilobyte.
	`mapped`	Zugeordneter Speicher	Die Gesamtmenge der Dateisysteminhalte, die Arbeitsspeicher in einem Prozess-Adressraum zugeordnet ist, in Kilobyte.
	`pageTables`	Seitentabellen	Umfang des von Page-Tabellen verwendeten Arbeitsspeichers in Kilobyte.
	`slab`	Plattenspeicher	Umfang der wiederverwendbaren Kernel-Datenstrukturen in Kilobyte.
	`total`	Gesamtspeicher	Gesamtumfang des Arbeitsspeichers in Kilobyte.
	`writeback`	Writeback-Speicher	Die Menge der fehlerhaften SeitenRAM, die noch in den Backing-Speicher geschrieben werden, in Kilobyte.
`network`	`interface`	Nicht zutreffend	Die Kennung für die Netzwerkschnittstelle, die für die DB-Instance verwendet wird.
	`rx`	RX	Die Anzahl der pro Sekunde empfangenen Byte.
	`tx`	TX	Die Anzahl der pro Sekunde hochgeladenen Byte.
`processList`	`cpuUsedPc`	CPU %	Der Prozentsatz der vom CPU Prozess verwendeten.
	`id`	Nicht zutreffend	Die ID des Prozesses.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`memoryUsedPc`	MEM%	Prozentsatz des insgesamt vom Prozess benutzten Speichers.
	`name`	Nicht zutreffend	Der Name des Prozesses.
	`parentID`	Nicht zutreffend	Die Prozess-ID für den übergeordneten Prozess des aktuellen Prozesses.
	`rss`	RES	Die dem Prozess RAM zugewiesene Menge in Kilobyte.
	`tgid`	Nicht zutreffend	Die Thread-Gruppen-ID als Zahl, die die Prozess-ID darstellt, zu der ein Thread gehört. Diese Kennung wird verwendet, um Threads aus demselben Prozess zu gruppieren.
	`vss`	VIRT	Umfang des dem Prozess zugeteilten virtuellen Speichers in Kilobyte.
`swap`	`total`	Auslagerung	Die Menge des verfügbaren Swap-Arbeitsspeichers in Kilobyte.
	`in`	Auslagerungen in	Die Menge des von der Festplatte ausgelagerten Speichers in Kilobyte.
	`out`	Auslagerungen aus	Die Menge des auf die Festplatte ausgelagerten Speichers in Kilobyte.
	`free`	Kostenlose Auslagerung	Die Menge des freien Swap-Arbeitsspeichers in Kilobyte.
	`cached`	Committed Auslagerung	Umfang des Swap-Arbeitsspeichers, der als Cache-Speicher verwendet wird, in Kilobyte.

Gruppe	Metrik	Name der Konsole	Beschreibung
tasks	blocked	Gesperrte Aufgaben	Die Anzahl von blockierten Aufgaben.
	running	Laufende Aufgaben	Die Anzahl von laufenden Aufgaben.
	sleeping	Ruhende Aufgaben	Die Anzahl von Aufgaben im Ruhezustand.
	stopped	Gestoppte Aufgaben	Die Anzahl von angehaltenen Aufgaben.
	total	Aufgaben insgesamt	Die Gesamtanzahl der Aufgaben.
	zombie	Aufgaben Zombie	Die Anzahl der untergeordneten Aufgaben, die unter einer aktiven übergeordneten Aufgabe inaktiv sind.

Betriebssystemmetriken für Microsoft SQL Server

Gruppe	Metrik	Name der Konsole	Beschreibung
General	engine	Nicht zutreffend	Die Datenbank-Engine für die DB-Instance.
	instanceID	Nicht zutreffend	Die DB-Instance-Kennung.
	instanceResourceID	Nicht zutreffend	Eine unveränderliche Kennung für die DB-Instance, die für eine AWS Region eindeutig ist und auch als Log-Stream-ID verwendet wird.
	numVCPUs	Nicht zutreffend	Die Anzahl der virtuellen Dateien CPUs für die DB-Instance.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`timestamp`	Nicht zutreffend	Die Uhrzeit, zu der die Metriken erfasst wurden.
	`uptime`	Nicht zutreffend	Die Zeitdauer, für die die DB-Instance aktiv war.
	`version`	Nicht zutreffend	Die Version des JSON Stream-Formats der Betriebssystemmetriken.
`cpuUtilization`	`idle`	CPUIm Leerlauf	Der Prozentsatz CPU davon ist ungenutzt.
	`kern`	CPUKernel	Der Prozentsatz, CPU der vom Kernel verwendet wird.
	`user`	CPUBenutzer	Der Prozentsatz, CPU der von Benutzerprogrammen verwendet wird.
`disks`	`name`	Nicht zutreffend	Die ID für den Datenträger.
	`totalKb`	Festplattenspeicher insgesamt	Gesamtspeicherplatz des Datenträgers in Kilobyte.
	`usedKb`	Belegter Festplattenspeicher	Auf dem Datenträger verwendeter Speicherplatz in Kilobyte.
	`usedPc`	Belegter Festplattenspeicher %	Prozentsatz des auf dem Datenträger verwendeten Speicherplatzes.
	`availKb`	Verfügbarer Speicherplatz	Auf dem Datenträger verfügbarer Speicherplatz in Kilobyte.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`availPc`	% verfügbarer Festplattenspeicher	Prozentsatz des auf dem Datenträger verfügbaren Speicherplatzes.
	`rdCountPS`	Lesevorgänge/s	Die Anzahl der Lesevorgänge pro Sekunde
	`rdBytesPS`	Gelesene KB/s	Die Anzahl der pro Sekunde gelesenen Byte.
	`wrCountPS`	Geschriebene IO/s	Die Anzahl der Schreibvorgänge pro Sekunde.
	`wrBytesPS`	Geschriebene KB/s	Die Anzahl der pro Sekunde geschriebenen Byte.
`memory`	`commitTotKb`	Commit insgesamt	Umfang des durch Auslagerungsdateien abgesicherten belegten virtuellen Adressraums, d. h. der aktuelle Commit Charge. Dieser Wert setzt sich aus Hauptspeicher (RAM) und Festplatte (Seitendateien) zusammen.
	`commitLimitKb`	Maximales Commit	Der maximal mögliche Wert für die `commitTotKb` -Metrik. Dieser Wert ist die Summe aus der aktuellen Größe der Auslagerungsdatei zuzüglich des physischen Speichers, der für Seiteninhalte verfügbar ist, mit Ausnahme des Speichers, der nicht RAM auslagerungsfähigen Bereichen zugewiesen ist.
	`commitPeakKb`	Commit Spitze	Der Höchstwert der `commitTotKb` -Metrik seit dem letzten Start des Betriebssystems.

Gruppe	Metrik	Name der Konsole	Beschreibung
	kernTotKb	Kernelspeicher insgesamt	Die Summe des Arbeitsspeichers in den auslagerungsfähigen und nicht auslagerungsfähigen Kernel-Pools, in Kilobyte.
	kernPagedKb	Ausgelagerter Kernel-Speicher	Umfang des im auslagerungsfähigen Kernel-Pool verwendeten Arbeitsspeichers in Kilobyte.
	kernNonpagedKb	Nicht ausgelagerter Kernel-Speicher	Umfang des im nicht auslagerungsfähigen Kernel-Pool verwendeten Arbeitsspeichers in Kilobyte.
	pageSize	Seitengröße	Die Größe einer Auslagerungsdatei in Byte.
	physTotKb	Gesamtspeicher	Umfang des physischen Arbeitsspeichers in Kilobyte.
	physAvailKb	Verfügbarer Speicher	Umfang des verfügbaren physischen Arbeitsspeichers in Kilobyte.
	sqlServerTotKb	SQLGesamtspeicher des Servers	Die Menge an Speicher, die dem SQL Server zugewiesen wurde, in Kilobyte.
	sysCacheKb	System-Cache	Umfang des nicht Systemcache-Arbeitsspeichers in Kilobyte.
network	interface	Nicht zutreffend	Die Kennung für die Netzwerkschnittstelle, die für die DB-Instance verwendet wird.
	rdBytesPS	Im Netzwerk gelesen KB/s	Die Anzahl der pro Sekunde empfangenen Byte.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`wrBytesPS`	Im Netzwerk geschrieben KB/s	Die Anzahl der pro Sekunde gesendeten Byte.
`processList`	`cpuUsedPc`	Used%	Der Prozentsatz, der vom CPU Prozess verwendet wird.
	`memUsedPc`	MEM%	Prozentsatz des insgesamt vom Prozess benutzten Speichers.
	`name`	Nicht zutreffend	Der Name des Prozesses.
	`pid`	Nicht zutreffend	Die ID des Prozesses. Dieser Wert ist für Prozesse, die Amazon gehören, nicht vorhandenRDS.
	`ppid`	Nicht zutreffend	Die Prozess-ID für den übergeordneten Prozess dieses Prozesses. Dieser Wert ist nur für untergeordnete Prozesse vorhanden.
	`tid`	Nicht zutreffend	Die Thread-Kennung. Dieser Wert ist nur für Threads vorhanden. Der Eigentümerprozess kann mithilfe des `pid`-Werts identifiziert werden.
	`workingSetKb`	Nicht zutreffend	Der Speicherumfang in der privaten Datenmenge plus der Speicherumfang, der durch den Prozess verwendet wird und mit anderen Prozessen geteilt werden kann, in Kilobyte.
	`workingSetPrivKb`	Nicht zutreffend	Der Speicherumfang, der durch einen Prozess verwendet wird, aber nicht mit anderen Prozessen geteilt werden kann, in Kilobyte.

Gruppe	Metrik	Name der Konsole	Beschreibung
	`workingSetShareableKb`	Nicht zutreffend	Der Speicherumfang, der durch einen Prozess verwendet wird und mit anderen Prozessen geteilt werden kann, in Kilobyte.
	`virtKb`	Nicht zutreffend	Umfang des vom Prozess verwendeten virtuellen Adressbereichs in Kilobyte. Die Nutzung eines virtuellen Adressbereichs impliziert nicht unbedingt die Verwendung von Datenträger- oder Hauptspeicher-Pages.
`system`	`handles`	Handles	Die Anzahl der Handles, die das System verwendet.
	`processes`	Prozesse	Die Anzahl der Prozesse, die im System ablaufen.
	`threads`	Threads	Die Anzahl der Threads, die im System ablaufen.

Überwachung von Ereignissen, Protokollen und Streams in einem Amazon RDS Amazon

Wenn Sie Ihre Amazon RDS Amazon und Ihre anderen AWS Lösungen überwachen, ist es Ihr Ziel, Folgendes aufrechtzuerhalten:

- Zuverlässigkeit
- Verfügbarkeit
- Leistung
- Sicherheit

Metriken in einer RDS Amazon-Instance überwachen erklärt, wie Sie die Instance mithilfe von Metriken überwachen. Eine vollständige Lösung muss auch Datenbankereignisse, Protokolldateien und Aktivitätsströme überwachen. AWS bietet Ihnen die folgenden Überwachungstools:

- Amazon EventBridge ist ein serverloser Event-Bus-Service, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. EventBridge liefert einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, oftware-as-a S-Service (SaaS) - Anwendungen und AWS Diensten. EventBridge leitet diese Daten an Ziele weiter wie AWS Lambda. Auf diese Weise können Sie Ereignisse überwachen, die in Services auftreten, und ereignisgesteuerte Architekturen erstellen. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.
- Amazon CloudWatch Logs bietet eine Möglichkeit, Ihre Protokolldateien von Amazon RDS Amazon und anderen Quellen zu überwachen AWS CloudTrail, zu speichern und darauf zuzugreifen. Amazon CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.
- AWS CloudTrailerfasst API Anrufe und damit verbundene Ereignisse, die von Ihnen oder in Ihrem Namen getätigt wurden AWS-Konto. CloudTrail übermittelt die Protokolldateien an einen Amazon S3 S3-Bucket, den Sie angeben. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Anrufe erfolgten. Weitere Informationen finden Sie im AWS CloudTrail -Benutzerhandbuch.

- Database Activity Streams ist eine Amazon RDS , die einen Stream der Aktivitäten in Ihrer nahezu in Echtzeit bereitstellt. Amazon RDS überträgt Aktivitäten in einen Amazon Kinesis Kinesis-Datenstream. Der Kinesis Stream wird automatisch erstellt. Von Kinesis aus können Sie AWS Dienste wie Amazon Data Firehose konfigurieren und AWS Lambda den Stream nutzen und die Daten speichern.

Themen

- Protokolle, Ereignisse und Streams in der RDS Amazon-Konsole anzeigen
- Überwachung von Amazon-Ereignissen
- Überwachung von Amazon RDS Amazon
- überwachen Amazon RDS API ruft an AWS CloudTrail
- Überwachung von Amazon RDS mithilfe von Datenbankaktivitätsstreams
- Überwachung von Bedrohungen mit Amazon GuardDuty RDS Protection

Protokolle, Ereignisse und Streams in der RDS Amazon-Konsole anzeigen

Amazon RDS lässt sich integrieren, AWS-Services um Informationen zu Protokollen, Ereignissen und Datenbankaktivitätsströmen in der RDS Konsole anzuzeigen.

Auf der Registerkarte Logs & Events für Ihren werden die folgenden Informationen angezeigt:

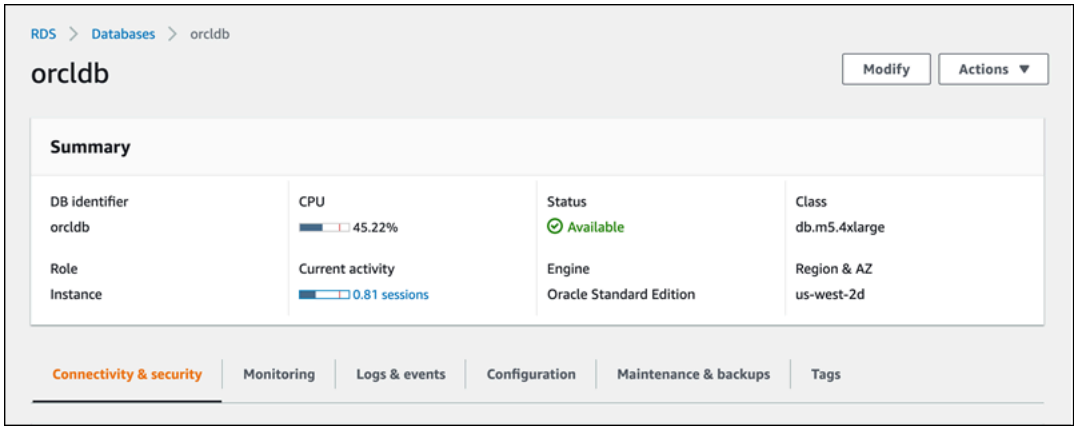
- CloudWatch Amazon-Alarme — Zeigt alle metrischen Alarme an, die Sie für die DB-Instance konfiguriert haben. Wenn Sie keine Alarme konfiguriert haben, können Sie sie in der RDS Konsole erstellen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mit Amazon CloudWatch.
- Aktuelle Ereignisse — Zeigt eine Zusammenfassung der Ereignisse (Umgebungsänderungen) für Ihre an. Weitere Informationen finden Sie unter RDSAmazon-Ereignisse anzeigen.
- Protokolle – Zeigt Datenbankprotokolldateien an, die von einer DB-Instance generiert wurden. Weitere Informationen finden Sie unter Überwachung von Amazon RDS Amazon.

Die Registerkarte Konfiguration zeigt Informationen über Datenbankaktivitäts-Streams an.

Um Logs, Ereignisse und Streams für Ihren in der RDS Konsole anzuzeigen

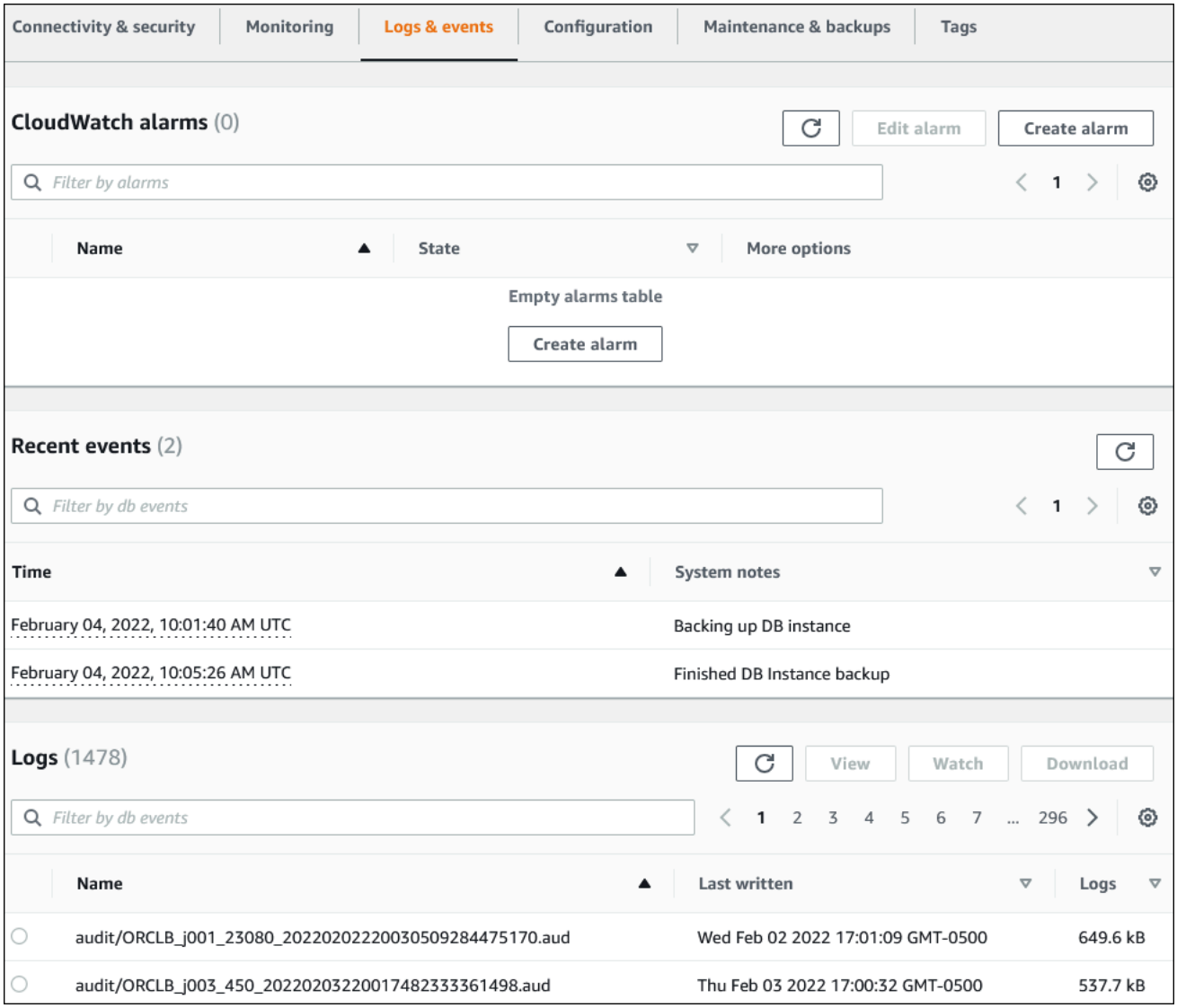
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen des -DB-Instance- an, das Sie überwachen möchten.

   Der Bereich Datenbanken wird angezeigt. Das folgende Beispiel zeigt eine Oracle-Datenbank mit dem Namen `orclb` .



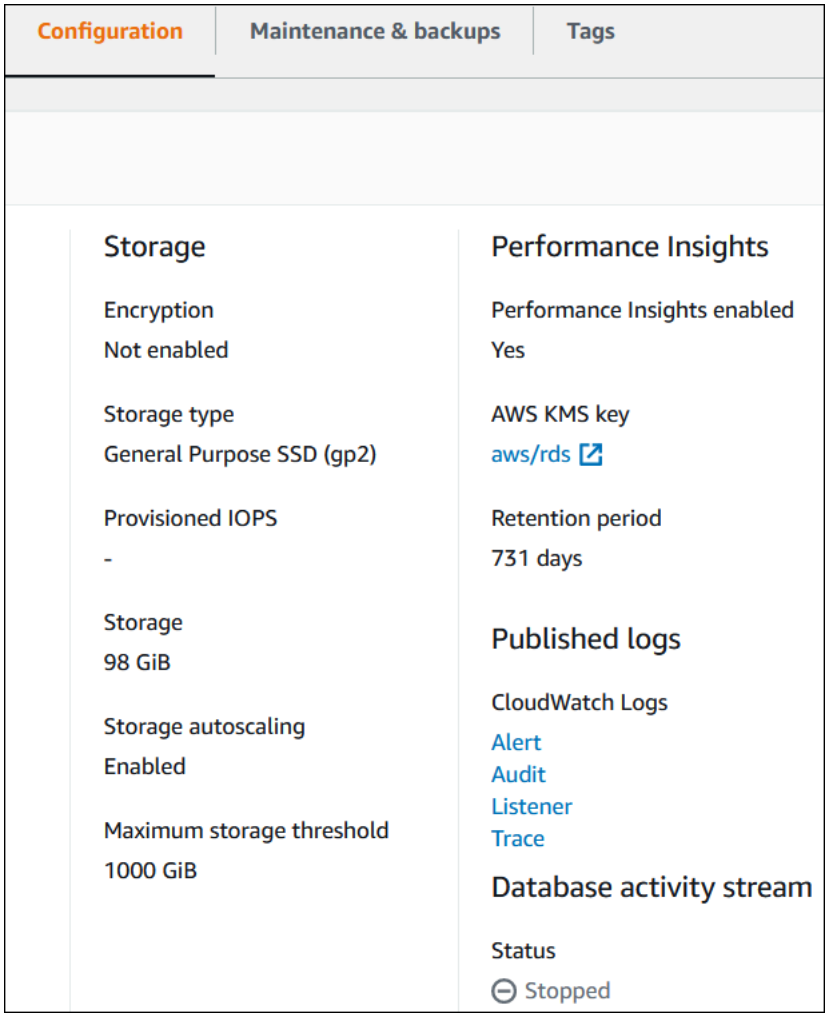
4. Wählen Sie Logs & Events (Protokolle und Ereignisse).

   Der Abschnitt Logs & Events (Protokolle und Ereignisse) wird angezeigt.



5. Wählen Sie Konfiguration.

   Das folgende Beispiel zeigt den Status der Datenbankaktivitätsstreams für Ihre DB-Instance an.


Configuration
Maintenance & backups
Tags
Storage
Encryption
Not enabled
Storage type
General Purpose SSD (gp2)
Provisioned IOPS
-
Storage
98 GiB
Storage autoscaling
Enabled
Maximum storage threshold
1000 GiB
Performance Insights
Performance Insights enabled
Yes
AWS KMS key
aws/rds
Retention period
731 days
Published logs
CloudWatch Logs
Alert
Audit
Listener
Trace
Database activity stream
Status
Stopped

Überwachung von Amazon-Ereignissen

Ein Ereignis weist auf eine Änderung in einer Umgebung hin. Dies kann eine AWS Umgebung, ein SaaS-Partnerdienst oder eine Anwendung oder eine benutzerdefinierte Anwendung oder ein benutzerdefinierter Dienst sein. Eine Beschreibung der finden Sie unterAmazon RDS-Ereigniskategorien und Ereignisnachrichten .

Themen

- Übersicht der Ereignisse für Amazon RDS
- RDSAmazon-Ereignisse anzeigen
- Mit RDS Amazon-Event-Benachrichtigungen arbeiten
- Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird
- Amazon RDS-Ereigniskategorien und Ereignisnachrichten

Übersicht der Ereignisse für Amazon RDS

Ein RDSEreignis weist auf eine Änderung in der Amazon-Umgebung hin. Amazon RDS Amazon generiert beispielsweise ein Ereignis, wenn ändert sich von „Ausstehend“ in „Wird ausgeführt“. Amazon RDS liefert Ereignisse nahezu EventBridge in Echtzeit an.

> **Note**
>
> Amazon RDS sendet Ereignisse nach bestem Wissen und Gewissen aus. Wir empfehlen Ihnen, keine Programme zu schreiben, die von der Reihenfolge oder dem Vorhandensein von Benachrichtigungsereignissen abhängen, da diese möglicherweise nicht in der richtigen Reihenfolge vorliegen oder fehlen.

Amazon RDS zeichnet Ereignisse auf, die sich auf die folgenden Ressourcen beziehen:

- DB-Instances

  Eine Liste der DB-Instance-Ereignisse finden Sie unter DB-Instance-Ereignisse.
- DB-Parametergruppen

  Eine Liste der Ereignisse der DB-Parametergruppe finden Sie unter DB-Parametergruppenereignisse.

- DB-Sicherheitsgruppen

  Eine Liste der Ereignisse der DB-Sicherheitsgruppe finden Sie unter DB-Sicherheitsgruppenereignisse.
- DB-Snapshots

  Eine Liste der DB-Snapshot-Ereignisse finden Sie unter DB-Snapshot-Ereignisse.
- RDSProxy-Ereignisse

  Eine Liste der RDS Proxy-Ereignisse finden Sie unterRDS-Proxy-Ereignisse.
- Blau/Grün-Bereitstellungsereignisse

  Eine Liste der Blau/Grün-Bereitstellungsereignisse finden Sie unter Blau/Grün-Bereitstellungsereignisse.

Diese Informationen beinhalten Folgendes:

- Das Datum und die Uhrzeit der Veranstaltung
- Der Quellname und der Quelltyp des Ereignisses
- Eine Nachricht, die mit dem Ereignis verknüpft ist
- Ereignisbenachrichtigungen enthalten Tags zum Zeitpunkt, an dem die Nachricht gesendet wurde, und geben möglicherweise nicht die Tags zum Zeitpunkt des Eintritts des Ereignisses wieder.

RDSAmazon-Ereignisse anzeigen

Sie können die folgenden Veranstaltungsinformationen für Ihre Amazon RDS Amazon abrufen:

- Ressourcenname
- Ressourcentyp
- Zeitpunkt des Ereignisses
- Zusammenfassung der Ereignisbenachrichtigung

Sie können in den folgenden Bereichen von auf Ereignisse zugreifen AWS Management Console:

- Die Registerkarte „Ereignisse“, auf der Ereignisse der letzten 24 Stunden angezeigt werden.
- Die Tabelle Aktuelle Ereignisse im Abschnitt Protokolle und Ereignisse auf der Registerkarte Datenbanken, in der Ereignisse der letzten 2 Wochen angezeigt werden können.

Sie können Ereignisse auch mithilfe des AWS CLI Befehls describe-events oder der DescribeEventsRDSAPIOperation abrufen. Wenn Sie das AWS CLI oder das verwenden, RDS API um Ereignisse anzuzeigen, können Sie Ereignisse der letzten 14 Tage abrufen.

> **Note**
>
> Wenn Sie Ereignisse für einen längeren Zeitraum speichern müssen, können Sie RDS Amazon-Ereignisse an senden EventBridge. Weitere Informationen finden Sie unter Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird

Eine Beschreibung der Amazon RDS Amazon finden Sie unterAmazon RDS-Ereigniskategorien und Ereignisnachrichten .

Ausführliche Informationen zu Ereignissen AWS CloudTrail, einschließlich Anforderungsparametern, finden Sie unterCloudTrail Ereignisse.

Konsole

Um alle RDS Amazon-Ereignisse der letzten 24 Stunden zu sehen

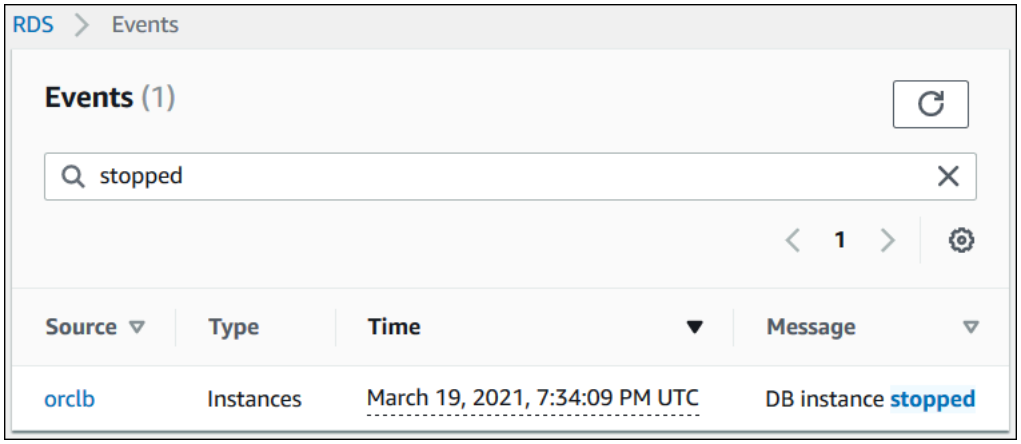
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich die Option Events.

   Die verfügbaren Ereignisse erscheinen in einer Liste.

3. (Optional) Geben Sie einen Suchbegriff ein, um Ihre Ergebnisse zu filtern.

   Das folgende Beispiel zeigt eine Liste von Ereignissen, die nach den Zeichen **`stopped`** gefiltert sind.



AWS CLI

Wenn Sie alle Ereignisse anzeigen möchten, die in der letzten Stunde generiert wurden, rufen Sie describe-events ohne Parameter auf.

```
aws rds describe-events
```

Die folgende Beispielausgabe zeigt, dass eine DB-Instance gestoppt wurde.

```
{
    "Events": [
        {
            "EventCategories": [
                "notification"
            ],
            "SourceType": "db-instance",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:testinst",
            "Date": "2022-04-22T21:31:00.681Z",
            "Message": "DB instance stopped",
            "SourceIdentifier": "testinst"
        }
    ]
```

```
}
```

Um alle RDS Amazon-Ereignisse der letzten 10080 Minuten (7 Tage) anzuzeigen, rufen Sie den AWS CLI Befehl describe-events auf und setzen Sie den `--duration` Parameter auf. `10080`

```
aws rds describe-events --duration 10080
```

Das folgende Beispiel zeigt die Ereignisse im angegebenen Zeitraum für die DB-Instance *`test-instance`*.

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance \
    --start-time 2022-03-13T22:00Z \
    --end-time 2022-03-13T23:59Z
```

Die folgende Beispielausgabe zeigt den Status eines Backups.

```
{
    "Events": [
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Backing up DB instance",
            "Date": "2022-03-13T23:09:23.983Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        },
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Finished DB Instance backup",
            "Date": "2022-03-13T23:15:13.049Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        }
    ]
```

```
}
```

API

Sie können alle RDS Amazon-Instance-Ereignisse der letzten 14 Tage anzeigen, indem Sie den DescribeEventsRDSAPIVorgang aufrufen und den `Duration` Parameter auf setzen`20160`.

Mit RDS Amazon-Event-Benachrichtigungen arbeiten

Amazon RDS verwendet den Amazon Simple Notification Service (AmazonSNS), um Benachrichtigungen bereitzustellen, wenn ein RDS Amazon-Ereignis eintritt. Diese Benachrichtigungen können in jedem von Amazon SNS für eine AWS Region unterstützten Benachrichtigungsformular erfolgen, z. B. in einer E-Mail, einer Textnachricht oder einem Anruf an einen HTTP Endpunkt.

Themen

- Übersicht über RDS Amazon-Event-Benachrichtigungen
- Erteilen von Berechtigungen zum Veröffentlichen von Benachrichtigungen in einem Amazon-SNS-Thema
- RDSAmazon-Veranstaltungsbenachrichtigung abonnieren
- Tags und Attribute für RDS Amazon-Event-Benachrichtigungen
- Amazon-Abonnements für RDS Event-Benachrichtigungen auflisten
- Ein Abonnement für RDS Amazon-Event-Benachrichtigungen ändern
- Hinzufügen einer Quell-ID zu einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen
- Entfernen einer Quell-ID aus einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen
- Auflistung der Kategorien Amazon RDS Amazon-Event-Benachrichtigungen
- Löschen eines Abonnements für RDS Amazon-Veranstaltungsbenachrichtigungen

Übersicht über RDS Amazon-Event-Benachrichtigungen

Amazon RDS gruppiert Ereignisse in Kategorien, die Sie abonnieren können, sodass Sie benachrichtigt werden können, wenn ein Ereignis in dieser Kategorie eintritt.

Themen

- RDSRessourcen, die für ein Event-Abonnement in Frage kommen
- Grundlegendes Verfahren für das Abonnieren von RDS Amazon-Event-Benachrichtigungen
- Zustellung von RDS Ereignisbenachrichtigungen
- Abrechnung für RDS Amazon-Event-Benachrichtigungen
- Beispiele für mit Amazon EventBridge

RDSRessourcen, die für ein Event-Abonnement in Frage kommen

Sie können eine Veranstaltungskategorie für die folgenden Ressourcen abonnieren:

- DB-Instance
- DB-Snapshot
- DB-Parametergruppe
- DB-Sicherheitsgruppe
- RDSProxy
- Kundenspezifische Motorversionen

Wenn Sie zum Beispiel die Backup-Kategorie für eine bestimmte DB-Instance abonnieren, werden Sie immer dann benachrichtigt, wenn ein Backup-bezogenes Ereignis eintritt, das die DB-Instance betrifft. Wenn Sie eine Konfigurationsänderungskategorie für eine DB-Instance abonnieren, werden Sie benachrichtigt, sobald die DB-Instance geändert wird. Außerdem erhalten Sie eine Benachrichtigung, wenn ein Abonnement für Ereignisbenachrichtigungen geändert wird.

Möglicherweise möchten Sie mehrere verschiedene Abonnements erstellen. Sie könnten beispielsweise ein Abonnement erstellen, das alle Ereignisbenachrichtigungen für alle DB-Instances empfängt, und ein anderes, das nur kritische Ereignisse für eine Teilmenge der DB-Instances enthält. Geben Sie für das zweite Abonnement eine oder mehrere DB-Instances im Filter an.

Grundlegendes Verfahren für das Abonnieren von RDS Amazon-Event-Benachrichtigungen

Das Verfahren zum Abonnieren der RDS Amazon-Veranstaltungsbenachrichtigung ist wie folgt:

1. Sie erstellen ein Abonnement für RDS Amazon-Ereignisbenachrichtigungen mithilfe der RDS Amazon-Konsole, AWS CLI, oderAPI.

   Amazon RDS verwendet ein SNS Amazon-Thema, um jedes Abonnement zu identifizieren. ARN Die RDS Amazon-Konsole erstellt das ARN für Sie, wenn Sie das Abonnement erstellen. Erstellen Sie ARN das mit der SNS Amazon-Konsole AWS CLI, dem oder dem Amazon SNSAPI.
2. Amazon RDS sendet eine Bestätigungs-E-Mail oder eine SMS Bestätigungsnachricht an die Adressen, die Sie mit Ihrem Abonnement angegeben haben.
3. Klicken Sie auf den Link in der erhaltenen Benachrichtigung, um das Abonnement zu bestätigen.
4. Die RDS Amazon-Konsole aktualisiert den Abschnitt Meine Event-Abonnements mit dem Status Ihres Abonnements.

5. Amazon RDS beginnt, die Benachrichtigungen an die Adressen zu senden, die Sie bei der Erstellung des Abonnements angegeben haben.

Weitere Informationen zur Identitäts- und Zugriffsverwaltung bei der Nutzung von Amazon SNS finden Sie unter Identitäts- und Zugriffsverwaltung SNS in Amazon im Amazon Simple Notification Service Developer Guide.

Sie können AWS Lambda es verwenden, um Ereignisbenachrichtigungen von einer DB-Instance aus zu verarbeiten. Weitere Informationen finden Sie unter Using AWS Lambda with Amazon RDS im AWS Lambda Developer Guide.

Zustellung von RDS Ereignisbenachrichtigungen

Amazon RDS sendet Benachrichtigungen an die Adressen, die Sie bei der Erstellung des Abonnements angeben. Die Benachrichtigung kann Nachrichtenattribute mit einschließen, die strukturierte Metadaten zu der Nachricht zur Verfügung stellen. Weitere Informationen über Nachrichtenattribute finden Sie unter Amazon RDS-Ereigniskategorien und Ereignisnachrichten .

Es kann bis zu fünf Minuten dauern, bis Ereignisbenachrichtigungen zugestellt werden.

> **Important**
>
> Amazon garantiert RDS nicht die Reihenfolge der Ereignisse, die in einem Event-Stream gesendet werden. Die Reihenfolge der Ereignisse kann sich ändern.

Wenn Amazon eine Benachrichtigung SNS an ein Abonnement HTTP oder einen HTTPS Endpunkt sendet, hat die POST an den Endpunkt gesendete Nachricht einen Nachrichtentext, der ein JSON Dokument enthält. Weitere Informationen finden Sie unter SNSAmazon-Nachrichten und JSON Formate im Amazon Simple Notification Service Developer Guide.

Sie können so konfigurierenSNS, dass Sie mit Textnachrichten benachrichtigt werden. Weitere Informationen finden Sie unter Mobile Textnachrichten (SMS) im Amazon Simple Notification Service Developer Guide.

Um Benachrichtigungen zu deaktivieren, ohne ein Abonnement zu löschen, wählen Sie in der RDS Amazon-Konsole Nein für Aktiviert. Oder Sie können den `Enabled` Parameter so einstellen, dass er AWS CLI oder Amazon `false` verwendet RDSAPI.

Abrechnung für RDS Amazon-Event-Benachrichtigungen

Die Abrechnung der RDS Amazon-Veranstaltungsbenachrichtigung erfolgt über AmazonSNS. Bei der Nutzung von Event Notification SNS fallen Amazon-Gebühren an. Weitere Informationen zur SNS Amazon-Abrechnung finden Sie unter Amazon Simple Notification Service — Preise.

Beispiele für mit Amazon EventBridge

Die folgenden Beispiele veranschaulichen verschiedene Arten von Amazon-Veranstaltungen im JSON Format. Ein Tutorial, das Ihnen zeigt, wie Sie Ereignisse im JSON Format erfassen und anzeigen können, finden Sie unterTutorial: Statusänderungen der DB-Instance mithilfe von Amazon protokollieren EventBridge.

Themen

- Beispiel für ein DB-Instance-Ereignis
- Beispiel für ein Ereignis der DB-Parametergruppe
- Beispiel für ein DB-Snapshot-Ereignis

Beispiel für ein DB-Instance-Ereignis

Das Folgende ist ein Beispiel für ein DB-Instance-Ereignis im JSON Format. Das Ereignis zeigt, dass ein Multi-AZ-Failover für die angegebene Instance RDS durchgeführt wurde. `my-db-instance` Die Ereignis-ID lautet RDS - EVENT -0049.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Instance Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
  ],
  "detail": {
    "EventCategories": [
      "failover"
    ],
    "SourceType": "DB_INSTANCE",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
```

```
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "A Multi-AZ failover has completed.",
    "SourceIdentifier": "my-db-instance",
    "EventID": "RDS-EVENT-0049"
  }
}
```

Beispiel für ein Ereignis der DB-Parametergruppe

Im Folgenden finden Sie ein Beispiel für ein DB-Parametergruppen-Ereignis im JSON Format. Das Ereignis zeigt, dass der Parameter `time_zone` in der Parametergruppe `my-db-param-group` aktualisiert wurde. Die Ereignis-ID lautet RDS - EVENT -0037.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Parameter Group Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PARAM",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Updated parameter time_zone to UTC with apply method immediate",
    "SourceIdentifier": "my-db-param-group",
    "EventID": "RDS-EVENT-0037"
  }
}
```

Beispiel für ein DB-Snapshot-Ereignis

Im Folgenden finden Sie ein Beispiel für ein DB-Snapshot-Ereignis im JSON Format. Das Ereignis zeigt das Löschen des Snapshots mit dem Namen `my-db-snapshot`. Die Ereignis-ID lautet RDS - EVENT -0041.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Snapshot Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot"
  ],
  "detail": {
    "EventCategories": [
      "deletion"
    ],
    "SourceType": "SNAPSHOT",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Deleted manual snapshot",
    "SourceIdentifier": "my-db-snapshot",
    "EventID": "RDS-EVENT-0041"
  }
}
```

Erteilen von Berechtigungen zum Veröffentlichen von Benachrichtigungen in einem Amazon-SNS-Thema

Um Amazon RDS Berechtigungen zur Veröffentlichung von Benachrichtigungen zu einem Amazon Simple Notification Service (Amazon SNS) -Thema zu gewähren, fügen Sie dem Zielthema eine AWS Identity and Access Management (IAM) -Richtlinie hinzu. Weitere Informationen zu Berechtigungen finden Sie unter Beispielfälle für die Zugriffskontrolle von Amazon Simple Notification Service im Entwicklerhandbuch für Amazon Simple Notification .

Standardmäßig verfügt ein Amazon-SNS-Thema über eine Richtlinie, die es allen Amazon-RDS-Ressourcen innerhalb desselben Kontos ermöglicht, Benachrichtigungen darin zu veröffentlichen. Sie können eine benutzerdefinierte Richtlinie anfügen, um kontoübergreifende Benachrichtigungen zuzulassen oder den Zugriff auf bestimmte Ressourcen einzuschränken.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die Sie dem Amazon-SNS-Zielthema anfügen. Sie beschränkt das Thema auf DB-Instances mit Namen, die dem angegebenen Präfix entsprechen. Geben Sie die folgenden Werte an, um diese Richtlinie zu verwenden:

- `Resource` – Den Amazon-Ressourcennamen (ARN) für das Amazon-SNS-Thema
- `SourceARN` – Ihren RDS-Ressourcen-ARN
- `SourceAccount`— Ihre ID AWS-Konto

Eine Liste der Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Von Amazon RDS definierte Ressourcen in der Service Authorization Reference.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.rds.amazonaws.com"
      },
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic_name",
```

```
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:prefix-*"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

RDSAmazon-Veranstaltungsbenachrichtigung abonnieren

Am einfachsten können Sie ein Abonnement mit der RDS Konsole erstellen. Wenn Sie Abonnements für Ereignisbenachrichtigungen mithilfe von CLI oder erstellen möchtenAPI, müssen Sie ein Amazon Simple Notification Service-Thema erstellen und dieses Thema über die SNS Amazon-Konsole oder Amazon abonnieren SNSAPI. Sie müssen auch den Amazon-Ressourcennamen (ARN) des Themas beibehalten, da er beim Senden von CLI Befehlen oder API Vorgängen verwendet wird. Informationen zum Erstellen und Abonnieren eines SNS Themas finden Sie unter Erste Schritte mit Amazon SNS im Amazon Simple Notification Service Developer Guide.

Sie können die Art der Quelle angeben, über die Sie benachrichtigt werden möchten, und die RDS Amazon-Quelle, die das Ereignis auslöst:

Source type (Quellentyp)

Der Quelltyp. Beispiel: Source type (Quellentyp) könnte Instances sein. Sie müssen einen Quellentyp auswählen.

Resources einbeziehen

Die RDS Amazon-Ressourcen, die die Ereignisse generieren. Sie könnten beispielsweise Bestimmte Instanzen auswählen und dann myDBInstance1 auswählen.

In der folgenden Tabelle wird das Ergebnis erläutert, wenn Sie angeben oder nicht angeben ***Resources*** einbeziehen.

Einzuschließende Ressourcen	Beschreibung	Beispiel
Angegeben	RDSbenachrichtigt Sie nur über alle Ereignisse für die angegebene Ressource.	Wenn Ihr Quelltyp Instances und Ihre Ressource myDBInstance1 ist, werden Sie RDS nur über alle Ereignisse für myDBInstance 1 benachrichtigt.
Nicht angegeben	RDSbenachrichtigt Sie über die Ereignisse für den angegebenen Quelltyp für all Ihre RDS Amazon-Ressourcen.	Wenn Ihr Quelltyp Instances ist, werden Sie über alle instanzbe

Einzuschließende Ressourcen	Beschreibung	Beispiel
		zogenen Ereignisse in Ihrem Konto RDS benachrichtigt.

Ein SNS Amazon-Thema-Abonnent erhält standardmäßig jede Nachricht, die zu dem Thema veröffentlicht wurde. Um eine Teilmenge der Nachrichten zu erhalten, müssen Abonnenten dem Themen-Abonnement eine Filterrichtlinie zuweisen. Weitere Informationen zur SNS Nachrichtenfilterung finden Sie unter SNSAmazon-Nachrichtenfilterung im Amazon Simple Notification Service Developer Guide.

Konsole

Um eine RDS Ereignisbenachrichtigung zu abonnieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Event subscriptions (Ereignisabonnements).
3. Wählen Sie im Bereich Ereignisabonnements Ereignisabonnement erstellen aus.
4. Geben Sie Ihre Abonnementdetails wie folgt ein:

    a. Geben Sie unter Name einen Namen für das Abonnement für Ereignisbenachrichtigungen ein.

    b. Führen Sie für den Abschnitt Send notifications to (Benachrichtigung senden an) einen der folgenden Schritte aus:

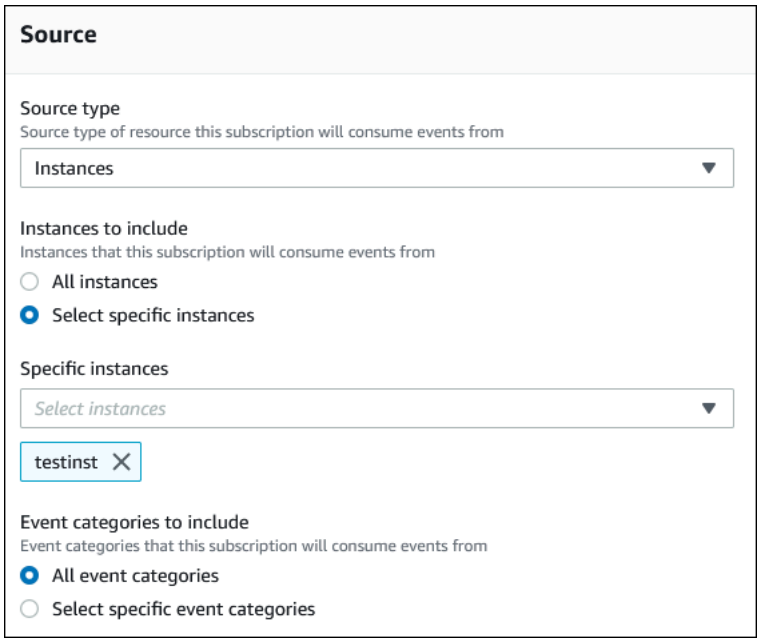
    - Wählen Sie New email topic (Neues E-Mail-Thema) aus. Geben Sie einen Namen für Ihr E-Mail-Thema und eine Liste der Empfänger ein. Wir empfehlen, dass Sie die Ereignisabonnements mit derselben E-Mail-Adresse konfigurieren wie Ihren primären Kontokontakt. Die Empfehlungen, Serviceereignisse und persönlichen Zustandsnachrichten werden über verschiedene Kanäle gesendet. Die Abonnements für dieselbe E-Mail-Adresse stellen sicher, dass alle Nachrichten an einem Ort zusammengefasst werden.
    - Wählen Sie Amazon-Ressourcenname (ARN). Wählen Sie dann ein vorhandenes Amazon SNS ARN für ein SNS Amazon-Thema aus.

Wenn Sie ein Thema verwenden möchten, für das serverseitige Verschlüsselung aktiviert wurde (SSE), gewähren Sie Amazon RDS die erforderlichen Zugriffsberechtigungen für das AWS KMS key. Weitere Informationen finden Sie unter Aktivieren der Kompatibilität zwischen Ereignisquellen von AWS Diensten und verschlüsselten Themen im Amazon Simple Notification Service Developer Guide.

c. Wählen Sie unter Quelltyp einen Quelltyp aus. Wählen Sie beispielsweise Instances oder Parameter groups (Parametergruppen) aus.

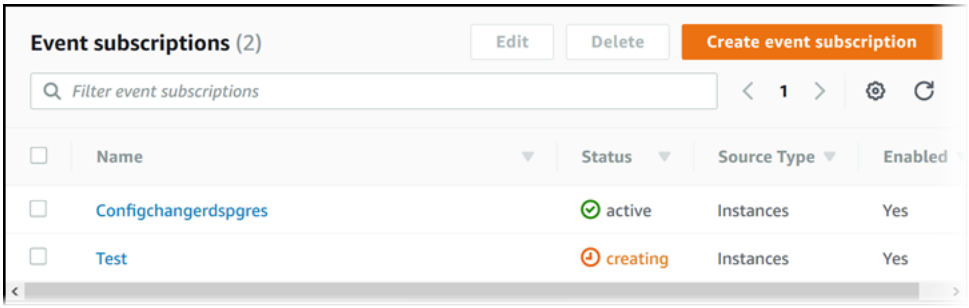
d. Wählen Sie die Ereigniskategorien und Ressourcen aus, für die Sie Ereignisbenachrichtigungen erhalten möchten.

Im folgenden Beispiel werden Ereignisbenachrichtigungen für die DB-Instance mit dem Namen `testinst` konfiguriert.



e. Wählen Sie Create (Erstellen) aus.

Die RDS Amazon-Konsole zeigt an, dass das Abonnement erstellt wird.



AWS CLI

Verwenden Sie den AWS CLI `create-event-subscription`Befehl, um die RDS Ereignisbenachrichtigung zu abonnieren. Nutzen Sie die folgenden erforderlichen Parameter:

- `--subscription-name`
- `--sns-topic-arn`

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-event-subscription \
    --subscription-name myeventsubscription \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS \
    --enabled
```

Windows:

```
aws rds create-event-subscription ^
    --subscription-name myeventsubscription ^
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS ^
    --enabled
```

API

Rufen Sie die RDS API Amazon-Funktion auf, um die RDS Amazon-Ereignisbenachrichtigung zu abonnieren `CreateEventSubscription`. Nutzen Sie die folgenden erforderlichen Parameter:

- `SubscriptionName`
- `SnsTopicArn`

Tags und Attribute für RDS Amazon-Event-Benachrichtigungen

Wenn Amazon eine Ereignisbenachrichtigung an Amazon Simple Notification Service (SNS) oder Amazon RDS sendet EventBridge, enthält die Benachrichtigung Nachrichtenattribute und Ereignis-Tags. RDSsendet die Nachrichtenattribute separat zusammen mit der Nachricht, wobei sich die Ereignis-Tags im Hauptteil der Nachricht befinden. Verwenden Sie die Nachrichtenattribute und die RDS Amazon-Tags, um Metadaten zu Ihren Ressourcen hinzuzufügen. Sie können diese Tags mit Ihren eigenen Notationen über die DB-Instances ändern. Weitere Informationen zum Taggen von RDS Amazon-Ressourcen finden Sie unterVerschlagwortung von Amazon-Ressourcen RDS.

Standardmäßig erhalten Amazon SNS und Amazon jede EventBridge an sie gesendete Nachricht. SNSund EventBridge kann die Nachricht filtern und die Benachrichtigungen an den bevorzugten Kommunikationsmodus senden, z. B. eine E-Mail, eine Textnachricht oder einen Anruf an einen HTTP Endpunkt.

> **Note**
>
> Die Benachrichtigung, die in einer E-Mail oder einer SMS gesendet wird, enthält keine Ereignis-Tags.

Die folgende Tabelle zeigt die Nachrichtenattribute für RDS Ereignisse, die an den Abonnenten des Themas gesendet wurden.

RDSAmazon-Ereignisattribut	Beschreibung
EventID	Bezeichner für die RDS Ereignisnachricht, zum Beispiel RDS - EVENT -0006.
Ressource	Der ARN Bezeichner für die Ressource, die das Ereignis ausgibt, `arn:aws:rds:ap-southeast-2:123456789012:db:database-1` z. B.

Die RDS Tags enthalten Daten über die Ressource, die von dem Dienstereignis betroffen war. RDSfügt den aktuellen Status der Tags im Nachrichtentext hinzu, wenn die Benachrichtigung an SNS oder gesendet wird EventBridge.

Weitere Informationen zum Filtern von Nachrichtenattributen für SNS finden Sie unter SNSAmazon-Nachrichtenfilterung im Amazon Simple Notification Service Developer Guide.

Weitere Informationen zum Filtern von Event-Tags für EventBridge finden Sie unter Vergleichsoperatoren zur Verwendung in Ereignismustern EventBridge in Amazon im EventBridge Amazon-Benutzerhandbuch.

Weitere Informationen zum Filtern von nutzlastbasierten Tags für SNS finden Sie unter Einführung der nutzlastbasierten Nachrichtenfilterung für Amazon SNS

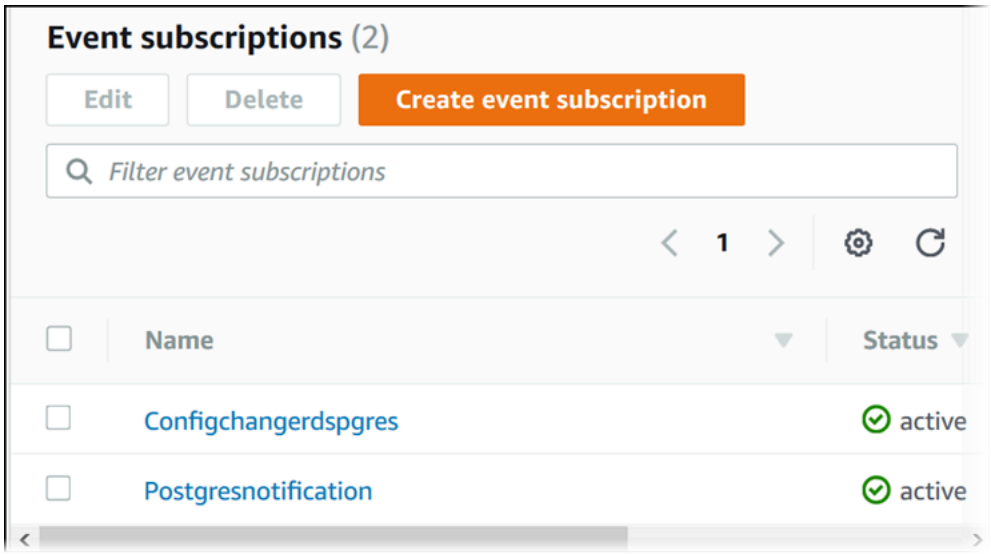
Amazon-Abonnements für RDS Event-Benachrichtigungen auflisten

Sie können Ihre aktuellen Abonnements für RDS Amazon-Veranstaltungsbenachrichtigungen auflisten.

Konsole

Um Ihre aktuellen Abonnements für RDS Amazon-Veranstaltungsbenachrichtigungen aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus. Im Bereich Ereignisabonnements werden all Ihre Abonnements für Ereignisbenachrichtigungen angezeigt.



AWS CLI

Verwenden Sie den AWS CLI `describe-event-subscriptions`Befehl, um Ihre aktuellen Abonnements für RDS Amazon-Event-Benachrichtigungen aufzulisten.

Example

Im folgenden Beispiel werden alle Ereignisabonnements beschrieben.

```
aws rds describe-event-subscriptions
```

Das folgende Beispiel beschreibt den Stack `myfirsteventsubscription`.

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

Rufen Sie Amazon RDS API DescribeEventSubscriptionsAction auf, um Ihre aktuellen Abonnements für RDS Amazon-Veranstaltungsbenachrichtigungen aufzulisten.

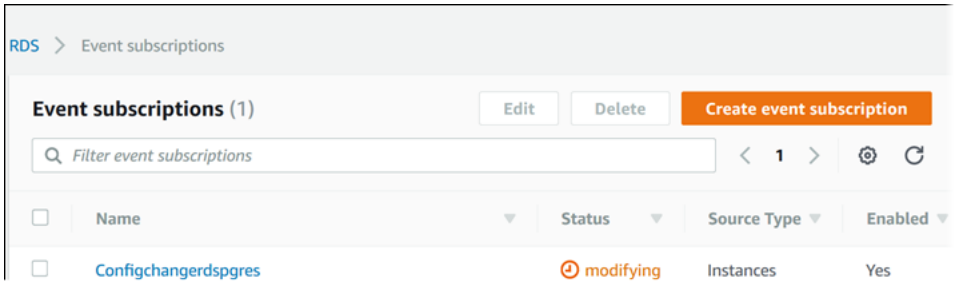
Ein Abonnement für RDS Amazon-Event-Benachrichtigungen ändern

Nachdem Sie ein Abonnement erstellt haben, können Sie den Abonnementnamen, die Quell-ID, die Kategorien oder das Thema ändernARN.

Konsole

Um ein Abonnement für RDS Amazon-Event-Benachrichtigungen zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus.
3. Wählen Sie im Bereich Ereignisabonnements das Abonnement, das Sie modifizieren möchten, und klicken Sie auf Bearbeiten.
4. Nehmen Sie Ihre Änderungen am Abonnement im Bereich Ziel oder Quelle vor.
5. Wählen Sie Edit (Bearbeiten) aus. Die RDS Amazon-Konsole zeigt an, dass das Abonnement geändert wird.



AWS CLI

Verwenden Sie den AWS CLI `modify-event-subscription`Befehl, um ein Abonnement für RDS Amazon-Ereignisbenachrichtigungen zu ändern. Verwenden Sie den folgenden erforderlichen Parameter:

- `--subscription-name`

Example

Mit folgendem Code wird aktivier `myeventsubscription`.

Für LinuxmacOS, oderUnix:

```
aws rds modify-event-subscription \
    --subscription-name myeventsubscription \
    --enabled
```

Windows:

```
aws rds modify-event-subscription ^
    --subscription-name myeventsubscription ^
    --enabled
```

API

Um ein RDS Amazon-Ereignis zu ändern, rufen Sie den RDS API Amazon-Betrieb auf `ModifyEventSubscription`. Verwenden Sie den folgenden erforderlichen Parameter:

- `SubscriptionName`

Hinzufügen einer Quell-ID zu einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen

Sie können einem bestehenden Abonnement eine Quell-ID (die RDS Amazon-Quelle, die das Ereignis generiert hat) hinzufügen.

Konsole

Sie können Quellkennungen ganz einfach über die RDS Amazon-Konsole hinzufügen oder entfernen, indem Sie sie bei der Änderung eines Abonnements auswählen oder deren Auswahl aufheben. Weitere Informationen finden Sie unter Ein Abonnement für RDS Amazon-Event-Benachrichtigungen ändern.

AWS CLI

Verwenden Sie den AWS CLI `add-source-identifier-to-subscription`Befehl, um einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen eine Quell-ID hinzuzufügen. Nutzen Sie die folgenden erforderlichen Parameter:

- `--subscription-name`
- `--source-identifier`

Example

Im folgenden Beispiel wird die Quell-ID `mysqldb` zum Abonnement `myrdseventsubscription` hinzugefügt.

Für LinuxmacOS, oderUnix:

```
aws rds add-source-identifier-to-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Windows:

```
aws rds add-source-identifier-to-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Rufen Sie Amazon an, um einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen eine Quell-ID hinzuzufügen RDS API `AddSourceIdentifierToSubscription`. Nutzen Sie die folgenden erforderlichen Parameter:

- `SubscriptionName`
- `SourceIdentifier`

Entfernen einer Quell-ID aus einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen

Sie können eine Quell-ID (die RDS Amazon-Quelle, die das Ereignis generiert hat) aus einem Abonnement entfernen, wenn Sie nicht mehr über Ereignisse für diese Quelle informiert werden möchten.

Konsole

Sie können Quellkennungen ganz einfach über die RDS Amazon-Konsole hinzufügen oder entfernen, indem Sie sie bei der Änderung eines Abonnements auswählen oder deren Auswahl aufheben. Weitere Informationen finden Sie unter Ein Abonnement für RDS Amazon-Event-Benachrichtigungen ändern.

AWS CLI

Verwenden Sie den AWS CLI `remove-source-identifier-from-subscription`Befehl, um eine Quell-ID aus einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen zu entfernen. Nutzen Sie die folgenden erforderlichen Parameter:

- `--subscription-name`
- `--source-identifier`

Example

Im folgenden Beispiel wird die Quell-ID `mysqldb` aus dem Abonnement `myrdseventsubscription` entfernt.

Für LinuxmacOS, oderUnix:

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Windows:

```
aws rds remove-source-identifier-from-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Verwenden Sie den RDS API `RemoveSourceIdentifierFromSubscription`Amazon-Befehl, um eine Quell-ID aus einem Abonnement für RDS Amazon-Ereignisbenachrichtigungen zu entfernen. Nutzen Sie die folgenden erforderlichen Parameter:

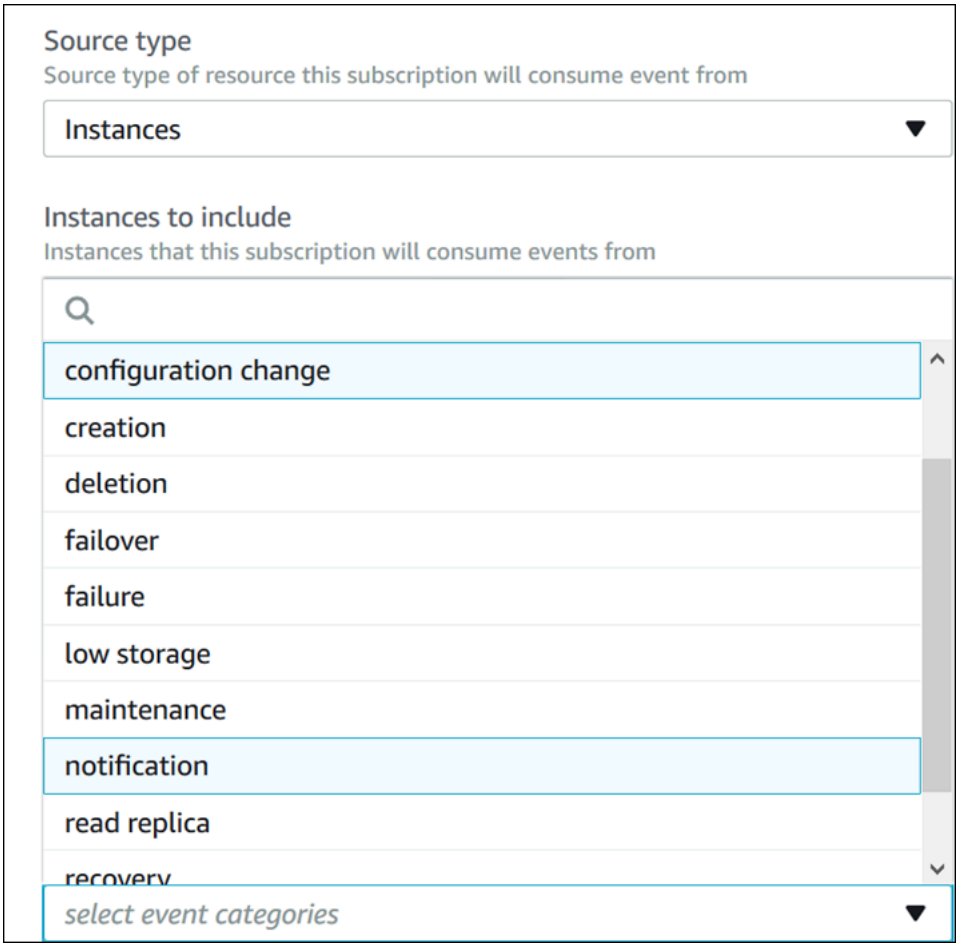
- `SubscriptionName`
- `SourceIdentifier`

Auflistung der Kategorien Amazon RDS Amazon-Event-Benachrichtigungen

Alle Ereignisse für einen Ressourcentyp sind in Kategorien gruppiert. Gehen Sie wie folgt vor, um die verfügbaren Kategorien in einer Liste anzuzeigen.

Konsole

Wenn Sie ein Abonnement für Ereignisbenachrichtigungen erstellen oder ändern, werden die Ereigniskategorien in der RDS Amazon-Konsole angezeigt. Weitere Informationen finden Sie unter Ein Abonnement für RDS Amazon-Event-Benachrichtigungen ändern.



AWS CLI

Verwenden Sie den AWS CLI `describe-event-categories`Befehl, um die Kategorien von RDS Amazon-Ereignisbenachrichtigungen aufzulisten. Dieser Befehl hat keine erforderlichen Parameter.

Example

```
aws rds describe-event-categories
```

API

Verwenden Sie den RDS API `DescribeEventCategories`Amazon-Befehl, um die Kategorien der RDS Amazon-Ereignisbenachrichtigungen aufzulisten. Dieser Befehl hat keine erforderlichen Parameter.

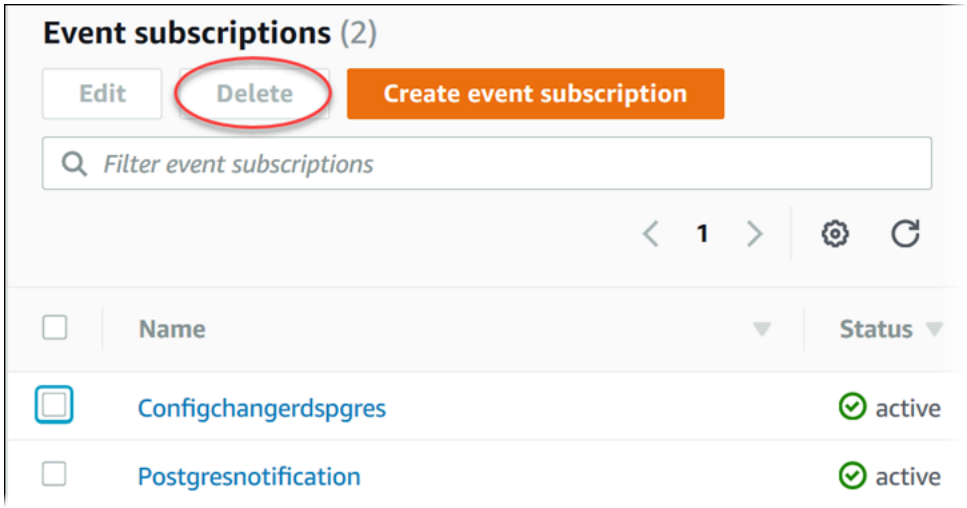
Löschen eines Abonnements für RDS Amazon-Veranstaltungsbenachrichtigungen

Sie können ein Abonnement löschen, wenn Sie es nicht mehr benötigen. Alle Abonnenten des Themas erhalten dann keine weiteren Ereignisbenachrichtigungen, die über dieses Abonnement ausgegeben wurden.

Konsole

Um ein Abonnement für RDS Amazon-Event-Benachrichtigungen zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich DB Event Subscriptions (DB-Ereignisabonnements) aus.
3. Wählen Sie im Bereich My DB Event Subscriptions (Meine DB-Ereignisabonnements) das Abonnement aus, das Sie löschen möchten.
4. Wählen Sie Löschen.
5. Die RDS Amazon-Konsole zeigt an, dass das Abonnement gelöscht wird.



AWS CLI

Verwenden Sie den AWS CLI `delete-event-subscription`Befehl, um ein Abonnement für RDS Amazon-Ereignisbenachrichtigungen zu löschen. Verwenden Sie den folgenden erforderlichen Parameter:

- `--subscription-name`

Example

Im folgenden Beispiel wird das Abonnement gelösch `myrdssubscription`.

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

Verwenden Sie den RDS API `DeleteEventSubscription`Befehl, um ein Abonnement für RDS Amazon-Ereignisbenachrichtigungen zu löschen. Verwenden Sie den folgenden erforderlichen Parameter:

- `SubscriptionName`

Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird

Mit Amazon EventBridge können Sie AWS Services automatisieren und auf Systemereignisse wie Probleme mit der Anwendungsverfügbarkeit oder Ressourcenänderungen reagieren.

Themen

- Regeln für das Senden von Amazon RDS-Ereignissen an Amazon erstellen EventBridge
- Tutorial: Statusänderungen der DB-Instance mithilfe von Amazon protokollieren EventBridge

Regeln für das Senden von Amazon RDS-Ereignissen an Amazon erstellen EventBridge

Sie können einfache Regeln schreiben, um anzugeben, welche Amazon-RDS-Ereignisse Sie interessieren und welche automatisierten Aktionen zu ergreifen sind, wenn ein Ereignis mit einer Regel übereinstimmt. Sie können eine Vielzahl von Zielen festlegen, z. B. eine AWS Lambda Funktion oder ein Amazon SNS SNS-Thema, die Ereignisse im JSON-Format empfangen. Sie können Amazon RDS Amazon beispielsweise so konfigurieren, dass Ereignisse an Amazon gesendet werden, EventBridge wenn eine DB-Instance erstellt oder gelöscht wird. Weitere Informationen finden Sie im Amazon CloudWatch Events-Benutzerhandbuch und im EventBridge Amazon-Benutzerhandbuch.

So erstellen Sie eine Regel, die bei einem Ereignis ausgelöst wird:

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich unter Events (Ereignisse) die Option Rules (Regeln) aus.
3. Wählen Sie Regel erstellen aus.
4. Führen Sie für Ereignisquelle folgende Schritte aus:
    a. Wählen Sie Event Pattern aus.
    b. Wählen Sie für Service Name (Servicename) die Option Relational Database Service (RDS) aus.
    c. Wählen Sie unter Event Type (Ereignistyp) den Amazon-RDS-Ressourcentyp aus, der das Ereignis auslöst. Wenn beispielsweise eine DB-Instance das Ereignis auslöst, wählen Sie RDS DB Instance Event (RDS-DB-Instance-Ereignis) aus.
5. Wählen Sie für Ziele die Option Ziel hinzufügen und wählen Sie den AWS Dienst aus, der reagieren soll, wenn ein Ereignis des ausgewählten Typs erkannt wird.

6. Geben Sie in die anderen Felder in diesem Abschnitt Informationen ein, die für diesen Zieltyp spezifisch sind, sofern vorhanden.
7. Für viele Zieltypen sind EventBridge Berechtigungen erforderlich, um Ereignisse an das Ziel zu senden. In diesen Fällen EventBridge kann die IAM-Rolle erstellt werden, die für die Ausführung Ihrer Veranstaltung erforderlich ist:

   - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Create a new role for this specific resource (Eine neue Rolle für diese spezifische Ressource erstellen).
   - Wenn Sie eine zuvor erstellte IAM-Rolle verwenden möchten, wählen Sie Use existing role (Vorhandene Rolle verwenden).
8. Optional können Sie die Schritte 5 bis 7 wiederholen, um ein weiteres Ziel für diese Regel hinzuzufügen.
9. Wählen Sie Configure details. Geben Sie für Rule definition einen Namen und eine Beschreibung für die Regel ein.

   Der Regelname muss innerhalb dieser Region eindeutig sein.
10. Wählen Sie Regel erstellen aus.

Weitere Informationen finden Sie unter Erstellen einer EventBridge Regel, die bei einem Ereignis ausgelöst wird im CloudWatch Amazon-Benutzerhandbuch.

Tutorial: Statusänderungen der DB-Instance mithilfe von Amazon protokollieren EventBridge

In diesem Tutorial erstellen Sie eine AWS Lambda Funktion, die die Statusänderungen für eine Amazon RDS-Instance protokolliert. Anschließend erstellen Sie eine Regel, die die Funktion ausführt, sobald eine Statusänderung einer vorhandenen RDS-DB-Instance stattfindet. Das Tutorial geht davon aus, dass Sie eine kleine laufende Test-Instance haben, die Sie vorübergehend herunterfahren können.

> **Important**
>
> Führen Sie dieses Tutorial nicht für eine laufende Produktions-DB-Instance durch.

Themen

- Schritt 1: Erstellen Sie eine AWS Lambda Funktion
- Schritt 2: Erstellen einer Regel
- Schritt 3: Testen der Regel

Schritt 1: Erstellen Sie eine AWS Lambda Funktion

Erstellen Sie eine Lambda-Funktion, um die Statusänderungsereignisse zu protokollieren. Sie geben diese Funktion beim Erstellen der Regel an.

So erstellen Sie eine Lambda-Funktion:

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wenn Sie noch nicht mit Lambda gearbeitet haben, wird Ihnen eine Willkommensseite angezeigt. Wählen Sie Get Started Now. Andernfalls, wählen Sie Create function (Funktion erstellen) aus.
3. Wählen Sie Von Grund auf neu schreiben aus.
4. Gehen Sie auf der Seite Create function (Funktion erstellen) wie folgt vor:
   a. Geben Sie einen Namen und eine Beschreibung für die Lambda-Funktion ein. Geben Sie der Funktion beispielsweise den Namen **`RDSInstanceStateChange`**.
   b. Wählen Sie in Runtime Node.js 16x aus.
   c. Wählen Sie für Architecture (Architektur) x86_64 aus.
   d. Führen Sie für Execution role (Ausführungsrolle) einen der folgenden Schritte aus:
      - Wählen Sie Create a new role with basic Lambda permissions (Eine neue Rolle mit den grundlegenden Lambda-Berechtigungen erstellen) aus.
      - Wählen Sie für Execution role (Ausführungsrolle) die Option Use an existing role (Vorhandene Rolle verwenden) aus. Wählen Sie die Rolle aus.
   e. Wählen Sie Funktion erstellen.
5. Gehen Sie auf der InstanceStateChangeRDS-Seite wie folgt vor:
   a. In Code-Quelle wählen Sie index.js.
   b. Im Ausschnitt index.js löschen Sie den vorhandenen Code.
   c. Geben Sie den folgenden Code ein:

```
console.log('Loading function');
```

```
exports.handler = async (event, context) => {
    console.log('Received event:', JSON.stringify(event));
};
```

d. Wählen Sie Deploy (Bereitstellen) aus.

Schritt 2: Erstellen einer Regel

Erstellen Sie eine Regel, damit die Lambda-Funktion ausgeführt wird, wenn Sie eine Amazon RDS-Instance starten.

Um die EventBridge Regel zu erstellen

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Rules aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein. Geben Sie z. B. ei **`RDSInstanceStateChangeRule`**.
5. Wählen Sie Rule with an event pattern 8Regel mit einem Ereignismuster) und dann Next (Weiter) aus.
6. Wählen Sie als Quelle für Ereignisse die Option AWS Veranstaltungen oder EventBridge Partnerveranstaltungen aus.
7. Scrollen Sie nach unten zum Abschnitt Event pattern (Ereignismuster).
8. Wählen Sie für Ereignisquelle die Option AWS-Services aus.
9. Wählen Sie für AWS -Service die Option Relational Database Service (RDS) aus.
10. Für Ereignistyp wählen Sie RDS DB-Instance-Ereignis.
11. Übernehmen Sie das Standard-Ereignismuster. Wählen Sie anschließend Weiter.
12. Bei Target types (Zieltypen) wählen Sie AWS -Service aus.
13. Für Select a target (Ein Ziel auswählen), wählen die Option Lambda function (Lambda-Funktion) aus.
14. Wählen Sie für Function (Funktion) die Lambda-Funktion aus, die Sie erstellt haben. Wählen Sie anschließend Weiter.
15. Wählen Sie in Configure tags (Tags konfigurieren) Next (Weiter) aus.
16. Überprüfen Sie die Schritte in Ihrer Regel. Wählen Sie dann Create rule (Regel erstellen) aus.

Schritt 3: Testen der Regel

Um Ihre Regel zu testen, fahren Sie eine RDS-DB-Instance herunter. Warten Sie einige Minuten, bis die Instance heruntergefahren wurde, und prüfen Sie dann, ob Ihre Lambda-Funktion aufgerufen wurde.

Testen der Regel durch Anhalten einer DB-Instance

1. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Stopp einer RDS-DB-Instance.
3. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
4. Wählen Sie im Navigationsbereich Rules (Regeln), den Namen der von Ihnen erstellten Regel aus.
5. Wählen Sie unter Regeldetails die Option Überwachung aus.

   Sie werden zur CloudWatch Amazon-Konsole weitergeleitet. Wenn Sie nicht weitergeleitet werden, klicken Sie auf Metriken anzeigen in CloudWatch.
6. In Alle Metriken wählen Sie den Namen der Regel aus, die Sie erstellt haben.

   Das Diagramm sollte darauf hinweisen, dass die Regel aufgerufen wurde.
7. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
8. Wählen Sie den Namen der Protokollgruppe für die Lambda-Funktion aus (/aws/lambda/***function-name***).
9. Wählen Sie den Namen des Protokoll-Streams aus, um die von der Funktion für die von Ihnen gestartete Instance bereitgestellten Daten anzuzeigen. Das empfangene Ergebnis sollte in etwa wie folgt aussehen:

```
{
    "version": "0",
    "id": "12a345b6-78c9-01d2-34e5-123f4ghi5j6k",
    "detail-type": "RDS DB Instance Event",
    "source": "aws.rds",
    "account": "111111111111",
    "time": "2021-03-19T19:34:09Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:111111111111:db:testdb"
    ],
    "detail": {
```

```
        "EventCategories": [
            "notification"
        ],
        "SourceType": "DB_INSTANCE",
        "SourceArn": "arn:aws:rds:us-east-1:111111111111:db:testdb",
        "Date": "2021-03-19T19:34:09.293Z",
        "Message": "DB instance stopped",
        "SourceIdentifier": "testdb",
        "EventID": "RDS-EVENT-0087"
    }
}
```

Weitere Beispiele für RDS-Ereignisse im JSON-Format finden Sie unter Übersicht der Ereignisse für Amazon RDS.

10. (Optional) Zum Abschluss können Sie die Amazon RDS-Konsole öffnen und die von Ihnen gestoppte Instance starten.

Amazon RDS-Ereigniskategorien und Ereignisnachrichten

Amazon RDS generiert eine beträchtliche Anzahl von Ereignissen in Kategorien, die Sie über die Amazon RDS-Konsole oder die API abonnieren können. AWS CLI

Themen

- DB-Cluster-Ereignisse
- DB-Cluster-Snapshot-Ereignisse
- DB-Instance-Ereignisse
- DB-Parametergruppenereignisse
- DB-Sicherheitsgruppenereignisse
- DB-Snapshot-Ereignisse
- RDS-Proxy-Ereignisse
- Blau/Grün-Bereitstellungsereignisse
- Benutzerdefinierte Engine-Versionsereignisse

DB-Cluster-Ereignisse

Die folgende Tabelle zeigt den Ereignistyp sowie eine Liste der Ereignisse für den Fall, dass ein DB-Cluster der Quelltyp ist.

Weitere Informationen zu Multi-AZ-DB-Cluster-Bereitstellungen finden Sie unter. Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0016	Die Anmeldeinformationen des Hauptbenutzers werden zurückgesetzt.	Keine
Erstellung	RDS-EVENT-0170	DB-Cluster erstellt.	Keine
Failover	RDS-EVENT-0069	Das Cluster-Failover ist fehlgeschlagen. Überprüfen Sie den Zustand Ihrer	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
		Cluster-Instances und versuchen Sie es erneut.	
Failover	RDS-EVENT-0070	Erneutes Heraufstufen des vorherigen Primärservers:. *name*	Keine
Failover	RDS-EVENT-0071	Failover zur DB-Instance abgeschlossen:*name*.	Keine
Failover	RDS-EVENT-0072	Derselbe AZ-Failover zur DB-Instance gestartet:. *name*	Keine
Failover	RDS-EVENT-0073	Cross-AZ-Failover zur DB-Instance gestartet:. *name*	Keine
Ausfall	RDS-EVENT-0354	Sie können den DB-Cluster aufgrund inkompatibler Ressourcen nicht erstellen. *message*.	Das *message* beinhaltet Details zum Fehler.
Ausfall	RDS-EVENT-0355	Der DB-Cluster kann aufgrund unzureichender Ressourcenlimits nicht erstellt werden. *message*.	Das *message* beinhaltet Details zum Fehler.
Wartung	RDS-EVENT-0156	Der DB-Cluster verfügt über ein Upgrade der DB-Engine für kleinere Versionen.	Keine
Wartung	RDS-EVENT-0173	Die Version der Datenbank-Cluster-Engine wurde aktualisiert.	Das Patchen des DB-Clusters ist abgeschlossen.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wartung	RDS-EVENT-0174	Der Datenbankcluster befindet sich in einem Zustand, der nicht aktualisiert werden kann.	Keine
Wartung	RDS-EVENT-0176	Die Hauptversion der Datenbank-Cluster-Engine wurde aktualisiert.	Keine
Wartung	RDS-EVENT-0177	Das Upgrade des Datenbank-Clusters ist im Gange.	Keine
Wartung	RDS-EVENT-0286	Das *version_number* Versionsupgrade der Datenbankcluster-Engine wurde gestartet. Der Cluster bleibt online.	Keine
Wartung	RDS-EVENT-0287	Es wurde festgestellt, dass ein Betriebssystem-Upgrade erforderlich ist.	Keine
Wartung	RDS-EVENT-0288	Das Upgrade des Cluster-Betriebssystems wird gestartet.	Keine
Wartung	RDS-EVENT-0289	Das Upgrade des Cluster-Betriebssystems wurde abgeschlossen.	Keine
Wartung	RDS-EVENT-0290	Der Datenbank-Cluster wurde gepatcht: Quellversion *version_number* =>*new_version_number* .	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wartung	RDS-EVENT-0410	Die Vorabprüfung für das Versionsupgrade der Datenbank-Cluster-Engine wurde gestartet.	Keine
Wartung	RDS-EVENT-0412	Die Vorabprüfung für das Versionsupgrade der Datenbank-Cluster-Engine ist fehlgeschlagen oder das Zeitlimit wurde überschri tten.	Keine
Wartung	RDS-EVENT-0413	Die Aufgaben vor dem Upgrade des DB-Clusters sind im Gange.	Keine
Wartung	RDS-EVENT-0414	Die Aufgaben des DB-Clusters nach dem Upgrade sind im Gange.	Keine
Wartung	RDS-EVENT-0417	Das Upgrade der Version der Datenbank-Cluster-Engine wurde gestartet.	Keine
Benachric htigung	RDS-EVENT-0172	Der Cluster wurde von *name* bis *name* umbenannt.	Keine
Read Replica	RDS-EVENT-0411	Die Vorabprüfung für das Versionsupgrade der Datenbank-Cluster-Engine wurde abgeschlossen.	Keine

DB-Cluster-Snapshot-Ereignisse

In der folgenden Tabelle werden die Ereigniskategorie und die Ereignisse für den Quelltyp „DB-Cluster-Snapshot“ aufgeführt.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
backup	RDS-EVENT-0074	Ein manueller Cluster-Snapshot wird erstellt.	Keine
backup	RDS-EVENT-0075	Ein manueller Cluster-Snapshot wurde erstellt.	Keine
Benachrichtigung	RDS-EVENT-0162	Bei der Aufgabe zum Exportieren von Cluster-Snapshots ist ein Fehler aufgetreten.	Keine
Benachrichtigung	RDS-EVENT-0163	Die Aufgabe zum Exportieren von Cluster-Snapshots wurde abgebrochen.	Keine
Benachrichtigung	RDS-EVENT-0164	Die Aufgabe zum Exportieren von Cluster-Snapshots ist abgeschlossen.	Keine
backup	RDS-EVENT-0168	Erstellen eines automatisierten Cluster-Snapshots.	Keine
backup	RDS-EVENT-0169	Automatisierter Cluster-Snapshot erstellt.	Keine

DB-Instance-Ereignisse

In der folgenden Tabelle werden die Ereigniskategorie und die Ereignisse für den Quelltyp "DB-Instance" aufgeführt.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Verfügbarkeit	RDS-EVENT-0004	DB-Instance-Shutdown.	Keine
Verfügbarkeit	RDS-EVENT-0006	Die DB-Instance wurde neu gestartet.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Verfügbarkeit	RDS-EVENT-0022	Fehler beim Neustart von MySQL:. *message*	Während des Neustarts von ist ein Fehler aufgetreten.
Verfügbarkeit	RDS-EVENT-0221	Die DB-Instance hat den Schwellenwert für „Speicher-voll“ erreicht und die Datenbank wurde heruntergefahren. Sie können den zugewiesenen Speicher erhöhen, um dieses Problem zu beheben.	Keine
Verfügbarkeit	RDS-EVENT-0222	Die freie Speicherkapazität für die DB-Instance *name* ist im Verhältnis *percentage* zum zugewiesenen Speicher gering [Zugewiesener Speicher:*amount*, Freier Speicher:*amount*]. Die Datenbank wird heruntergefahren, um Beschädigungen zu verhindern, wenn der freie Speicherplatz niedriger ist als*amount*. Sie können den zugewiesenen Speicher erhöhen, um dieses Problem zu beheben.	Gilt nur für RDS for MySQL, wenn eine DB-Instance mehr als 90% des zugewiesenen Speichers belegt. Überwachen Sie den Speicherplatz für eine DB-Instance mithilfe der Metrik „Kostenloser Speicherplatz“. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Verfügbarkeit	RDS-EVENT-0330	Die freie Speicherkapazität des dedizierten Transaktionsprotokollvolumens ist für eine DB-Instance zu gering. *name* Der freie Speicherplatz für das Protokollvolumen *percentage* entspricht dem zugewiesenen Speicher. [Zugewiesener Speicher:*amount*, Freier Speicher:*amount*] Die Datenbank wird heruntergefahren, um Beschädigungen zu verhindern, wenn der freie Speicherplatz niedriger ist als*amount*. Sie können das dedizierte Transaktionslog-Volume deaktivieren, um dieses Problem zu beheben.	Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Verfügbarkeit	RDS-EVENT-0331	Die freie Speicherkapazität des dedizierten Transaktionsprotokollvolumens ist für eine DB-Instance zu gering. *name* Der freie Speicher für das Protokollvolumen *percentage* gehört zum bereitgestellten Speicher. [Bereitgestellter Speicher:*amount*, Freier Speicherplatz:*amount*] Sie können das dedizierte Transaktionslog-Volume deaktivieren, um dieses Problem zu beheben.	Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).
Verfügbarkeit	RDS-EVENT-0396	Amazon RDS hat im nächsten Wartungsfenster dieser Instance nach der internen Benutzerpasswortrotation einen Neustart für diese Read Replica geplant.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Verfügbarkeit	RDS-EVENT-0419	Amazon RDS konnte nicht auf den KMS Verschlüs selungsschlüssel für die Datenbank-Instance zugreifen*name*. Diese Datenbank wird in einen Zustand versetzt, auf den nicht zugegriffen werden kann. Weitere Informationen finden Sie im Abschnitt zur Fehlerbeh ebung in der Amazon-RDS-Dokumentation.	Keine
Backup	RDS-EVENT-0001	Sichern einer DB-Instance	Keine
Sicherung	RDS-EVENT-0002	DB-Instance-Backup wurde beendet.	Keine
backup	RDS-EVENT-0086	Wir können die Optionsgr uppe nicht *name* mit der Datenbankinstanz *name* verknüpfen. Vergewiss ern Sie sich, dass die Optionsgruppe *name* in Ihrer DB-Instance-Klasse und Konfiguration unterstüt zt wird. Wenn ja, überprüfe n Sie alle Optionsgr uppeneinstellungen und versuchen Sie es erneut.	Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.
Konfigura tionsände rung	RDS-EVENT-0011	Es wurde aktualisiert, um DBParameter Group zu verwenden*name*.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0012	Die Änderung wird auf die Datenbank-Instance-Klasse angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0014	Die Änderung wurde auf die DB-Instance-Klasse angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0016	Die Anmeldeinformationen des Hauptbenutzers werden zurückgesetzt.	Keine
Konfigurationsänderung	RDS-EVENT-0017	Die Änderung wurde auf den zugewiesenen Speicher angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0018	Die Änderung wird auf den zugewiesenen Speicher angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0024	Die Änderung zur Konvertierung in eine Multi-AZ-DB-Instance wird angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0025	Die Änderung zur Konvertierung in eine Multi-AZ-DB-Instance wurde angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0028	Automatisierte Backups wurden deaktiviert.	Keine
Konfigurationsänderung	RDS-EVENT-0029	Die Änderung zur Konvertierung in eine standardmäßige (Single-AZ-)DB-Instance wurde angewendet.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0030	Die Änderung zur Konvertierung in eine standardmäßige (Single-AZ-)DB-Instance wird angewendet.	Keine
Konfigurationsänderung	RDS-EVENT-0032	Automatisierte Backups wurden aktiviert.	Keine
Konfigurationsänderung	RDS-EVENT-0033	Es gibt *number* Benutzer, die dem Master-Benutzernamen entsprechen. Es wird nur der Benutzer zurückgesetzt, der nicht an einen bestimmten Host gebunden ist.	Keine
Konfigurationsänderung	RDS-EVENT-0067	Ihr Passwort konnte nicht zurückgesetzt werden. Informationen zum Fehler:. *message*	Keine
Konfigurationsänderung	RDS-EVENT-0078	Das Überwachungsintervall wurde geändert auf*number*.	Die Konfiguration von „Enhanced Monitoring“ (Erweiterte Überwachung) wurde geändert.
Konfigurationsänderung	RDS-EVENT-0092	Die Aktualisierung der DB-Parametergruppe ist abgeschlossen.	Keine
Konfigurationsänderung	RDS-EVENT-0217	Anwenden von Auto-Scaling-initiierten Modifikationen auf zugewiesenen Speicher.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0218	Die Anwendung von Auto-Scaling-initiierten Modifikationen auf den zugewiesenen Speicher wurde abgeschlossen.	Keine
Konfigurationsänderung	RDS-EVENT-0295	Das Upgrade der Speicherkonfiguration wurde gestartet.	Keine
Konfigurationsänderung	RDS-EVENT-0296	Das Upgrade der Speicherkonfiguration wurde abgeschlossen.	Keine
Konfigurationsänderung	RDS-EVENT-0332	Das dedizierte Protokollvolume ist deaktiviert.	Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).
Konfigurationsänderung	RDS-EVENT-0333	Die Deaktivierung des dedizierten Protokollvolumes wurde gestartet.	Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).
Konfigurationsänderung	RDS-EVENT-0334	Die Aktivierung des dedizierten Protokollvolumes wurde gestartet.	Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).
Konfigurationsänderung	RDS-EVENT-0335	Das dedizierte Protokollvolume ist aktiviert.	Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0383	*engine version* unterstützt das Memcached-Plugin nicht. RDS wird Ihre DB-Instance weiter aktualisieren und dieses Plugin entfernen.	Ab MySQL 8.3.0 wird das Memcached-Plugin nicht unterstützt. Weitere Informationen finden Sie unter Änderungen in MySQL 8.3.0 (2024-01-16, Innovation Release).
Erstellung	RDS-EVENT-0005	Die DB-Instance wurde erstellt.	Keine
Löschung	RDS-EVENT-0003	Die DB-Instance wurde gelöscht.	Keine
Failover	RDS-EVENT-0013	Das Multi-AZ-Instance-Failover wurde gestartet.	Ein Multi-AZ-Failover, das zur Hochstufung einer Standby-DB-Instance geführt hat, wurde gestartet.
Failover	RDS-EVENT-0015	Das Multi-AZ-Failover auf Standby ist abgeschlossen. Die DNS-Verteilung kann einige Minuten dauern.	Ein Multi-AZ-Failover, das zur Hochstufung einer Standby-DB-Instance geführt hat, wurde abgeschlossen. Es kann einige Minuten dauern, bis die DNS-Übertragung auf die neue primäre DB-Instance abgeschlossen ist.
Failover	RDS-EVENT-0034	Abbrechender Benutzer hat ein Failover angefordert, da erst vor Kurzem ein Failover in der Datenbank-Instance erfolgt ist.	Amazon RDS führt ein angefordertes Failover nicht aus, da erst vor Kurzem ein Failover auf dieser DB-Instance erfolgt ist.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Failover	RDS-EVENT-0049	Das Multi-AZ-Instance-Failover wurde abgeschlossen.	Keine
Failover	RDS-EVENT-0050	Die Aktivierung der Multi-AZ-Instance wurde gestartet.	Nach einer erfolgreichen Wiederherstellung der DB-Instance wurde eine Multi-AZ-Aktivierung gestartet. Dieses Ereignis tritt auf, wenn Amazon RDS die primäre DB-Instance auf dieselbe AZ hochstuft wie die vorherige primäre DB-Instance.
Failover	RDS-EVENT-0051	Die Aktivierung der Multi-AZ-Instance wurde abgeschlossen.	Die Multi-AZ-Aktivierung wurde abgeschlossen. Sie sollten nun Zugriff auf Ihre Datenbank haben.
Failover	RDS-EVENT-0065	Wiederherstellung nach teilweisem Failover ist erfolgt.	Keine
Ausfall	RDS-EVENT-0031	DB-Instance in den *name* Status versetzt. RDS empfiehlt, dass Sie eine initiieren point-in-time-restore.	Die DB-Instance ist aufgrund einer inkompatiblen Konfiguration oder eines zugrunde liegenden Speicherproblems ausgefallen. Starten Sie a point-in-time-restore für die DB-Instance.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0035	Datenbankinstanz, in die eingefügt wurde*state*. *message*.	Einige Parameter der DB-Instance sind ungültig. Wenn beispielsweise die DB-Instance nicht gestartet werden konnte, da der Wert eines speicherbezogenen Parameters zu hoch für diese Instance-Klasse ist, sollten Sie den Speicherparameter ändern und die DB-Instance neu starten.
Ausfall	RDS-EVENT-0036	Datenbankinstanz in*state*. *message*.	Die DB-Instance befindet sich in einem inkompatiblen Netzwerk. Einige der angegebenen Subnetze IDs sind ungültig oder existieren nicht.
Ausfall	RDS-EVENT-0058	Die Statspack-Installation ist fehlgeschlagen. *message*.	Beim Erstellen des Statspack-Benutzerkontos PERFSTAT ist ein Fehler aufgetreten. Verwerfen Sie das Konto, bevor Sie die STATSPACK -Option hinzufügen.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0079	Amazon RDS konnte keine Anmeldeinformationen für die erweiterte Überwachung erstellen und diese Funktion wurde deaktiviert. Dies liegt wahrscheinlich daran, dass es in Ihrem Konto rds-monitoring-role nicht vorhanden und nicht korrekt konfiguriert ist. Weitere Informationen finden Sie im Abschnitt zur Fehlerbehebung in der Amazon-RDS-Dokumentation.	Die erweiterte Überwachung kann nicht ohne die IAM-Rolle für erweiterte Überwachung aktiviert werden. Weitere Informationen zum Erstellen der IAM-Rolle finden Sie unter So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung) in Amazon RDS.
Ausfall	RDS-EVENT-0080	Amazon RDS konnte die erweiterte Überwachung für Ihre Instance nicht konfigurieren: *name* und diese Funktion wurde deaktiviert. Dies liegt wahrscheinlich daran, dass es in Ihrem Konto rds-monitoring-role nicht vorhanden und nicht richtig konfiguriert ist. Weitere Informationen finden Sie im Abschnitt zur Fehlerbehebung in der Amazon-RDS-Dokumentation.	Die erweiterte Überwachung wurde aufgrund eines Fehlers während der Konfigurationsänderung deaktiviert. Wahrscheinlich ist die IAM-Rolle für erweiterte Überwachung falsch konfiguriert. Weitere Informationen über das Erstellen der IAM-Rolle für erweiterte Überwachung finden Sie unter So erstellen Sie eine IAM-Rolle für „Enhanced Monitoring“ (Erweiterte Überwachung) in Amazon RDS.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0081	Amazon RDS konnte keine Anmeldeinformationen für die *name* Option erstellen . Dies liegt daran, dass die *name* IAM-Rolle in Ihrem Konto nicht korrekt konfiguriert ist. Weitere Informationen finden Sie im Abschnitt zur Fehlerbeh ebung in der Amazon-RDS-Dokumentation.	Die IAM-Rolle, die Sie zum Zugriff auf den Amazon S3-Bucket für die SQL Server-nativen Backups und Wiederherstellunge n verwenden, ist falsch konfiguriert. Weitere Informationen finden Sie unter Einrichtung für native Backups und Wiederher stellungen.
Ausfall	RDS-EVENT-0165	Die RDS-Custom-DB-Instance befindet sich außerhalb des Support-U mfangs.	Es liegt in Ihrer Verantwor tung, Konfigurationsprob leme zu beheben, die Ihre RDS Custom DB-Instance in den Zustand `unsupport ed-configuration` . Wenn das Problem in der AWS Infrastruktur liegt, können Sie die Konsole oder die verwenden AWS CLI , um es zu beheben. Wenn das Problem mit dem Betriebssystem oder der Datenbankkonfiguration besteht, können Sie sich beim Host anmelden, um es zu beheben. Weitere Informationen finden Sie unter Support-P erimeter in RDS Custom.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0188	Die DB-Instance befindet sich in einem Zustand, der nicht aktualisiert werden kann. *message*	Amazon RDS konnte eine MySQL-DB-Instance aufgrund von Inkompatibilitäten im Zusammenhang mit dem Datenwörterbuch nicht aktualisieren. Die DB-Instance wurde auf MySQL Version 5.7 zurückgesetzt, weil ein Upgrade-Versuch auf Version 8.0 fehlschlug, oder sie wurde auf MySQL Version 8.0 zurückgesetzt, weil ein Upgrade-Versuch auf Version 8.4 fehlschlug. Weitere Informationen finden Sie unter Rollback nach einem fehlgeschlagenen Upgrade.
Ausfall	RDS-EVENT-0219	Die DB-Instance hat den Status „Ungültig“. Es sind keine Aktionen erforderlich. Die Auto Scaling wird später erneut versucht.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0220	Die DB-Instance befindet sich in der Bedenkzeit für einen früheren Skalenspeichervorgang. Wir optimieren Ihre DB-Instance. Dies dauert mindestens 6 Stunden. Es sind keine Aktionen erforderlich. Das Auto Scaling wird nach der Bedenkzeit erneut versucht.	Keine
Ausfall	RDS-EVENT-0223	Die automatische Speicherskalierung kann den Speicher aus dem folgenden Grund nicht skalieren:*reason*.	Keine
Ausfall	RDS-EVENT-0224	Die automatische Speicherskalierung hat eine ausstehende Skalierungsspeicheraufgabe ausgelöst, die den maximalen Speichers chwellenwert erreichen oder überschreiten wird. Erhöhen Sie den maximalen Speichers chwellenwert.	Keine
Ausfall	RDS-EVENT-0237	Der Speichertyp der DB-Instance ist derzeit in der Availability Zone nicht verfügbar. Die Auto Scaling wird später erneut versucht.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0254	Das zugrunde liegende Speicherkontingent für dieses Kundenkonto hat das Limit überschri tten. Bitte erhöhen Sie das zulässige Speicherk ontingent, damit die Skalierung auf der Instance durchgeführt werden kann.	Keine
Ausfall	RDS-EVENT-0278	Die Erstellung der DB-Instance ist fehlgeschlagen. *message*	Das *message* beinhaltet Details zum Fehler.
Ausfall	RDS-EVENT-0279	Die Heraufstufung der RDS Custom Read Replica ist fehlgeschlagen. *message*	Das *message* beinhaltet Details zum Fehler.
Ausfall	RDS-EVENT-0280	RDS Custom konnte die DB-Instance nicht aktualisi eren, da die Vorabprüf ung fehlgeschlagen ist. *message*	Das *message* beinhaltet Details zum Fehler.
Ausfall	RDS-EVENT-0281	RDS Custom konnte die DB-Instance nicht ändern, da die Vorabprüf ung fehlgeschlagen ist. *message*	Das *message* beinhaltet Details zum Fehler.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0282	RDS Custom konnte die DB-Instance nicht ändern, da die Elastic-IP-Berechtigungen nicht korrekt sind. Bitte vergewissern Sie sich, dass die Elastic-IP-Adresse mit `AWSRDSCustom` getaggt ist.	Keine
Ausfall	RDS-EVENT-0283	RDS Custom konnte die DB-Instance nicht ändern, da das Elastic-IP-Limit in Ihrem Konto erreicht wurde. Geben Sie unbenutztes Elastic frei IPs oder fordern Sie eine Erhöhung des Kontingents für Ihr Elastic-IP-Adresslimit an.	Keine
Ausfall	RDS-EVENT-0284	RDS Custom konnte die Instance nicht in Hochverfügbarkeit konvertieren, da die Vorabprüfung fehlgeschlagen ist. *`message`*	Das *`message`* beinhaltet Details zum Fehler.
Ausfall	RDS-EVENT-0285	RDS Custom konnte aus folgendem Grund keinen endgültigen Snapshot für die DB-Instance erstellen *`message`*.	Das *`message`* beinhaltet Details zum Fehler.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0421	RDS Custom konnte die DB-Instance nicht in eine Multi-AZ-Bereitstellung konvertieren:. *message* Die Instance bleibt eine Single-AZ-Bereitstellung. Informationen zu Multi-AZ-Bereitstellungen für RDS Custom for Oracle finden Sie im RDS-Benutzerhandbuch.	Das *message* enthält Einzelheiten zum Fehler.
Ausfall	RDS-EVENT-0306	Das Upgrade der Speicherkonfiguration ist fehlgeschlagen. Bitte versuchen Sie das Upgrade erneut.	Keine
Ausfall	RDS-EVENT-0315	Die Datenbank mit inkompatiblem Netzwerk, , konnte nicht in den verfügbaren *name* Status versetzt werden: *message*	Die Datenbank-Netzwerkkonfiguration ist ungültig. Die Datenbank konnte nicht vom Status Inkompatibles Netzwerk in den Status Verfügbar wechseln.
Ausfall	RDS-EVENT-0328	Fehler beim Hinzufügen eines Hosts zu einer Domäne. Der Domänenmitgliedschaftsstatus *instancename* wurde beispielsweise auf Fehlgeschlagen gesetzt.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0329	Fehler beim Hinzufüge n eines Hosts zu Ihrer Domain. Während des Domänenbe itritts gab Microsoft Windows den Fehlercod e zurück*message*. Überprüfen Sie Ihre Netzwerk- und Berechtig ungskonfigurationen und `modify-db-instance` fordern Sie erneut an, den Domänenbeitritt zu versuchen.	Wenn Sie ein selbstver waltetes Active Directory verwenden, finden Sie weitere Informationen unter Fehlerbehebung für selbstverwaltetes Active Directory
Ausfall	RDS-EVENT-0353	Die DB-Instance kann aufgrund unzureichender Ressourcenlimits nicht erstellt werden. *message*.	Das *message* beinhaltet Details zum Fehler.
Ausfall	RDS-EVENT-0356	RDS konnte den Kerberos-Endpunkt in Ihrer Domäne nicht konfigurieren. Dies könnte die Kerberos-Authentifizierung für Ihre DB-Instance verhinder n. Überprüfen Sie die Netzwerkkonfiguration zwischen Ihrer DB-Instan ce und den Domänenco ntrollern.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0418	Amazon RDS kann nicht auf den KMS Verschlüsselungsschlüssel für die Datenbank-Instance zugreifen*name*. Dies liegt wahrscheinlich daran, dass der Schlüssel deaktiviert ist oder Amazon RDS nicht darauf zugreifen kann. Wenn dies so weitergeht, wird die Datenbank in einen Zustand versetzt, auf den nicht zugegriffen werden kann. Weitere Informationen finden Sie im Abschnitt zur Fehlerbehebung in der Amazon-RDS-Dokumentation.	Keine
Ausfall	RDS-EVENT-0420	Amazon RDS kann jetzt erfolgreich auf den KMS Verschlüsselungsschlüssel für die Datenbank-Instance zugreifen*name*.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wenig Speicherplatz	RDS-EVENT-0007	Der zugewiesene Speicherplatz ist ausgeschöpft. Weisen Sie zusätzlichen Speicherplatz zu, um das Problem zu lösen.	Der Speicherplatz, der für die DB-Instance zugewiesen wurde, ist aufgebraucht. Sie sollten der DB-Instance weiteren Speicher zuordnen, um dieses Problem zu lösen. Weitere Informationen finden Sie unter RDS FAQ. Sie können den Speicherplatz für eine DB-Instance mit der Metrik Freier Speicherplatz überwachen.
Wenig Speicherplatz	RDS-EVENT-0089	Die freie Speicherkapazität für die DB-Instance: *name* ist im Verhältnis *percentage* zum bereitgestellten Speicher gering [Provisioned Storage:*size*, Free Storage:*size*]. Erhöhen Sie ggf. den bereitgestellten Speicher, um dieses Problem zu beheben.	Die DB-Instance hat mehr als 90% ihres zugewiesenen Speichers verbraucht. Sie können den Speicherplatz für eine DB-Instance mit der Metrik Freier Speicherplatz überwachen.
Wenig Speicherplatz	RDS-EVENT-0227	Der Speicherplatz Ihres Aurora-Clusters ist gefährlich knapp, es sind nur noch *amount* Terabyte übrig. Bitte ergreifen Sie Maßnahmen, um die Speicherlast auf Ihrem Cluster zu reduzieren.	Das Aurora-Speicher-Subsystem hat wenig Speicherplatz.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wartung	RDS-EVENT-0026	Es werden Offline-P atches auf die DB-Instance angewendet.	Die Offline-Wartung der DB-Instance wird gerade durchgeführt. Die DB-Instance ist zurzeit nicht verfügbar.
Wartung	RDS-EVENT-0027	Es wurden Offline-P atches auf die DB-Instance angewendet.	Die Offline-Wartung der DB-Instance ist abgeschlo ssen. Die DB-Instance ist nun verfügbar.
Wartung	RDS-EVENT-0047	Die Datenbank-Instance wurde gepatcht.	Keine
Wartung	RDS-EREIG NIS-0155	Die DB-Instance verfügt über ein Upgrade der DB-Engine für kleinere Versionen.	Keine
Wartung	RDS-EVENT-0178	Das Upgrade der Datenbankinstanz ist im Gange.	Keine
Wartung	RDS-EVENT-0264	Die Vorabprüfung für das Versions-Upgrade der DB-Engine wurde gestartet.	Keine
Wartung	RDS-EVENT-0265	Die Vorabprüfung für das Versions-Upgrade der DB-Engine wurde beendet.	Keine
Wartung	RDS-EVENT-0266	Die Ausfallzeit für die DB-Instance hat begonnen.	Keine
Wartung	RDS-EVENT-0267	Das Upgrade der Engine-Version wurde gestartet.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wartung	RDS-EVENT-0268	Das Upgrade der Engine-Version ist abgeschlossen.	Keine
Wartung	RDS-EVENT-0269	Die Aufgaben nach dem Upgrade sind in Bearbeitung.	Keine
Wartung	RDS-EVENT-0270	Das Versions-Upgrade der DB-Engine ist fehlgeschlagen. Das Rollback der Engine-Version war erfolgreich.	Keine
Wartung	RDS-EVENT-0398	Es wird gewartet, bis das Upgrade der DB-Engine-Version auf der primären DB-Instance abgeschlossen ist.	Wird während eines Upgrades einer größeren Engine-Version auf einer Read Replica ausgegeben.
Wartung	RDS-EVENT-0399	Es wird gewartet, bis das Upgrade der DB-Engine-Version für die Read Replicas abgeschlossen ist.	Wird während eines Upgrades einer größeren Engine-Version auf der Quell-DB-Engine ausgegeben.
Wartung	RDS-EVENT-0422	RDS ersetzt den Host der DB-Instance *name* aufgrund einer ausstehenden Wartungsmaßnahme.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wartung, Ausfall	RDS-EVENT-0195	*message*	Das Update der Oracle-Zeitzonendatei ist fehlgeschlagen. Weitere Informationen finden Sie unter Automatische Aktualisierung der Oracle-Zeitzonendatei.
Wartung, Benachrichtigung	RDS-EVENT-0191	Eine neue Version der Zeitzonendatei steht zum Update zur Verfügung.	Wenn Sie Ihre DB-Engine von RDS für Oracle aktualisieren, generiert Amazon RDS dieses Ereignis, falls Sie kein Zeitzonendatei-Upgrade ausgewählt haben und die Datenbank nicht die neueste DST-Zeitzonendatei verwendet, die auf der Instance verfügbar ist. Weitere Informationen finden Sie unter Automatische Aktualisierung der Oracle-Zeitzonendatei.
Wartung, Benachrichtigung	RDS-EVENT-0192	Das Update Ihrer Zeitzonendatei wurde gestartet.	Das Upgrade Ihrer Oracle-Zeitzonendatei wurde gestartet. Weitere Informationen finden Sie unter Automatische Aktualisierung der Oracle-Zeitzonendatei.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wartung, Benachrichtigung	RDS-EVENT-0193	Für die aktuelle Version der Zeitzonendatei steht kein Update zur Verfügung.	Ihre Oracle DB-Instance verwendet die neueste Version der Zeitzonendatei und eine der folgenden Aussagen gilt: • Sie haben kürzlich die `TIMEZONE_FILE_AUTOUPGRADE` -Option hinzugefügt. • Ein Upgrade Ihrer Oracle DB-Engine wird durchgeführt. Weitere Informationen finden Sie unter Automatische Aktualisierung der Oracle-Zeitzonendatei.
Wartung, Benachrichtigung	RDS-EVENT-0194	Das Update Ihrer Zeitzonendatei wurde beendet.	Das Update Ihrer Oracle-Zeitzonendatei wurde abgeschlossen. Weitere Informationen finden Sie unter Automatische Aktualisierung der Oracle-Zeitzonendatei.
Benachrichtigung	RDS-EVENT-0044	*message*	Dies ist eine vom Betreiber ausgegebene Benachrichtigung. Weitere Informationen finden Sie in der Ereignismeldung.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0048	Das Upgrade der Datenbank-Engine verzögert sich, da diese Instance über Lesereplikate verfügt, die zuerst aktualisiert werden müssen.	Patchen der DB-Instance wurde verzögert.
Benachrichtigung	RDS-EVENT-0054	*message*	Sie verwenden nicht InnoDB, die für Amazon RDS empfohlene MySQL-Speicher-Engine. Informationen über MySQL-Speicher-Engines finden Sie unter Unterstützte Speicher-Engines für RDS for MySQL.
Benachrichtigung	RDS-EVENT-0055	*message*	Die Tabellenanzahl auf der DB-Instance ist höher als in den empfohlenen bewährten Methoden von Amazon RDS. Reduzieren Sie die Anzahl der Tabellen auf Ihrer DB-Instance. Weitere Informationen zu empfohlenen bewährten Methoden finden Sie unter Grundlegende Anleitungen für den Amazon RDS-Betrieb.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0056	*message*	Die Datenbankanzahl auf der DB-Instance ist höher als in den empfohlenen bewährten Methoden von Amazon RDS. Reduzieren Sie die Anzahl der Datenbanken auf Ihrer DB-Instance. Weitere Informationen zu empfohlenen bewährten Methoden finden Sie unter Grundlegende Anleitungen für den Amazon RDS-Betrieb.
Benachrichtigung	RDS-EVENT-0064	Der TDE-Verschlüsselungsschlüssel wurde erfolgreich rotiert.	Weitere Informationen zu empfohlenen bewährten Methoden finden Sie unter Grundlegende Anleitungen für den Amazon RDS-Betrieb.
Benachrichtigung	RDS-EVENT-0084	Die DB-Instance konnte nicht in Multi-AZ konvertiert werden:*message*.	Sie haben versucht, eine DB-Instance mit In-Memory-Dateigruppen in eine Multi-AZ zu konvertieren; diese werden jedoch in einer Multi-AZ nicht unterstützt. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.
Benachrichtigung	RDS-EVENT-0087	Die DB-Instance wurde gestoppt.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0088	Die DB-Instance wurde gestartet.	Keine
Benachrichtigung	RDS-EVENT-0154	Die DB-Instance wird gestartet, da sie die maximal zulässige Anhaltezeit überschritten hat.	Keine
Benachrichtigung	RDS-EVENT-0157	Die DB-Instance-Klasse konnte nicht geändert werden. *message*.	RDS kann die DB-Instance-Klasse nicht ändern, da die Ziel-Instance-Klasse die Anzahl der Datenbanken auf der Quell-DB-Instance nicht unterstützt. Es wird folgende Fehlermeldung angezeigt: „The instance has N databases, but after conversion it would only support N.“ Weitere Informationen finden Sie unter Einschränkungen für Microsoft SQL Server-DB-Instances.
Benachrichtigung	RDS-EVENT-0158	Die Datenbankinstanz befindet sich in einem Zustand, der nicht aktualisiert werden kann:*message*.	Keine
Benachrichtigung	RDS-EVENT-0167	*message*	Die RDS-Custom-Support-Perimeterkonfiguration wurde geändert.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0189	Die Gutschriften für den gp2-Kontostand für die RDS-Datenbank-Instance sind niedrig. Um dieses Problem zu beheben, reduzieren Sie die IOPS-Nutzung oder ändern Sie Ihre Speichereinstellungen, um mehr Leistung zu erzielen.	Die Gutschriften für den gp2-Kontostand für die RDS-Datenbank-Instance sind niedrig. Um dieses Problem zu beheben, reduzieren Sie die IOPS-Nutzung oder ändern Sie Ihre Speichereinstellungen, um mehr Leistung zu erzielen. Weitere Informationen finden Sie unter I/O-Guthaben und Spitzenlastleistung im Benutzerhandbuch zu Amazon Elastic Compute Cloud.
Benachrichtigung	RDS-EVENT-0225	Die Größe des zugewiesenen Speichers in *amount* GB nähert sich dem maximalen Speichers chwellenwert *amount* GB. Erhöhen Sie den maximalen Speichers chwellenwert.	Dieses Ereignis wird aufgerufen, wenn der Speicher 80 % des maximalen Speichers chwellenwerts erreicht. Erhöhen Sie den maximalen Speichers chwellenwert, um das Ereignis zu vermeiden.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0231	Bei der Speicheränderung Ihrer DB-Instance trat ein interner Fehler auf. Die Änderungsanforderung steht noch aus und wird später erneut versucht.	Bei der Read-Replikation ist ein Fehler aufgetreten. Weitere Informationen finden Sie in der Ereignismeldung. Weitere Informationen finden Sie im Abschnitt zur Problembehandlung für Read Replicas für Ihre DB-Engine. • Fehlerbehebung bei Problemen mit einer MariaDB Read Replica • Behebung eines SQL Server-Read-Replica-Problems • Behebung eines MySQL-Read Replica-Problems • Fehlerbehebung bei Replikaten von RDS für Oracle

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0253	Die Datenbank verwendet den Doublewrite-Puffer . *message*. Weitere Informationen finden Sie in der *name* Dokumentation RDS Optimized Writes.	RDS Optimized Writes ist nicht mit der Speicherkonfiguration der Instance kompatibel. Weitere Informationen erhalten Sie unter Verbesserung der Schreibleistung mit RDS Optimized Writes for My SQL und Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB. Sie können ein Upgrade der Speicherkonfiguration durchführen, um optimierte Schreibvorgänge zu aktivieren, indem Sie eine blue/green Bereitstellung erstellen.
Benachrichtigung	RDS-EVENT-0297	Die Speicherkonfiguration für die DB-Instance *name* unterstützt eine maximale Größe von 16384 GiB. Führen Sie ein Upgrade der Speicherkonfiguration durch, um Speichergrößen von mehr als 16 384 GiB zu unterstützen.	Sie können die zugewiesene Speichergröße der DB-Instance nicht über 16 384 GiB hinaus erhöhen. Um diese Einschränkung zu umgehen, führen Sie ein Upgrade der Speicherkonfiguration durch. Weitere Informationen finden Sie unter Aktualisieren des Speicherdateisystems für eine DB-Instance.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0298	Die Speicherkonfiguration für die DB-Instance *name* unterstützt eine maximale Tabellengröße von 2048 GiB. Führen Sie ein Upgrade der Speicherkonfiguration durch, um Tabellengrößen von mehr als 2048 GiB zu unterstützen.	RDS-MySQL- und MariaDB-Instances mit dieser Einschränkung dürfen keine Tabellengröße von mehr als 2048 GiB haben. Um diese Einschränkung zu umgehen, führen Sie ein Upgrade der Speicherkonfiguration durch. Weitere Informationen finden Sie unter Aktualisieren des Speicherdateisystems für eine DB-Instance.
Benachrichtigung	RDS-EVENT-0327	Amazon RDS konnte das Geheimnis nicht find*SECRET ARN*. *message.*	Keine
Benachrichtigung	RDS-EREIGNIS-0365	Zeitzonendateien wurden aktualisiert. Starten Sie Ihre RDS-Instanz neu, damit die Änderungen wirksam werden.	Keine
Benachrichtigung	RDS-EVENT-0385	Die Cluster-Topologie wurde aktualisiert.	Es gibt DNS-Änderungen am DB-Cluster für die DB-Instance. Dazu gehört auch, wenn neue DB-Instances hinzugefügt oder gelöscht werden oder wenn es zu einem Failover kommt.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0403	Eine Datenbank-Arbeitslast führt dazu, dass dem System extrem wenig Arbeitsspeicher zur Verfügung steht. Um das Problem zu beheben, setzt RDS den Wert von innodb_buffer_pool_size automatisch auf. *amount*	Gilt nur für RDS für MySQL und RDS für MariaDB-DB-Instances.
Benachrichtigung	RDS-EVENT-0404	Eine Datenbank-Arbeitslast führt dazu, dass dem System extrem wenig Arbeitsspeicher zur Verfügung steht. Um das Problem zu beheben, hat RDS den Wert von shared_buffers automatisch auf gesetzt. *amount*	Gilt nur für RDS für PostgreSQL-DB-Instances.
Read Replica	RDS-EVENT-0045	Die Replikation wurde gestoppt.	Diese Meldung wird angezeigt, wenn bei der Replikation ein Fehler auftritt. Informationen zur Bestimmung der Art des Fehlers finden Sie unter Problembehandlung bei einem MySQL-Read Replica-Problem.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Read Replica	RDS-EVENT-0046	Die Replikation für das Lesereplikat wurde fortgesetzt.	Diese Meldung wird beim ersten Erstellen eines Lesereplikats sowie als Überwachungsmeldung zur Bestätigung der ordnungsgemäßen Replikationsausführung angezeigt. Falls diese Meldung auf die Benachrichtigung `RDS-EVENT-0045` folgt, wurde die Replikation (nach einem Fehler oder nachdem sie gestoppt wurde) fortgesetzt.
Read Replica	RDS-EVENT-0057	Das Replikationsstreaming wurde beendet.	Keine
Read Replica	RDS-EVENT-0062	Die Replikation für das Lesereplikat wurde manuell gestoppt.	Keine
Read Replica	RDS-EVENT-0063	Die Replikation von einer Nicht-RDS-Instance wurde zurückgesetzt.	Keine
Read Replica	RDS-EVENT-0202	Read Replica-Erstellung fehlgeschlagen.	Keine
Read Replica	RDS-EVENT-0233	Der Switchover zur Read Replica wurde gestartet.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Read Replica	RDS-EVENT-0357	Der Replikationskanal wurde gestartet. *name*	Hinweise zu Replikationskanälen finden Sie unterthe section called “Konfiguration der Replikation mit mehreren Quellen”.
Read Replica	RDS-EVENT-0358	Der Replikationskanal wurde gestoppt. *name*	Hinweise zu Replikationskanälen finden Sie unterthe section called “Konfiguration der Replikation mit mehreren Quellen”.
Read Replica	RDS-EVENT-0359	Der Replikationskanal wurde manuell gestoppt*name*.	Hinweise zu Replikationskanälen finden Sie unterthe section called “Konfiguration der Replikation mit mehreren Quellen”.
Read Replica	RDS-EVENT-0360	Der Replikationskanal wurde zurückgesetzt. *name*	Hinweise zu Replikationskanälen finden Sie unterthe section called “Konfiguration der Replikation mit mehreren Quellen”.
Read Replica	RDS-EVENT-0415	Der Upgrade-Vorgang hat die Replikation auf der Read Replica wieder aufgenommen.	Keine
Read Replica	RDS-EVENT-0416	Durch den Upgrade-Vorgang wurde die Replikation auf der Read Replica gestoppt.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wiederherstellung	RDS-EVENT-0020	Wiederherstellung der DB-Instance wurde gestartet. Die Wiederherstellungsdauer variiert je nach zu wiederherstellender Datenmenge.	Keine
Wiederherstellung	RDS-EVENT-0021	Wiederherstellung der DB-Instance ist abgeschlossen.	Keine
Wiederherstellung	RDS-EVENT-0023	Auftretende Snapshot-Anfrage:. *message*	Ein manuelles Backup wurde angefordert, jedoch erstellt Amazon RDS gerade einen DB-Snapshot. Senden Sie die Anforderung erneut, nachdem Amazon RDS den DB-Snapshot abgeschlossen hat.
Wiederherstellung	RDS-EVENT-0052	Die Wiederherstellung der Multi-AZ-Instance wurde gestartet.	Die Wiederherstellungsdauer variiert je nach zu wiederherstellender Datenmenge.
Wiederherstellung	RDS-EVENT-0053	Die Wiederherstellung der Multi-AZ-Instance wurde abgeschlossen. Failover oder Aktivierung steht noch aus.	Diese Meldung weist darauf hin, dass Amazon RDS Ihre DB-Instance darauf vorbereitet hat, bei Bedarf ein Failover zur sekundären Instance einzuleiten.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wiederherstellung	RDS-EVENT-0066	Die Instance wird heruntergestuft, während die Spiegelung wiederhergestellt wird:. *message*	Die SQL Server-DB-Instance stellt ihre Spiegelung wieder her. Die Leistung ist beeinträchtigt, bis die Spiegelung wiederhergestellt ist. Datenbank mit einem anderen Wiederherstellungsmodell als FULL gefunden. Das Wiederherstellungsmodell wurde zurück zu FULL geändert und die Spiegelungswiederherstellung wurde gestartet. (<dbname>: <recovery model found>[,…])”
Wiederherstellung	RDS-EVENT-0166	*message*	Die RDS-Custom-DB-Instance befindet sich innerhalb des Support-Umfangs.
Wiederherstellung	RDS-EVENT-0361	Die Wiederherstellung der Standby-DB-Instance wurde gestartet.	Die Standby-DB-Instance wird während des Wiederherstellungsvorgangs neu erstellt. Die Datenbankleistung wird während des Wiederherstellungsprozesses beeinträchtigt.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Wiederherstellung	RDS-EVENT-0362	Die Wiederherstellung der Standby-DB-Instance ist abgeschlossen.	Die Standby-DB-Instanc e wird während des Wiederherstellungs vorgangs neu erstellt. Die Datenbankleistung wird während des Wiederher stellungsprozesses beeinträchtigt.
Wiederherstellung	RDS-EVENT-0019	Von der DB-Instance *name* auf *name* wiederhergestellt.	Die DB-Instance wurde aus einem point-in-time Backup wiederhergestellt.
Sicherheit	RDS-EVENT-0068	Das Passwort der HSM-Partition wird entschlüsselt, um die Instance zu aktualisieren.	RDS entschlüsselt das AWS CloudHSM Partition spasswort, um Aktualisierungen an der DB-Instance vorzunehmen. Weitere Informationen finden Sie unter Oracle Database Transparent Data Encryption (TDE) mit AWS CloudHSM im AWS CloudHSM -Benutzerhandbuch.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausführen von Sicherheits-Patches	RDS-EVENT-0230	Für Ihre DB-Instance ist ein System-Update verfügbar. Informationen zum Anwenden von Updates finden Sie unter „Warten einer DB-Instance“ im RDS-Benutzerhandbuch.	Ein neues Betriebssystem-Update ist verfügbar. Eine neues Nebenversions-Update des Betriebssystems ist für Ihre DB-Instance verfügbar. Informationen zum Anwenden von Aktualisierungen finden Sie unter Betriebssystem-Updates für RDS-DB-Instances.
Wartung	RDS-EVENT-0425	Amazon RDS kann das Betriebssystem-Upgrade nicht durchführen, da in den angegebenen Subnetzen keine IP-Adressen verfügbar sind. Wählen Sie Subnetze mit verfügbaren IP-Adressen aus und versuchen Sie es erneut.	Keine
Wartung	RDS-EVENT-0429	Amazon RDS kann das Betriebssystem-Upgrade nicht durchführen, da für den *type* Instance-Typ in der *zone* Availability Zone nicht genügend Kapazität verfügbar ist	Keine

DB-Parametergruppenereignisse

In der folgenden Tabelle werden die Ereigniskategorie und die Ereignisse für den Quelltyp "DB-Parametergruppe" aufgeführt.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0037	Der Parameter wurde *name* auf die Methode „*value*Mit Anwenden“ aktualisiert*method*.	Keine

DB-Sicherheitsgruppenereignisse

In der folgenden Tabelle werden die Ereigniskategorie und die Ereignisse für den Quelltyp "DB-Sicherheitsgruppe" aufgeführt.

> **Note**
>
> DB-Sicherheitsgruppen sind Ressourcen für EC2 -Classic. EC2-Classic wurde am 15. August 2022 eingestellt. Wenn Sie noch nicht von EC2 -Classic zu einer VPC migriert haben, empfehlen wir Ihnen, so bald wie möglich zu migrieren. Weitere Informationen finden Sie unter Migration von EC2 -Classic zu einer VPC im EC2 Amazon-Benutzerhandbuch und im Blog EC2-Classic Networking is Retiring — So bereiten Sie sich vor.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0038	Die Änderung wurde auf die Sicherheitsgruppe angewendet.	Keine
Ausfall	RDS-EVENT-0039	Autorisierung widerrufen als. *user*	Die Sicherheitsgruppe, deren Eigentümer *user* ist, existiert nicht. Die Autorisierung für die Sicherheitsgruppe wurde widerrufen, weil sie ungültig ist.

DB-Snapshot-Ereignisse

In der folgenden Tabelle werden die Ereigniskategorie und die Ereignisse für den Quelltyp "DB-Snapshot" aufgeführt.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Erstellung	RDS-EVENT-0040	Ein manueller Snapshot wird erstellt.	Keine
Erstellung	RDS-EVENT-0042	Ein manueller Snapshot wurde erstellt.	Keine
Erstellung	RDS-EVENT-0090	Ein automatisierter Snapshot wird erstellt.	Keine
Erstellung	RDS-EVENT-0091	Ein automatisierter Snapshot wurde erstellt.	Keine
Löschung	RDS-EVENT-0041	Der Benutzer-Snapshot wurde gelöscht.	Keine
Benachrichtigung	RDS-EVENT-0059	Die Kopie des Snapshots *name* aus der Region wurde gestartet*name*.	Dies ist eine regionenübergreifenden Snapshot-Kopie.
Benachrichtigung	RDS-EVENT-0060	Die Kopie des Snapshots *name* aus der Region wurde *name* in *number* wenigen Minuten fertiggestellt.	Dies ist eine regionenübergreifenden Snapshot-Kopie.
Benachrichtigung	RDS-EVENT-0061	Die Anforderung zum Kopieren von Snapshots *name* aus der Region wurde storniert*name*.	Dies ist eine regionenübergreifenden Snapshot-Kopie.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Benachrichtigung	RDS-EVENT-0159	Bei der Aufgabe zum Exportieren von Snapshots ist ein Fehler aufgetreten.	Keine
Benachrichtigung	RDS-EVENT-0160	Die Aufgabe zum Exportieren von Snapshots wurde abgebrochen.	Keine
Benachrichtigung	RDS-EVENT-0161	Die Aufgabe zum Exportieren von Snapshots ist abgeschlossen.	Keine
Benachrichtigung	RDS-EVENT-0196	Die Kopie des Snapshots *name* in der Region wurde gestartet*name*.	Dies ist eine lokale Snapshot-Kopie.
Benachrichtigung	RDS-EVENT-0197	Die Kopie des Snapshots *name* in der Region wurde abgeschlossen*name*.	Dies ist eine lokale Snapshot-Kopie.
Benachrichtigung	RDS-EVENT-0190	Die Anfrage zum Kopieren des Snapshots von *name* in der Region wurde abgebrochen*name*.	Dies ist eine lokale Snapshot-Kopie.
Wiederherstellung	RDS-EVENT-0043	Aus dem Snapshot wiederhergestellt*name*.	Eine DB-Instance wird aus einem DB-Snapshot wiederhergestellt.

RDS-Proxy-Ereignisse

Die folgende Tabelle zeigt die Ereigniskategorie und eine Liste von Ereignissen für den Fall, dass ein RDS Proxy der Quelltyp ist.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS-EVENT-0204	RDS-modifizierter DB-Proxy*name*.	Keine
Konfigurationsänderung	RDS-EVENT-0207	RDS hat den Endpunkt des DB-Proxys geändert*name*.	Keine
Konfigurationsänderung	RDS-EVENT-0213	RDS hat das Hinzufügen der DB-Instance erkannt und sie automatisch der Zielgruppe des DB-Proxys hinzugefügt*name*.	Keine
Konfigurationsänderung	RDS-EVENT-0214	RDS hat das Löschen der DB-Instance erkannt *name* und sie automatisch aus *name* der Zielgruppe des DB-Proxys entfernt*name*.	Keine
Konfigurationsänderung	RDS-EVENT-0215	RDS hat das Löschen des DB-Clusters erkannt *name* und ihn automatisch aus *name* der Zielgruppe des DB-Proxys entfernt*name*.	Keine
Erstellung	RDS-EVENT-0203	RDS hat einen DB-Proxy erstellt*name*.	Keine
Erstellung	RDS-EVENT-0206	RDS hat einen Endpunkt *name* für den DB-Proxy erstellt*name*.	Keine
Löschung	RDS-EVENT-0205	RDS hat den DB-Proxy gelöscht*name*.	Keine

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Löschung	RDS-EVENT-0208	RDS hat den Endpunkt *name* für den DB-Proxy gelöscht*name*.	Keine
Ausfall	RDS-EVENT-0243	RDS konnte keine Kapazität für den Proxy bereitstellen, *name* da in Ihren Subnetzen nicht genügend IP-Adressen verfügbar sind:*name*. Stellen Sie sicher, dass Ihre Subnetze die Mindestanzahl unbenutzter IP-Adressen aufweisen, wie in der RDS-Proxy-Dokumentation empfohlen.	Informationen zur Bestimmung der empfohlenen Anzahl für Ihre Instance-Klasse finden Sie unter Planen der Kapazität von IP-Adressen.
Ausfall	RDS-EVENT-0275	RDS hat einige Verbindungen zum DB-Proxy gedrosselt. *name* Die Anzahl der gleichzeitigen Verbindungsanfragen vom Client zum Proxy hat das Limit überschritten.	Keine

Blau/Grün-Bereitstellungsereignisse

Die folgende Tabelle zeigt die Ereigniskategorie und eine Liste von Ereignissen, wenn es sich bei einer blue/green Bereitstellung um den Quelltyp handelt.

Weitere Informationen zu blue/green Bereitstellungen finden Sie unterVerwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Kategorie	Amazon RDS-Ereignis-ID	Fehlermeldung	Hinweise
Erstellung	RDS-EVENT-0244	Die Blau/Grün-Bereitstellungsaufgaben sind abgeschlossen. Sie können weitere Änderungen an den Datenbanken der grünen Umgebung vornehmen oder die Bereitstellung umstellen.	Keine
Ausfall	RDS-EVENT-0245	Die Erstellung der blue/green Bereitstellung ist fehlgeschlagen, weil*reason*.	Keine
Löschung	RDS-EVENT-0246	Die Blau/Grün-Bereitstellung wurde gelöscht.	Keine
Benachrichtigung	RDS-EVENT-0247	Der Switchover von *blue* zu *green* wurde gestartet.	Keine
Benachrichtigung	RDS-EVENT-0248	Der Switchover wurde bei der Bereitstellung abgeschlossen. blue/green	Keine
Ausfall	RDS-EVENT-0249	Der Switchover wurde bei der Bereitstellung abgebrochen. blue/green	Keine
Benachrichtigung	RDS-EVENT-0250	*blue green*	Keine
Benachrichtigung	RDS-EVENT-0251	*blue green* Umbenannt in *blue* und in. *blue-old green blue*	Keine
Ausfall	RDS-EVENT-0252	Der Switchover vom Primär-/Read *blue* auf	Keine

Kategorie	Amazon RDS-Ereignis-ID	Fehlermeldung	Hinweise
		green wurde aufgrund von abgebrochen. *reason*	
Benachrichtigung	RDS-EVENT-0307	Die Sequenzsynchronisierung für den Switchover des . *blue* *green* Die Umstellung kann bei der Verwendung von Sequenzen zu längeren Ausfallzeiten führen.	Keine
Benachrichtigung	RDS-EVENT-0308	Die Sequenzsynchronisierung für den Switchover des *blue* auf *green* ist abgeschlossen.	Keine
Ausfall	RDS-EVENT-0310	Die Sequenzsynchronisierung für den Switchover des *blue* auf *green* wurde abgebrochen, da die Sequenzen nicht synchronisiert werden konnten.	Keine
Benachrichtigung	RDS-EVENT-0405	Ihre Speichervolumes werden initialisiert.	Keine
Benachrichtigung	RDS-EVENT-0406	Ihre Speichervolumes wurden initialisiert.	Keine
Benachrichtigung	RDS-EVENT-0409	*message*	Keine

Benutzerdefinierte Engine-Versionsereignisse

Die folgende Tabelle zeigt den Ereignistyp sowie eine Liste der Ereignisse für den Fall, dass der Quelltyp eine benutzerdefinierte Engine-Version ist.

Kategorie	Amazon RDS-Ereignis-ID	Fehlermeldung	Hinweise
Erstellung	RDS-EVENT-0316	Vorbereitung der Erstellung einer benutzerdefinierten Engine-Version *name* Der gesamte Erstellungsvorgang kann bis zu vier Stunden dauern.	Keine
Erstellung	RDS-EVENT-0317	Benutzerdefinierte Engine-Version wird erstellt*name*.	Keine
Erstellung	RDS-EVENT-0318	Die benutzerdefinierte Engine-Version *name* wird validiert.	Keine
Erstellung	RDS-EVENT-0319	Die benutzerdefinierte Engine-Version *name* wurde erfolgreich erstellt.	Keine
Erstellung	RDS-EVENT-0320	RDS kann *name* aufgrund eines internen Problems keine benutzerdefinierte Engine-Version erstellen. Wir kümmern uns um das Problem und werden uns bei Bedarf mit Ihnen in Verbindung setzen. Wenn Sie weitere Unterstützung benötigen, wenden Sie sich an den AWS Premium Support/.	Keine

Kategorie	Amazon RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS-EVENT-0198	Die Erstellung für die benutzerdefinierte Engine-Version ist fehlgeschlagen*name*. *message*	Das *message* enthält Details zum Fehler, z. B. fehlende Dateien.
Ausfall	RDS-EVENT-0277	Fehler beim Löschen der benutzerdefinierten Engine-Version*name*. *message*	Das *message* beinhaltet Details zum Fehler.
Wiederherstellen	RDS-EVENT-0352	Die maximale Datenbankanzahl, die für point-in-time die Wiederherstellung unterstützt wird, hat sich geändert.	Das *message* beinhaltet Einzelheiten über das Ereignis.

Überwachung von Amazon RDS Amazon

Jede RDS Datenbank-Engine generiert Protokolle, auf die Sie zur Prüfung und Problembehandlung zugreifen können. Der Typ der Protokolle hängt von Ihrer Datenbank-Engine ab.

Sie können mit dem AWS Management Console, dem AWS Command Line Interface (AWS CLI) oder dem Amazon auf Datenbankprotokolle für DB-Instances zugreifen RDSAPI. Sie können Transaktionsprotokolle nicht anzeigen, ansehen oder herunterladen.

Themen

- Anzeigen und Auflisten von Datenbank-Protokolldateien
- Herunterladen einer Datenbank-Protokolldatei
- Überwachen einer Datenbank-Protokolldatei
- Datenbankprotokolle in Amazon CloudWatch Logs veröffentlichen
- Lesen von Protokolldateiinhalten mit REST
- Amazon RDS for Db2-Datenbank-Protokolldateien
- MariaDB-Datenbank-Protokolldateien
- Datenbankprotokolldateien von Amazon RDS für Microsoft SQL Server
- MySQL-Datenbank-Protokolldateien
- Datenbankprotokolldateien von Amazon RDS für Oracle
- Datenbank-Protokolldateien von RDS für PostgreSQL

Anzeigen und Auflisten von Datenbank-Protokolldateien

Sie können Datenbank-Protokolldateien für Ihre Amazon RDS Amazon anzeigen, indem Sie den verwenden AWS Management Console. Sie können auflisten, welche Protokolldateien zum Herunterladen oder zur Überwachung verfügbar sind, indem Sie das AWS CLI oder Amazon verwenden RDSAPI.

> **Note**
>
> Wenn Sie die Liste der Protokolldateien für eine bestehende RDS Oracle-DB-Instance nicht anzeigen können, starten Sie die Instance neu, um die Liste anzuzeigen.

Konsole

So zeigen Sie eine Datenbank-Protokolldatei an

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, welche die anzuzeigende Protokolldatei enthält.
4. Wählen Sie die Registerkarte Logs & events (Protokolle und Ereignisse).
5. Scrollen Sie nach unten bis zum Abschnitt Protokolle.
6. (Optional) Geben Sie einen Suchbegriff ein, um Ihre Ergebnisse zu filtern.
7. Wählen Sie das gewünschte Protokoll und dann View (Anzeigen) aus.

AWS CLI

Verwenden Sie den AWS CLI `describe-db-log-files`Befehl, um die verfügbaren Datenbank-Protokolldateien für eine DB-Instance aufzulisten.

Das folgende Beispiel gibt eine Liste von Protokolldateien für eine DB-Instance namens zurüc `my-db-instance`.

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

RDS API

Verwenden Sie die RDS API `DescribeDBLogFiles`Amazon-Aktion, um die verfügbaren Datenbank-Protokolldateien für eine DB-Instance aufzulisten.

Herunterladen einer Datenbank-Protokolldatei

Sie können das AWS Management Console, oder verwenden AWS CLI, API um eine Datenbank-Logdatei herunterzuladen.

Konsole

So laden Sie eine Datenbank-Protokolldatei herunter

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, welche die anzuzeigende Protokolldatei enthält.
4. Wählen Sie die Registerkarte Logs & events (Protokolle und Ereignisse).
5. Scrollen Sie nach unten bis zum Abschnitt Protokolle.
6. Klicken Sie im Bereich Protokolle auf die Schaltfläche neben dem gewünschten Protokoll und wählen Sie Herunterladen.
7. Öffnen Sie das Kontextmenü (rechte Maustaste) für den bereitgestellten Link und wählen Sie Save Link As (Link speichern unter) aus. Geben Sie den Speicherort für die Protokolldatei ein und klicken Sie dann auf Speichern.



AWS CLI

Verwenden Sie den AWS CLI Befehl, um eine Datenbank-Logdatei herunterzuladen `download-db-log-file-portion`. Standardmäßig lädt dieser Befehl nur den neuesten Teil einer Protokolldatei herunter. Sie können jedoch eine ganze Datei herunterladen, indem Sie den Parameter angebe `--starting-token 0`.

Das folgende Beispiel zeigt, wie der gesamte Inhalt einer Protokolldatei namens log/ ERROR .4 heruntergeladen und in einer lokalen Datei namens errorlog.txt gespeichert wird.

Example

FürLinux, odermacOS: Unix

```
aws rds download-db-log-file-portion \
```

```
    --db-instance-identifier myexampledb \
    --starting-token 0 --output text \
    --log-file-name log/ERROR.4 > errorlog.txt
```

Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier myexampledb ^
    --starting-token 0 --output text ^
    --log-file-name log/ERROR.4 > errorlog.txt
```

RDS API

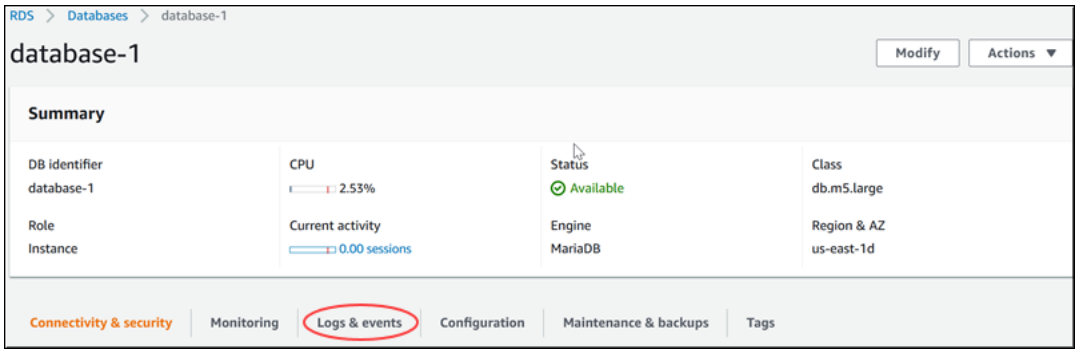
Verwenden Sie die RDS API DownloadDBLogFilePortionAmazon-Aktion, um eine Datenbank-Protokolldatei herunterzuladen.

Überwachen einer Datenbank-Protokolldatei

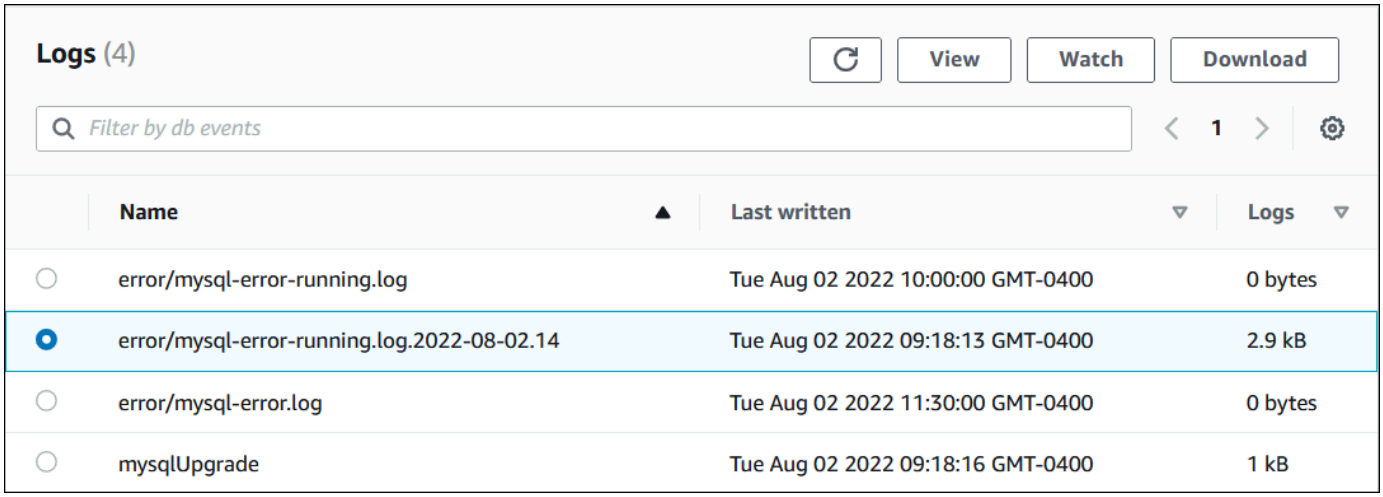
Das Ansehen einer Datenbank-Logdatei entspricht dem Verfolgen der Datei auf einem UNIX oder Linux-System. Sie können eine Protokolldatei mithilfe der AWS Management Consoleüberwachen. RDSaktualisiert das Ende des Protokolls alle 5 Sekunden.

So überwachen Sie eine Datenbank-Protokolldatei

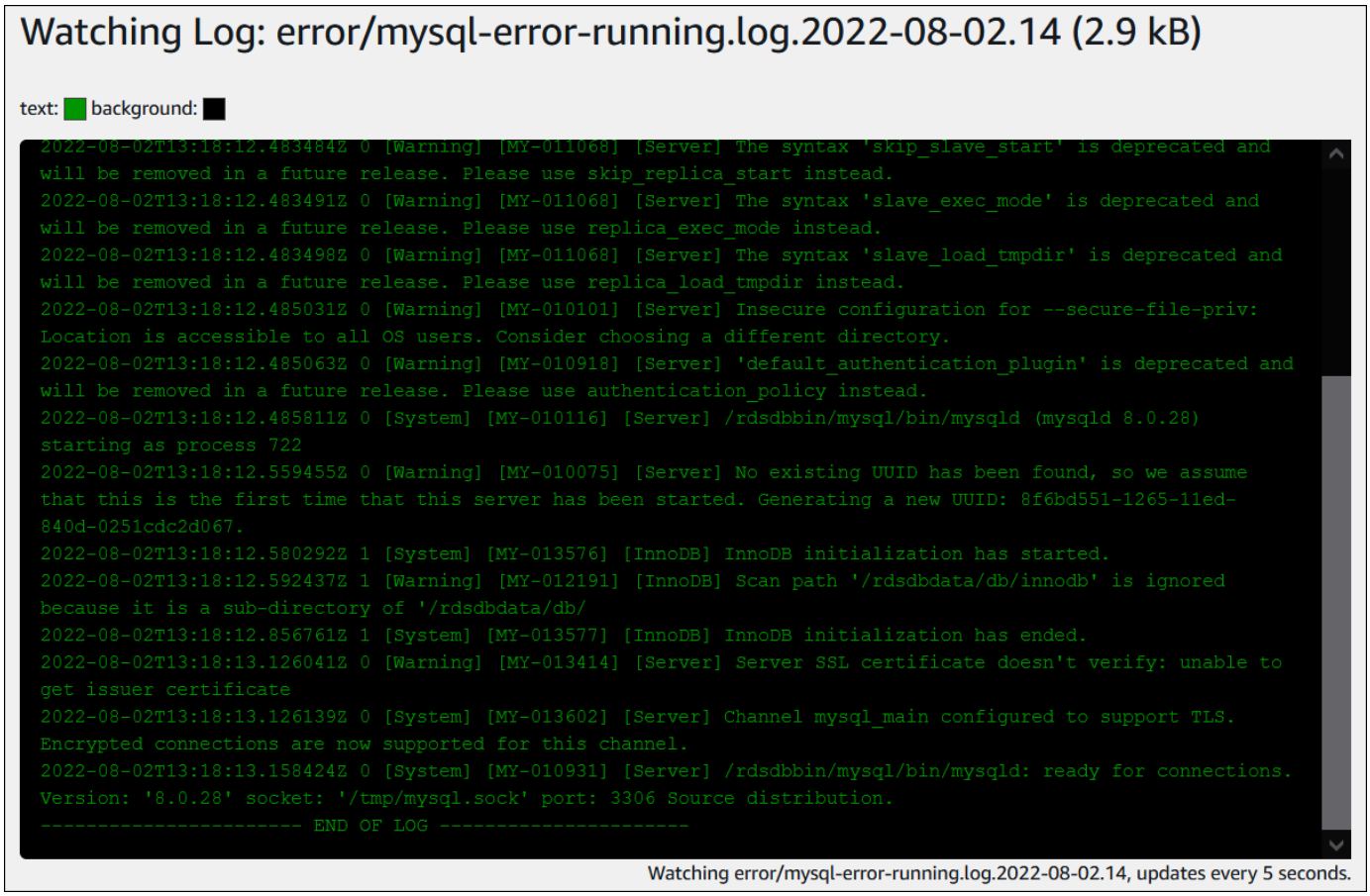
1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, welche die anzuzeigende Protokolldatei enthält.
4. Wählen Sie die Registerkarte Logs & events (Protokolle und Ereignisse).



5. Wählen Sie im Abschnitt Protokolle eine Protokolldatei und wählen Sie dann Watch (Beobachten).



RDSzeigt das Ende des Protokolls, wie im folgenden SQL Beispiel.



Datenbankprotokolle in Amazon CloudWatch Logs veröffentlichen

In einer On-Premises-Datenbank befinden sich die Datenbankprotokolle im Dateisystem. Amazon RDS bietet keinen Host-Zugriff auf die Datenbankprotokolle im Dateisystem Ihrer DB-Instance. Aus

diesem Grund können Sie mit Amazon RDS Datenbankprotokolle nach Amazon CloudWatch Logs exportieren. Mit CloudWatch Logs können Sie eine Echtzeitanalyse der Protokolldaten durchführen. Sie können die Daten auch in einem äußerst langlebigen Speicher speichern und die Daten mit dem CloudWatch Logs Agent verwalten.

Themen

- Überblick über die RDS-Integration mit CloudWatch Logs
- Entscheiden, welche Protokolle in CloudWatch Logs veröffentlicht werden sollen
- Angabe der Protokolle, die in Logs veröffentlicht werden CloudWatch sollen
- Suchen und Filtern Ihrer Logs in CloudWatch Logs

Überblick über die RDS-Integration mit CloudWatch Logs

In CloudWatch Logs ist ein Protokollstream eine Folge von Protokollereignissen, die dieselbe Quelle haben. Jede einzelne Protokollquelle in CloudWatch Logs bildet einen separaten Protokollstream. Eine Protokollgruppe ist eine Gruppe von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen.

Amazon RDS streamt kontinuierlich die Protokolldatensätze Ihrer DB-Instance in eine Protokollgruppe. Sie haben z. B. eine Protokollgruppe `/aws/rds/instance/`*`instance_name`*`/`*`log_type`* für jeden Protokolltyp, den Sie veröffentlichen. Diese Protokollgruppe befindet sich in derselben AWS Region wie die Datenbankinstanz, die das Protokoll generiert.

AWS speichert Protokolldaten, die in CloudWatch Logs veröffentlicht wurden, auf unbestimmte Zeit, sofern Sie keinen Aufbewahrungszeitraum angeben. Weitere Informationen finden Sie unter Aufbewahrung von Protokolldaten ändern in CloudWatch Logs.

Entscheiden, welche Protokolle in CloudWatch Logs veröffentlicht werden sollen

Jede RDS-Datenbank-Engine unterstützt einen eigene Gruppe von Protokollen. Lesen Sie die folgenden Themen, um mehr über die Optionen für Ihre Datenbank-Engine zu erfahren:

- the section called “Veröffentlichen von Db2-Protokollen in Amazon CloudWatch Logs”
- the section called “Veröffentlichen von MariaDB-Protokollen in Amazon Logs CloudWatch ”
- the section called “Veröffentlichen von MySQL-Protokollen in Amazon CloudWatch Logs”
- the section called “Oracle-Logs in Amazon CloudWatch Logs veröffentlichen”

- the section called “PostgreSQL-Protokolle in Amazon Logs veröffentlichen CloudWatch ”
- the section called “SQLServerprotokolle in Amazon CloudWatch Logs veröffentlichen”

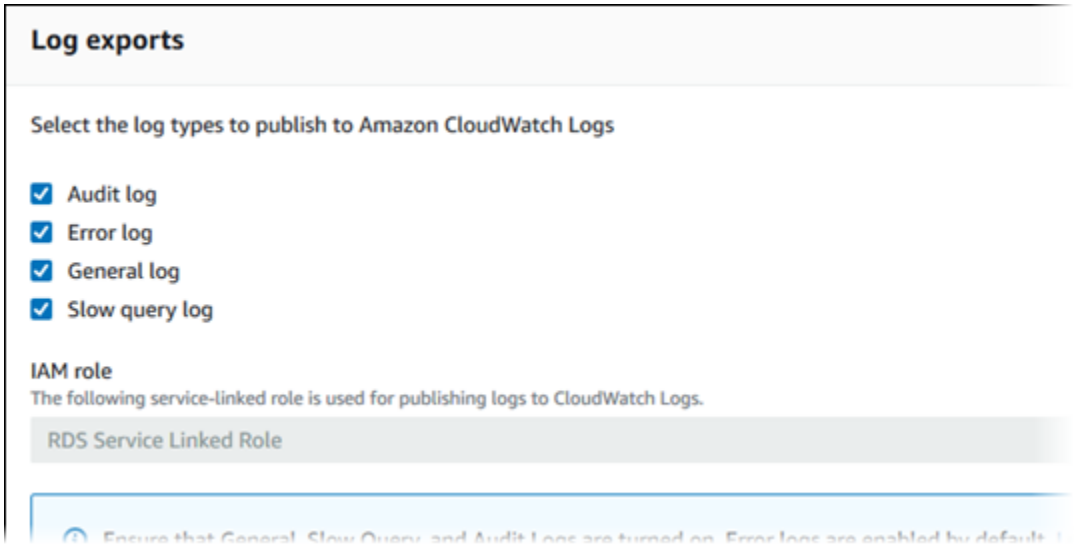
Angabe der Protokolle, die in Logs veröffentlicht werden CloudWatch sollen

Sie geben an, welche Protokolle in der Konsole veröffentlicht werden sollen. Stellen Sie sicher, dass Sie eine dienstbezogene Rolle in AWS Identity and Access Management (IAM) haben. Weitere Informationen zu serviceverknüpften Rollen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon RDS.

So geben Sie die Protokolle an, die veröffentlicht werden sollen

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Führen Sie eine der folgenden Aufgaben aus:
   - Wählen Sie Datenbank erstellen aus.
   - Wählen Sie eine Datenbank in der Liste und dann Modify (Ändern) aus.
4. Wählen Sie in Logs exports (Protokollexporte) die Protokolle aus, die veröffentlicht werden sollen.

   Das folgende Beispiel spezifiziert das Auditprotokoll, die Fehlerprotokolle, das allgemeine Protokoll, das für langsame Abfragen für eine für MySQL-DB-Instance.



Suchen und Filtern Ihrer Logs in CloudWatch Logs

In der CloudWatch Logs-Konsole können Sie nach Protokolleinträgen suchen, die bestimmte Kriterien erfüllen. Sie können entweder über die RDS-Konsole, die Sie zur Logs-Konsole führt, oder direkt über die CloudWatch CloudWatch Logs-Konsole auf die Protokolle zugreifen.

So suchen Sie Ihre RDS-Protokolle mithilfe der RDS-Konsole

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie ein DB-Instance aus.
4. Wählen Sie Konfiguration.
5. Wählen Sie unter Published logs (Veröffentlichte Protokolle) das Datenbankprotokoll aus, das Sie anzeigen möchten.

Um Ihre RDS-Logs mit der CloudWatch Logs-Konsole zu durchsuchen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Protokollgruppen aus.
3. Geben Sie im Filterfeld **`/aws/rds`** ein.
4. Wählen Sie für Log Groups den Namen der Protokollgruppe mit dem Protokoll-Stream aus, nach dem gesucht werden soll.
5. Wählen Sie für Log Streams den Namen des zu suchenden Protokoll-Streams.
6. Geben Sie unter Protokollereignisse die zu verwendende Filtersyntax ein.

Weitere Informationen finden Sie unter Suchen und Filtern von Protokolldaten im Amazon CloudWatch Logs-Benutzerhandbuch. Ein Blog-Tutorial zur Überwachung von RDS-Protokollen finden Sie unter Aufbau einer proaktiven Datenbanküberwachung für Amazon RDS mit Amazon CloudWatch Logs, AWS Lambda und Amazon SNS.

Lesen von Protokolldateiinhalten mit REST

Amazon RDS bietet einen REST Endpunkt, der den Zugriff auf DB-Instance-Protokolldateien ermöglicht. Dies ist nützlich, wenn Sie eine Anwendung schreiben müssen, um den Inhalt der RDS Amazon-Protokolldatei zu streamen.

Die Syntax lautet wie folgt:

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

Die folgenden Parameter sind erforderlich:

- *DBInstanceIdentifier*: der Name der DB-Instance, welche die Protokolldatei enthält, die Sie herunterladen möchten.
- *LogFileName*: der Name der Protokolldatei, die heruntergeladen werden soll.

Die Antwort enthält die Inhalte der angeforderten Protokolldatei als Stream zurück.

Im folgenden Beispiel wird die Protokolldatei mit dem Namen log/ ERROR .6 für die DB-Instance mit dem Namen sample-sql in der Region us-west-2 heruntergeladen.

```
GET /v13/downloadCompleteLogFile/sample-sql/log/ERROR.6 HTTP/1.1
host: rds.us-west-2.amazonaws.com
X-Amz-Security-Token: AQoDYXdzEIH//////////
wEa0AIXLhngC5zp9CyB1R6abwKrXHVR5efnAVN3XvR7IwqKYalFSn6UyJuEFTft9nObglx4QJ+GXV9cpACkETq=
X-Amz-Date: 20140903T233749Z
X-Amz-Algorithm: AWS4-HMAC-SHA256
X-Amz-Credential: AKIADQKE4SARGYLE/20140903/us-west-2/rds/aws4_request
X-Amz-SignedHeaders: host
X-Amz-Content-SHA256: e3b0c44298fc1c229afbf4c8996fb92427ae41e4649b934de495991b7852b855
X-Amz-Expires: 86400
X-Amz-Signature: 353a4f14b3f250142d9afc34f9f9948154d46ce7d4ec091d0cdabbcf8b40c558
```

Wenn Sie eine nicht vorhandene DB-Instance angeben, besteht die Antwort aus dem folgenden Fehler:

- `DBInstanceNotFound`: *DBInstanceIdentifier* bezieht sich nicht auf eine vorhandene DB-Instance. HTTP(Statuscode: 404)

Amazon RDS for Db2-Datenbank-Protokolldateien

Sie können auf Db2-Diagnose- und Benachrichtigungsprotokolle zugreifenRDS, indem Sie die RDS Amazon-Konsole verwenden, AWS CLI, oder RDSAPI. Weitere Informationen zum Anzeigen und Herunterladen von dateibasierten Datenbankprotokollen finden Sie unter Überwachung von Amazon RDS Amazon.

Themen

- Aufbewahrungsplan
- Veröffentlichen von Db2-Protokollen in Amazon CloudWatch Logs

Aufbewahrungsplan

Protokolldateien werden jeden Tag rotiert und jedes Mal, wenn Ihre DB-Instance neu gestartet wird. Im Folgenden finden Sie den Aufbewahrungszeitplan RDS für Db2-Protokolle bei AmazonRDS.

Protokolltyp	Aufbewahrungsplan
Diagnoseprotokolle	Db2 löscht Protokolle außerhalb der Aufbewahrungseinstellungen in der Konfiguration auf Instanzebene. Amazon RDS setzt den `diagsize` Parameter auf 1000.
Protokolle benachrichtigen	Db2 löscht Protokolle außerhalb der Aufbewahrungseinstellungen in der Konfiguration auf Instanzebene. Amazon RDS setzt den `diagsize` Parameter auf 1000.

Veröffentlichen von Db2-Protokollen in Amazon CloudWatch Logs

Mit RDS for Db2 können Sie Diagnose- und Benachrichtigungsprotokollereignisse direkt in Amazon CloudWatch Logs veröffentlichen. Analysieren Sie die Protokolldaten mit CloudWatch Logs und verwenden Sie sie dann, CloudWatch um Alarme zu erstellen und Metriken anzuzeigen.

Mit CloudWatch Logs können Sie Folgendes tun:

- Speichern von Protokollen in hoch dauerhaften Speichern mit einem von Ihnen festgelegten Aufbewahrungszeitraum.
- Durchsuchen und Filtern von Protokolldaten

- Protokolldateien zwischen Konten freigeben.
- Exportieren von Protokollen zu Amazon S3.
- Streamen Sie Daten an Amazon OpenSearch Service.
- Verarbeiten von Protokolldaten in Echtzeit mit Amazon Kinesis Data Streams. Weitere Informationen finden Sie unter Arbeiten mit Amazon CloudWatch Logs im Amazon Managed Service for Apache Flink for SQL Applications Developer Guide.

Amazon RDS veröffentlicht jedes Datenbankprotokoll RDS für DB2 als separaten Datenbank-Stream in der Protokollgruppe. Wenn Sie beispielsweise die Diagnose-Logs und Notify-Logs veröffentlichen, werden die Diagnosedaten in einem Diagnose-Log-Stream in der `/aws/rds/instance/my_instance/diagnostic` Log-Gruppe gespeichert, und die Notify-Logdaten werden in der `/aws/rds/instance/my_instance/notify` Log-Gruppe gespeichert.

> **Note**
>
> Das RDS Veröffentlichen von CloudWatch DB2-Protokollen in Logs ist standardmäßig nicht aktiviert. Das Veröffentlichen von Statistikprotokollen mit automatischer Optimierung des Speichermanagers (STMM) und des Optimierers wird nicht unterstützt. Das RDS Veröffentlichen von Db2-Protokollen in CloudWatch Logs wird in allen Regionen unterstützt, mit Ausnahme von Asien-Pazifik (Hongkong).

Konsole

RDSZum Veröffentlichen von DB2-Protokollen in Logs aus dem CloudWatch AWS Management Console

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus.
4. Wählen Sie im Abschnitt Protokollexporte die Protokolle aus, die Sie in CloudWatch Logs veröffentlichen möchten.

   Sie können diag.log, notify.log oder beide auswählen.

5. Wählen Sie Weiter und dann auf der zusammenfassenden Seite Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Um RDS für Db2-Protokolle zu veröffentlichen, können Sie den `modify-db-instance`Befehl mit den folgenden Parametern verwenden:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Eine Änderung der Option `--cloudwatch-logs-export-configuration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund sind die Optionen `--apply-immediately` und `--no-apply-immediately` wirkungslos.

Mit den folgenden Befehlen können Sie auch RDS für DB2-Protokolle veröffentlichen:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

Im folgenden Beispiel wird eine DB-Instance RDS für DB2 erstellt, bei der die Veröffentlichung von CloudWatch Protokollen aktiviert ist. Der `--enable-cloudwatch-logs-exports` Wert ist ein JSON Array von Zeichenfolgen, die `diag.lognotify.log`, oder beides enthalten können.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["diag.log","notify.log"]' \
    --db-instance-class db.m4.large \
    --engine db2-se
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports "[\"diag.log\",\"notify.log\"]" ^
    --db-instance-class db.m4.large ^
    --engine db2-se
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

Example

Im folgenden Beispiel wird eine bestehende DB-Instance RDS für Db2 so geändert, dass Protokolldateien in Logs veröffentlicht werden. CloudWatch Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Objekt. Der Schlüssel für dieses Objekt ist `EnableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit `diag.log`, `notify.log` oder beiden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["diag.log","notify.log"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"EnableLogTypes\":[\"diag.log\",
\"notify.log\"]}"
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

Example

Im folgenden Beispiel wird eine bestehende DB-Instance RDS für Db2 dahingehend geändert, dass die Veröffentlichung von Diagnoseprotokolldateien in Logs deaktiviert wird. CloudWatch Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Objekt. Der Schlüssel für dieses Objekt ist `DisableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit `diag.log`, `notify.log` oder beiden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["diag.log"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"diag.log\"]}"
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

MariaDB-Datenbank-Protokolldateien

Sie können das MariaDB-Fehlerprotokoll, das langsame Abfrageprotokoll, das Fehlerprotokoll der IAM-Datenbankauthentifizierung und das allgemeine Protokoll überwachen. Das MariaDB-Fehlerprotokoll wird standardmäßig generiert; Sie können die langsame Abfrage und allgemeine Protokolle generieren, indem Sie Parameter in Ihrer DB-Parametergruppe festlegen. Amazon RDS rotiert alle MariaDB-Protokolldateien; die Intervalle für jeden Typ sind im Folgenden angegeben.

Sie können die MariaDB-Protokolle direkt über die Amazon RDS-Konsole, Amazon RDS-API, Amazon RDS-CLI oder überwachen. AWS SDKs Sie können auf MariaDB-Protokolle auch direkt zugreifen, indem Sie die Protokolle in eine Datenbank-Tabelle in der Hauptdatenbank weiterleiten und diese Tabelle abfragen. Mit dem Dienstprogramm "mysqlbinlog" können Sie ein binäres Protokoll herunterladen.

Weitere Informationen zum Anzeigen und Herunterladen von dateibasierten Datenbankprotokollen finden Sie unter Überwachung von Amazon RDS Amazon.

Themen

- Zugriff auf MariaDB-Fehlerprotokolle
- Zugriff auf die MariaDB-Slow-Query- und allgemeinen Protokolle
- Veröffentlichen von MariaDB-Protokollen in Amazon Logs CloudWatch
- Protokollrotation und Aufbewahrung für MariaDB
- Verwalten von tabellenbasierten MariaDB-Protokollen
- Konfiguration der MariaDB-Binärprotokollierung
- Zugriff auf binäre MariaDB-Protokolle
- MariaDB-Binärlog-Annotation aktivieren

Zugriff auf MariaDB-Fehlerprotokolle

Das MariaDB-Fehlerprotokoll wird in die `<host-name>.err`-Datei geschrieben. Sie können diese Datei mit der RDS Amazon-Konsole anzeigen. Sie können das Protokoll auch über Amazon RDS API RDSCLI, Amazon oder abrufen AWS SDKs. Die Datei `<host-name>.err` wird alle 5 Minuten bereinigt und ihre Inhalte werden an `mysql-error-running.log` angefügt. Die `mysql-error-running.log`-Datei wird dann jede Stunde rotiert und die stündlich erstellten Dateien der letzten 24 Stunden werden aufbewahrt. An den Namen jeder Protokolldatei wird die Stunde angehängt, zu

der sie generiert wurde (inUTC). Die Protokolldateien verfügen auch über einen Zeitstempel, anhand dessen Sie feststellen können, wann die Protokolleinträge geschrieben wurden.

MariaDB schreibt das Fehlerprotokoll nur beim Startup, Herunterfahren und beim Auftreten von Fehlern. Eine DB-Instance kann Stunden oder Tage lang laufen, ohne dass neue Einträge in das Fehlerprotokoll geschrieben werden. Wenn Sie keine neuen Einträge sehen, sind im Server keine Fehler aufgetreten, die zu einem Eintrag in das Protokoll geführt hätten.

Zugriff auf die MariaDB-Slow-Query- und allgemeinen Protokolle

Das Slow-Query-Protokoll von MariaDB und das allgemeine Protokoll können in eine Datei oder in eine Datenbanktabelle mithilfe der Parametereinstellungen in Ihrer DB-Parametergruppe geschrieben werden. Weitere Informationen zum Erstellen und Ändern einer DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS. Sie müssen diese Parameter festlegen, bevor Sie das Protokoll für langsame Abfragen oder das allgemeine Protokoll in der RDS Amazon-Konsole oder mithilfe von Amazon RDSAPI, AWS CLI, oder anzeigen können AWS SDKs.

Sie können MariaDB-Protokolle mithilfe der Parameter in dieser Liste kontrollieren:

- `slow_query_log`Oder`log_slow_query`: Um das Protokoll für langsame Abfragen zu erstellen, setzen Sie den Wert auf 1. Der Standardwert ist 0.
- `general_log` Um das allgemeine Protokoll zu erstellen, auf 1 setzen. Der Standardwert ist 0.
- `long_query_time`Oder`log_slow_query_time`: Um zu verhindern, dass Abfragen mit schneller Ausführung im Protokoll für langsame Abfragen protokolliert werden, geben Sie einen Wert für die kürzeste zu protokollierende Abfragelaufzeit in Sekunden an. Der Standardwert liegt bei 10 Sekunden, der Mindestwert bei 0. Wenn log_output = istFILE, können Sie einen Fließkommawert mit einer Auflösung im Mikrosekundenbereich angeben. Wenn log_output = istTABLE, müssen Sie einen Ganzzahlwert mit zweiter Auflösung angeben. Nur Abfragen, deren Laufzeit den `log_slow_query_time` Wert `long_query_time` oder überschreitet, werden protokolliert. Wenn Sie beispielsweise `long_query_time` oder `log_slow_query_time` auf 0,1 setzen, wird verhindert, dass Abfragen, die weniger als 100 Millisekunden lang ausgeführt werden, protokolliert werden.
- `log_queries_not_using_indexes`: Um alle Abfragen zu protokollieren, die keinen Index für das Slow-Query-Protokoll verwenden, setzen Sie diesen Parameter auf 1. Der Standardwert ist 0. Abfragen, die keinen Index verwenden, werden protokolliert, auch wenn ihre Laufzeit niedriger als der Wert des `long_query_time`-Parameters ist.
- `log_output` *`option`*: Sie können eine der folgenden Optionen für den `log_output`-Parameter festlegen:

- TABLE(Standard) — Schreiben Sie allgemeine Abfragen in die `mysql.general_log` Tabelle und langsame Abfragen in die Tabelle. `mysql.slow_log`
- FILE— Schreibt sowohl allgemeine als auch langsame Abfrageprotokolle in das Dateisystem. Protokolldateien werden stündlich rotiert.
- NONE— Deaktiviert die Protokollierung.

Wenn die Protokollierung aktiviert ist, RDS rotiert Amazon Tabellenprotokolle oder löscht Protokolldateien in regelmäßigen Abständen. Dies ist eine Vorsichtsmaßnahme, um möglichst zu vermeiden, dass eine umfangreiche Protokolldatei die Datenbanknutzung blockiert oder die Leistung beeinträchtigt. `FILE`- und `TABLE`-Protokollierung führen das Rotieren und Löschen wie folgt aus:

- Wenn die `FILE`-Protokollierung aktiviert ist, werden Protokolldateien stündlich geprüft und Protokolldateien, die älter als 24 Stunden sind, werden gelöscht. In einigen Fällen kann die Größe der verbleibenden kombinierten Protokolldatei nach dem Löschen die Schwelle von 2 % des zugeteilten Speicherplatzes für eine DB-Instance überschreiten. In diesen Fällen werden die umfangreichsten Protokolldateien gelöscht, bis die Größe den Schwellenwert nicht mehr überschreitet.
- Wenn die `TABLE`-Protokollierung aktiviert ist, werden in einigen Fällen Protokolltabellen alle 24 Stunden überschrieben. Diese Rotation erfolgt, wenn der von den Tabellenprotokollen verwendete Speicherplatz mehr als 20 Prozent des zugeteilten Speicherplatzes ausmacht. Sie tritt auch auf, wenn die Größe aller kombinierten Protokolle mehr als 10 GB beträgt. Wenn der für eine DB-Instance verwendete Speicherplatz 90 Prozent des Speicherplatzes überschreitet, der der DB-Instance zugeteilt ist, werden die Grenzwerte für die Protokollrotation reduziert. Protokolltabellen werden dann rotiert, wenn der von den Tabellenprotokollen verwendete Speicherplatz mehr als 10 Prozent des zugeteilten Speicherplatzes ausmacht. Sie werden auch rotiert, wenn die Größe aller kombinierten Protokolle mehr als 5 GB beträgt.

  Beim Rotieren von Protokolldateien wird die aktuelle Protokolltabelle in eine Sicherungsprotokolltabelle kopiert, und die Einträge in der aktuellen Protokolltabelle werden entfernt. Sofern bereits eine Sicherungsprotokolltabelle vorhanden ist, wird diese gelöscht, bevor die aktuelle Protokolltabelle ins Backup kopiert wird. Sie können die Sicherungsprotokolltabelle abfragen, wenn dies nötig ist. Die Backup-Protokolltabelle für die `mysql.general_log`-Tabelle ist als `mysql.general_log_backup` benannt. Die Backup-Protokolltabelle für die `mysql.slow_log`-Tabelle ist als `mysql.slow_log_backup` benannt.

Sie können die `mysql.general_log`-Tabelle rotieren, wenn Sie die Prozedur `mysql.rds_rotate_general_log` aufrufen. Sie können die `mysql.slow_log`-Tabelle rotieren, wenn Sie die Prozedur `mysql.rds_rotate_slow_log` aufrufen.

Tabellenprotokolle werden während des Upgrades einer Datenbankversion rotiert.

Amazon RDS zeichnet beides auf `TABLE` und `FILE` protokolliert die Rotation in einem RDS Amazon-Ereignis und sendet Ihnen eine Benachrichtigung.

Um mit den Protokollen von der RDS Amazon-Konsole, Amazon RDSAPI, Amazon RDS CLI oder zu arbeiten AWS SDKs, setzen Sie den `log_output` Parameter aufFILE. Wie das MariaDB-Fehlerprotokoll werden auch diese Protokolldateien stündlich rotiert. Die Protokolldateien, die während der vorherigen 24 Stunden angelegt wurden, werden aufbewahrt.

Weitere Informationen zu den Slow-Query- und allgemeinen Protokollen finden Sie in den folgenden Themen in der MariaDB-Dokumentation:

- Slow-Query-Protokoll
- Allgemeines Abfrageprotokoll

Veröffentlichen von MariaDB-Protokollen in Amazon Logs CloudWatch

Sie können Ihre MariaDB-DB-Instance so konfigurieren, dass Protokolldaten in einer Protokollgruppe in Amazon CloudWatch Logs veröffentlicht werden. Mit CloudWatch Logs können Sie eine Echtzeitanalyse der Protokolldaten durchführen und diese CloudWatch zum Erstellen von Alarmen und zum Anzeigen von Metriken verwenden. Sie können CloudWatch Logs verwenden, um Ihre Protokolldatensätze in einem äußerst langlebigen Speicher zu speichern.

Amazon RDS veröffentlicht jedes MariaDB-Datenbankprotokoll als separaten Datenbank-Stream in der Protokollgruppe. Angenommen, Sie konfigurieren die Exportfunktion so, dass sie das Slow-Query-Protokoll enthält. Dann werden Slow-Query-Daten in einem Slow-Query-Protokollstream in der `/aws/rds/instance/`*`my_instance`*`/slowquery`-Protokollgruppe gespeichert.

Das Fehlerprotokoll ist standardmäßig aktiviert. Die folgende Tabelle fasst die Anforderungen für die anderen MariaDB-Protokolle zusammen.

Protokoll	Anforderung
Prüfungsprotokoll	Die DB-Instance muss eine benutzerdefinierte Optionsgruppe mit der Option `MARIADB_AUDIT_PLUGIN` verwenden.
Allgemeines Protokoll	Die DB-Instance muss eine benutzerdefinierte Parametergruppe mit der Parametereinstellung `general_log = 1` verwenden, um das allgemeine Protokoll zu aktivieren.
Slow-Query-Protokoll	Die DB-Instance muss eine benutzerdefinierte Parametergruppe mit der Parametereinstellung `slow_query_log = 1` oder `log_slow_query = 1` zur Aktivierung des langsamen Abfrageprotokolls verwenden.
Fehlerprotokoll für die IAM-Datenbankauthentifizierung	Sie müssen den Protokolltyp `iam-db-auth-error` für eine DB-Instance aktivieren, indem Sie eine DB-Instance erstellen oder ändern.
Protokollausgabe	Die DB-Instance muss eine benutzerdefinierte Parametergruppe mit der Parametereinstellung verwenden`log_output = FILE` , um Protokolle in das Dateisystem zu schreiben und sie in CloudWatch Logs zu veröffentlichen.

Konsole

Um MariaDB-Protokolle von der Konsole aus in CloudWatch Logs zu veröffentlichen

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus.
4. Wählen Sie im Abschnitt Protokollexporte die Protokolle aus, die Sie in Logs veröffentlichen möchten CloudWatch .

5. Wählen Sie Weiter und dann auf der zusammenfassenden Seite Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Sie können ein MariaDB-Protokoll mit dem veröffentlichen. AWS CLI Sie können den Befehl `modify-db-instance` mit den folgenden Parametern aufrufen:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Eine Änderung der Option `--cloudwatch-logs-export-configuration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund sind die Optionen `--apply-immediately` und `--no-apply-immediately` wirkungslos.

Sie können MariaDB-Protokolle auch veröffentlichen, indem Sie die folgenden AWS CLI Befehle aufrufen:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Führen Sie einen dieser AWS CLI Befehle mit den folgenden Optionen aus:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Je nachdem, welchen AWS CLI Befehl Sie ausführen, sind möglicherweise andere Optionen erforderlich.

Example

Das folgende Beispiel ändert eine bestehende MariaDB-DB-Instance, um Protokolldateien in Logs zu veröffentlichen. CloudWatch Der `--cloudwatch-logs-export-configuration`-Wert ist ein JSON-Objekt. Der Schlüssel für dieses Objekt ist `EnableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit einer beliebigen Kombination aus `audit`, `error`, `general` und `slowquery`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

Der folgende Befehl erstellt eine MariaDB-DB-Instance und veröffentlicht Protokolldateien in Logs. CloudWatch Der Wert `--enable-cloudwatch-logs-exports` ist ein JSON-Array mit Zeichenfolgen. Die Zeichenfolgen können eine beliebige Kombination von `audit`, `error`, `general` und `slowquery` sein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine mariadb
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine mariadb
```

RDS-API

Sie können MariaDB-Protokolle mithilfe der RDS-API veröffentlichen. Sie können die ModifyDBInstance-Operation mit folgenden Parametern aufrufen:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Eine Änderung des Parameters `CloudwatchLogsExportConfiguration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund ist der Parameter `ApplyImmediately` wirkungslos.

Sie können MariaDB-Protokolle auch veröffentlichen, indem Sie die folgenden RDS-API-Operationen aufrufen:

- CreateDBInstance
- RestoreDBInstanceFromDBSnapshot
- RestoreDBInstanceFromS3
- RestoreDBInstanceToPointInTime

Führen Sie eine dieser RDS-API-Operationen mit den folgenden Parametern aus:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Je nachdem, welchen AWS CLI Befehl Sie ausführen, sind möglicherweise andere Parameter erforderlich.

Protokollrotation und Aufbewahrung für MariaDB

Wenn Protokollierung aktiviert ist, rotiert Amazon RDS Tabellenprotokolle oder löscht Protokolldateien in regelmäßigen Intervallen. Dies ist eine Vorsichtsmaßnahme, um möglichst zu vermeiden, dass eine umfangreiche Protokolldatei die Datenbanknutzung blockiert oder die Leistung beeinträchtigt.

Die MariaDB-Slow-Query-Protokolldatei, die Fehlerprotokolldatei und die allgemeine Protokolldatei sind auf eine Größe beschränkt, die 2 Prozent des zugeteilten Speicherplatzes für eine DB-Instance nicht überschreiten darf. Um diesen Schwellwert einzuhalten, werden Protokolle automatisch stündlich rotiert und Protokolldateien, die älter als 24 Stunden sind, werden entfernt. Wenn die kombinierte Größe der Protokolle nach dem Löschen von alten Protokolldateien den Schwellwert überschreitet, werden die umfangreichsten Protokolldateien gelöscht, bis die Größe den Schwellwert nicht mehr überschreitet.

Amazon RDS rotiert Protokolldateien für Fehler bei der IAM-Datenbankauthentifizierung, die größer als 10 MB sind. Amazon RDS entfernt Protokolldateien für Fehler bei der IAM-Datenbankauthentifizierung, die älter als fünf Tage oder größer als 100 MB sind.

Verwalten von tabellenbasierten MariaDB-Protokollen

Sie können die allgemeinen und Slow-Query-Protokolle an Tabellen auf der DB-Instance weiterleiten. Erstellen Sie dazu eine DB-Parametergruppe und legen Sie den `log_output`-Serverparameter auf `TABLE` fest. Allgemeine Abfragen werden anschließend in der `mysql.general_log`-Tabelle und Slow-Queries in der`mysql.slow_log`-Tabelle protokolliert. Sie können die Tabellen abfragen, um auf Protokollinformationen zuzugreifen. Durch Aktivieren dieser Protokollierung wird die Datenmenge erhöht, die in die Datenbank geschrieben wird, was die Performance beeinträchtigen kann.

Das allgemeine Protokoll und das Slow-Query-Protokoll sind standardmäßig deaktiviert. Um die Protokollierung in Tabellen zu aktivieren, müssen Sie außerdem die folgenden Serverparameter auf einstellen: 1

- `general_log`
- `slow_query_log` oder `log_slow_query`

Protokolltabellen wachsen stetig, bis die entsprechenden Protokollierungsaktivitäten ausgeschaltet werden, indem der entsprechende Parameter auf gesetzt wir `0`. Mit der Zeit sammelt sich häufig eine große Datenmenge an und belegt einen beträchtlichen Anteil Ihres zugeteilten Speicherplatzes. Amazon erlaubt Ihnen RDS nicht, die Protokolltabellen zu kürzen, aber Sie können ihren Inhalt verschieben. Beim Rotieren einer Tabelle wird deren Inhalt in einer Sicherungstabelle gespeichert, und anschließend wird eine neue leere Protokolldatei angelegt. Sie können Protokolltabellen mithilfe der folgenden Befehlszeilenprozeduren manuell rotieren, wobei die Eingabeaufforderung mit bezeichnet is `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Um alte Daten komplett zu entfernen und den Speicherplatz zurückzugewinnen, rufen Sie die entsprechende Prozedur zweimal nacheinander auf.

Konfiguration der MariaDB-Binärprotokollierung

Das Binärprotokoll besteht aus einer Reihe von Protokolldateien, die Informationen über Datenänderungen enthalten, die an einer MariaDB-Serverinstanz vorgenommen wurden. Das Binärprotokoll enthält Informationen wie die folgenden:

- Ereignisse, die Datenbankänderungen wie Tabellenerstellungen oder Zeilenänderungen beschreiben
- Informationen über die Dauer jeder Anweisung, durch die Daten aktualisiert wurden
- Ereignisse für Anweisungen, durch die Daten aktualisieren werden hätten können, aber nicht wurden

Das binäre Protokoll zeichnet Anweisungen auf, die während der Replikation gesendet werden. Es ist auch für einige Wiederherstellungsvorgänge erforderlich. Weitere Informationen finden Sie unter Binary Log in der MariaDB-Dokumentation.

Die automatische Backup-Funktion bestimmt, ob die Binärprotokollierung für MariaDB ein- oder ausgeschaltet ist. Ihnen stehen folgende Optionen zur Verfügung:

Aktivieren der Binärprotokollierung

Legen Sie den Aufbewahrungszeitraum für Backups auf einen positiven Wert größer 0 fest.

Deaktivieren der Binärprotokollierung

Legen Sie den Aufbewahrungszeitraum für Backups auf 0 fest.

Weitere Informationen finden Sie unter Aktivieren von automatisierten Backups.

MariaDB auf Amazon RDS unterstützt die zeilenbasierten, anweisungsbasierten und gemischten binären Protokollierungsformate. Das standardmäßige binäre Protokollierungsformat ist mixed. Weitere Details zu anderen binären Protokollierungsformaten in MariaDB finden Sie unter Binary Log Formats in der MariaDB-Dokumentation.

Wenn Sie die Replikation verwenden möchten, ist das Binärprotokollformat wichtig. Dies liegt daran, dass es den Datensatz der Datenänderungen bestimmt, der in der Quelle aufgezeichnet und an die Replikationsziele gesendet wird. Informationen zu den Vor- und Nachteilen verschiedener binärer Protokollierungsformate für die Replikation finden Sie unter Vor- und Nachteile der anweisungsbasierten und zeilenbasierten Replikation in der Dokumentation Meine Dokumentation. SQL

> **Important**
>
> Wenn das binäre Protokollierungsformat auf "row-based" eingestellt ist, kann das zu sehr umfangreichen binären Protokolldateien führen. Große binäre Protokolldateien verringern den Speicherplatz, der einer DB-Instance zur Verfügung steht. Sie können auch die Zeitspanne erhöhen, um einen Wiederherstellungsvorgang einer DB-Instance auszuführen.
> Die Replikation vom Typ „statement-based“ kann zu Inkonsistenzen zwischen der Quell-DB-Instance und einem Lesereplikat führen. Weitere Informationen finden Sie unter Unsafe Statements for Statement-based Replication in der MariaDB-Dokumentation.
> Durch die Aktivierung der binären Protokollierung wird die Anzahl der Festplatten-I/O-Schreibvorgänge in der DB-Instance erhöht. Sie können die IOPS Nutzung mit der `WriteIOPS` CloudWatch Metrik überwachen.

Stellen Sie das binäre Protokollierungsformat für MariaDB wie folgt ein:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

3. Wählen Sie in der Liste die zu ändernde Parametergruppe, die von der DB-Instance verwendet wird.

   Eine Standard-Parametergruppe kann nicht modifiziert werden. Erstellen Sie eine neue Parametergruppe und ordnen Sie diese der DB-Instance zu, wenn die DB-Instance eine Standardparametergruppe verwendet.

   Weitere Informationen zu DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.
4. Wählen Sie für Parameter group actions (Parametergruppenaktionen) die Option Bearbeiten.
5. Legen Sie den Parameter `binlog_format` auf das binäre Protokollierungsformat Ihrer Wahl fest (ROW, STATEMENT oder MIXED).

   Sie können die Binärprotokollierung deaktivieren, indem Sie den Aufbewahrungszeitraum für Backups einer DB-Instance auf Null festlegen. Dadurch werden jedoch tägliche automatische Backups deaktiviert. Durch das Deaktivieren automatisierter Backups wird die `log_bin` Sitzungsvariable ausgeschaltet oder deaktiviert. Dadurch wird die binäre Protokollierung auf der RDS for MariaDB-DB-Instance deaktiviert, was wiederum die `binlog_format` Sitzungsvariable auf den Standardwert von `ROW` in der Datenbank zurücksetzt. Wir empfehlen, Backups nicht zu deaktivieren. Weitere Informationen zur Einstellung Aufbewahrungszeitraums für Backups finden Sie unter Einstellungen für DB-Instances.
6. Wählen Sie Änderungen speichern, um die Aktualisierungen in dieser DB-Parametergruppe zu speichern.

Da der `binlog_format` Parameter RDS für MariaDB dynamisch ist, müssen Sie die DB-Instance nicht neu starten, damit die Änderungen wirksam werden.

> **Important**
>
> Das Ändern einer DB-Parametergruppe wirkt sich auf alle DB-Instances aus, die diese Parametergruppe verwenden. Wenn Sie unterschiedliche binäre Logging-Formate für verschiedene MariaDB-DB-Instances in einer AWS Region angeben möchten, müssen die DB-Instances unterschiedliche DB-Parametergruppen verwenden. Diese Parametergruppen identifizieren unterschiedliche Protokollierungsformate. Weisen Sie den einzelnen DB-Instances die entsprechende DB-Parametergruppe zu.

Zugriff auf binäre MariaDB-Protokolle

Mithilfe des Dienstprogramms mysqlbinlog können Sie Binärprotokolle in Textformat aus DB-Instances in MariaDB herunterladen. Das binäre Protokoll wird auf Ihren lokalen Computer heruntergeladen. Weitere Informationen über die Verwendung des Dienstprogramms mysqlbinlog finden Sie unter Using mysqlbinlog in der MariaDB-Dokumentation.

Verwenden Sie die folgenden Optionen, um das Hilfsprogramm mysqlbinlog für eine RDS Amazon-Instance auszuführen:

- Legen Sie die Option `--read-from-remote-server` fest.
- `--host`: Geben Sie den DNS Namen vom Endpunkt der Instance an.
- `--port`Geben Sie den von der Instance verwendeten Port an.
- `--user`: Geben Sie einen MariaDB-Benutzer an, dem die Slave-Berechtigung für Replikation erteilt wurde.
- `--password`: Geben Sie das Passwort für den Benutzer an oder lassen Sie einen Passwortwert aus, damit das Hilfsprogramm zur Eingabe eines Passworts auffordert.
- `--result-file`: Geben Sie die lokale Datei an, die den Output empfängt.
- Geben Sie die Namen einer oder mehrerer Binärprotokolldateien an. Verwenden Sie den SQL Befehl, um eine Liste der verfügbaren Protokolle abzurufen SHOW BINARYLOGS.

Weitere Informationen über Optionen für mysqlbinlog finden Sie unter mysqlbinlog Options in der MariaDB-Dokumentation.

Im Folgenden wird ein Beispiel gezeigt:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password <password> \
    --result-file=/tmp/binlog.txt
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password <password> ^
    --result-file=/tmp/binlog.txt
```

Amazon RDS löscht normalerweise so schnell wie möglich ein Binärprotokoll. Allerdings muss das binäre Protokoll immer noch auf der Instance verfügbar sein, sodass mysqlbinlog darauf zugreifen kann. Verwenden Sie die `mysql.rds_set_configuration` gespeicherte Prozedur, RDS um die Anzahl der Stunden für die Aufbewahrung von Binärprotokollen anzugeben. Geben Sie einen Zeitraum an, in dem Sie genügend Zeit haben, um die Protokolle herunterzuladen. Nachdem Sie den Aufbewahrungszeitraum festgelegt haben, überwachen Sie die Speichernutzung für die DB-Instance, um sicherzustellen, dass die aufbewahrten binären Protokolle nicht zu viel Speicherplatz beanspruchen.

Das folgende Beispiel setzt den Aufbewahrungszeitraum auf 1 Tag.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Verwenden Sie die gespeicherte Prozedur `mysql.rds_show_configuration`, um die aktuelle Einstellung anzeigen zu lassen.

```
call mysql.rds_show_configuration;
```

MariaDB-Binärlog-Annotation aktivieren

In einer MariaDB-DB-Instance können Sie das `Annotate_rows` Ereignis verwenden, um ein Zeilenereignis mit einer Kopie der SQL Abfrage zu annotieren, die das Zeilenereignis verursacht hat. Dieser Ansatz bietet ähnliche Funktionen wie die Aktivierung des `binlog_rows_query_log_events` Parameters auf einer RDS for My SQL DB-Instance.

Sie können Binärprotokoll-Anmerkungen global aktivieren, indem Sie eine benutzerdefinierte Parametergruppe erstellen und den Parameter `binlog_annotate_row_events` auf **1** setzen. Sie können Anmerkungen auch auf Sitzungsebene aktivieren, indem Sie aufrufe `SET SESSION binlog_annotate_row_events = 1`. Verwenden Sie `replicate_annotate_row_events`, um Binärprotokoll-Anmerkungen auf der Replika-Instance zu replizieren, falls binäre Protokollierung

darauf aktiviert ist. Für die Nutzung dieser Einstellungen sind keine besonderen Berechtigungen erforderlich.

Nachfolgend sehen Sie ein Beispiel für eine zeilenbasierte Transaktion in MariaDB. Die Verwendung von zeilenbasierter Protokollierung wird ausgelöst, indem der Transaktionsisolationslevel auf "read-committed" eingestellt wird.

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
```

Ohne Anmerkungen sehen die Einträge des Binärprotokolls für die Transaktion wie folgt aus:

```
BEGIN
/*!*/;
# at 1163
# at 1209
#150922  7:55:57 server id 1855786460  end_log_pos 1209         Table_map:
 `test`.`square` mapped to number 76
#150922  7:55:57 server id 1855786460  end_log_pos 1247         Write_rows: table id 76
 flags: STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 1247
#150922  7:56:01 server id 1855786460  end_log_pos 1274         Xid = 62
COMMIT/*!*/;
```

Die folgende Anweisung aktiviert Anmerkungen auf Sitzungsebene für diese Transaktion und deaktiviert die Anmerkungen nach dem Übertragen der Transaktion:

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION binlog_annotate_row_events = 1;
BEGIN;
```

```
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
SET SESSION binlog_annotate_row_events = 0;
```

Mit Anmerkungen sehen die Einträge des Binärprotokolls für die Transaktion wie folgt aus:

```
BEGIN
/*!*/;
# at 423
# at 483
# at 529
#150922  8:04:24 server id 1855786460  end_log_pos 483  Annotate_rows:
#Q> INSERT INTO square(x, y) VALUES(5, 5 * 5)
#150922  8:04:24 server id 1855786460  end_log_pos 529  Table_map: `test`.`square`
 mapped to number 76
#150922  8:04:24 server id 1855786460  end_log_pos 567  Write_rows: table id 76 flags:
 STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 567
#150922  8:04:26 server id 1855786460  end_log_pos 594  Xid = 88
COMMIT/*!*/;
```

Datenbankprotokolldateien von Amazon RDS für Microsoft SQL Server

Sie können auf Microsoft SQL Server-Fehlerprotokolle, Agentenprotokolle, Trace-Dateien und Dump-Dateien zugreifen, indem Sie die RDS Amazon-Konsole verwenden, AWS CLI, oder RDSAPI. Weitere Informationen zum Anzeigen und Herunterladen von dateibasierten Datenbankprotokollen finden Sie unter Überwachung von Amazon RDS Amazon.

Aufbewahrungsplan

Protokolldateien werden jeden Tag rotiert und jedes Mal, wenn Ihre DB-Instance neu gestartet wird. Im Folgenden finden Sie den Aufbewahrungszeitplan für SQL Microsoft-Serverprotokolle bei AmazonRDS.

Protokolltyp	Aufbewahrungsplan
Fehlerprotokolle	Es werden maximal 30 Fehlerprotokolle aufbewahrt. Amazon löscht RDS möglicherweise Fehlerprotokolle, die älter als 7 Tage sind.
Agent-Protokolle	Es werden maximal 10 Agent-Protokolle aufbewahrt. Amazon löscht RDS möglicherweise Agentenprotokolle, die älter als 7 Tage sind.
Trace-Dateien	Trace-Dateien werden entsprechend dem Aufbewahrungszeitraum Ihrer DB-Instance für Trace-Dateien aufbewahrt. Der Standardaufbewahrungszeitraum für Trace-Dateien beträgt 7 Tage. Informationen zum Ändern des Aufbewahrungszeitraums für Trace-Dateien für Ihre DB-Instance finden Sie unter Festlegen des Aufbewahrungszeitraums für Trace- und Dump-Dateien.
Dump-Dateien	Dump-Dateien werden entsprechend dem Aufbewahrungszeitraum Ihrer DB-Instance für Dump-Dateien aufbewahrt. Der Standardaufbewahrungszeitraum für Dump-Dateien beträgt 7 Tage. Informationen zum Ändern des Aufbewahrungszeitraums für Dump-Dateien für Ihre DB-Instance finden Sie unter Festlegen des Aufbewahrungszeitraums für Trace- und Dump-Dateien.

Das SQL Serverfehlerprotokoll mithilfe des Verfahrens rds_read_error_log anzeigen

Sie können die RDS gespeicherte Amazon-Prozedur verwenden`rds_read_error_log`, um Fehler- und Agentenprotokolle einzusehen. Weitere Informationen finden Sie unter Anzeigen von Fehler- und Agent-Protokollen.

SQLServerprotokolle in Amazon CloudWatch Logs veröffentlichen

Mit Amazon RDS for SQL Server können Sie Fehler- und Agentenprotokollereignisse direkt in Amazon CloudWatch Logs veröffentlichen. Analysieren Sie die Protokolldaten mit CloudWatch Logs und verwenden Sie sie dann, CloudWatch um Alarme zu erstellen und Metriken anzuzeigen.

Mit CloudWatch Logs können Sie Folgendes tun:

- Speichern von Protokollen in hoch dauerhaften Speichern mit einem von Ihnen festgelegten Aufbewahrungszeitraum.
- Durchsuchen und Filtern von Protokolldaten
- Protokolldateien zwischen Konten freigeben.
- Exportieren von Protokollen zu Amazon S3.
- Streamen Sie Daten an Amazon OpenSearch Service.
- Verarbeiten von Protokolldaten in Echtzeit mit Amazon Kinesis Data Streams. Weitere Informationen finden Sie unter Arbeiten mit Amazon CloudWatch Logs im Amazon Managed Service for Apache Flink for SQL Applications Developer Guide.

Amazon RDS veröffentlicht jedes SQL Server-Datenbankprotokoll als separaten Datenbank-Stream in der Protokollgruppe. Wenn Sie beispielsweise die Agentenprotokolle und Fehlerprotokolle veröffentlichen, werden Fehlerdaten in einem Fehlerprotokoll-Stream in der `/aws/rds/instance/`*`my_instance.node1`*`/error` Protokollgruppe gespeichert, und Agenten-Protokolldaten werden in der `/aws/rds/instance/`*`my_instance.node1`*`/agent` Protokollgruppe gespeichert.

Bei Multi-AZ-DB-Instances RDS veröffentlicht Amazon das Datenbankprotokoll als zwei separate Streams in der Protokollgruppe. Wenn Sie zum Beispiel Fehlerprotokolle veröffentlichen, werden die Fehlerdaten in dem Fehlerprotokollstream `/aws/rds/instance/`*`my_instance.node1`*`/error` bzw. `/aws/rds/instance/`*`my_instance.node2`*`/error` gespeichert. Die Protokollstreams ändern sich während eines Failovers nicht und der Fehlerprotokollstream jedes Knotens kann Fehlerprotokolle von der primären oder sekundären Instance enthalten. Bei Multi-AZ wird automatisch

ein Protokollstream `/aws/rds/instance/my_instance/rds-events` zum Speichern von Ereignisdaten wie DB-Instance-Failovers erstellt.

> **Note**
>
> Das Veröffentlichen von SQL CloudWatch Serverprotokollen in Logs ist standardmäßig nicht aktiviert. Die Veröffentlichung von Trace- und Dump-Dateien wird nicht unterstützt. Das Veröffentlichen von SQL CloudWatch Serverprotokollen in Logs wird in allen Regionen unterstützt, mit Ausnahme von Asien-Pazifik (Hongkong).

Konsole

Um SQL Server-DB-Protokolle in Logs zu CloudWatch veröffentlichen, klicken Sie auf AWS Management Console

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus.
4. Wählen Sie im Abschnitt Protokollexporte die Protokolle aus, die Sie in CloudWatch Logs veröffentlichen möchten.

   Sie können Agentenprotokoll, Fehlerprotokoll oder beides wählen.
5. Wählen Sie Weiter und dann auf der zusammenfassenden Seite Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Um SQL Serverprotokolle zu veröffentlichen, können Sie den `modify-db-instance`Befehl mit den folgenden Parametern verwenden:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Eine Änderung der Option `--cloudwatch-logs-export-configuration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund sind die Optionen `--apply-immediately` und `--no-apply-immediately` wirkungslos.

Sie können SQL Serverprotokolle auch mit den folgenden Befehlen veröffentlichen:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

Im folgenden Beispiel wird eine SQL Server-DB-Instance mit aktivierter CloudWatch Protokollveröffentlichung erstellt. Der `--enable-cloudwatch-logs-exports` Wert ist ein JSON Array von Zeichenfolgen, die `erroragent`, oder beides enthalten können.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["error","agent"]' \
    --db-instance-class db.m4.large \
    --engine sqlserver-se
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports "[\"error\",\"agent\"]" ^
    --db-instance-class db.m4.large ^
    --engine sqlserver-se
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

Example

Im folgenden Beispiel wird eine bestehende SQL Server-DB-Instance so geändert, dass Protokolldateien in Logs veröffentlicht werden. CloudWatch Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Objekt. Der Schlüssel für dieses Objekt ist `EnableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit `error`, `agent` oder beiden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["error","agent"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"EnableLogTypes\":[\"error\",\"agent\"]}"
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

Example

Im folgenden Beispiel wird eine bestehende SQL Server-DB-Instance dahingehend geändert, dass Agent-Protokolldateien nicht in Logs veröffentlicht werden. CloudWatch Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Objekt. Der Schlüssel für dieses Objekt ist `DisableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit `error`, `agent` oder beiden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["agent"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"agent\"]}"
```

> **Note**
>
> Wenn Sie die Windows-Befehlszeile verwenden, müssen Sie doppelte Anführungszeichen („) im JSON Code umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

MySQL-Datenbank-Protokolldateien

Sie können die MySQL-Protokolle direkt über die Amazon RDS-Konsole, die Amazon RDS-API AWS CLI, oder überwachen AWS SDKs. Sie können auf MySQL-Protokolle auch direkt zugreifen, indem Sie die Protokolle in eine Datenbank-Tabelle in der Hauptdatenbank weiterleiten und diese Tabelle abfragen. Mit dem Dienstprogramm "mysqlbinlog" können Sie ein binäres Protokoll herunterladen.

Weitere Informationen zum Anzeigen und Herunterladen von dateibasierten Datenbankprotokollen finden Sie unter Überwachung von Amazon RDS Amazon.

Themen

- Überblick über RDS-for-MySQL-Datenbankprotokolle
- Veröffentlichen von MySQL-Protokollen in Amazon CloudWatch Logs
- My SQL Log-Ausgabe an Tabellen senden
- Konfiguration von RDS für MySQL-Binärprotokollierung für Single-AZ-Datenbanken
- Konfiguration von My SQL Binary Logging für Multi-AZ-DB-Cluster
- Zugriff auf MySQL-Binärprotokolle

Überblick über RDS-for-MySQL-Datenbankprotokolle

Sie können die folgenden Arten von RDS-for-MySQL-Protokolldateien überwachen:

- Fehler-log
- Slow-Query-Protokoll
- Allgemeines Protokoll
- Prüfungsprotokoll
- Instanzprotokoll
- Fehlerprotokoll für die IAM-Datenbankauthentifizierung

Das RDS-for-MySQL-Fehlerprotokoll wird standardmäßig generiert. Sie können die langsamen Abfrage- und allgemeinen Protokolle generieren, indem Sie Parameter in Ihrer DB-Parametergruppe festlegen.

Themen

- RDS-for-MySQL-Fehlerprotokolle

- RDS-für-MySQL-Protokolle für langsame Abfragen und allgemeine Protokolle
- MySQL Audit-Protokoll
- Protokollrotation und -aufbewahrung für RDS for MySQL
- Größenbeschränkungen für Redo-Protokolle

RDS-for-MySQL-Fehlerprotokolle

RDS for MySQL schreibt Fehler in die `mysql-error.log`-Datei. An den Namen jeder Protokolldatei wird die Stunde ihrer Erstellung (in UTC) angefügt. Die Protokolldateien verfügen auch über einen Zeitstempel, anhand dessen Sie feststellen können, wann die Protokolleinträge geschrieben wurden.

RDS f0r MySQL schreibt das Fehlerprotokoll nur beim Startup, Herunterfahren und beim Auftreten von Fehlern. Eine DB-Instance kann Stunden oder Tage lang laufen, ohne dass neue Einträge in das Fehlerprotokoll geschrieben werden. Wenn Sie keine neuen Einträge sehen, sind im Server keine Fehler aufgetreten, die zu einem Eintrag in das Protokoll führen würden.

Konstruktionsbedingt werden die Fehlerprotokolle gefiltert, sodass nur unerwartete Ereignisse wie Fehler angezeigt werden. Die Fehlerprotokolle enthalten jedoch auch einige zusätzliche Datenbankinformationen, z. B. den Abfragefortschritt, die nicht angezeigt werden. Daher kann die Größe der Fehlerprotokolle auch ohne tatsächliche Fehler aufgrund laufender Datenbankaktivitäten zunehmen. Und obwohl Ihnen für die Fehlerprotokolle möglicherweise eine bestimmte Größe in Byte oder Kilobyte angezeigt wird AWS Management Console, haben sie beim Herunterladen möglicherweise 0 Byte.

RDS for MySQL schreibt `mysql-error.log` alle 5 Minuten auf die Festplatte. Es fügt den Inhalt des Protokolls `mysql-error-running.log` an.

RDS for MySQL rotiert die Datei `mysql-error-running.log` stündlich. Es behält die in den letzten zwei Wochen erzeugten Protokolle bei.

> **Note**
>
> Der Aufbewahrungszeitraum für das Protokoll ist zwischen Amazon RDS und unterschiedlich Aurora.

RDS-für-MySQL-Protokolle für langsame Abfragen und allgemeine Protokolle

Das Slow-Query-Protokoll von RDS für MySQL und das allgemeine Protokoll können in eine Datei oder in eine Datenbanktabelle geschrieben werden. Legen Sie dazu die Parameter in Ihrer DB-Parametergruppe fest. Weitere Informationen zum Erstellen und Ändern einer DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS. Sie müssen diese Parameter festlegen, bevor Sie das Protokoll für langsame Abfragen oder das allgemeine Protokoll in der Amazon RDS-Konsole oder mithilfe der Amazon RDS-API, Amazon RDS-CLI oder anzeigen können AWS SDKs.

Sie können RDS-for-MySQL-Protokolle mithilfe der Parameter in dieser Liste kontrollieren:

- `slow_query_log` Um das Slow-Query-Protokoll zu erstellen, auf 1 setzen. Der Standardwert ist 0.
- `general_log` Um das allgemeine Protokoll zu erstellen, auf 1 setzen. Der Standardwert ist 0.
- `long_query_time`: Damit vermieden wird, dass schnell ausgeführte Abfragen im Slow-Query-Protokoll aufgenommen werden, legen Sie die kürzeste Laufzeit für eine zu protokollierende Abfrage in Sekunden fest. Der Standardwert liegt bei 10 Sekunden, der Mindestwert bei 0. Wenn log_output = FILE, können Sie einen Gleitkommawert angeben, der die Mikrosekundenauflösung festlegt. Wenn log_output = TABLE, können Sie einen Ganzzahlwert angeben, der die Sekundenauflösung festlegt. Nur Abfragen, deren Laufzeit den `long_query_time`-Wert übersteigt, werden im Protokoll aufgenommen. Wenn Sie beispielsweise `long_query_time` auf 0,1 setzen, verhindert dies Einträge von allen Abfragen, die weniger als 100 Millisekunden lang ausgeführt werden.
- `log_queries_not_using_indexes`: Um alle Abfragen, die keinen Index für das Slow-Query-Protokoll verwenden im Protokoll aufzunehmen, auf 1 setzen. Abfragen, die keinen Index verwenden, werden protokolliert, auch wenn ihre Laufzeit niedriger als der Wert des Parameters `long_query_time` ist. Der Standardwert ist 0.
- `log_output` *`option`*: Sie können eine der folgenden Optionen für den `log_output`-Parameter festlegen.
  - TABLE (Standard)– schreibt allgemeine Abfragen in die `mysql.general_log`-Tabelle und langsame Abfragen in die `mysql.slow_log`-Tabelle.
  - FILE– schreibt Protokolle allgemeiner und langsamer Abfragen in das Dateisystem.
  - NONE– Die Protokollierung ist deaktiviert.

Damit langsame Abfragedaten in Amazon CloudWatch Logs angezeigt werden, müssen die folgenden Bedingungen erfüllt sein:

- CloudWatch Protokolle müssen so konfiguriert sein, dass sie langsame Abfrageprotokolle enthalten.
- `slow_query_log`muss aktiviert sein.
- muss `log_output` auf `FILE` festgelegt sein.
- Die Abfrage muss länger dauern als die konfigurierte Zeit`long_query_time`.

Weitere Informationen zu den Slow-Query- und allgemeinen Protokollen finden Sie in den folgenden Themen in der MySQL-Dokumentation:

- Das Slow-Query-Protokoll
- Das allgemeine Abfrageprotokoll

MySQL Audit-Protokoll

Für den Zugriff auf das Audit-Protokoll muss die DB-Instance eine benutzerdefinierte Optionsgruppe mit der Option `MARIADB_AUDIT_PLUGIN` verwenden. Weitere Informationen finden Sie unter MariaDB Audit Plugin-Unterstützung für My SQL.

Protokollrotation und -aufbewahrung für RDS for MySQL

Wenn Protokollierung aktiviert ist, rotiert Amazon RDS Tabellenprotokolle oder löscht Protokolldateien in regelmäßigen Intervallen. Dies ist eine Vorsichtsmaßnahme, um möglichst zu vermeiden, dass eine umfangreiche Protokolldatei die Datenbanknutzung blockiert oder die Leistung beeinträchtigt. RDS for MySQL behandelt Rotation und Löschen wie folgt:

- Die MySQL-Slow-Query-Protokolldatei, Fehlerprotokolldatei und allgemeine Protokolldatei sind auf eine Größe beschränkt, die 2 % des zugewiesenen Speicherplatzes für eine DB-Instance nicht überschreiten darf. Um diesen Schwellenwert einzuhalten, werden die Protokolle automatisch stündlich gedreht. MySQL entfernt Protokolldateien, die älter als zwei Wochen sind. Wenn die kombinierte Größe der Protokolle nach dem Löschen von alten Protokolldateien den Schwellwert überschreitet, werden die ältesten Protokolldateien gelöscht, bis die Größe den Schwellwert nicht mehr überschreitet.
- Wenn die `FILE`-Protokollierung aktiviert ist, werden Protokolldateien stündlich untersucht und Protokolldateien, die älter als zwei Wochen sind, werden gelöscht. In einigen Fällen kann die Größe der verbleibenden kombinierten Protokolldatei nach dem Löschen die Schwelle von 2 % des zugeteilten Speicherplatzes für eine DB-Instance überschreiten. In diesen Fällen werden die ältesten Protokolldateien gelöscht, bis die Größe den Schwellenwert nicht mehr überschreitet.

- Wenn die TABLE-Protokollierung aktiviert ist, werden in einigen Fällen Protokolltabellen alle 24 Stunden überschrieben. Diese Rotation erfolgt, wenn der von den Tabellenprotokollen verwendete Speicherplatz mehr als 20 Prozent des zugeteilten Speicherplatzes ausmacht. Sie tritt auch auf, wenn die Größe aller kombinierten Protokolle mehr als 10 GB beträgt. Wenn der für eine DB-Instance verwendete Speicherplatz 90 Prozent des Speicherplatzes überschreitet, der der DB-Instance zugeteilt ist, werden die Schwellen für die Protokollrotation reduziert. Protokolltabellen werden dann rotiert, wenn der von den Tabellenprotokollen verwendete Speicherplatz mehr als 10 Prozent des zugeteilten Speicherplatzes ausmacht. Sie werden auch rotiert, wenn die Größe aller kombinierten Protokolle mehr als 5 GB beträgt. Sie können die `low storage` Ereigniskategorie abonnieren, um benachrichtigt zu werden, wenn die Protokolltabellen rotiert werden, um Speicherplatz freizugeben. Weitere Informationen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.

  Wenn Protokolltabellen rotiert werden, wird die aktuelle Protokolltabelle zuerst in eine Backup-Protokolltabelle kopiert. Dann werden die Einträge aus der aktuellen Protokolltabelle entfernt. Sofern bereits eine Sicherungsprotokolltabelle vorhanden ist, wird diese gelöscht, bevor die aktuelle Protokolltabelle ins Backup kopiert wird. Sie können die Sicherungsprotokolltabelle abfragen, wenn dies nötig ist. Die Backup-Protokolltabelle für die `mysql.general_log`-Tabelle ist als `mysql.general_log_backup` benannt. Die Backup-Protokolltabelle für die `mysql.slow_log`-Tabelle ist als `mysql.slow_log_backup` benannt.

  Sie können die `mysql.general_log`-Tabelle rotieren, wenn Sie die Prozedur `mysql.rds_rotate_general_log` aufrufen. Sie können die `mysql.slow_log`-Tabelle rotieren, wenn Sie die Prozedur `mysql.rds_rotate_slow_log` aufrufen.

  Tabellenprotokolle werden während des Upgrades einer Datenbankversion rotiert.

Um mit den Protokollen von der Amazon RDS-Konsole, der Amazon RDS-API, der Amazon RDS-CLI oder zu arbeiten AWS SDKs, setzen Sie den `log_output` Parameter auf FILE. So wie das MySQL-Fehlerprotokoll, werden diese Protokolldateien stündlich rotiert. Die Protokolldateien, die in den letzten zwei Wochen generiert wurden, werden aufbewahrt. Beachten Sie, dass der Aufbewahrungszeitraum bei Amazon RDS und Aurora jeweils unterschiedlich ist.

Größenbeschränkungen für Redo-Protokolle

Für RDS for MySQL Version 8.0.32 und niedriger ist der Standardwert dieses Parameters 256 MB. Dieser Betrag wird abgeleitet, indem der Standardwert des `innodb_log_file_size` Parameters (128 MB) mit dem Standardwert des Parameters (2) multipliziert `innodb_log_files_in_group`

wird. Weitere Informationen finden Sie unter Bewährte Methoden für die Konfiguration von Parametern für Amazon RDS for MySQL, Teil 1: Leistungsparameter.

Für RDS für MySQL Version 8.0.33 und höhere Nebenversionen verwendet Amazon RDS den `innodb_redo_log_capacity` Parameter anstelle des `innodb_log_file_size` Parameters. Der Amazon RDS-Standardwert des `innodb_redo_log_capacity` Parameters ist 2 GB. Weitere Informationen finden Sie unter Änderungen in MySQL 8.0.30 in der MySQL-Dokumentation.

Ab MySQL 8.4 aktiviert Amazon RDS den `innodb_dedicated_server` Parameter standardmäßig. Mit dem `innodb_dedicated_server` Parameter berechnet die Datenbank-Engine die `innodb_redo_log_capacity` Parameter `innodb_buffer_pool_size` und. Weitere Informationen finden Sie unter Konfiguration der Pufferpoolgröße und der Redo-Log-Kapazität in My 8.4 SQL.

Veröffentlichen von MySQL-Protokollen in Amazon CloudWatch Logs

Sie können Ihre MySQL-DB-Instance so konfigurieren, dass Protokolldaten in einer Protokollgruppe in Amazon CloudWatch Logs veröffentlicht werden. Mit CloudWatch Logs können Sie eine Echtzeitanalyse der Protokolldaten durchführen und diese CloudWatch zum Erstellen von Alarmen und zum Anzeigen von Metriken verwenden. Sie können CloudWatch Logs verwenden, um Ihre Protokolldatensätze in einem äußerst langlebigen Speicher zu speichern.

Amazon RDS veröffentlicht jedes MySQL-Datenbankprotokoll als separaten Datenbank-Stream in der Protokollgruppe. Wenn Sie beispielsweise die Exportfunktion so konfigurieren, dass das Slow-Query-Protokoll berücksichtigt wird, werden Slow-Query-Daten in einem Slow-Query-Protokollstream in der Protokollgruppe `/aws/rds/instance/`*`my_instance`*`/slowquery` gespeichert.

Das Fehlerprotokoll ist standardmäßig aktiviert. Die folgende Tabelle fasst die Anforderungen für die anderen MySQL-Protokolle zusammen.

Protokoll	Anforderung
Prüfungsprotokoll	Die DB-Instance muss eine benutzerdefinierte Optionsgruppe mit der Option `MARIADB_AUDIT_PLUGIN` verwenden.
Allgemeines Protokoll	Die DB-Instance muss eine benutzerdefinierte Parametergruppe mit der Parametereinstellu

Protokoll	Anforderung
	ng `general_log = 1` verwenden, um das allgemeine Protokoll zu aktivieren.
Slow-Query-Protokoll	Die DB-Instance muss eine benutzerdefinierte Parametergruppe mit der Parametereinstellung `slow_query_log = 1` verwenden, um das Slow-Query-Protokoll zu aktivieren.
Fehlerprotokoll für die IAM-Datenbankauthentifizierung	Sie müssen den Protokolltyp `iam-db-auth-error` für eine DB-Instance aktivieren, indem Sie eine DB-Instance erstellen oder ändern.
Protokollausgabe	Die DB-Instance muss eine benutzerdefinierte Parametergruppe mit der Parametereinstellung verwenden`log_output = FILE` , um Protokolle in das Dateisystem zu schreiben und sie in CloudWatch Logs zu veröffentlichen.

Konsole

Um MySQL-Logs mit der Konsole in CloudWatch Logs zu veröffentlichen

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus.
4. Wählen Sie im Abschnitt Protokollexporte die Protokolle aus, die Sie in CloudWatch Logs veröffentlichen möchten.
5. Wählen Sie Weiter und dann auf der zusammenfassenden Seite Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Sie können MySQL-Logs mit dem veröffentlichen AWS CLI. Sie können den Befehl `modify-db-instance` mit den folgenden Parametern aufrufen:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Eine Änderung der Option `--cloudwatch-logs-export-configuration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund sind die Optionen `--apply-immediately` und `--no-apply-immediately` wirkungslos.

Sie können MySQL-Logs auch veröffentlichen, indem Sie die folgenden AWS CLI Befehle aufrufen:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Führen Sie einen dieser AWS CLI Befehle mit den folgenden Optionen aus:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Je nachdem, welchen AWS CLI Befehl Sie ausführen, sind möglicherweise andere Optionen erforderlich.

Example

Das folgende Beispiel ändert eine bestehende MySQL-DB-Instance, um Protokolldateien in CloudWatch Logs zu veröffentlichen. Der `--cloudwatch-logs-export-configuration`-Wert ist ein JSON-Objekt. Der Schlüssel für dieses Objekt ist `EnableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit einer beliebigen Kombination aus `audit`, `error`, `general` und `slowquery`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

Das folgende Beispiel erstellt eine MySQL-DB-Instance und veröffentlicht Protokolldateien in CloudWatch Logs. Der Wert `--enable-cloudwatch-logs-exports` ist ein JSON-Array mit Zeichenfolgen. Die Zeichenfolgen können eine beliebige Kombination von `audit`, `error`, `general` und `slowquery` sein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine MySQL
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine MySQL
```

RDS-API

Sie können MySQL-Protokolle über die RDS-API veröffentlichen. Die Aktion `ModifyDBInstance` kann dazu mit den folgenden Parametern aufgerufen werden:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Eine Änderung des Parameters `CloudwatchLogsExportConfiguration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund ist der Parameter `ApplyImmediately` wirkungslos.

Sie können MySQL-Protokolle auch veröffentlichen, indem Sie in der RDS-API die folgenden Operationen aufrufen:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Führen Sie eine dieser RDS-API-Operationen mit den folgenden Parametern aus:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Je nachdem, welchen AWS CLI Befehl Sie ausführen, sind möglicherweise andere Parameter erforderlich.

My SQL Log-Ausgabe an Tabellen senden

Sie können die allgemeinen und Slow-Query-Protokolle an Tabellen in der DB-Instance weiterleiten, indem Sie eine DB-Parametergruppe erstellen und den `log_output`-Serverparameter auf `TABLE` setzen. Allgemeine Abfragen werden anschließend in der `mysql.general_log`-Tabelle und Slow-Queries in der`mysql.slow_log`-Tabelle protokolliert. Sie können die Tabellen abfragen, um auf Protokollinformationen zuzugreifen. Durch Aktivieren dieser Protokollierung wird die Datenmenge erhöht, die in die Datenbank geschrieben wird, was die Performance beeinträchtigen kann.

Das allgemeine Protokoll und das Slow-Query-Protokoll sind standardmäßig deaktiviert. Um die Protokollierung in Tabellen zu aktivieren, müssen Sie auch die Serverparameter `general_log` und `slow_query_log` auf `1` setzen.

Protokolltabellen wachsen stetig, bis die entsprechenden Protokollierungsaktivitäten ausgeschaltet werden, indem der entsprechende Parameter auf gesetzt wir `0`. Mit der Zeit sammelt sich häufig eine große Datenmenge an und belegt einen beträchtlichen Anteil Ihres zugeteilten Speicherplatzes. Amazon RDS erlaubt es Ihnen nicht, die Protokolltabellen zu kürzen, aber Sie können ihren Inhalt verschieben. Beim Rotieren einer Tabelle wird deren Inhalt in einer Sicherungstabelle gespeichert, und anschließend wird eine neue leere Protokolldatei angelegt. Sie können Protokolltabellen mithilfe der folgenden Befehlszeilenprozeduren manuell rotieren, wobei die Eingabeaufforderung mit bezeichnet is `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Um alte Daten komplett zu entfernen und den Speicherplatz zurückzugewinnen, rufen Sie die entsprechende Prozedur zweimal nacheinander auf.

Konfiguration von RDS für MySQL-Binärprotokollierung für Single-AZ-Datenbanken

Das Binärprotokoll ist eine Reihe von Protokolldateien, die Informationen zu Datenänderungen enthalten, die an einer MySQL-Server-Instance vorgenommen wurden. Das Binärprotokoll enthält Informationen wie die folgenden:

- Ereignisse, die Datenbankänderungen wie Tabellenerstellungen oder Zeilenänderungen beschreiben
- Informationen über die Dauer jeder Anweisung, durch die Daten aktualisiert wurden
- Ereignisse für Anweisungen, durch die Daten aktualisieren werden hätten können, aber nicht wurden

Das binäre Protokoll zeichnet Anweisungen auf, die während der Replikation gesendet werden. Es ist auch für einige Wiederherstellungsvorgänge erforderlich. Weitere Informationen finden Sie unter The Binary Log in der MySQL-Dokumentation.

Die Funktion für automatisierte Backups bestimmt, ob die binäre Protokollierung für MySQL ein- oder ausgeschaltet wird. Ihnen stehen folgende Optionen zur Verfügung:

Aktivieren der Binärprotokollierung

Legen Sie den Aufbewahrungszeitraum für Backups auf einen positiven Wert größer 0 fest.

Deaktivieren der Binärprotokollierung

Legen Sie den Aufbewahrungszeitraum für Backups auf 0 fest.

Weitere Informationen finden Sie unter Aktivieren von automatisierten Backups.

MySQL in Amazon RDS unterstützt die binären Protokollformate row-based, statement-based und mixed. Wir empfehlen gemischt, sofern Sie kein spezifisches Format des Binärprotokolls benötigen. Einzelheiten zu den verschiedenen binären Protokollformaten von MySQL finden Sie unter Binäre Protokollierungsformate in der MySQL-Dokumentation.

Zur Verwendung der Replikation ist das binäre Protokollierungsformat wichtig, da es den Datensatz der Datenänderungen bestimmt, der in der Quelle aufgezeichnet und an die Replikationsziele gesendet wird. Weitere Informationen über Vor- und Nachteile verschiedener binärer Protokollierungsformate finden Sie unter Vorteile und Nachteile einer auf Anweisungen und einer auf Zeilen basierenden Replikation in der MySQL-Dokumentation.

Important

Mit MySQL 8.0.34 hat MySQL den Parameter als veraltet eingestuft. `binlog_format` In späteren MySQL-Versionen plant MySQL, den Parameter zu entfernen und nur die zeilenbasierte Replikation zu unterstützen. Daher empfehlen wir, für neue MySQL-Replikations-Setups die zeilenbasierte Protokollierung zu verwenden. Weitere Informationen finden Sie unter binlog_format in der MySQL-Dokumentation.
Die MySQL-Versionen 8.0 und 8.4 akzeptieren den Parameter`binlog_format`. Bei Verwendung dieses Parameters gibt MySQL eine Verfallswarnung aus. In einer future Hauptversion wird MySQL den Parameter entfernen`binlog_format`.
Die anweisungsbasierte Replikation kann zu Inkonsistenzen zwischen dem Quell-DB-Instance- und einem Lese-Replikat führen. Weitere Informationen finden Sie

unter Determination of Safe and Unsafe Statements in Binary Logging in der MySQL-Dokumentation.
Durch die Aktivierung der binären Protokollierung wird die Anzahl der Write-Disk-I/O-Operationen für die DB-Instance erhöht. Sie können die IOPS-Nutzung mit der `WriteIOPS` CloudWatch Metrik überwachen.

Stellen Sie das MySQL-binäres-Protokollierungsformat wie folgt ein:

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie die aus, die dem zugeordnet ist und den Sie ändern möchten.

   Eine Standard-Parametergruppe kann nicht modifiziert werden. Erstellen Sie eine neue Parametergruppe und ordnen Sie diese dem DB-Instance- zu, wenn der DB-Instance- eine Standardparametergruppe verwendet.

   Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.
4. Wählen Sie unter Aktionen die Option Bearbeiten aus.
5. Legen Sie den Parameter `binlog_format` auf das binäre Protokollierungsformat Ihrer Wahl fest (`ROW`, `STATEMENT` oder `MIXED`).

   Sie können die Binärprotokollierung deaktivieren, indem Sie den Aufbewahrungszeitraum für Backups einer DB-Instance auf Null festlegen. Dadurch werden jedoch tägliche automatische Backups deaktiviert. Durch die Deaktivierung automatisierter Backups wird die Sitzungsvariable ausgeschaltet oder deaktiviert. `log_bin` Dadurch wird die binäre Protokollierung auf der RDS for MySQL-DB-Instance deaktiviert, wodurch wiederum die `binlog_format` Sitzungsvariable auf den Standardwert von `ROW` in der Datenbank zurückgesetzt wird. Wir empfehlen, Backups nicht zu deaktivieren. Weitere Informationen zur Einstellung Aufbewahrungszeitraums für Backups finden Sie unter Einstellungen für DB-Instances.
6. Wählen Sie Save changes (Änderungen speichern), um die Aktualisierungen in dieser DB--Parametergruppe zu speichern.

Da der `binlog_format` Parameter in RDS for MySQL dynamisch ist, müssen Sie die DB-Instance nicht neu starten, damit die Änderungen wirksam werden. (Beachten Sie, dass dieser

Parameter in Aurora MySQL statisch ist. Weitere Informationen finden Sie unter Konfiguration der Binärprotokollierung von Aurora MySQL.)

> **Important**
>
> Das Ändern einer DB-Parametergruppe wirkt sich auf alle DB-Instances aus, die diese Parametergruppe verwenden. Wenn Sie unterschiedliche binäre Logging-Formate für verschiedene MySQL-DB-Instances in einer AWS Region angeben möchten, müssen die DB-Instances unterschiedliche DB-Parametergruppen verwenden. Diese Parametergruppen identifizieren unterschiedliche Protokollierungsformate. Weisen Sie den einzelnen DB-Instances die entsprechende DB-Parametergruppe zu.

Konfiguration von My SQL Binary Logging für Multi-AZ-DB-Cluster

Die binäre Protokollierung in Amazon RDS for My SQL Multi-AZ-DB-Clustern zeichnet alle Datenbankänderungen auf, um Replikation, point-in-time Wiederherstellung und Prüfung zu unterstützen. In Multi-AZ-DB-Clustern synchronisieren Binärprotokolle sekundäre Knoten mit dem primären Knoten, wodurch die Datenkonsistenz in allen Availability Zones gewährleistet und reibungslose Failover ermöglicht werden.

Um die Binärprotokollierung zu optimieren, RDS unterstützt Amazon die Komprimierung von Binärprotokolltransaktionen, wodurch der Speicherbedarf für Binärprotokolle reduziert und die Replikationseffizienz verbessert wird.

Themen

- Komprimierung binärer Protokolltransaktionen für Multi-AZ-DB-Cluster
- Konfiguration der Komprimierung von binären Protokolltransaktionen für Multi-AZ-DB-Cluster

Komprimierung binärer Protokolltransaktionen für Multi-AZ-DB-Cluster

Bei der Komprimierung von Transaktionen für binäre Protokolle wird der zstd-Algorithmus verwendet, um die Größe der in Binärprotokollen gespeicherten Transaktionsdaten zu reduzieren. Wenn diese Option aktiviert ist, komprimiert die My SQL Database-Engine Transaktionsnutzlasten zu einem einzigen Ereignis, wodurch der I/O- und Speicheraufwand minimiert wird. Diese Funktion verbessert die Datenbankleistung, reduziert die Größe des Binärprotokolls und optimiert die Ressourcennutzung für die Verwaltung und Replikation von Protokollen in Multi-AZ-DB-Clustern.

Amazon RDS bietet die Komprimierung von Binärprotokolltransaktionen RDS für My SQL Multi-AZ-DB-Cluster über die folgenden Parameter:

- `binlog_transaction_compression`— Wenn diese Option aktiviert ist (1), komprimiert die Datenbank-Engine Transaktionsnutzlasten und schreibt sie als einzelnes Ereignis in das Binärprotokoll. Dies reduziert die Speichernutzung und den I/O-Overhead. Der Parameter ist standardmäßig deaktiviert.
- `binlog_transaction_compression_level_zstd`— Konfiguriert die zstd-Komprimierungsstufe für binäre Protokolltransaktionen. Höhere Werte erhöhen das Komprimierungsverhältnis, wodurch der Speicherbedarf weiter reduziert wird, aber auch der Speicherverbrauch für die Komprimierung steigtCPU. Der Standardwert ist 3 mit einem Bereich von 1 bis 22.

Mit diesen Parametern können Sie die Komprimierung von Binärprotokollen auf der Grundlage der Workload-Merkmale und der Ressourcenverfügbarkeit fein abstimmen. Weitere Informationen finden Sie in der SQL Dokumentation Meine Dokumentation unter Komprimierung von Binärprotokolltransaktionen.

Die Komprimierung von binären Protokolltransaktionen bietet die folgenden Hauptvorteile:

- Durch die Komprimierung wird die Größe von Binärprotokollen verringert, insbesondere bei Workloads mit großen Transaktionen oder hohen Schreibvolumen.
- Kleinere Binärprotokolle reduzieren den Netzwerk- und I/O-Overhead und verbessern so die Replikationsleistung.
- Der `binlog_transaction_compression_level_zstd` Parameter ermöglicht die Steuerung des Kompromisses zwischen dem Komprimierungsverhältnis und dem Ressourcenverbrauch.

Konfiguration der Komprimierung von binären Protokolltransaktionen für Multi-AZ-DB-Cluster

Um die Komprimierung von binären Protokolltransaktionen für einen RDS for My SQL Multi-AZ-DB-Cluster zu konfigurieren, ändern Sie die entsprechenden Cluster-Parametereinstellungen entsprechend Ihren Workload-Anforderungen.

Konsole

Um die Komprimierung von binären Protokolltransaktionen zu aktivieren

1. Ändern Sie die DB-Cluster-Parametergruppe, um den `binlog_transaction_compression` Parameter auf festzulegen1.
2. (Optional) Passen Sie den Wert des `binlog_transaction_compression_level_zstd` Parameters an Ihre Workload-Anforderungen und die Ressourcenverfügbarkeit an.

Weitere Informationen finden Sie unter the section called “Ändern von Parametern in einer DB-Cluster-Parametergruppe”.

AWS CLI

Verwenden Sie den Befehl modify-db-cluster-parameter-group AWS CLI, um die Komprimierung von binären Protokolltransaktionen mit dem zu konfigurieren.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name your-cluster-parameter-group \
  --parameters
 "ParameterName=binlog_transaction_compression,ParameterValue=1,ApplyMethod=pending-
reboot"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-cluster-parameter-group ^
  --db-cluster-parameter-group-name your-cluster-parameter-group ^
  --parameters
 "ParameterName=binlog_transaction_compression,ParameterValue=1,ApplyMethod=pending-
reboot"
```

RDS API

Verwenden Sie den Vorgang, um die Komprimierung von Binärprotokolltransaktionen mithilfe von Amazon RDS API zu konfigurieren. `ModifyDBClusterParameterGroup`

Zugriff auf MySQL-Binärprotokolle

Sie können das Dienstprogramm mysqlbinlog verwenden, um Binärprotokolle aus RDS-for-MySQL-DB-Instances herunterzuladen oder zu streamen. Das Binärprotokoll wird auf den lokalen Computer heruntergeladen, von wo aus Sie Aktionen, wie die Wiedergabe eines Protokolls mithilfe des Hilfsprogramms mysql ausführen können. Weitere Informationen über die Verwendung des Dienstprogramms mysqlbinlog finden Sie unter Verwenden von mysqlbinlog zum Sichern binärer Protokolldateien in der MySQL-Dokumentation.

Verwenden Sie zum Ausführen des Dienstprogramms mysqlbinlog mit einer Amazon RDS-Instance die folgenden Optionen:

- `--read-from-remote-server` – Erforderlich.
- `--host` – der DNS-Name vom Endpunkt der Instance.
- `--port` – der von der Instance verwendete Port.
- `--user` – ein MySQL-Benutzer, dem die Berechtigung `REPLICATION SLAVE` erteilt wurde.
- `--password` – das Passwort für den MySQL-Benutzer oder lassen Sie einen Passwortwert aus, damit das Dienstprogramm zur Eingabe eines Passworts auffordert.
- `--raw` – Laden Sie die Datei im Binärformat herunter.
- `--result-file` – die lokale Datei, die den raw-Output empfangen soll.
- `--stop-never` – Streamen Sie die binären Protokolldateien.
- `--verbose` – Wenn Sie das Binlog-Format `ROW` verwenden, schließen Sie diese Option ein, um die Zeilenereignisse als Pseudo-SQL-Anweisungen anzuzeigen. Weitere Informationen zur Option `--verbose` finden Sie unter mysqlbinlog row event display in der MySQL-Dokumentation.
- Geben Sie die Namen einer oder mehrerer Binärprotokolldateien an. Verwenden Sie den SQL-Befehl `SHOW BINARY LOGS`, um eine Liste der verfügbaren Protokolle abzurufen.

Weitere Informationen über mysqlbinlog-Optionen finden Sie unter mysqlbinlog – Hilfsprogramm für die Verarbeitung binärer Protokolldateien in der MySQL-Dokumentation.

Die folgenden Beispiele veranschaulichen die Verwendung des Dienstprogramms mysqlbinlog.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
mysqlbinlog \
    --read-from-remote-server \
```

```
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password \
    --raw \
    --verbose \
    --result-file=/tmp/ \
    binlog.00098
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password ^
    --raw ^
    --verbose ^
    --result-file=/tmp/ ^
    binlog.00098
```

Binärprotokolle müssen auf der DB-Instance verfügbar bleiben, damit das Hilfsprogramm mysqlbinlog auf sie zugreifen kann. Um ihre Verfügbarkeit sicherzustellen, verwenden Sie die mysql.rds_set_configuration gespeicherte Prozedur und geben Sie einen Zeitraum an, in dem Sie die Protokolle herunterladen können. Wenn diese Konfiguration nicht festgelegt ist, löscht Amazon RDS die Binärprotokolle so schnell wie möglich, was zu Lücken in den Binärprotokollen führt, die das Hilfsprogramm mysqlbinlog abruft.

Das folgende Beispiel setzt den Aufbewahrungszeitraum auf 1 Tag.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Verwenden Sie die gespeicherte Prozedur mysql.rds_show_configuration, um die aktuelle Einstellung anzeigen zu lassen.

```
call mysql.rds_show_configuration;
```

Datenbankprotokolldateien von Amazon RDS für Oracle

Sie können über die RDS Amazon-Konsole oder auf Oracle-Alert-Logs, Prüfdateien und Trace-Dateien zugreifenAPI. Weitere Informationen zum Anzeigen und Herunterladen von dateibasierten Datenbankprotokollen finden Sie unter Überwachung von Amazon RDS Amazon.

Die bereitgestellten Oracle-Prüfungsdateien sind die Oracle-Standardprüfungsdateien. Amazon RDS unterstützt die detaillierte Auditing-Funktion (FGA) von Oracle. Der Protokollzugriff bietet jedoch keinen Zugriff auf FGA Ereignisse, die in der `SYS.FGA_LOG$` Tabelle gespeichert sind und auf die über die Ansicht zugegriffen werden kann. `DBA_FGA_AUDIT_TRAIL`

Der `DescribeDBLogFiles`APIVorgang, der die Oracle-Protokolldateien auflistet, die für eine DB-Instance verfügbar sind, ignoriert den `MaxRecords` Parameter und gibt bis zu 1.000 Datensätze zurück. Der Aufruf wird `LastWritten` als POSIX Datum in Millisekunden zurückgegeben.

Themen

- Aufbewahrungsplan
- Arbeiten mit Oracle-Trace-Dateien
- Oracle-Logs in Amazon CloudWatch Logs veröffentlichen
- Zugreifen auf Warn- und Listener-Logs

Aufbewahrungsplan

Die Oracle-Datenbank-Engine kann Protokolldateien rotieren, wenn diese sehr groß werden. Um Audit- oder Trace-Dateien aufzubewahren, laden Sie diese herunter. Wenn Sie die Dateien lokal speichern, reduzieren Sie Ihre RDS Amazon-Speicherkosten und stellen mehr Speicherplatz für Ihre Daten zur Verfügung.

Die folgende Tabelle zeigt den Aufbewahrungszeitplan für Oracle-Alert-Logs, Audit-Dateien und Trace-Dateien bei AmazonRDS.

Protokolltyp	Aufbewahrungsplan
Alarmprotokolle	Das Textwarnungsprotokoll wird täglich rotiert, wobei die Aufbewahr ung für 30 Tage von Amazon RDS verwaltet wird. Das XML Warnproto koll wird mindestens sieben Tage lang aufbewahrt. Sie können über die Ansicht `ALERTLOG` auf dieses Protokoll zugreifen.

Protokolltyp	Aufbewahrungsplan
Audit-Dateien	Der Standardaufbewahrungszeitraum für Prüfungsdateien beträgt sieben Tage. Amazon löscht RDS möglicherweise Auditdateien, die älter als sieben Tage sind.
Trace-Dateien	Der Standardaufbewahrungszeitraum für Trace-Dateien beträgt sieben Tage. Amazon löscht RDS möglicherweise Protokolldateien, die älter als sieben Tage sind.
Listener-Protokolle	Der Standardaufbewahrungszeitraum für Listener-Protokolle beträgt sieben Tage. Amazon löscht RDS möglicherweise Listener-Logs, die älter als sieben Tage sind.

> **Note**
>
> Für Audit-Dateien und Trace-Dateien gilt die gleiche Aufbewahrungskonfiguration.

Arbeiten mit Oracle-Trace-Dateien

Im Folgenden finden Sie Beschreibungen der RDS Amazon-Verfahren zum Erstellen, Aktualisieren, Zugreifen und Löschen von Trace-Dateien.

Themen

- Auflisten von Dateien
- Erzeugen von Trace-Dateien und Nachverfolgen einer Sitzung
- Abrufen von Trace-Dateien
- Bereinigen von Trace-Dateien

Auflisten von Dateien

Sie können jedes der beiden Verfahren verwenden, um Zugriff auf eine Datei im Pfad `background_dump_dest` zuzulassen. Das erste Verfahren aktualisiert eine Ansicht mit einer Liste aller Dateien in `background_dump_dest`.

```
EXEC rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

Nachdem die Ansicht aktualisiert wurde, fragen Sie die folgende Ansicht ab, um auf die Ergebnisse zuzugreifen.

```
SELECT * FROM rdsadmin.tracefile_listing;
```

Eine Alternative zur vorherigen Vorgehensweise ist die Verwendung von `FROM table` zum Streamen von nicht-relationalen Daten in einem tabellenähnlichen Format, um den Inhalt des Datenbankverzeichnisses aufzulisten.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('BDUMP'));
```

Die folgende Abfrage zeigt den Text einer Protokolldatei.

```
SELECT text FROM
 TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_dbname.log.date'));
```

Rufen Sie bei einer Read Replica den Namen des BDUMP Verzeichnisses durch eine Abfrage `V$DATABASE.DB_UNIQUE_NAME` ab. Wenn der eindeutige Name lautet`DATABASE_B`, dann ist es das BDUMP Verzeichnis. `BDUMP_B` Im folgenden Beispiel wird der BDUMP Name eines Replikats abgefragt und dann mit diesem Namen der Inhalt von `alert_DATABASE.log.2020-06-23` abgefragt.

```
SELECT 'BDUMP' || (SELECT regexp_replace(DB_UNIQUE_NAME,'.*(_[A-Z])', '\1') FROM V
$DATABASE) AS BDUMP_VARIABLE FROM DUAL;

BDUMP_VARIABLE
--------------
BDUMP_B

SELECT TEXT FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP_B','alert_DATABASE.log.2020-06-23'));
```

Erzeugen von Trace-Dateien und Nachverfolgen einer Sitzung

Da es keine Einschränkungen gibt`ALTER SESSION`, stehen viele Standardmethoden zur Generierung von Trace-Dateien in Oracle weiterhin für eine RDS Amazon-DB-Instance zur

Verfügung. Die folgenden Methoden sind für Trace-Dateien vorgesehen, auf die in größerem Umfang zugegriffen werden muss.

Oracle-Methode	RDSAmazon-Methode
`oradebug hanganalyze 3`	`EXEC rdsadmin.manage_tracefiles.hanganalyze;`
`oradebug dump systemstate 266`	`EXEC rdsadmin.manage_tracefiles.dump_systemstate;`

Sie können viele Standardmethoden verwenden, um einzelne Sitzungen zu verfolgen, die mit einer Oracle-DB-Instance in Amazon verbunden sindRDS. Um die Ablaufverfolgung für eine Sitzung zu aktivieren, können Sie Unterprogramme in SQL PL/-Paketen ausführen, die von Oracle bereitgestellt werden, z. B. und. `DBMS_SESSION` `DBMS_MONITOR` Weitere Informationen finden Sie unter Enabling Tracing for a Session in der Oracle-Dokumentation.

Abrufen von Trace-Dateien

Sie können jede Trace-Datei `background_dump_dest` mithilfe einer SQL Standardabfrage in einer von Amazon RDS verwalteten externen Tabelle abrufen. Für diese Methode müssen Sie die Prozedur ausführen, um den Speicherort für diese Tabelle auf die spezifische Trace-Datei festzulegen.

Sie können beispielsweise die oben erwähnte Ansicht `rdsadmin.tracefile_listing` verwenden, um alle Trace-Dateien im System aufzulisten. Danach kann die `tracefile_table`-Ansicht so konfiguriert werden, dass sie auf die gewünschte Trace-Datei zeigt.

```
EXEC
 rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

Im folgenden Beispiel wird eine externe Tabelle im aktuellen Schema mit dem auf die angegebene Datei eingestellten Speicherort erstellt. Sie können den Inhalt mithilfe einer SQL Abfrage in eine lokale Datei abrufen.

```
SPOOL /tmp/tracefile.txt
SELECT * FROM tracefile_table;
```

```
SPOOL OFF;
```

Bereinigen von Trace-Dateien

Trace-Dateien können sich ansammeln und viel Festplattenspeicher belegen. Amazon RDS löscht standardmäßig Trace-Dateien und Protokolldateien, die älter als sieben Tage sind. Sie können den Aufbewahrungszeitraum für Trace-Dateien mit der Prozedur `show_configuration` anzeigen und festlegen. Sie sollten den Befehl `SET SERVEROUTPUT ON` ausführen, damit Sie die Konfigurationsergebnisse sehen können.

Im folgenden Beispiel wird zunächst der aktuelle Aufbewahrungszeitraum angezeigt und dann ein neuer Zeitraum festgelegt.

```
# Show the current tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:10080
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.

# Set the tracefile retention to 24 hours:
SQL> EXEC rdsadmin.rdsadmin_util.set_configuration('tracefile retention',1440);
SQL> commit;

#show the new tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:1440
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.
```

Sie können zusätzlich zur regelmäßigen automatischen Bereinigung manuell Dateien aus entferne `background_dump_dest`. Im folgenden Beispiel werden alle Dateien gelöscht, die älter als fünf Minuten sind.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

Sie können auch alle Dateien löschen, die mit einem bestimmten Muster übereinstimmen (geben Sie in diesem Fall keine Dateinamenserweiterung wie .trc an). Im folgenden Beispiel werden alle Dateien gelöscht, die mit beginne `SCHPOC1_ora_5935`.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

Oracle-Logs in Amazon CloudWatch Logs veröffentlichen

Sie können Ihre RDS Oracle-DB-Instance so konfigurieren, dass Protokolldaten in einer Protokollgruppe in Amazon CloudWatch Logs veröffentlicht werden. Mit CloudWatch Logs können Sie die Protokolldaten analysieren und zur Erstellung von Alarmen und CloudWatch zum Anzeigen von Metriken verwenden. Sie können CloudWatch Logs verwenden, um Ihre Protokolldatensätze in einem äußerst langlebigen Speicher zu speichern.

Amazon RDS veröffentlicht jedes Oracle-Datenbankprotokoll als separaten Datenbank-Stream in der Protokollgruppe. Wenn Sie beispielsweise die Exportfunktion so konfigurieren, dass das Audit-Protokoll berücksichtigt wird, werden Audit-Daten in einem Audit-Protokollstream in der Protokollgruppe `/aws/rds/instance/my_instance/audit` gespeichert. In der folgenden Tabelle sind die Anforderungen zusammengefasst, die RDS für die Veröffentlichung von Protokollen durch Oracle in Amazon CloudWatch Logs gelten.

Protokollnamen	Anforderung	Standard
Alert-Protokoll	Keine. Sie können dieses Protokoll nicht deaktivieren.	Aktiviert
Trace-Protokoll	Setzen Sie den `trace_enabled` Parameter auf die Standardeinstellung `TRUE` oder lassen Sie ihn unverändert.	`TRUE`
Prüfungsprotokoll	Setzen Sie den `audit_trail` Parameter auf einen der folgenden zulässigen Werte: `{ none | os | db [, extended] | xml [, extended] }`	`none`
Listener-Protokoll	Keine. Sie können dieses Protokoll nicht deaktivieren.	Aktiviert
Oracle Management Agent-Protokoll	Keine. Sie können dieses Protokoll nicht deaktivieren.	Aktiviert

Dieses Protokoll von Oracle Management Agent besteht aus den in der folgenden Tabelle aufgeführten Protokollgruppen.

Protokollnamen	CloudWatch Protokollgruppe
emctl.log	oemagent-emctl
emdctlj.log	oemagent-emdctlj
gcagent.log	oemagent-gcagent
gcagent_errors.log	oemagent-gcagent-errors
emagent.nohup	oemagent-emagent-nohup
secure.log	oemagent-secure

Weitere Informationen finden Sie unter Locating Management Agent Log and Trace Files (Management Agent Protokoll- und Trace-Dateien anzeigen) in der Oracle-Dokumentation.

Konsole

Um Oracle-DB-Logs in CloudWatch Logs zu veröffentlichen AWS Management Console

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus.
4. Wählen Sie im Abschnitt Protokollexporte die Protokolle aus, die Sie in CloudWatch Logs veröffentlichen möchten.
5. Wählen Sie Weiter und dann auf der zusammenfassenden Seite Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Um Oracle-Protokolle zu veröffentlichen, können Sie den Befehl `modify-db-instance` mit den folgenden Parametern verwenden:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Eine Änderung der Option `--cloudwatch-logs-export-configuration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund sind die Optionen `--apply-immediately` und `--no-apply-immediately` wirkungslos.

Sie können Oracle-Protokolle auch mit den folgenden Befehlen veröffentlichen:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Example

Im folgenden Beispiel wird eine Oracle-DB-Instance erstellt, bei der die Veröffentlichung von CloudWatch Logs aktiviert ist. Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Array von Zeichenketten. Die Zeichenfolgen können eine beliebige Kombination von `alert`, `audit`, `listener` und `trace` sein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration
 '["trace","audit","alert","listener","oemagent"]' \
    --db-instance-class db.m5.large \
    --allocated-storage 20 \
    --engine oracle-ee \
    --engine-version 19.0.0.0.ru-2024-04.rur-2024-04.r1 \
    --license-model bring-your-own-license \
    --master-username myadmin \
    --manage-master-user-password
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration trace alert audit listener oemagent ^
    --db-instance-class db.m5.large ^
    --allocated-storage 20 ^
    --engine oracle-ee ^
    --engine-version 19.0.0.0.ru-2024-04.rur-2024-04.r1 ^
    --license-model bring-your-own-license ^
    --master-username myadmin ^
    --manage-master-user-password
```

Example

Das folgende Beispiel ändert eine bestehende Oracle-DB-Instance, um Protokolldateien in CloudWatch Logs zu veröffentlichen. Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Objekt. Der Schlüssel für dieses Objekt ist `EnableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit einer beliebigen Kombination aus `alert`, `audit`, `listener` und `trace`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["trace","alert","audit","listener","oemagent"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration EnableLogTypes=\"trace\",\"alert\",\"audit
\",\"listener\",\"oemagent\"
```

Example

Im folgenden Beispiel wird eine bestehende Oracle-DB-Instance dahingehend geändert, dass die Veröffentlichung von Audit- und Listener-Logdateien in Logs deaktiviert wird. CloudWatch Der `--cloudwatch-logs-export-configuration` Wert ist ein JSON Objekt. Der Schlüssel für dieses

Objekt ist `DisableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit einer beliebigen Kombination aus `alert`, `audit`, `listener` und `trace`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration DisableLogTypes=\"audit\",\"listener\"
```

RDS API

Sie können Oracle-DB-Logs mit dem veröffentlichen RDSAPI. Die Aktion `ModifyDBInstance` kann dazu mit den folgenden Parametern aufgerufen werden:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Eine Änderung des Parameters `CloudwatchLogsExportConfiguration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund ist der Parameter `ApplyImmediately` wirkungslos.

Sie können Oracle-Logs auch veröffentlichen, indem Sie die folgenden RDS API Operationen aufrufen:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Führen Sie eine dieser RDS API Operationen mit den folgenden Parametern aus:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Abhängig von dem RDS Vorgang, den Sie ausführen, sind möglicherweise andere Parameter erforderlich.

Zugreifen auf Warn- und Listener-Logs

Sie können das Alert-Protokoll über die RDS Amazon-Konsole einsehen. Sie können auch die folgende SQL Anweisung verwenden.

```
SELECT message_text FROM alertlog;
```

Greifen Sie mit Amazon CloudWatch Logs auf das Listener-Protokoll zu.

> **Note**
>
> Oracle rotiert die Alert- und Listener-Logs, wenn sie 10 MB überschreiten. Ab diesem Zeitpunkt sind sie in Amazon RDS Views nicht mehr verfügbar.

Datenbank-Protokolldateien von RDS für PostgreSQL

Sie können die folgenden Typen von überwachen:

- PostgreSQL-Protokoll
- Protokoll aktualisieren
- Fehlerprotokoll für die IAM-Datenbankauthentifizierung

> **Note**
>
> Um Fehlerprotokolle für die IAM-Datenbankauthentifizierung zu aktivieren, müssen Sie zunächst die IAM-Datenbankauthentifizierung für Ihre aktivieren. Weitere Hinweise zur Aktivierung der IAM-Datenbankauthentifizierung finden Sie unter. Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung

RDS für PostgreSQL protokolliert Datenbankaktivitäten in der PostgreSQL-Standardprotokolldatei. Bei einer On-Premises PostgreSQL-DB-Instance werden diese Nachrichten in `log/postgresql.log` lokal gespeichert. Für eine DB-Instance von RDS für PostgreSQL ist die Protokolldatei auf der Amazon-RDS-Instance verfügbar. Auf diese Protokolle kann auch über die zugegriffen werden AWS Management Console, wo Sie sie anzeigen oder herunterladen können. Die Standardprotokollierungsebene erfasst Anmeldefehler, schwerwiegende Serverfehler, Deadlocks und Abfragefehler.

Weitere Informationen zum Anzeigen, Herunterladen und Überwachen von dateibasierten Datenbankprotokollen finden Sie unter Überwachung von Amazon RDS Amazon. Weitere Informationen zu PostgreSQL-Protokollen finden Sie unter Arbeiten mit Amazon RDS und Aurora-PostgreSQL-Protokollen: Teil 1 und Arbeiten mit Amazon RDS und Aurora-PostgreSQL-Protokollen: Teil 2.

Zusätzlich zu den in diesem Thema behandelten PostgreSQL-Standardprotokollen unterstützt RDS für PostgreSQL auch die PostgreSQL-Audit-Erweiterung (`pgAudit`). Die meisten regulierten Branchen und Regierungsbehörden müssen ein Auditprotokoll oder einen Audit-Trail für die Änderungen von Daten führen, um die gesetzlichen Bestimmungen zu erfüllen. Weitere Informationen zur Installation und Verwendung von pgAudit finden Sie unter Wird verwendet pgAudit , um Datenbankaktivitäten zu protokollieren.

Themen

- Parameter für die Protokollierung in
- Aktivieren der Abfrageprotokollierung für Ihre DB-Instance von RDS für PostgreSQL
- PostgreSQL-Protokolle in Amazon Logs veröffentlichen CloudWatch

Parameter für die Protokollierung in

Sie können das Protokollierungsverhalten für Ihre DB-Instance von RDS für PostgreSQL anpassen, indem Sie verschiedene Parameter ändern. In der folgenden Tabelle finden Sie unter anderem die Parameter, die sich darauf auswirken, wie lange die Protokolle gespeichert werden, wann das Protokoll rotiert werden soll und ob das Protokoll im CSV-Format (Comma-Separated Value) ausgegeben werden soll. Außerdem ist abgesehen von anderen Einstellungen die Textausgabe angegeben, die an STDERR gesendet wurde. Wenn Sie die Einstellungen für die modifizierbaren Parameter ändern möchten, verwenden Sie eine benutzerdefinierte DB-Parametergruppe für Ihren . Instance von RDS für PostgreSQL Weitere Informationen finden Sie unter DB-Parametergruppen für Amazon RDS Amazon.

Parameter	Standard	Beschreibung
log_destination	stderr	Legt das Ausgabeformat für das Protokoll fest. Die Standardeinstellung ist `stderr`, aber Sie können auch das CSV-Format angeben, indem Sie der Einstellung `csvlog` hinzufügen. Weitere Informationen finden Sie unter Festlegen des Protokollziels (`stderr`, `csvlog`).
log_filename	postgresql.log.%Y-%m-%d-%H	Gibt das Muster für den Namen der Protokolldatei an. Zusätzlich zur Standardeinstellung unterstützt dieser Parameter `postgresql.log.%Y-%m-%d` und `postgresql.log.%Y-%m-%d-%H%M` für das Dateinamenmuster.
log_line_prefix	%t:%r:%u@%d:[%p]:	Definiert das Präfix für jede Protokollzeile, die in `stderr` geschrieben wird, um die Uhrzeit (%t), den Remote-Host (%r), den Benutzer

Parameter	Standard	Beschreibung
		(%u), die Datenbank (%d) und die Prozess-ID (%p) anzugeben.
log_rotation_age	60	Minuten, nach denen die Protokolldatei automatisch rotiert wird. Sie können diesen Wert im Bereich von 1 bis 1440 Minuten ändern. Weitere Informationen finden Sie unter Festlegen der Rotation der Protokolldatei.
log_rotation_size	–	Die Größe (kB), bei der das Protokoll automatis ch rotiert wird. Standardmäßig wird dieser Parameter nicht verwendet, da die Protokolle auf der Grundlage des `log_rotation_age` Parameters rotiert werden. Weitere Informati onen hierzu finden Sie unter Festlegen der Rotation der Protokolldatei.
rds.log_retention_ period	4320	PostgreSQL-Protokolle, die älter als die angegebene Anzahl von Minuten sind, werden gelöscht. Mit dem Standardwert von 4.320 Minuten werden Protokolldateien nach 3 Tagen gelöscht. Weitere Informationen finden Sie unter Festlegen des Aufbewahrungszeitraums für Protokolle.

Anwendungsprobleme können Sie identifizieren, indem Sie im Protokoll nach Abfragefehlern, Anmeldefehlern, Deadlocks und schwerwiegenden Serverfehlern suchen. Angenommen, Sie haben eine Legacy-Anwendung von Oracle in Amazon RDS PostgreSQL konvertiert, wobei jedoch nicht alle Abfragen ordnungsgemäß umgewandelt wurden. Diese falsch formatierten Abfragen generieren Fehlermeldungen in den Protokollen, mit denen Sie Probleme identifizieren können. Weitere Informationen zur Protokollierung von Abfragen finden Sie unter Aktivieren der Abfrageprotokollierung für Ihre DB-Instance von RDS für PostgreSQL.

In den folgenden Themen finden Sie Informationen darüber, wie Sie verschiedene Parameter festlegen, die die grundlegenden Details Ihrer PostgreSQL-Protokolle steuern.

Themen

- Festlegen des Aufbewahrungszeitraums für Protokolle
- Festlegen der Rotation der Protokolldatei
- Festlegen des Protokollziels (stderr, csvlog)
- Grundlagen des Parameters log_line_prefix

Festlegen des Aufbewahrungszeitraums für Protokolle

Der `rds.log_retention_period`-Parameter gibt an, wie lange Ihre DB-Instance von RDS für PostgreSQL die entsprechenden Protokolldateien aufbewahrt. Die Standardeinstellung ist 3 Tage (4 320 Minuten). Sie können diese Einstellung jedoch auf einen beliebigen Wert zwischen 1 Tag (1 440 Minuten) und 7 Tagen (10 080 Minuten) festlegen. Stellen Sie sicher, dass Ihre DB-Instance von RDS für PostgreSQL über ausreichend Speicherplatz verfügt, um die Protokolldateien für diesen Zeitraum zu speichern.

Wir empfehlen, dass Sie Ihre Protokolle routinemäßig in Amazon CloudWatch Logs veröffentlichen, damit Sie Systemdaten noch lange nach dem Entfernen der Protokolle aus Ihrem anzeigen und analysieren können. DB-Instance von RDS für PostgreSQL Weitere Informationen finden Sie unter PostgreSQL-Protokolle in Amazon Logs veröffentlichen CloudWatch .

Festlegen der Rotation der Protokolldatei

Neue Protokolldateien werden von Amazon RDS standardmäßig jede Stunde erstellt. Das Timing wird vom Parameter `log_rotation_age` kontrolliert. Dieser Parameter hat einen Standardwert von 60 (Minuten). Sie können ihn jedoch auf jeden beliebigen Wert zwischen 1 Minute und 24 Stunden (1 440 Minuten) festlegen. Wenn die Rotation ansteht, wird eine neue eindeutige Protokolldatei erstellt. Die Datei wird nach dem Muster benannt, das durch den Parameter `log_filename` angegeben wird.

Protokolldateien können auch entsprechend ihrer Größe gedreht werden, wie im Parameter `log_rotation_size` angegeben. Dieser Parameter gibt an, dass das Protokoll rotiert werden soll, wenn es die angegebene Größe (in Kilobyte) erreicht. Bei einer RDS-for-PostgreSQL-DB-Instance wird `log_rotation_size` nicht festgelegt, das heißt, es wird kein Wert angegeben. Der Parameter ermöglicht jedoch die Einstellung von 0-2 097 151 kB (Kilobyte).

Die Protokolldateinamen basieren auf dem Dateinamenmuster des Parameters `log_filename`. Die verfügbaren Einstellungen für diesen Parameter lauten wie folgt:

- `postgresql.log.%Y-%m-%d` – Standardformat für den Namen der Protokolldatei. Nimmt das Jahr, den Monat und das Datum in den Namen der Protokolldatei auf.
- `postgresql.log.%Y-%m-%d-%H` – Nimmt die Stunde in das Format des Protokolldateinamens auf.

Weitere Informationen finden Sie unter `log_rotation_age` und `log_rotation_size` in der PostgreSQL-Dokumentation.

Festlegen des Protokollziels (**`stderr`**, **`csvlog`**)

Standardmäßig generiert Amazon RDS PostgreSQL Protokolle im Standardfehlerformat (stderr). Dieses Format ist die Standardeinstellung des Parameters `log_destination`. Jede Nachricht erhält ein Präfix nach dem im `log_line_prefix`-Parameter angegebenen Muster. Weitere Informationen finden Sie unter Grundlagen des Parameters log_line_prefix.

RDS für PostgreSQL kann die Protokolle auch im `csvlog`-Format generieren. Das `csvlog`-Format ist nützlich, um die Protokolldaten als CSV-Daten zu analysieren. Angenommen, Sie verwenden die Erweiterung `log_fdw`, um mit Ihren Protokollen als Fremdtabellen zu arbeiten. Die Fremdtabelle, die für `stderr`-Protokolldateien erstellt wurde, enthält eine einzelne Spalte mit Protokollereignisdaten. Durch Hinzufügen von `csvlog` zum `log_destination`-Parameter erhalten Sie die Protokolldatei im CSV-Format mit Abgrenzungen für die verschiedenen Spalten der Fremdtabelle. Sie können Ihre Protokolle jetzt einfacher sortieren und analysieren. Informationen dazu, wie Sie `log_fdw` mit `csvlog` verwenden, finden Sie unter Verwenden der Erweiterung log_fdw für den Zugriff auf das DB-Protokoll mit SQL.

Wenn Sie `csvlog` für diesen Parameter angeben, beachten Sie, dass sowohl `stderr`- als auch `csvlog`-Dateien generiert werden. Achten Sie auf den von den Protokollen verbrauchten Speicher und berücksichtigen Sie dabei die `rds.log_retention_period`und andere Einstellungen, die sich auf den Protokollspeicher und Turnover auswirken. Wenn Sie `stderr` und `csvlog` verwenden, verdoppelt sich der von den Protokollen verbrauchte Speicher.

Wenn Sie `csvlog` zu `log_destination` hinzufügen und zu `stderr` allein zurückkehren möchten, müssen Sie den Parameter zurücksetzen. Rufen Sie dazu die Amazon-RDS-Konsole auf und öffnen Sie die benutzerdefinierte DB-Parametergruppe für Ihre Instance. Wählen Sie den `log_destination`-Parameter, die Option Edit parameter (Parameter bearbeiten) und dann Reset (Zurücksetzen) aus.

Weitere Informationen zum Konfigurieren der Protokollierung finden Sie unter Arbeiten mit Amazon-RDS- und Aurora-PostgreSQL-Protokollen: Teil 1.

Grundlagen des Parameters log_line_prefix

Das `stderr` Protokollformat stellt jeder Protokollnachricht die durch den `log_line_prefix` Parameter angegebenen Details voran. Der Standardwert ist:

```
%t:%r:%u@%d:[%p]:t
```

Ab Aurora PostgreSQL Version 16 können Sie auch wählen:

```
%m:%r:%u@%d:[%p]:%l:%e:%s:%v:%x:%c:%q%a
```

Jeder an stderr gesendete Protokolleintrag enthält die folgenden Informationen, die auf dem ausgewählten Wert basieren:

- `%t`— Uhrzeit des Protokolleintrags ohne Millisekunden
- `%m`— Uhrzeit des Logeintrags in Millisekunden
- `%r` – Adresse des Remote-Hosts
- `%u@%d` – Benutzername und Datenbankname
- `[%p]` – Prozess-ID, falls verfügbar
- `%l`— Nummer der Protokollzeile pro Sitzung
- `%e`— SQL-Fehlercode
- `%s`— Zeitstempel für den Prozessstart
- `%v`— Virtuelle Transaktions-ID
- `%x`— Transaktions-ID
- `%c`— Sitzungs-ID
- `%q`— Terminator, der keine Sitzung beendet
- `%a`— Name der Anwendung

Aktivieren der Abfrageprotokollierung für Ihre DB-Instance von RDS für PostgreSQL

Sie können detailliertere Informationen über Ihre Datenbankaktivitäten sammeln, einschließlich Abfragen, Abfragen, die auf Sperren warten, Prüfpunkte und viele andere Details, indem Sie einige der in der folgenden Tabelle aufgeführten Parameter festlegen. Dieses Thema konzentriert sich auf das Protokollieren von Abfragen.

Parameter	Standard	Beschreibung
log_connections	–	Protokolliert jede erfolgreiche Verbindung.
log_disconnections	–	Protokolliert das Ende jeder Sitzung und ihre Dauer.
log_checkpoints	1	Protokolliert jeden Checkpoint.
log_lock_waits	–	Protokolliert lange Sperrenwartezeiten. Dieser Parameter ist standardmäßig nicht festgelegt.
log_min_duration_sample	–	(ms) Legt die Mindestausführungszeit fest, jenseits der stichprobenartig Anweisungen protokolliert werden. Die Stichprobengröße wird mit dem `log_statement_sample_rate` -Parameter festgelegt.
log_min_duration_statement	–	Jede SQL-Anweisung, die mindestens für die angegebene Zeit oder länger ausgeführt wird, wird protokolliert. Dieser Parameter ist standardmäßig nicht festgelegt. Die Aktivierung dieses Parameters kann Sie dabei unterstützen, nicht optimierte Abfragen zu finden.
log_statement	–	Legt den Typ der protokollierten Anweisungen fest. Standardmäßig ist dieser Parameter nicht festgelegt, aber Sie können ihn in `all`, `ddl` oder `mod` ändern, um die Typen von SQL-Anweisungen anzugeben, die protokolliert werden sollen. Wenn Sie etwas anderes als `none` für diesen Parameter angeben, sollten Sie auch zusätzliche Maßnahmen ergreifen, um die Offenlegung von Passwörtern in den Protokolldateien zu verhindern. Weitere Informationen finden Sie unter Reduzieren des Risikos der Offenlegung von Passwörtern bei der Verwendung der Abfrageprotokollierung.

Parameter	Standard	Beschreibung
log_statement_sample_rate	–	Der Prozentsatz der Anweisungen, die die in `log_min_duration_sample` angegebene Zeit bei der Protokollierung überschreiten. Diese Angabe wird als Gleitkommawert zwischen 0,0 und 1,0 ausgedrückt.
log_statement_stats	–	Schreibt kumulative Leistungsstatistiken in das Serverprotokoll.

Verwendung der Protokollierung, um Abfragen mit langsamer Ausführung zu suchen

Sie können SQL-Anweisungen und Abfragen protokollieren, um Abfragen zu finden, die langsam ausgeführt werden. Sie aktivieren diese Funktion, indem Sie die Einstellungen der Parameter `log_statement` und `log_min_duration` wie in diesem Abschnitt beschrieben ändern. Bevor Sie die Abfrageprotokollierung für Ihre DB-Instance von RDS für PostgreSQL aktivieren, sollten Sie sich der möglichen Offenlegung von Passwörtern in den Protokollen bewusst sein und wissen, wie Sie die Risiken minimieren können. Weitere Informationen finden Sie unter Reduzieren des Risikos der Offenlegung von Passwörtern bei der Verwendung der Abfrageprotokollierung.

Nachstehend finden Sie Referenzinformationen zu den Parametern `log_statement` und `log_min_duration`.

log_statement

Dieser Parameter gibt den Typ der SQL-Anweisungen an, die an das Protokoll gesendet werden sollen. Der Standardwert ist `none`. Wenn Sie diesen Parameter in `all`, `ddl` oder `mod` ändern, stellen Sie sicher, dass Sie die empfohlenen Maßnahmen ergreifen, um das Risiko einer Offenlegung von Passwörtern in den Protokollen zu reduzieren. Weitere Informationen finden Sie unter Reduzieren des Risikos der Offenlegung von Passwörtern bei der Verwendung der Abfrageprotokollierung.

all

Protokolliert alle Anweisungen. Diese Einstellung wird für Debugging-Zwecke empfohlen.

ddl

Protokolliert alle Data Definition Language (DDL)-Anweisungen wie CREATE, ALTER, DROP usw.

mod

Protokolliert alle DDL-Anweisungen und Data Manipulation Language (DML)-Anweisungen wie INSERT, UPDATE und DELETE, die die Daten modifizieren.

Keine

Es werden keine SQL-Anweisungen protokolliert. Wir empfehlen diese Einstellung, um das Risiko zu vermeiden, dass Passwörter in den Protokollen offengelegt werden.

log_min_duration_statement

Jede SQL-Anweisung, die mindestens für die angegebene Zeit oder länger ausgeführt wird, wird protokolliert. Dieser Parameter ist standardmäßig nicht festgelegt. Die Aktivierung dieses Parameters kann Sie dabei unterstützen, nicht optimierte Abfragen zu finden.

–1–2147483647

Die Laufzeit in Millisekunden (ms), in der eine Anweisung protokolliert wird.

So richten Sie die Abfrageprotokollierung ein

Bei diesen Schritten wird davon ausgegangen, dass Ihr Die DB-Instance von RDS für PostgreSQL verwendet eine benutzerdefinierte DB-Parametergruppe.

1. Stellen Sie den Parameter `log_statement` auf `all` ein. Im folgenden Beispiel werden die Informationen gezeigt, die bei dieser Parametereinstellung in die Datei `postgresql.log` geschrieben werden.

```
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: statement:
 SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: QUERY
 STATISTICS
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:DETAIL: ! system
 usage stats:
! 0.017355 s user, 0.000000 s system, 0.168593 s elapsed
! [0.025146 s user, 0.000000 s system total]
! 36644 kB max resident size
! 0/8 [0/8] filesystem blocks in/out
```

```
! 0/733 [0/1364] page faults/reclaims, 0 [0] swaps
! 0 [0] signals rcvd, 0/0 [0/0] messages rcvd/sent
! 19/0 [27/0] voluntary/involuntary context switches
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: SELECT
 feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:ERROR: syntax error
 at or near "ORDER" at character 1
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: ORDER BY
 s.confidence DESC;
---------------------- END OF LOG ----------------------
```

2. Legen Sie den Parameter `log_min_duration_statement` fest. Im folgenden Beispiel werden die Informationen gezeigt, die in die Datei `postgresql.log` geschrieben werden, wenn der Parameter auf 1 festgelegt wird.

   Abfragen, die die im `log_min_duration_statement`-Parameter angegebene Dauer überschreiten, werden protokolliert. Es folgt ein Beispiel. Sie können die Protokolldatei für Ihre DB-Instance von RDS für PostgreSQL in der Amazon-RDS-Konsole anzeigen.

   ```
   2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: statement: DROP
    table comments;
   2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: duration:
    167.754 ms
   2022-10-05 19:08:07 UTC::@:[355]:LOG: checkpoint starting: time
   2022-10-05 19:08:08 UTC::@:[355]:LOG: checkpoint complete: wrote 11 buffers
    (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.013 s, sync=0.006 s,
    total=1.033 s; sync files=8, longest=0.004 s, average=0.001 s; distance=131028 kB,
    estimate=131028 kB
   ---------------------- END OF LOG ----------------------
   ```

Reduzieren des Risikos der Offenlegung von Passwörtern bei der Verwendung der Abfrageprotokollierung

Wir empfehlen, für `log_statement` die Einstellung none beizubehalten, um zu vermeiden, dass Passwörter offengelegt werden. Wenn Sie `log_statement` auf `all`, `ddl` oder mod festlegen, sollten Sie einen oder mehrere der folgenden Schritte auszuführen.

- Verschlüsseln Sie vertrauliche Informationen für den Client. Weitere Informationen finden Sie unter Verschlüsselungsoptionen der PostgreSQL-Dokumentation. Verwenden Sie die Optionen

`ENCRYPTED` und `UNENCRYPTED` der `CREATE`- und `ALTER`-Anweisungen. Weitere Informationen finden Sie im Abschnitt CREATE USER der PostgreSQL-Dokumentation.

- Richten Sie für Ihre DB-Instance von RDS für PostgreSQL die PostgreSQL Auditing (PGAudit)-Erweiterung ein und verwenden Sie sie. Diese Erweiterung redigiert sensible Informationen in CREATE- und ALTER-Anweisungen, die an das Protokoll gesendet werden. Weitere Informationen finden Sie unter Wird verwendet pgAudit , um Datenbankaktivitäten zu protokollieren.
- Beschränken Sie den Zugriff auf die Protokolle CloudWatch .
- Verwenden Sie stärkere Authentifizierungsmechanismen wie IAM.

PostgreSQL-Protokolle in Amazon Logs veröffentlichen CloudWatch

Um Ihre PostgreSQL-Protokolldatensätze in einem äußerst langlebigen Speicher zu speichern, können Sie Amazon CloudWatch Logs verwenden. Mit CloudWatch Logs können Sie auch Protokolldaten in Echtzeit analysieren und diese zur Anzeige von Metriken und CloudWatch zur Erstellung von Alarmen verwenden. Wenn Sie beispielsweise `log_statement` auf `ddl` festlegen, können Sie einen Alarm einrichten, der immer dann ausgelöst wird, wenn eine DDL-Anweisung ausgeführt wird. Sie können wählen, ob Ihre CloudWatch PostgreSQL-Protokolle während der Erstellung Ihrer RDS for PostgreSQL-DB-Instance in Logs hochgeladen werden sollen. Wenn Sie zu diesem Zeitpunkt keine Protokolle hochladen möchten, können Sie Ihre Instance später so ändern, dass ab diesem Zeitpunkt mit dem Hochladen von Protokollen begonnen wird. Mit anderen Worten, vorhandene Protokolle werden nicht hochgeladen. Es werden nur neue Protokolle hochgeladen, da sie auf Ihrer modifizierten DB-Instance von RDS für PostgreSQL erstellt werden.

Alle derzeit verfügbaren Versionen von RDS für PostgreSQL unterstützen die Veröffentlichung von Protokolldateien in Logs. CloudWatch Weitere Informationen finden Sie unter Aktualisierungen von Amazon RDS für PostgreSQL im Abschnitt Versionshinweise für Amazon RDS für PostgreSQL.

Um mit CloudWatch Logs zu arbeiten, konfigurieren Sie Ihre RDS for PostgreSQL-DB-Instance so, dass Protokolldaten in einer Protokollgruppe veröffentlicht werden.

Sie können die folgenden Protokolltypen in CloudWatch Logs for RDS for PostgreSQL veröffentlichen:

- PostgreSQL-Protokoll
- Protokoll aktualisieren
- Fehlerprotokoll für die IAM-Datenbankauthentifizierung

Nachdem Sie die Konfiguration abgeschlossen haben, veröffentlicht Amazon RDS die Protokollereignisse, um Streams innerhalb einer CloudWatch Protokollgruppe zu protokollieren. Beispielsweise werden die PostgreSQL-Protokolldaten innerhalb der Protokollgruppe gespeicher `/aws/rds/instance/my_instance/postgresql`. Um Ihre Protokolle einzusehen, öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

Konsole

So veröffentlichen Sie PostgreSQL-Protokolle mithilfe der Konsole in CloudWatch Logs

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die zu ändernde DB-Instance aus und klicken Sie anschließend auf Modify (Ändern).
4. Wählen Sie im Abschnitt Protokollexporte die Protokolle aus, die Sie in Logs veröffentlichen möchten CloudWatch .

   Der Abschnitt Protokollexporte ist nur für PostgreSQL-Versionen verfügbar, die das Veröffentlichen in Logs unterstützen. CloudWatch
5. Wählen Sie Weiter und dann auf der zusammenfassenden Seite Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Sie können PostgreSQL-Protokolle mit dem veröffentlichen. AWS CLI Sie können den Befehl `modify-db-instance` mit den folgenden Parametern aufrufen.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Eine Änderung der Option `--cloudwatch-logs-export-configuration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund sind die Optionen `--apply-immediately` und `--no-apply-immediately` wirkungslos.

Sie können PostgreSQL-Protokolle auch veröffentlichen, indem Sie die folgenden CLI-Befehle aufrufen:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Führen Sie einen dieser CLI-Befehle mit den folgenden Optionen aus:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Je nach verwendetem CLI-Befehl müssen möglicherweise noch weitere Optionen angegeben werden.

Example Ändern Sie eine Instanz, um Protokolle in Logs zu veröffentlichen CloudWatch

Das folgende Beispiel ändert eine bestehende PostgreSQL-DB-Instance, um Protokolldateien in Logs zu veröffentlichen. CloudWatch Der `--cloudwatch-logs-export-configuration`-Wert ist ein JSON-Objekt. Der Schlüssel für dieses Objekt ist `EnableLogTypes` und dessen Wert ist ein Array von Zeichenfolgen mit einer beliebigen Kombination aus `postgresql` und `upgrade`.

FürLinux, oder: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql",
 "upgrade"]}'
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["postgresql","upgrade"]}'
```

Example Erstellen Sie eine Instanz, um Logs in Logs zu CloudWatch veröffentlichen

Das folgende Beispiel erstellt eine PostgreSQL-DB-Instance und veröffentlicht Protokolldateien in Logs. CloudWatch Der Wert `--enable-cloudwatch-logs-exports` ist ein JSON-Array mit Zeichenfolgen. Die Zeichenfolgen können eine beliebige Kombination aus `postgresql` und `upgrade` sein.

FürLinux, odermacOS: Unix

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' \
    --db-instance-class db.m4.large \
    --engine postgres
```

Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' ^
    --db-instance-class db.m4.large ^
    --engine postgres
```

RDS-API

Sie können PostgreSQL-Protokolle mit der RDS-API veröffentlichen. Die Aktion `ModifyDBInstance` kann dazu mit den folgenden Parametern aufgerufen werden:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Eine Änderung des Parameters `CloudwatchLogsExportConfiguration` wird immer sofort auf die DB-Instance angewendet. Aus diesem Grund ist der Parameter `ApplyImmediately` wirkungslos.

Sie können PostgreSQL-Protokolle auch veröffentlichen, indem Sie die folgenden RDS-API-Operationen aufrufen:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

Führen Sie eine dieser RDS-API-Operationen mit den folgenden Parametern aus:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Je nach ausgeführter Operation müssen möglicherweise noch weitere Parameter angegeben werden.

überwachen Amazon RDS API ruft an AWS CloudTrail

AWS CloudTrail ist ein AWS Service, der Sie bei der Prüfung Ihres AWS Kontos unterstützt. AWS CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie es erstellen. Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail Benutzerhandbuch.

Themen

- CloudTrail Integration mit Amazon RDS
- Amazon-Protokolldateieinträge

CloudTrail Integration mit Amazon RDS

Alle Amazon-Aktionen werden von protokolliert CloudTrail. CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Service in ausgeführt wurdenRDS.

CloudTrail Ereignisse

CloudTrail erfasst API Anrufe für Amazon RDS als Ereignisse. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. Zu den Ereignissen gehören Anrufe von der RDS Amazon-Konsole und von Codeaufrufen an den RDS API Amazon-Betrieb.

Amazon-Aktivitäten werden in einem CloudTrail Ereignis im Eventverlauf aufgezeichnet. Sie können die CloudTrail Konsole verwenden, um die aufgezeichneten API Aktivitäten und Ereignisse der letzten 90 Tage in einer AWS Region einzusehen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

CloudTrail Pfade

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Amazon RDS, erstellen Sie einen Trail. Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen an einen bestimmten Amazon S3 S3-Bucket ermöglicht. CloudTrail übermittelt Protokolldateien in der Regel innerhalb von 15 Minuten nach Kontoaktivität.

> **Note**
>
> Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen.

Du kannst zwei Arten von Wanderwegen für ein AWS Konto erstellen: eine Route, die für alle Regionen gilt, oder eine Route, die für eine Region gilt. Wenn Sie ein Trail in der Konsole anlegen, gilt dieser für alle Regionen.

Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Trails
- CloudTrail unterstützte Dienste und Integrationen
- Konfiguration von SNS Amazon-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Amazon-Protokolldateieinträge

CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. CloudTrail Logdateien sind kein geordneter Stack-Trace der öffentlichen API Aufrufe, sie erscheinen also nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateDBInstance` Aktion demonstriert.

```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
```

```
    },
    "eventTime": "2018-07-30T22:14:06Z",
    "eventSource": "rds.amazonaws.com",
    "eventName": "CreateDBInstance",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",
    "requestParameters": {
        "enableCloudwatchLogsExports": [
            "audit",
            "error",
            "general",
            "slowquery"
        ],
        "dBInstanceIdentifier": "test-instance",
        "engine": "mysql",
        "masterUsername": "myawsuser",
        "allocatedStorage": 20,
        "dBInstanceClass": "db.m1.small",
        "masterUserPassword": "****"
    },
    "responseElements": {
        "dBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance",
        "storageEncrypted": false,
        "preferredBackupWindow": "10:27-10:57",
        "preferredMaintenanceWindow": "sat:05:47-sat:06:17",
        "backupRetentionPeriod": 1,
        "allocatedStorage": 20,
        "storageType": "standard",
        "engineVersion": "8.0.28",
        "dbInstancePort": 0,
        "optionGroupMemberships": [
            {
                "status": "in-sync",
                "optionGroupName": "default:mysql-8-0"
            }
        ],
        "dBParameterGroups": [
            {
                "dBParameterGroupName": "default.mysql8.0",
                "parameterApplyStatus": "in-sync"
            }
        ],
        "monitoringInterval": 0,
```

```
        "dBInstanceClass": "db.m1.small",
        "readReplicaDBInstanceIdentifiers": [],
        "dBSubnetGroup": {
            "dBSubnetGroupName": "default",
            "dBSubnetGroupDescription": "default",
            "subnets": [
                {
                    "subnetAvailabilityZone": {"name": "us-east-1b"},
                    "subnetIdentifier": "subnet-cbfff283",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1e"},
                    "subnetIdentifier": "subnet-d7c825e8",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1f"},
                    "subnetIdentifier": "subnet-6746046b",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1c"},
                    "subnetIdentifier": "subnet-bac383e0",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1d"},
                    "subnetIdentifier": "subnet-42599426",
                    "subnetStatus": "Active"
                },
                {
                    "subnetAvailabilityZone": {"name": "us-east-1a"},
                    "subnetIdentifier": "subnet-da327bf6",
                    "subnetStatus": "Active"
                }
            ],
            "vpcId": "vpc-136a4c6a",
            "subnetGroupStatus": "Complete"
        },
        "masterUsername": "myawsuser",
        "multiAZ": false,
        "autoMinorVersionUpgrade": true,
        "engine": "mysql",
```

```
        "cACertificateIdentifier": "rds-ca-2015",
        "dbiResourceId": "db-ETDZIIXHEWY5N7GXVC4SH7H5IA",
        "dBSecurityGroups": [],
        "pendingModifiedValues": {
            "masterUserPassword": "****",
            "pendingCloudwatchLogsExports": {
                "logTypesToEnable": [
                    "audit",
                    "error",
                    "general",
                    "slowquery"
                ]
            }
        },
        "dBInstanceStatus": "creating",
        "publiclyAccessible": true,
        "domainMemberships": [],
        "copyTagsToSnapshot": false,
        "dBInstanceIdentifier": "test-instance",
        "licenseModel": "general-public-license",
        "iAMDatabaseAuthenticationEnabled": false,
        "performanceInsightsEnabled": false,
        "vpcSecurityGroups": [
            {
                "status": "active",
                "vpcSecurityGroupId": "sg-f839b688"
            }
        ]
    },
    "requestID": "daf2e3f5-96a3-4df7-a026-863f96db793e",
    "eventID": "797163d3-5726-441d-80a7-6eeb7464acd4",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


Wie im vorangegangenen Beispiel im Element `userIdentity` gezeigt, enthält jeder Ereignis- oder Protokolleintrag Informationen darüber, wer die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder IAM-Benutzeranmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.

- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen zu dem `userIdentity` finden Sie im CloudTrail userIdentity Element. Weitere Informationen zu `CreateDBInstance` und anderen von Amazon finden Sie in der RDSAPIAmazon-Referenz.

Überwachung von Amazon RDS mithilfe von Datenbankaktivitätsstreams

Mithilfe von Datenbankaktivitätsstreams können Sie nahezu in Echtzeit Datenbankaktivitätsstreams überwachen.

Themen

- Übersicht über Datenbankaktivitätsstreams
- Konfigurieren von einheitlicher Prüfung für Oracle-Datenbank
- Konfiguration der Überwachungsrichtlinie für Amazon RDS für Microsoft SQL Server
- Starten eines Datenbankaktivitäts-Streams
- Ändern eines Datenbankaktivitätsstroms für Amazon RDS
- Abrufen des Status eines Datenbank-Aktivitätsstreams
- Stoppen eines Datenbankaktivitäts-Streams
- Überwachen von Datenbankaktivitäts-Streams
- Beispiele für IAM-Richtlinien für Datenbankaktivitätsstreams

Übersicht über Datenbankaktivitätsstreams

Als Amazon RDS-Datenbankadministrator müssen Sie Ihre Datenbank schützen und Compliance- und regulatorische Anforderungen erfüllen. Eine Strategie besteht darin, Datenbankaktivitätsstrreams in Ihre Überwachungstools zu integrieren. Auf diese Weise können Sie die Überwachungsaktivitäten in Ihrer überwachen und Alarme einrichten.

Sicherheitsbedrohungen sind sowohl extern als auch intern. Zum Schutz vor internen Bedrohungen können Sie den Administratorzugriff auf Datenströme durch die Konfiguration der Funktion „Datenbankaktivitätsstreams“ steuern. Amazon RDS hat DBAs keinen Zugriff auf die Erfassung, Übertragung, Speicherung und Verarbeitung der Streams.

Inhalt

- Funktionsweise von Datenbankaktivitätsstreams
- Prüfungen in Oracle-Datenbanken und Microsoft-SQL-Server-Datenbanken
  - Einheitliche Prüfung in Oracle Database

- Prüfungen in Microsoft SQL Server
- Nicht-native Prüfungsfelder für Oracle Database und SQL Server
- Überschreiben von DB-Parametergruppen
- Asynchroner Modus für Datenbankaktivitätsstreams
- Anforderungen und Einschränkungen für Datenbankaktivitätsstreams
- Verfügbarkeit von Regionen und Versionen
- Unterstützte DB-Instance-Klassen für Datenbankaktivitätsstreams

Funktionsweise von Datenbankaktivitätsstreams

Amazon RDS sendet Aktivitäten nahezu in Echtzeit per Push in einen Amazon Kinesis Data Stream. Der Kinesis Stream wird automatisch erstellt. Von Kinesis aus können Sie AWS Dienste wie Amazon Data Firehose konfigurieren und AWS Lambda den Stream nutzen und die Daten speichern.

> **Important**
>
> Die Verwendung der Datenbankaktivitätsstreams-Funktion in Amazon RDS ist kostenlos, Amazon Kinesis erhebt jedoch Gebühren für einen Datenstrom. Weitere Informationen finden Sie unter Amazon Kinesis Data Streams – Preise.

Sie können Anwendungen für das Compliance-Management für den Verbrauch von Datenbankaktivitäts-Streams konfigurieren. Solche Anwendungen können den Datenstrom verwenden, um Warnungen zu generieren und -Datenbank zu prüfen.

RDS für Oracle unterstützt Datenbankaktivitätsstreams in Multi-AZ-Bereitstellungen. In diesem Fall überwachen Datenbankaktivitätsstreams sowohl die primäre als auch die Standby-Instances.

Prüfungen in Oracle-Datenbanken und Microsoft-SQL-Server-Datenbanken

Das Auditing ist die Überwachung und Aufzeichnung von konfigurierten Datenbankaktionen. Amazon RDS erfasst standardmäßig keine Datenbankaktivitäten. Sie erstellen und verwalten Audit-Richtlinien in Ihrer Datenbank selbst.

Themen

- Einheitliche Prüfung in Oracle Database

- Prüfungen in Microsoft SQL Server
- Nicht-native Prüfungsfelder für Oracle Database und SQL Server
- Überschreiben von DB-Parametergruppen

Einheitliche Prüfung in Oracle Database

In einer Oracle-Datenbank ist eine einheitliche Prüfungsrichtlinie eine benannte Gruppe von Prüfungseinstellungen, die Sie verwenden können, um einen Aspekt des Benutzerverhaltens zu prüfen. Eine Richtlinie kann so einfach sein wie die Prüfung der Aktivitäten eines einzelnen Benutzers. Sie können auch komplexe Prüfungsrichtlinien erstellen, die Bedingungen verwenden.

Eine Oracle-Datenbank schreibt Audit-Datensätze, einschließlich SYS-Audit-Datensätzen an den einheitlichen Audit-Trail. Wenn beispielsweise während einer `INSERT`-Anweisung ein Fehler auftritt, zeigt die Standardprüfung die Fehlernummer und die ausgeführte SQL an. Der Audit-Trail befindet sich in einer schreibgeschützten Tabelle im Schema `AUDSYS`. Um auf diese Datensätze zuzugreifen, fragen Sie die `UNIFIED_AUDIT_TRAIL`-Datenwörterbuchansicht ab.

In der Regel konfigurieren Sie Datenbankaktivitätsstreams wie folgt:

1. Erstellen Sie eine Oracle-Datenbank-Prüfungsrichtlinie mithilfe des `CREATE AUDIT POLICY`-Befehls.

   Die Oracle-Datenbank generiert Audit-Datensätze.
2. Aktivieren Sie die Audit-Richtlinie mithilfe des `AUDIT POLICY`-Befehls.
3. Datenbankaktivitätsstreams konfigurieren.

   Nur Aktivitäten, die den Prüfungs-Richtlinien der Oracle-Datenbank entsprechen, werden erfasst und an den Amazon Kinesis Data Stream gesendet. Wenn Datenbankaktivitäts-Streams aktiviert sind, kann ein Oracle-Datenbankadministrator die Prüfungsrichtlinie weder ändern noch Prüfungsprotokolle entfernen.

Weitere Informationen zu einheitlichen Audit-Richtlinien finden Sie unter Prüfungsaktivitäten mit einheitlichen Prüfungsrichtlinien und AUDIT im Oracle-Datenbank-Sicherheitshandbuch.

Prüfungen in Microsoft SQL Server

Database Activity Stream verwendet SQLAudit eine Funktion zur Prüfung der SQL Server-Datenbank.

Die RDS-für-SQL-Server-Instance enthält Folgendes:

- Server-Audit – Das SQL-Server-Audit erfasst eine einzelne Instance von Aktionen auf Server- oder Datenbankebene und eine Gruppe von Aktionen, die überwacht werden sollen. Die Audits auf Serverebene `RDS_DAS_AUDIT` und `RDS_DAS_AUDIT_CHANGES` werden von RDS verwaltet.
- Server-Audit-Spezifikation – Die Server-Audit-Spezifikation zeichnet die Ereignisse auf Serverebene auf. Sie können die `RDS_DAS_SERVER_AUDIT_SPEC`-Spezifikation ändern. Diese Spezifikation ist mit dem Server-Audit `RDS_DAS_AUDIT` verknüpft. Die `RDS_DAS_CHANGES_AUDIT_SPEC`-Spezifikation wird von RDS verwaltet.
- Datenbank-Audit-Spezifikation — Die Datenbank-Audit-Spezifikation zeichnet die Ereignisse auf Datenbankebene auf. Sie können eine Datenbank-Audit-Spezifikation `RDS_DAS_DB_<name>` erstellen und sie mit dem `RDS_DAS_AUDIT`-Server-Audit verknüpfen.

Sie können Datenbankaktivitätsstreams mithilfe der Konsole oder der CLI konfigurieren. In der Regel konfigurieren Sie Datenbankaktivitätsstreams wie folgt:

1. (Optional) Erstellen Sie mit dem `CREATE DATABASE AUDIT SPECIFICATION`-Befehl eine Datenbank-Audit-Spezifikation und verknüpfen Sie sie mit dem `RDS_DAS_AUDIT`-Server-Audit.
2. (Optional) Ändern Sie die Server-Audit-Spezifikation mit dem `ALTER SERVER AUDIT SPECIFICATION`-Befehl und definieren Sie die Richtlinien.
3. Aktivieren Sie die Datenbank- und Server-Auditrichtlinien. Zum Beispiel:

   ```
   ALTER DATABASE AUDIT SPECIFICATION [<Your database specification>] WITH
   (STATE=ON)

   ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC] WITH
   (STATE=ON)
   ```

4. Datenbankaktivitätsstreams konfigurieren.

   Nur Aktivitäten, die den Audit-Richtlinien des Servers und der Datenbank entsprechen, werden erfasst und an den Amazon Kinesis Data Stream gesendet. Wenn Datenbankaktivitätsstreams aktiviert und die Richtlinien gesperrt sind, kann ein Datenbankadministrator die Audit-Richtlinie weder ändern noch Audit-Protokolle entfernen.

> **Important**
>
> Wenn die Datenbank-Audit-Spezifikation für eine bestimmte Datenbank aktiviert ist und sich die Richtlinie in einem gesperrten Zustand befindet, kann die Datenbank nicht gelöscht werden.

Weitere Informationen zum SQL-Server-Auditing finden Sie unter SQL-Server-Audit-Komponenten in der Microsoft-SQL-Server-Dokumentation.

Nicht-native Prüfungsfelder für Oracle Database und SQL Server

Wenn Sie einen Datenbankaktivitätsstream starten, generiert jedes Datenbankereignis ein entsprechendes Aktivitätsstream-Ereignis. Beispielsweise kann ein Datenbankbenutzer `SELECT` und `INSERT`-Anweisungen ausführen. Diese Ereignisse werden von der Datenbank überprüft und an einen Amazon Kinesis Data Stream gesendet.

Die Ereignisse werden im Stream als JSON-Objekte dargestellt. Ein JSON-Objekt enthält `DatabaseActivityMonitoringRecord` mit einem `databaseActivityEventList`-Array. Vordefinierte Felder im Array sind etwa `class`, `clientApplication` und `command`.

Standardmäßig enthält ein Aktivitätsstream keine systemeigenen Audit-Felder. Sie können Amazon RDS für Oracle und SQL Server so konfigurieren, dass es diese zusätzlichen Felder in das `engineNativeAuditFields`-JSON-Objekt einschließt.

In Oracle Database sind die meisten Ereignisse im einheitlichen Audit-Trail Feldern im RDS-Datenaktivitätsstream zugeordnet. Beispielsweise wird das `UNIFIED_AUDIT_TRAIL.SQL_TEXT`-Feld bei einheitlichem Audit dem `commandText`-Feld in einem Datenbankaktivitätsstream zugeordnet. Audit-Felder der Oracle-Datenbank wie `OS_USERNAME` werden jedoch nicht vordefinierten Feldern in einem Datenbankaktivitätsstream zugeordnet.

In SQL Server werden die meisten Felder des Ereignisses, die von der Datenbank aufgezeichnet werden, SQLAudit den Feldern im Aktivitätsstream der RDS-Datenbank zugeordnet. Beispielsweise ist das `code`-Feld von `sys.fn_get_audit_file` in den Audits dem `commandText`-Feld in einem Datenbankaktivitätsstream zugeordnet. SQL-Server-Datenbank-Audit-Felder wie `permission_bitmask` werden jedoch nicht vordefinierten Feldern in einem Datenbankaktivitätsstream zugeordnet.

Weitere Informationen zu databaseActivityEvent List finden Sie unterdatabaseActivityEventJSON-Array für Datenbank-Aktivitätsstreams auflisten.

Überschreiben von DB-Parametergruppen

In der Regel aktivieren Sie das einheitliche Auditing in RDS für Oracle, indem Sie eine Parametergruppe anhängen. Datenbankaktivitätsstreams erfordern jedoch eine zusätzliche Konfiguration. Um Ihren Kundenkomfort zu verbessern, führt Amazon RDS folgende Schritte aus:

- Wenn Sie einen Aktivitätsstream aktivieren, ignoriert RDS für Oracle die Auditing-Parameter in der Parametergruppe.
- Wenn Sie einen Aktivitätsstream deaktivieren, ignoriert RDS für Oracle die Auditing-Parameter nicht mehr.

Der Datenbankaktivitätsstream für SQL Server ist unabhängig von allen Parametern, die Sie in der SQL-Audit-Option festgelegt haben.

Asynchroner Modus für Datenbankaktivitätsstreams

Aktivitätsstreams in Amazon RDS sind immer asynchron. Wenn eine Datenbanksitzung ein Aktivitätsstream-Ereignis generiert, kehrt die Sitzung sofort zu normalen Aktivitäten zurück. Im Hintergrund verwandelt Amazon RDS den Aktivitätsstream in einen dauerhaften Datensatz.

Wenn bei der Hintergrundaufgabe ein Fehler auftritt, generiert Amazon RDS ein Ereignis. Dieses Ereignis gibt Anfang und Ende der Zeitfenster an, in denen möglicherweise Datensätze des Aktivitäts-Stream-Ereignisses verloren gegangen sind. Der asynchrone Modus beschleunigt die Datenbankleistung, verschlechtert jedoch die Genauigkeit des Aktivitätsstreams.

Anforderungen und Einschränkungen für Datenbankaktivitätsstreams

In RDS gelten für Datenbankaktivitätsstreams die folgenden Anforderungen und Einschränkungen:

- Amazon Kinesis ist für Datenaktivitätsstreams erforderlich.
- AWS Key Management Service (AWS KMS) ist für Datenbank-Aktivitätsstreams erforderlich, da sie immer verschlüsselt sind.
- Das Anwenden zusätzlicher Verschlüsselung auf Ihren Amazon Kinesis Kinesis-Datenstrom ist nicht mit Datenbankaktivitätsströmen kompatibel, die bereits mit Ihrem AWS KMS Schlüssel verschlüsselt sind.

- Sie erstellen und verwalten Audit-Richtlinien selbst. Im Gegensatz zu Amazon Aurora erfasst RDS for Oracle standardmäßig keine Datenbankaktivitäten.
- Sie erstellen und verwalten Audit-Richtlinie und -Spezifikationenn selbst. Im Gegensatz zu Amazon Aurora erfasst Amazon RDS standardmäßig keine Datenbankaktivitäten.
- Starten Sie in einer Multi-AZ-Bereitstellung den Datenbankaktivitätsstream nur auf der primären DB-Instance. Der Aktivitätsstream überwacht sowohl die primäre als auch die Standby-DB-Instances automatisch. Bei einem Failover sind keine weiteren Schritte erforderlich.
- Durch das Umbenennen einer DB-Instance wird kein neuer Kinesis-Stream erstellt.
- CDBs werden für RDS for Oracle nicht unterstützt.
- Read Replicas werden nicht unterstützt.

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Verfügbarkeit von Versionen und Regionen mit Datenbankaktivitätsstreams finden Sie unter Unterstützte Regionen und DB-Engines für Datenbankaktivitätsstreams in Amazon RDS.

Unterstützte DB-Instance-Klassen für Datenbankaktivitätsstreams

Für RDS für Oracle können Sie Datenbankaktivitätsstreams mit den folgenden DB-Instance-Klassen verwenden:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5.*large.tpc*.mem*x
- db.r5b.*large
- db.r5b.*large.tpc*.mem*x
- db.r5d.*large
- db.r6i.*large

- db.r6i. *groß.tpc*.mem*x
- db.x2idn.*large
- db.x2iedn.*large
- db.x2iezn.*large
- db.z1d.*large

Für RDS für SQL Server können Sie Datenbankaktivitätsstreams mit den folgenden DB-Instance-Klassen verwenden:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5b.*large
- db.r5d.*large
- db.r6i.*large
- db.x1e.*large
- db.x2iedn.*large
- db.z1d.*large

Weitere Informationen zu Instance-Klassentypen finden Sie unter .

Konfigurieren von einheitlicher Prüfung für Oracle-Datenbank

Wenn Sie die einheitliche Prüfung für die Verwendung mit Datenbankaktivitäts-Streams konfigurieren, sind die folgenden Situationen möglich:

- Die einheitliche Prüfung ist nicht für Ihre Oracle-Datenbank konfiguriert.

  Erstellen Sie in diesem Fall neue Richtlinien mit dem `CREATE AUDIT POLICY`-Befehl und aktivieren Sie sie dann mit dem Befehl `AUDIT POLICY`. Im folgenden Beispiel wird eine Richtlinie erstellt und aktiviert, um Benutzer mit bestimmten Berechtigungen und Rollen zu überwachen.

```
CREATE AUDIT POLICY table_pol
PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE
ROLES emp_admin, sales_admin;

AUDIT POLICY table_pol;
```

Vollständige Anweisungen finden Sie unter Konfigurieren von Prüfungsrichtlinien in der Oracle-Datenbank-Dokumentation.

- Die einheitliche Prüfung ist für Ihre Oracle-Datenbank konfiguriert.

  Wenn Sie einen Datenbankaktivitätsstream aktivieren, löscht RDS für Oracle automatisch vorhandene Audit-Daten. Es entzieht auch Prüfungs-Trail-Berechtigungen. RDS for Oracle kann die folgenden Aktionen nicht mehr ausführen:

  - Einheitliche Prüfungs-Trail-Datensätze löschen.
  - Hinzufügen, Löschen oder Ändern der einheitlichen Prüfungsrichtlinie.
  - Aktualisieren des letzten archivierten Zeitstempels.

  > **Important**
  >
  > Es wird dringend empfohlen, Ihre Audit-Daten zu sichern, bevor Sie einen Datenbankaktivitätsstream aktivieren.

  Eine Beschreibung der `UNIFIED_AUDIT_TRAIL`-Ansicht, siehe UNIFIED_AUDIT_TRAIL. Wenn Sie ein Konto bei Oracle Support haben, finden Sie weitere Informationen unter So löschen Sie den EINHEITLICHEN PRÜFUNGS-TRAIL.

Konfiguration der Überwachungsrichtlinie für Amazon RDS für Microsoft SQL Server

Eine SQL Server-Datenbank-Instance hat das Server-Audit`RDS_DAS_AUDIT`, das von Amazon verwaltet wird`RDS`. Sie können die Richtlinien zur Aufzeichnung von Serverereignissen in der Server-Audit-Spezifikation `RDS_DAS_SERVER_AUDIT_SPEC` definieren. Sie können eine Datenbank-Audit-Spezifikation erstellen, z. B. `RDS_DAS_DB_<name>`, und die Richtlinien für die Aufzeichnung von Datenbankereignissen definieren. Eine Liste der Audit-Aktionsgruppen auf Server- und

Datenbankebene finden Sie in der Microsoft SQLServer-Dokumentation unter Aktionsgruppen und Aktionen für die SQL Serverüberwachung.

Die Standard-Serverrichtlinie überwacht nur fehlgeschlagene Anmeldungen und Änderungen an Datenbank- oder Server-Audit-Spezifikationen für Datenbankaktivitätsstreams.

Zu den Einschränkungen für das Audit und die Audit-Spezifikationen gehören:

- Sie können die Server- oder Datenbank-Audit-Spezifikationen nicht ändern, wenn sich der Datenbankaktivitätsstream in einem gesperrten Zustand befindet.
- Sie können die Server-Audit-Spezifikation `RDS_DAS_AUDIT` nicht ändern.
- Sie können die Serverüberwachung `RDS_DAS_CHANGES` oder die zugehörige SQL Serverüberwachungsspezifikation nicht ändern`RDS_DAS_CHANGES_AUDIT_SPEC`.
- Wenn Sie eine Datenbank-Audit-Spezifikation erstellen, müssen Sie das Format `RDS_DAS_DB_<name>` verwenden, beispielsweise `RDS_DAS_DB_databaseActions`.

> **Important**
>
> Für kleinere Instance-Klassen empfehlen wir, nicht alle, sondern nur die erforderlichen Daten zu prüfen. Dies trägt dazu bei, die Leistungsauswirkungen von Datenbankaktivitätsstreams auf diese Instance-Klassen zu reduzieren.

Der folgende Beispielcode ändert die Server-Audit-Spezifikation `RDS_DAS_SERVER_AUDIT_SPEC` und überprüft alle Abmelde- und erfolgreichen Anmeldeaktionen:

```
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
      WITH (STATE=OFF);
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
      ADD (LOGOUT_GROUP),
      ADD (SUCCESSFUL_LOGIN_GROUP)
      WITH (STATE = ON );
```

Der folgende Beispielcode erstellt eine Datenbank-Audit-Spezifikation `RDS_DAS_DB_database_spec` und hängt sie an das Server-Audit `RDS_DAS_AUDIT` an:

```
USE testDB;
CREATE DATABASE AUDIT SPECIFICATION [RDS_DAS_DB_database_spec]
```

```
    FOR SERVER AUDIT [RDS_DAS_AUDIT]
    ADD ( INSERT, UPDATE, DELETE
         ON testTable BY testUser )
    WITH (STATE = ON);
```

Vergewissern Sie sich nach der Konfiguration der Audit-Spezifikationen, dass die Spezifikationen `RDS_DAS_SERVER_AUDIT_SPEC` und `RDS_DAS_DB_<name>` auf den Status `ON` gesetzt sind. Jetzt können sie die Audit-Daten an Ihren Datenbankaktivitätsstream senden.

Starten eines Datenbankaktivitäts-Streams

Wenn Sie einen Aktivitätsstream für die DB-Instance starten, generiert jedes Datenbankaktivitätsereignis, das Sie in der Audit-Richtlinie konfiguriert haben, ein Ereignis im Aktivitätsstream. Zugriffsereignisse werden von SQL-Befehlen wie `CONNECT` und `SELECT` generiert. Änderungsereignisse werden von SQL-Befehlen wie `CREATE` und `INSERT` generiert.

> **Important**
>
> Aktivieren eines Aktivitäts-Streams für eine Oracle-DB-Instance löscht vorhandene Prüfungsdaten. Es entzieht auch Prüfungs-Trail-Berechtigungen. Wenn der Stream aktiviert ist, kann RDS for Oracle Folgendes nicht mehr tun:
>
> - Einheitliche Prüfungs-Trail-Datensätze löschen.
> - Hinzufügen, Löschen oder Ändern der einheitlichen Prüfungsrichtlinie.
> - Aktualisieren des letzten archivierten Zeitstempels.

Console

Starten eines Datenbankaktivitäts-Streams

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die Amazon-RDS-Instance, für den/die Sie einen Aktivitätsstream starten möchten. Starten Sie in einer Multi-AZ-Bereitstellung den Stream nur auf der primären Instance. Der Aktivitätsstream überwacht sowohl die primäre als auch die Standby-Instances.
4. Wählen Sie für Actions (Aktionen) die Option Start activity stream (Aktivitäts-Stream starten) aus.

Das *name* Fenster Datenbankaktivitätsstream starten: wird angezeigt, in dem *name* sich Ihr befindet.

5. Geben Sie die folgenden Einstellungen ein:

   - Wählen Sie für AWS KMS keyeinen Schlüssel aus der Liste von AWS KMS keys.

     Amazon RDS verwendet den KMS-Schlüssel zur Verschlüsselung des Schlüssels, der wiederum die Datenbankaktivitäten verschlüsselt. Wählen Sie einen anderen KMS-Schlüssel als den Standardschlüssel. Weitere Informationen zu Verschlüsselungsschlüsseln und AWS KMS finden Sie unter Was ist AWS Key Management Service? im AWS Key Management Service Entwicklerhandbuch.
   - Wählen Sie für Datenbankaktivitätsereignisse die Option Engine-native Audit-Felder aktivieren, um engine-spezifische Audit-Felder einzuschließen.
   - Wählen Sie Sofort aus.

     Wenn Sie Sofort auswählen, wird die RDS-Instance sofort neu gestartet. Wenn Sie Während des nächsten Wartungsfensters auswählen, wird die RDS-Instance nicht sofort neu gestartet. In diesem Fall wird der Datenbankaktivitäts-Stream erst im nächsten Wartungsfenster gestartet.
6. Wählen Sie Start database activity stream (Datenbank-Aktivitätsstream starten) aus.

   Der Status für den die Datenbank zeigt an, dass der Aktivitätsstream gestartet wird.

   > **Note**
   >
   > Wenn Sie den Fehler `You can't start a database activity stream in this configuration` erhalten, überprüfen Sie Unterstützte DB-Instance-Klassen für Datenbankaktivitätsstreams, um festzustellen, ob Ihre RDS-Instance eine unterstützte Instance-Klasse verwendet.

AWS CLI

Um Datenbankaktivitätsstreams für zu starten, konfigurieren Sie und die Datenbank mit dem start-activity-stream AWS CLI folgenden Befehl.

- `--resource-arn` *arn* – Gibt den Amazon-Ressourcennamen (ARN) der DB-Instance an.

- `--kms-key-id` *`key`* – Gibt die KMS-Schlüssel-ID für die Verschlüsselung von Nachrichten im Datenbankaktivitäts-Stream an. Die AWS KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für AWS KMS key.
- `--engine-native-audit-fields-included` – Schließt Engine-spezifische einheitliche Audit-Felder in den Datenstrom ein. Um diese Felder auszuschließen, geben Sie `--no-engine-native-audit-fields-included` (Standard) an.

Das folgende Beispiel startet einen Datenbankaktivitätsstream für eine DB-Instance im asynchronen Modus.

Für LinuxmacOS, oderUnix:

```
aws rds start-activity-stream \
    --mode async \
    --kms-key-id my-kms-key-arn \
    --resource-arn my-instance-arn \
    --engine-native-audit-fields-included \
    --apply-immediately
```

Windows:

```
aws rds start-activity-stream ^
    --mode async ^
    --kms-key-id my-kms-key-arn ^
    --resource-arn my-instance-arn ^
    --engine-native-audit-fields-included ^
    --apply-immediately
```

Amazon RDS API

Um Datenbankaktivitätsstreams für eine zu starten, konfigurieren Sie und die Instance mithilfe des StartActivityStreamVorgangs.

Rufen Sie die Aktion mit den folgenden Parametern auf:

- `Region`
- `KmsKeyId`
- `ResourceArn`

- `Mode`
- `EngineNativeAuditFieldsIncluded`

Ändern eines Datenbankaktivitätsstroms für Amazon RDS

Möglicherweise möchten Sie Ihre RDS Amazon-Audit-Richtlinie anpassen, wenn Ihr Activity Stream gestartet wird. Wenn Sie keine Zeit und Daten verlieren möchten, indem Sie Ihren Aktivitätsstream stoppen, können Sie den Status der Prüfungsrichtlinie auf eine der folgenden Einstellungen festlegen:

Gesperrt (Standard)

Die Prüfungsrichtlinien in Ihrer Datenbank sind schreibgeschützt.

Entsperrt

Die Prüfungsrichtlinien in Ihrer Datenbank erlauben Lese- und Schreibzugriff.

Die grundlegenden Schritte sind wie folgt:

1. Ändern Sie den Status der Prüfungsrichtlinie auf „Entsperrt“.
2. Passen Sie Ihre Prüfungsrichtlinie an.
3. Ändern Sie den Status der Prüfungsrichtlinie auf „Gesperrt“.

Konsole

So ändern Sie den Status der Prüfungsrichtlinie Ihres Aktivitätsstreams

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie für Actions (Aktionen) die Option Modify database activity stream (Datenbankaktivitäts-Stream ändern) aus.

   Der Aktivitätsstream „Datenbank modifizieren“: ***name***Es erscheint ein Fenster, in dem *name* ist deine RDS Instanz.
4. Wählen Sie eine der folgenden Optionen:

Gesperrt

Wenn Sie Ihre Prüfungsrichtlinie sperren, wird sie schreibgeschützt. Sie können Ihre Überwachungsrichtlinie nur bearbeiten, wenn Sie die Richtlinie entsperren oder den Aktivitätsstream stoppen.

Entsperrt

Wenn Sie Ihre Prüfungsrichtlinie entsperren, erlaubt sie Lese-/Schreibzugriff. Sie können Ihre Prüfungsrichtlinie bearbeiten, während der Aktivitätsstream gestartet wird.

5. Wählen Sie DB-Aktivitätsstream ändern aus.

   Der Status für die RDS Amazon-Datenbank des zeigt den Aktivitätsstream Konfiguration an.

6. (Optional) Wählen Sie den DB-Instance-Link aus. Wechseln Sie zur Registerkarte Konfiguration.

   Das Feld Audit policy status (Status der Prüfungsrichtlinie) zeigt einen der folgenden Werte an:

   - Gesperrt
   - Entsperrt
   - Sperrrichtlinie
   - Entsperrrichtlinie

AWS CLI

Verwenden Sie den modify-activity-stream AWS CLI Befehl, um den Status des Activity Streams für die Datenbank-Instance zu ändern.

Option	Erforderlich?	Beschreibung
`--resource-arn` *my-instance-ARN*	Ja	Der Amazon-Ressourcenname (ARN) Ihrer RDS Datenbank-Instance.
`--audit-policy-state`	Nein	Der neue Status der Prüfungsrichtlinie für den Datenbankaktivitäts-Stream auf Ihrer Instance: `locked` oder `unlocked`.

Im folgenden Beispiel wird die Audit-Richtlinie für den Activity Stream entsperrt, der am gestartet wurde *my-instance-ARN*.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-activity-stream \
    --resource-arn my-instance-ARN \
    --audit-policy-state unlocked
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-activity-stream ^
    --resource-arn my-instance-ARN ^
    --audit-policy-state unlocked
```

Das folgende Beispiel beschreibt die Instanz *my-instance*. Die teilweise Beispielausgabe zeigt, dass die Überwachungsrichtlinie entsperrt ist.

```
aws rds describe-db-instances --db-instance-identifier my-instance

{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "started",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": "unlocked",
            ...
        }
    ]
}
```

RDS API

Verwenden Sie den ModifyActivityStreamVorgang, um den Richtlinienstatus Ihres Datenbankaktivitätsstroms zu ändern.

Rufen Sie die Aktion mit den folgenden Parametern auf:

- `AuditPolicyState`
- `ResourceArn`

Abrufen des Status eines Datenbank-Aktivitätsstreams

Sie können den Status eines Aktivitätsstreams für Ihre Amazon RDS-Datenbank-Instance über die Konsole oder AWS CLI abrufen.

Konsole

Abrufen des Status eines Datenbank-Aktivitäts-Streams

1. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den Link der DB-Instance aus.
3. Wählen Sie die Registerkarte Konfiguration aus und prüfen Sie die Statusangabe für Datenbank-Aktivitätsstream.

AWS CLI

Sie können die Aktivitätsstream-Konfiguration für eine Datenbank-Instance als Antwort auf eine CLI-Anforderung describe-db-instances abrufen.

Das folgende Beispiel beschreibt *my-instance*.

```
aws rds --region my-region describe-db-instances --db-instance-identifier my-db
```

Das folgende Beispiel zeigt eine JSON-Antwort. Die folgenden Felder werden angezeigt:

- `ActivityStreamKinesisStreamName`
- `ActivityStreamKmsKeyId`
- `ActivityStreamStatus`
- `ActivityStreamMode`
- `ActivityStreamPolicyStatus`

```
{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "starting",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": locked",
            ...
        }
    ]
}
```

RDS-API

Sie können die Aktivitätsstream-Konfiguration für eine Datenbank als Antwort auf einen DescribeDBInstances-Vorgang abrufen.

Stoppen eines Datenbankaktivitäts-Streams

Sie können den Status eines Aktivitäts-Streams über die Konsole oder AWS CLI abrufen.

Wenn Sie Ihre Amazon-RDS-Datenbank-Instance löschen, wird der Aktivitätsstream gestoppt und der zugrunde liegende Amazon-Kinesis-Stream wird automatisch gelöscht.

Konsole

So deaktivieren Sie einen Aktivitäts-Stream:

1. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die Datenbank aus, für das Sie den Datenbankaktivitäts-Stream stoppen möchten.
4. Wählen Sie für Actions (Aktionen) die Option Stop activity stream (Aktivitäts-Stream stoppen) aus. Das Fenster Database Activity Stream (Datenbankaktivitäts-Stream) wird aufgerufen.

    a. Wählen Sie Sofort aus.

Wenn Sie Sofort auswählen, wird die RDS-Instance sofort neu gestartet. Wenn Sie Während des nächsten Wartungsfensters auswählen, wird die RDS-Instance nicht sofort neu gestartet. In diesem Fall wird der Datenbankaktivitäts-Stream erst im nächsten Wartungsfenster gestoppt.

b. Klicken Sie auf Weiter.

AWS CLI

Um Datenbankaktivitäts-Streams für Ihre Datenbank zu stoppen, konfigurieren Sie die -DB-Instance mit dem AWS CLI Befehl stop-activity-stream. Bestimmen Sie die AWS-Region für die DB-Instance mit dem Parameter `--region`. Der Parameter `--apply-immediately` ist optional.

Für Linux, macOSoder Unix:

```
aws rds --region MY_REGION \
    stop-activity-stream \
    --resource-arn MY_DB_ARN \
    --apply-immediately
```

Windows:

```
aws rds --region MY_REGION ^
    stop-activity-stream ^
    --resource-arn MY_DB_ARN ^
    --apply-immediately
```

RDS-API

Um Datenbankaktivitäts-Streams für Ihre Datenbank zu stoppen, konfigurieren Sie die -DB-Instance mithilfe der -StopActivityStreamOperation. Bestimmen Sie die AWS-Region für die DB-Instance mit dem Parameter `Region`. Der Parameter `ApplyImmediately` ist optional.

Überwachen von Datenbankaktivitäts-Streams

Datenbankaktivitäts-Streams überwachen und melden Aktivitäten. Der Aktivitäts-Stream wird erfasst und an Amazon Kinesis übertragen. Von Kinesis aus können Sie den Aktivitäts-Stream überwachen, oder andere Dienste und Anwendungen können den Aktivitäts-Stream zur weiteren Analyse nutzen. Sie können den zugrunde liegenden Kinesis-Stream-Namen mithilfe des AWS CLI Befehls oder der RDS API Operation ermitteln.

Amazon RDS verwaltet den Kinesis-Stream für Sie wie folgt:

- Amazon RDS erstellt den Kinesis-Stream automatisch mit einer Aufbewahrungsfrist von 24 Stunden.
- Amazon RDS skaliert den Kinesis-Stream bei Bedarf.
- Wenn Sie den Datenbankaktivitätsstream beenden oder löschen, RDS löscht Amazon den Kinesis-Stream.

Die folgenden Kategorien von Aktivitäten werden überwacht und in das Prüfprotokoll des Aktivitäts-Streams aufgenommen:

- SQLBefehle — Alle SQL Befehle werden geprüft, ebenso vorbereitete Anweisungen, integrierte Funktionen und Funktionen in PL/. SQL Aufrufe von gespeicherten Prozeduren werden überprüft. Alle SQL Anweisungen, die innerhalb von gespeicherten Prozeduren oder Funktionen ausgegeben werden, werden ebenfalls geprüft.
- Weitere Datenbankinformationen — Zu den überwachten Aktivitäten gehören die vollständige SQL Anweisung, die Zeilenanzahl der betroffenen DML Befehlszeilen, Objekte, auf die zugegriffen wurde, und der eindeutige Datenbankname. Datenbankaktivitätsstreams überwachen auch die Bindungsvariablen und die Parameter der gespeicherten Prozeduren.

  > **Important**
  >
  > Der vollständige SQL Text jeder Anweisung ist im Audit-Log des Activity Streams sichtbar, einschließlich aller sensiblen Daten. Datenbankbenutzerkennwörter werden jedoch geschwärzt, wenn Oracle sie anhand des Kontextes ermitteln kann, wie in der folgenden SQL Anweisung.
  >
  > ```
  > ALTER ROLE role-name WITH password
  > ```

- Verbindungsinformationen – Die überwachte Aktivität umfasst Sitzungs- und Netzwerkinformationen, die Server-Prozess-ID und Beendigungscodes.

Wenn ein Activity Stream bei der Überwachung Ihrer DB-Instance ausfällt, werden Sie durch RDS Ereignisse benachrichtigt.

In den folgenden Abschnitten können Sie auf Datenbankaktivitätsstreams zugreifen, diese prüfen und verarbeiten.

Themen

- Zugreifen auf einen Activity Stream von Amazon Kinesis
- Inhalt des Auditprotokolls und Beispiele für Datenbankaktivitätsströme
- databaseActivityEventJSON-Array für Datenbank-Aktivitätsstreams auflisten
- Verarbeitung eines Datenbankaktivitätsstroms mit dem AWS SDK

Zugreifen auf einen Activity Stream von Amazon Kinesis

Wenn Sie einen Aktivitäts-Stream für eine Datenbank aktivieren, wird ein Kinesis-Stream für Sie erstellt. Von Kinesis aus können Sie die Datenbankaktivität in Echtzeit überwachen. Zur weiteren Analyse der Datenbankaktivität können Sie Ihren Kinesis-Stream mit Consumer-Anwendungen verbinden. Sie können den Stream auch mit Compliance-Management-Anwendungen wie verbinden.

Sie können entweder von der RDS Konsole oder der Kinesis-Konsole aus auf Ihren Kinesis-Stream zugreifen.

So greifen Sie über die Konsole auf einen Activity Stream von Kinesis aus zu RDS

1. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die aus, auf der Sie einen Activity Stream gestartet haben.
4. Wählen Sie Konfiguration.
5. Wählen Sie unter Database activity stream (Datenbank-Aktivitätsstream) den Link unter Kinesis stream (Kinesis-Stream) aus.
6. Wählen Sie in der Kinesis-Konsole Monitoring (Überwachung) aus, um mit der Überwachung der Datenbankaktivität zu beginnen.

So greifen Sie über die Kinesis-Konsole auf einen Aktivitätsstream von Kinesis zu

1. Öffnen Sie die Kinesis-Konsole unter.https://console.aws.amazon.com/kinesis.
2. Wählen Sie Ihren Aktivitäts-Stream aus der Liste der Kinesis-Streams aus.

Der Name eines Aktivitäts-Streams besteht aus dem Präfix `aws-rds-das-db-` gefolgt von der Ressourcen-ID der Datenbank. Im Folgenden wird ein Beispiel gezeigt.

```
aws-rds-das-db-NHVOV4PCLWHGF52NP
```

Um die RDS Amazon-Konsole zu verwenden, um die Ressourcen-ID für und die Datenbank zu finden, wählen Sie Ihre aus der Liste der Datenbanken aus und wählen Sie dann den Tab Konfiguration.

AWS CLI Um den vollständigen Kinesis-Stream-Namen für einen Aktivitätsstream zu ermitteln, verwenden Sie eine CLIAnfrage und notieren Sie sich den Wert von `ActivityStreamKinesisStreamName` in der Antwort.

3. Wählen Sie Monitoring (Überwachung) aus, um mit der Überwachung der Datenbankaktivität zu beginnen.

Weitere Informationen zur Verwendung von Amazon Kinesis finden Sie unter Was sind Amazon Kinesis Data Streams?.

Inhalt des Auditprotokolls und Beispiele für Datenbankaktivitätsströme

Überwachte Ereignisse werden im Datenbankaktivitätsstream als JSON-Zeichenfolgen dargestellt. Die Struktur besteht aus einem JSON-Objekt mit einem `DatabaseActivityMonitoringRecord`, der wiederum ein Array von Aktivitätsereignissen `databaseActivityEventList` enthält.

> **Note**
>
> Für Datenbank-Aktivitätsstreams enthält das `paramList` JSON-Array keine Nullwerte aus Hibernate-Anwendungen.

Themen

- Prüfungsprotokollbeispiele für Aktivitäts-Streams
- DatabaseActivityMonitoringRecordsJSON-Objekt
- databaseActivityEvents JSON-Objekt

Prüfungsprotokollbeispiele für Aktivitäts-Streams

Im Folgenden sehen Sie Beispiele für entschlüsselte JSON-Prüfprotokolle von Aktivitätsereignisdatensätzen.

Example Aktivitätsereignisdatensatz einer CONNECT SQL-Anweisung

Im Folgenden sehen Sie einen Aktivitätsereignisdatensatz einer Anmeldung unter Verwendung einer `CONNECT`-SQL-Anweisung (`command`) durch einen JDBC-Thin-Client (`clientApplication`) für Ihre Oracle-DB.


```
{
    "class": "Standard",
    "clientApplication": "JDBC Thin Client",
    "command": "LOGON",
    "commandText": null,
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:15:36.233787",
    "netProtocol": "tcp",
    "objectName": null,
    "objectType": null,
    "paramList": [],
    "pid": 17904,
    "remoteHost": "123.456.789.012",
    "remotePort": "25440",
    "rowCount": null,
    "serverHost": "987.654.321.098",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 987654321,
    "startTime": null,
    "statementId": 1,
    "substatementId": null,
    "transactionId": "0000000000000000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
```

```
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SESSION",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DBID": 123456789
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((ADDRESS
\u003d(PROTOCOL\u003dtcp)(HOST\u003d205.251.233.183)(PORT\u003d25440))));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "a1b2c3d4e5f6.amazon.com",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
```

```
        "TERMINAL": "unknown",
        "OS_USERNAME": "sumepate",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 1,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5124715
    }
}
```

Der folgende Aktivitätsereignisdatensatz zeigt einen Anmeldefehler für Ihre SQL-Server-DB.

```
{
    "type": "DatabaseActivityMonitoringRecord",
```

```
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "LOGIN",
            "clientApplication": "Microsoft SQL Server Management Studio",
            "command": "LOGIN FAILED",
            "commandText": "Login failed for user 'test'. Reason: Password did not
 match that for the login provided. [CLIENT: local-machine]",
            "databaseName": "",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 0,
            "logTime": "2022-10-06 21:34:42.7113072+00",
            "netProtocol": null,
            "objectName": "",
            "objectType": "LOGIN",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 0,
            "startTime": null,
            "statementId": "0x1eb0d1808d34a94b9d3dcf5432750f02",
            "substatementId": 1,
            "transactionId": "0",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 0,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "",
                "target_server_principal_name": "",
                "schema_name": "",
```

```
                "is_column_permission": false,
                "object_id": 0,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "<action_info "xmlns=\"http://
schemas.microsoft.com/sqlserver/2008/sqlaudit_data\"><pooled_connection>0</
pooled_connection><error>0x00004818</error><state>8</state><address>local machine</
address><PasswordFirstNibbleHash>B</PasswordFirstNibbleHash></action_info>"-->,
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "",
                "connection_id": "98B4F537-0F82-49E3-AB08-B9D33B5893EF",
                "audit_schema_version": 1,
                "database_principal_id": 0,
                "server_principal_sid": null,
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```

> **Note**
>
> Wenn ein Datenbankaktivitätsstream nicht aktiviert ist, ist das letzte Feld im JSON-Dokument `"engineNativeAuditFields": { }`.

Example Aktivitätsereignisdatensatz einer CREATE TABLE-Anweisung

Im Folgenden sehen Sie ein Beispiel eines `CREATE TABLE`-Ereignisses für Ihre Oracle-Datenbank.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "CREATE TABLE",
    "commandText": "CREATE TABLE persons(\n    person_id NUMBER GENERATED BY DEFAULT AS
 IDENTITY,\n    first_name VARCHAR2(50) NOT NULL,\n    last_name VARCHAR2(50) NOT NULL,
\n    PRIMARY KEY(person_id)\n)",
    "dbid": "0123456789",
    "databaseName": "ORCL",
```

```
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:22:49.535239",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.01",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1234567890,
    "startTime": null,
    "statementId": 43,
    "substatementId": null,
    "transactionId": "090011007F0D0000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SEQUENCE, CREATE TABLE",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
```

```
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-10-13-0-122",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
```

```
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 12,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5133083
    }
}
```

Das folgende Beispiel zeigt ein `CREATE TABLE`-Ereignis für Ihre SQL-Server-Datenbank.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "SCHEMA",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "ALTER",
            "commandText": "Create table [testDB].[dbo].[TestTable2](\r\ntextA
 varchar(6000),\r\n    textB varchar(6000)\r\n)",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:44:38.4120677+00",
            "netProtocol": null,
            "objectName": "dbo",
            "objectType": "SCHEMA",
            "paramList": null,
```

```
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 84,
            "startTime": null,
            "statementId": "0x5178d33d56e95e419558b9607158a5bd",
            "substatementId": 1,
            "transactionId": "4561864",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 1,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "EE1FE3FD-EF2C-41FD-AF45-9051E0CD983A",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
"0x01050000000000051500000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
```

```
}
```

Example Aktivitätsereignisdatensatz einer SELECT-Anweisung

Das folgende Beispiel zeigt ein SELECT-Ereignis für Ihre Oracle-DB.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "SELECT",
    "commandText": "select count(*) from persons",
    "databaseName": "1234567890",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:25:18.850375",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.09",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1080639707,
    "startTime": null,
    "statementId": 44,
    "substatementId": null,
    "transactionId": null,
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": null,
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
```

```
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-12-34-5-678",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
```

```
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 13,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5136972
    }
}
```

Das folgende Beispiel zeigt ein SELECT-Ereignis für Ihre SQL-Server-DB.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "TABLE",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "SELECT",
            "commandText": "select * from [testDB].[dbo].[TestTable]",
```

```
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:24:59.9422268+00",
            "netProtocol": null,
            "objectName": "TestTable",
            "objectType": "TABLE",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 62,
            "startTime": null,
            "statementId": "0x03baed90412f564fad640ebe51f89b99",
            "substatementId": 1,
            "transactionId": "4532935",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "dbo",
                "is_column_permission": true,
                "object_id": 581577110,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000001",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
```

```
                "connection_id": "AD3A5084-FB83-45C1-8334-E923459A8109",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```

DatabaseActivityMonitoringRecordsJSON-Objekt

Die Datenbank-Aktivitätsereignisdatensätze befinden sich in einem JSON-Objekt, das die folgenden Informationen enthält.

JSON-Feld	Datentyp	Beschreibung
`type`	string	Der Typ des JSON-Datensatzes. Der Wert ist `DatabaseActivityMonitoringRecords` .
`version`	string	Die Version der Datenbank-Aktivitätsüberwachungsdatensätze. Oracle DB verwendet Version 1.3 und SQL Server verwendet Version 1.4. Diese Engine-Versionen führen das `engineNativeAuditFields` -JSON-Objekt ein.
databaseActivityEvents	Zeichenfolge	Ein JSON-Objekt, das die Aktivitätsereignisse enthält.
Schlüssel	Zeichenfolge	Ein Verschlüsselungsschlüssel, den Sie zum Entschlüsseln des databaseActivityEventJSON-Array auflisten verwenden

databaseActivityEvents JSON-Objekt

Das `databaseActivityEvents`-JSON-Objekt enthält die folgenden Informationen.

Felder der obersten Ebene im JSON-Datensatz

Jedes Ereignis im Prüfprotokoll wird in einen Datensatz im JSON-Format verpackt. Dieser Datensatz enthält die folgenden Felder.

type

Dieses Feld hat immer den Wert `DatabaseActivityMonitoringRecords`.

Version

Dieses Feld stellt die Version des Datenprotokolls oder des Vertrags für die Datenbankaktivität dar. Es definiert, welche Felder verfügbar sind.

databaseActivityEvents

Eine verschlüsselte Zeichenfolge, die ein oder mehrere Aktivitätsereignisse darstellt. Sie wird als Base64-Byte-Array dargestellt. Wenn Sie die Zeichenfolge entschlüsseln, ist das Ergebnis ein Datensatz im JSON-Format mit Feldern, wie in den Beispielen in diesem Abschnitt gezeigt.

Schlüssel

Der verschlüsselte Datenschlüssel, der zum Verschlüsseln der `databaseActivityEvents`-Zeichenfolge verwendet wird. Dies ist dasselbe AWS KMS key , das Sie beim Start des Datenbank-Aktivitätsstreams angegeben haben.

Im folgenden Beispiel wird das Format dieses Datensatzes gezeigt.

```
{
  "type":"DatabaseActivityMonitoringRecords",
  "version":"1.3",
  "databaseActivityEvents":"encrypted audit records",
  "key":"encrypted key"
}
```

```
          "type":"DatabaseActivityMonitoringRecords",
          "version":"1.4",
          "databaseActivityEvents":"encrypted audit records",
          "key":"encrypted key"
```

Führen Sie die folgenden Schritte aus, um den Inhalt des `databaseActivityEvents`-Feldes zu entschlüsseln:

1. Entschlüsseln Sie den Wert im JSON-Feld `key` mit dem KMS-Schlüssel, den Sie beim Starten des Datenbankaktivitätsstroms angegeben haben. Dadurch wird der Datenverschlüsselungsschlüssel im Klartext zurückgegeben.
2. Base64-dekodieren Sie den Wert im `databaseActivityEvents`-JSON-Feld, um den Verschlüsselungstext der Prüfungsnutzlast im Binärformat zu erhalten.
3. Entschlüsseln Sie den binären Verschlüsselungstext mit dem Datenverschlüsselungsschlüssel, den Sie im ersten Schritt dekodiert haben.
4. Dekomprimieren Sie die entschlüsselte Nutzlast.
   - Die verschlüsselte Nutzlast befindet sich im `databaseActivityEvents`-Feld.
   - Das `databaseActivityEventList`-Feld enthält ein Array von Prüfdatensätzen. Die `type`-Felder im Array können `record` oder sein `heartbeat`.

Der Prüfprotokoll-Aktivitätsereignisdatensatz ist ein JSON-Objekt mit folgenden Informationen.

JSON-Feld	Datentyp	Beschreibung
`type`	string	Der Typ des JSON-Datensatzes. Der Wert ist `DatabaseActivityMonitoringRecord` .
`instanceId`	string	Die Ressourcen-ID der DB-Instance. Sie dem DB-Instance-Attribut `DbiResourceId` .
databaseActivityEventJSON-Array auflisten	string	Ein Array von Aktivitätsprüfdatensätzen oder Heartbeat-Nachrichten.

databaseActivityEventJSON-Array für Datenbank-Aktivitätsstreams auflisten

Die Prüfprotokollnutzlast ist ein verschlüsseltes JSON-Array `databaseActivityEventList`. In der folgenden Tabelle sind die Felder für jedes Aktivitätsereignis im entschlüsselten Array `DatabaseActivityEventList` eines Prüfprotokolls alphabetisch aufgelistet.

Wenn die einheitliche Prüfung in der Oracle-Datenbank aktiviert ist, werden die Prüfungsdatensätze in diesem neuen Prüfungs-Trail aufgefüllt. Die Ansicht `UNIFIED_AUDIT_TRAIL` zeigt Prüfungs-Datensätze in Tabellenform an, indem die Prüfungs-Datensätze aus dem Prüfungs-Trail abgerufen werden. Wenn Sie einen Datenbankaktivitätsstream starten, wird eine Spalte in `UNIFIED_AUDIT_TRAIL` einem Feld im Array `databaseActivityEventList` zugeordnet.

> **Important**
>
> Die Ereignisstruktur kann sich ändern. Amazon RDS könnte in Zukunft neue Felder zu Aktivitätsereignissen hinzufügen. Stellen Sie bei Anwendungen, welche die JSON-Daten analysieren, sicher, dass Ihr Code unbekannte Feldnamen ignorieren oder entsprechende Aktionen durchführen kann.

databaseActivityEventListenfelder für Amazon RDS for Oracle

Die folgenden `databaseActivityEventList` Felder gelten für Amazon RDS for Oracle.

Feld	Dater	Quelle	Beschreibung
`class`	string	`AUDIT_TYPE` -Spalte in `UNIFIED_AUDIT_TRAIL`	Die Aktivitätsereignis klasse. Dies entsprich t dem `AUDIT_TYPE` - Spalte in der `UNIFIED_A UDIT_TRAIL` -Ansicht. Gültige Werte für Amazon RDS for Oracle sind: • `Standard` • `FineGrainedAudit` • `XS` • `Database Vault` • `Label Security` • `RMAN_AUDIT` • `Datapump` • `Direct path API`

Feld	Dater	Quelle	Beschreibung
			Weitere Informationen finden Sie unter UNIFIED_AUDIT_TRAIL in der Oracle-Dokumentation.
clientApplication	string	CLIENT_PROGRAM_NAME in UNIFIED_AUDIT_TRAIL	Die Anwendung, die der Client laut Meldung für die Verbindung verwendet hat. Der Client muss diese Informationen nicht angeben, der Wert kann daher Null sein. Ein Beispielwert ist JDBC Thin Client.
command	string	ACTION_NAME -Spalte in UNIFIED_AUDIT_TRAIL	Name der Aktion, die vom Benutzer ausgeführt wird. Um die vollständige Aktion zu verstehen, lesen Sie sowohl den Befehlsnamen als auch den AUDIT_TYPE -Wert. Ein Beispielwert ist ALTER DATABASE.
commandText	string	SQL_TEXT-Spalte in UNIFIED_AUDIT_TRAIL	Die dem Ereignis zugeordnete SQL-Anweisung. Ein Beispielwert ist ALTER DATABASE BEGIN BACKUP.
databaseName	string	NAME-Spalte in V$DATABASE	Name der Datenbank.

Feld	Dater	Quelle	Beschreibung
`dbid`	Zahl	`DBID`-Spalte in `UNIFIED_AUDIT_TRAIL`	Numerische ID für die Datenbank. Ein Beispielwert ist `1559204751` .
`dbProtocol`	string	–	Das Datenbankprotokoll . In dieser Beta ist der Wert `oracle`.
`dbUserName`	string	`DBUSERNAME` -Spalte in `UNIFIED_AUDIT_TRAIL`	Name des Datenbankbenutzers, dessen Aktionen überwacht wurden. Ein Beispielwert ist `RDSADMIN`.
`endTime`	string	–	Dieses Feld wird für RDS for Oracle nicht verwendet und ist immer Null.

Feld	Dater	Quelle	Beschreibung

`engineNativeAuditFields`	Objek	`UNIFIED_AUDIT_TRAIL`	Standardmäßig ist dieses Objekt leer. Wenn Sie den Aktivitäts-Stream mit der Option `--engine-native-audit-fields-included` starten, enthält dieses Objekt die folgenden Spalten und deren Werte:
			`ADDITIONAL_INFO`
			`APPLICATION_CONTEXTS`
			`AUDIT_OPTION`
			`AUTHENTICATION_TYPE`
			`CLIENT_IDENTIFIER`
			`CURRENT_USER`
			`DBLINK_INFO`
			`DBPROXY_USERNAME`
			`DIRECT_PATH_NUM_COLUMNS_LOADED`
			`DP_BOOLEAN_PARAMETERS1`
			`DP_TEXT_PARAMETERS1`
			`DV_ACTION_CODE`
			`DV_ACTION_NAME`
			`DV_ACTION_OBJECT_NAME`
			`DV_COMMENT`
			`DV_EXTENDED_ACTION_CODE`
			`DV_FACTOR_CONTEXT`
			`DV_GRANTEE`
			`DV_OBJECT_STATUS`
			`DV_RETURN_CODE`
			`DV_RULE_SET_NAME`
			`ENTRY_ID`

Feld	Dater	Quelle	Beschreibung
			EXCLUDED_OBJECT EXCLUDED_SCHEMA EXCLUDED_USER EXECUTION_ID EXTERNAL_USERID FGA_POLICY_NAME GLOBAL_USERID INSTANCE_ID KSACL_SER VICE_NAME KSACL_SOURCE_LOCA TION KSACL_USER_NAME NEW_NAME NEW_SCHEMA OBJECT_EDITION OBJECT_PRIVILEGES OLS_GRANTEE OLS_LABEL_COM PONENT_NAME OLS_LABEL_COMPO NENT_TYPE OLS_MAX_READ_LABEL OLS_MAX_WRITE_LA BEL OLS_MIN_WRITE_ LABEL OLS_NEW_VALUE OLS_OLD_VALUE OLS_PARENT_ GROUP_NAME OLS_POLICY_NAME OLS_PRIVILEGES_GRA NTED OLS_PRIVILEGE S_USED OLS_PROGRAM _UNIT_NAME OLS_STRING_LABEL OS_USERNAME PROTOCOL_ACTIO N_NAME

Feld	Dater	Quelle	Beschreibung
			PROTOCOL_MESSAGE PROTOCOL_RET URN_CODE PROTOCOL_SESSION_I D PROTOCOL_USERHOST PROXY_SESSIONID RLS_INFO RMAN_DEVICE_TYPE RMAN_OBJECT_TYPE RMAN_OPERATION RMAN_SESSION_RECID RMAN_SESSION_STAMP ROLE SCN SYSTEM_PRIVILEGE SYSTEM_PRIVIL EGE_USED TARGET_USER TERMINAL UNIFIED_AUDIT_P OLICIES USERHOST XS_CALLBAC K_EVENT_TYPE XS_COOKIE XS_DATASEC_PO LICY_NAME XS_ENABLED_ROLE XS_ENTITY_TYPE XS_INACTIVITY _TIMEOUT XS_NS_ATTRIBUTE XS_NS_ATTRI BUTE_NEW_VAL XS_NS_ATTRIBUT E_OLD_VAL XS_NS_NAME XS_PACKAGE_NAME XS_PROCEDURE_NAME XS_PROXY_USER_NAME XS_SCHEMA_NAME

Feld	Dater	Quelle	Beschreibung

			`XS_SESSIONID`
			`XS_TARGET_PRINC IPAL_NAME`
			`XS_USER_NAME`
			
			Weitere Informati onen finden Sie unter UNIFIED_AUDIT_TRAIL in der Oracle-Datenbank-okumentation.

`errorMessage`	string	–	Dieses Feld wird für RDS for Oracle nicht verwendet und ist immer Null.
`exitCode`	Zahl	`RETURN_CODE` -Spalte in `UNIFIED_AUDIT_TRAIL`	Fehlercode der Oracle-Datenbank, der von der Aktion generiert wurde. Wenn die Aktion erfolgrei ch war, lautet der Wert `0`.
`logTime`	string	`EVENT_TIMESTAMP_UT C` -Spalte in `UNIFIED_A UDIT_TRAIL`	Zeitstempel der Erstellun g des Prüfungs-Trail-Eintrags. Ein Beispielw ert ist `2020-11-27 06:56:14.981404` .
`netProtocol`	string	`AUTHENTICATION_TYP E` -Spalte in `UNIFIED_A UDIT_TRAIL`	Das Netzwerkk ommunikationsprotokoll. Ein Beispielwert ist `TCP`.

Feld	Dater	Quelle	Beschreibung
objectName	string	OBJECT_NAME -Spalte in UNIFIED_AUDIT_TRAIL	Der Name des Objekts, das von der Aktion betroffen ist. Ein Beispielwert ist employees .
objectType	string	OBJECT_SCHEMA -Spalte in UNIFIED_AUDIT_TRAIL	Der Schemaname des von der Aktion betroffen en Objekts. Ein Beispielw ert ist hr.
paramList	table	SQL_BINDS -Spalte in UNIFIED_AUDIT_TRAIL	Die Liste der Bindungsv ariablen, falls vorhanden, die SQL_TEXT zugeordne t sind. Ein Beispielwert ist parameter_1,parame ter_2 .
pid	Zahl	OS_PROCESS -Spalte in UNIFIED_AUDIT_TRAIL	Betriebssystem-Pro zesskennung des Oracle-Datenbankprozesses Ein Beispielwert ist 22396.
remoteHost	string	AUTHENTICATION_TYP E -Spalte in UNIFIED_A UDIT_TRAIL	Entweder die Client-IP-Adresse oder der Name des Hosts, von dem die Sitzung ausgelöst wurde. Ein Beispielwert ist 123.456.789.123 .
remotePort	string	AUTHENTICATION_TYP E -Spalte in UNIFIED_A UDIT_TRAIL	Die Portnummer des Clients. Ein typischer Wert in Oracle-Da tenbank-Umgebungen ist 1521.

Feld	Dater	Quelle	Beschreibung
`rowCount`	Zahl	–	Dieses Feld wird für RDS for Oracle nicht verwendet und ist immer Null.
`serverHost`	string	Datenbank-Host	Die IP-Adresse des Datenbankserverhos ts. Ein Beispielwert ist `123.456.789.123` .
`serverType`	string	–	Der Datenbankservertyp . Dieser Wert ist immer `ORACLE`.
`serverVersion`	string	Datenbank-Host	Die Version von Amazon RDS for Oracle, Release Update (RU) und Release Update Revision (RUR). Ein Beispielwert ist `19.0.0.0.ru-2020-0 1.rur-202 0-01.r1.EE.3` .
`serviceName`	string	Datenbank-Host	Name des Service. Ein Beispielwert ist `oracle-ee` .
`sessionId`	Zahl	`SESSIONID` -Spalte in `UNIFIED_AUDIT_TRAIL`	Die Sitzungskennung der Prüfung. Ein Beispiel ist `1894327130` .
`startTime`	string	–	Dieses Feld wird für RDS for Oracle nicht verwendet und ist immer Null.

Feld	Dater	Quelle	Beschreibung
`statementId`	Zahl	`STATEMENT_ID` -Spalte in `UNIFIED_AUDIT_TRAIL`	Numerische ID für jeden Anweisungslauf. Eine Anweisung kann viele Aktionen verursach en. Ein Beispielwert ist `142197`.
`substatementId`	–	–	Dieses Feld wird für RDS for Oracle nicht verwendet und ist immer Null.
`transactionId`	string	`TRANSACTION_ID` -Spalte in `UNIFIED_AUDIT_TRAIL`	Die ID der Transaktion, in der das Objekt geändert wird. Ein Beispielwert ist `02000800D5030000` .

databaseActivityEventListenfelder für Amazon RDS for SQL Server

Die folgenden `databaseActivityEventList` Felder gelten für Amazon RDS for SQL Server.

Feld	Datenty	Quelle	Beschreibung
`class`	Zeicher lge	`sys.fn_get_audit_file.class _type` zugeordnet zu `sys.dm_au dit_class_type_map .class_type_desc`	Die Aktivitätsereignisklasse. Weitere Informationen finden Sie unter SQL-Server-Audit (Datenbank-Engine) in der Microsoft SQL Server-Do kumentation.
`clientApp lication`	Zeicher lge	`sys.fn_get_audit_file.appli cation_name`	Die Anwendung, mit der der Client eine Verbindung herstellt , wie vom Client gemeldet (SQL-Server-Version 14 und höher).

Feld	Datenty	Quelle	Beschreibung
			Dieses Feld ist in SQL-Server-Version 13 Null.
command	Zeicher lge	sys.fn_get_audit_file.actio n_id zugeordnet zu sys.dm_au dit_actions.name	Die allgemeine Kategorie der SQL-Anweisung. Die Werte für dieses Feld hängen vom Wert der Klasse ab.
commandTe xt	Zeicher lge	sys.fn_get_audit_file.state ment	Dieses Feld zeigt die SQL-Anwei sung an.
databaseN ame	Zeicher lge	sys.fn_get_audit_file.datab ase_name	Name der Datenbank.
dbProtoco l	Zeicher lge	–	Das Datenbankprotokoll. Dieser Wert ist SQLSERVER .
dbUserNam e	Zeicher lge	sys.fn_get_audit_file.serve r_principal_name	Der Datenbankbenutzer für die Client-Authentifizierung.
endTime	Zeicher lge	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.
engineNat iveAuditF ields	object	Jedes Feld in sys.fn_ge t_audit_file , das in dieser Spalte nicht aufgeführt ist.	Standardmäßig ist dieses Objekt leer. Wenn Sie den Aktivität sstream mit der --engine-native-audit-fields-included -Option starten, enthält dieses Objekt weitere systemeigene Engine-Audit-Felde r, die JSON-Map nicht zurückgibt.
errorMess age	Zeicher lge	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.

Feld	Datenty	Quelle	Beschreibung
exitCode	Ganzza	sys.fn_get_audit_file.succeeded	Gibt an, ob die Aktion, die das Ereignis ausgelöst hat, erfolgreich war. Dieses Feld darf nicht Null sein. Für alle Ereignisse außer Anmeldeereignissen gibt dieses Feld an, ob die Berechtigungsprüfung erfolgreich war oder fehlgeschlagen ist, nicht jedoch, ob der Vorgang erfolgreich war oder fehlgeschlagen ist. Gültige Werte sind: • 0 – Fehlschlag • 1 – Erfolg
logTime	Zeicher lge	sys.fn_get_audit_file.event_time	Der Ereigniszeitstempel, der vom SQL Server aufgezeichnet wird.
netProtocol	Zeicher lge	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.
objectNam e	Zeicher lge	sys.fn_get_audit_file.object_name	Der Name des Datenbankobjekts, wenn die SQL-Anweisung für ein Objekt ausgeführt wird.
objectTyp e	Zeicher lge	sys.fn_get_audit_file.class_type zugeordnet zu sys.dm_audit_class_type_map .class_type_desc	Der Name des Datenbankobjekts, wenn die SQL-Anweisung für einen Objekttyp ausgeführt wird.
paramList	Zeicher lge	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.

Feld	Datenty	Quelle	Beschreibung
`pid`	Ganzza	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.
`remoteHost`	Zeicher lge	`sys.fn_get_audit_file.client_ip`	Die IP-Adresse oder der Hostname des Clients, der die SQL-Anweisung ausgegeben hat (SQL-Server-Version 14 und höher). Dieses Feld ist in SQL-Server-Version 13 Null.
`remotePort`	Ganzza	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.
`rowCount`	Ganzza	`sys.fn_get_audit_file.affected_rows`	Die Anzahl der von der SQL-Anweisung betroffenen Tabellenzeilen (SQL-Server-Version 14 und höher). Dieses Feld ist in SQL-Server-Version 13 enthalten .
`serverHost`	Zeicher lge	Datenbank-Host	Die IP-Adresse des Host-Datenbankservers.
`serverType`	Zeicher lge	–	Der Datenbankservertyp. Der Wert ist `SQLSERVER` .
`serverVersion`	Zeicher lge	Datenbank-Host	Die Datenbankserverversion, z. B. 15.00.4073.23.v1.R1 für SQL Server 2017.
`serviceName`	Zeicher lge	Datenbank-Host	Der Name des -Dienstes. Ein Beispielwert ist `sqlserver-ee`.

Feld	Datenty	Quelle	Beschreibung
`sessionId`	Ganzza	`sys.fn_get_audit_file.session_id`	Eindeutige Kennung für die Sitzung.
`startTime`	Zeicher lge	N/A	Dieses Feld wird von Amazon RDS für SQL Server nicht verwendet und der Wert ist Null.
`statementId`	Zeicher lge	`sys.fn_get_audit_file.sequence_group_id`	Eine eindeutige Kennung für die SQL-Anweisung des Clients. Die Kennung ist für jedes generierte Ereignis unterschiedlich. Ein Beispielwert ist `0x38eaf4156267184094bb82071aaab644` .
`substatementId`	Ganzza	`sys.fn_get_audit_file.sequence_number`	Eine Kennung zur Bestimmung der Sequenznummer für eine Anweisung. Diese Kennung hilft, wenn große Datensätze in mehrere Datensätze aufgeteilt werden.
`transactionId`	Ganzza	`sys.fn_get_audit_file.transaction_id`	Eine Kennung für eine Transaktion. Wenn es keine aktiven Transaktionen gibt, ist der Wert Null.
`type`	Zeicher lge	Generierter Datenbankaktivitätsstream	Der Ereignistyp. Die Werte sind `record` oder `heartbeat` .

Verarbeitung eines Datenbankaktivitätsstroms mit dem AWS SDK

Sie können einen Aktivitätsstream programmgesteuert verarbeiten, indem Sie den verwenden. AWS SDK Im Folgenden sehen Sie vollständig funktionsfähige Java- und Python-Beispiele für die Verwendung von Datensätzen zu Datenbank-Aktivitätsstreams für die Instance-basierte Aktivierung.

Java

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.Security;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;
import java.util.zip.GZIPInputStream;

import javax.crypto.Cipher;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoInputStream;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;
import
 com.amazonaws.services.kinesis.clientlibrary.exceptions.InvalidStateException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ShutdownException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ThrottlingException;
import com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessor;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorCheckpointer;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorFactory;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.KinesisClientLibConfiguration;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.ShutdownReason;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker.Builder;
import com.amazonaws.services.kinesis.model.Record;
import com.amazonaws.services.kms.AWSKMS;
```

```
import com.amazonaws.services.kms.AWSKMSClientBuilder;
import com.amazonaws.services.kms.model.DecryptRequest;
import com.amazonaws.services.kms.model.DecryptResult;
import com.amazonaws.util.Base64;
import com.amazonaws.util.IOUtils;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.annotations.SerializedName;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class DemoConsumer {

    private static final String STREAM_NAME = "aws-rds-das-[instance-external-
resource-id]"; // aws-rds-das-db-ABCD123456
    private static final String APPLICATION_NAME = "AnyApplication"; //unique
 application name for dynamo table generation that holds kinesis shard tracking
    private static final String AWS_ACCESS_KEY =
 "[AWS_ACCESS_KEY_TO_ACCESS_KINESIS]";
    private static final String AWS_SECRET_KEY =
 "[AWS_SECRET_KEY_TO_ACCESS_KINESIS]";
    private static final String RESOURCE_ID = "[external-resource-id]"; // db-
ABCD123456
    private static final String REGION_NAME = "[region-name]"; //us-east-1, us-
east-2...
    private static final BasicAWSCredentials CREDENTIALS = new
 BasicAWSCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY);
    private static final AWSStaticCredentialsProvider CREDENTIALS_PROVIDER = new
 AWSStaticCredentialsProvider(CREDENTIALS);

    private static final AwsCrypto CRYPTO = new AwsCrypto();
    private static final AWSKMS KMS = AWSKMSClientBuilder.standard()
            .withRegion(REGION_NAME)
            .withCredentials(CREDENTIALS_PROVIDER).build();

    class Activity {
        String type;
        String version;
        String databaseActivityEvents;
        String key;
    }

    class ActivityEvent {
        @SerializedName("class") String _class;
        String clientApplication;
```

```
        String command;
        String commandText;
        String databaseName;
        String dbProtocol;
        String dbUserName;
        String endTime;
        String errorMessage;
        String exitCode;
        String logTime;
        String netProtocol;
        String objectName;
        String objectType;
        List<String> paramList;
        String pid;
        String remoteHost;
        String remotePort;
        String rowCount;
        String serverHost;
        String serverType;
        String serverVersion;
        String serviceName;
        String sessionId;
        String startTime;
        String statementId;
        String substatementId;
        String transactionId;
        String type;
    }

    class ActivityRecords {
        String type;
        String clusterId; // note that clusterId will contain an empty string on RDS
Oracle and RDS SQL Server
        String instanceId;
        List<ActivityEvent> databaseActivityEventList;
    }

    static class RecordProcessorFactory implements IRecordProcessorFactory {
        @Override
        public IRecordProcessor createProcessor() {
            return new RecordProcessor();
        }
    }
```

```
   static class RecordProcessor implements IRecordProcessor {

       private static final long BACKOFF_TIME_IN_MILLIS = 3000L;
       private static final int PROCESSING_RETRIES_MAX = 10;
       private static final long CHECKPOINT_INTERVAL_MILLIS = 60000L;
       private static final Gson GSON = new
GsonBuilder().serializeNulls().create();

       private static final Cipher CIPHER;
       static {
           Security.insertProviderAt(new BouncyCastleProvider(), 1);
           try {
               CIPHER = Cipher.getInstance("AES/GCM/NoPadding", "BC");
           } catch (NoSuchAlgorithmException | NoSuchPaddingException |
NoSuchProviderException e) {
               throw new ExceptionInInitializerError(e);
           }
       }

       private long nextCheckpointTimeInMillis;

       @Override
       public void initialize(String shardId) {
       }

       @Override
       public void processRecords(final List<Record> records, final
IRecordProcessorCheckpointer checkpointer) {
           for (final Record record : records) {
               processSingleBlob(record.getData());
           }

           if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
               checkpoint(checkpointer);
               nextCheckpointTimeInMillis = System.currentTimeMillis() +
CHECKPOINT_INTERVAL_MILLIS;
           }
       }

       @Override
       public void shutdown(IRecordProcessorCheckpointer checkpointer,
ShutdownReason reason) {
           if (reason == ShutdownReason.TERMINATE) {
               checkpoint(checkpointer);
```

```
            }
        }

        private void processSingleBlob(final ByteBuffer bytes) {
            try {
                // JSON $Activity
                final Activity activity = GSON.fromJson(new String(bytes.array(),
StandardCharsets.UTF_8), Activity.class);

                // Base64.Decode
                final byte[] decoded =
Base64.decode(activity.databaseActivityEvents);
                final byte[] decodedDataKey = Base64.decode(activity.key);

                Map<String, String> context = new HashMap<>();
                context.put("aws:rds:db-id", RESOURCE_ID);

                // Decrypt
                final DecryptRequest decryptRequest = new DecryptRequest()

.withCiphertextBlob(ByteBuffer.wrap(decodedDataKey)).withEncryptionContext(context);
                final DecryptResult decryptResult = KMS.decrypt(decryptRequest);
                final byte[] decrypted = decrypt(decoded,
getByteArray(decryptResult.getPlaintext()));

                // GZip Decompress
                final byte[] decompressed = decompress(decrypted);
                // JSON $ActivityRecords
                final ActivityRecords activityRecords = GSON.fromJson(new
String(decompressed, StandardCharsets.UTF_8), ActivityRecords.class);

                // Iterate throught $ActivityEvents
                for (final ActivityEvent event :
activityRecords.databaseActivityEventList) {
                    System.out.println(GSON.toJson(event));
                }
            } catch (Exception e) {
                // Handle error.
                e.printStackTrace();
            }
        }

        private static byte[] decompress(final byte[] src) throws IOException {
```

```
            ByteArrayInputStream byteArrayInputStream = new
ByteArrayInputStream(src);
            GZIPInputStream gzipInputStream = new
GZIPInputStream(byteArrayInputStream);
            return IOUtils.toByteArray(gzipInputStream);
        }

        private void checkpoint(IRecordProcessorCheckpointer checkpointer) {
            for (int i = 0; i < PROCESSING_RETRIES_MAX; i++) {
                try {
                    checkpointer.checkpoint();
                    break;
                } catch (ShutdownException se) {
                    // Ignore checkpoint if the processor instance has been shutdown
(fail over).
                    System.out.println("Caught shutdown exception, skipping
checkpoint." + se);
                    break;
                } catch (ThrottlingException e) {
                    // Backoff and re-attempt checkpoint upon transient failures
                    if (i >= (PROCESSING_RETRIES_MAX - 1)) {
                        System.out.println("Checkpoint failed after " + (i + 1) +
"attempts." + e);
                        break;
                    } else {
                        System.out.println("Transient issue when checkpointing -
attempt " + (i + 1) + " of " + PROCESSING_RETRIES_MAX + e);
                    }
                } catch (InvalidStateException e) {
                    // This indicates an issue with the DynamoDB table (check for
table, provisioned IOPS).
                    System.out.println("Cannot save checkpoint to the DynamoDB table
used by the Amazon Kinesis Client Library." + e);
                    break;
                }
                try {
                    Thread.sleep(BACKOFF_TIME_IN_MILLIS);
                } catch (InterruptedException e) {
                    System.out.println("Interrupted sleep" + e);
                }
            }
        }
    }
```

```
   private static byte[] decrypt(final byte[] decoded, final byte[] decodedDataKey)
throws IOException {
       // Create a JCE master key provider using the random key and an AES-GCM
encryption algorithm
       final JceMasterKey masterKey = JceMasterKey.getInstance(new
SecretKeySpec(decodedDataKey, "AES"),
               "BC", "DataKey", "AES/GCM/NoPadding");
       try (final CryptoInputStream<JceMasterKey> decryptingStream =
CRYPTO.createDecryptingStream(masterKey, new ByteArrayInputStream(decoded));
            final ByteArrayOutputStream out = new ByteArrayOutputStream()) {
           IOUtils.copy(decryptingStream, out);
           return out.toByteArray();
       }
   }

   public static void main(String[] args) throws Exception {
       final String workerId = InetAddress.getLocalHost().getCanonicalHostName() +
":" + UUID.randomUUID();
       final KinesisClientLibConfiguration kinesisClientLibConfiguration =
               new KinesisClientLibConfiguration(APPLICATION_NAME, STREAM_NAME,
CREDENTIALS_PROVIDER, workerId);

kinesisClientLibConfiguration.withInitialPositionInStream(InitialPositionInStream.LATEST);
       kinesisClientLibConfiguration.withRegionName(REGION_NAME);
       final Worker worker = new Builder()
               .recordProcessorFactory(new RecordProcessorFactory())
               .config(kinesisClientLibConfiguration)
               .build();

       System.out.printf("Running %s to process stream %s as worker %s...\n",
APPLICATION_NAME, STREAM_NAME, workerId);

       try {
           worker.run();
       } catch (Throwable t) {
           System.err.println("Caught throwable while processing data.");
           t.printStackTrace();
           System.exit(1);
       }
       System.exit(0);
   }

   private static byte[] getByteArray(final ByteBuffer b) {
       byte[] byteArray = new byte[b.remaining()];
```

```
        b.get(byteArray);
        return byteArray;
    }
}
```

Python

```
import base64
import json
import zlib
import aws_encryption_sdk
from aws_encryption_sdk import CommitmentPolicy
from aws_encryption_sdk.internal.crypto import WrappingKey
from aws_encryption_sdk.key_providers.raw import RawMasterKeyProvider
from aws_encryption_sdk.identifiers import WrappingAlgorithm, EncryptionKeyType
import boto3

REGION_NAME = '<region>'                    # us-east-1
RESOURCE_ID = '<external-resource-id>'      # db-ABCD123456
STREAM_NAME = 'aws-rds-das-' + RESOURCE_ID  # aws-rds-das-db-ABCD123456

enc_client =
 aws_encryption_sdk.EncryptionSDKClient(commitment_policy=CommitmentPolicy.FORBID_ENCRYPT_AL

class MyRawMasterKeyProvider(RawMasterKeyProvider):
    provider_id = "BC"

    def __new__(cls, *args, **kwargs):
        obj = super(RawMasterKeyProvider, cls).__new__(cls)
        return obj

    def __init__(self, plain_key):
        RawMasterKeyProvider.__init__(self)
        self.wrapping_key =
 WrappingKey(wrapping_algorithm=WrappingAlgorithm.AES_256_GCM_IV12_TAG16_NO_PADDING,
                                       wrapping_key=plain_key,
 wrapping_key_type=EncryptionKeyType.SYMMETRIC)

    def _get_raw_key(self, key_id):
        return self.wrapping_key


def decrypt_payload(payload, data_key):
```

```
    my_key_provider = MyRawMasterKeyProvider(data_key)
    my_key_provider.add_master_key("DataKey")
    decrypted_plaintext, header = enc_client.decrypt(
        source=payload,

 materials_manager=aws_encryption_sdk.materials_managers.default.DefaultCryptoMaterialsManag
    return decrypted_plaintext


def decrypt_decompress(payload, key):
    decrypted = decrypt_payload(payload, key)
    return zlib.decompress(decrypted, zlib.MAX_WBITS + 16)


def main():
    session = boto3.session.Session()
    kms = session.client('kms', region_name=REGION_NAME)
    kinesis = session.client('kinesis', region_name=REGION_NAME)

    response = kinesis.describe_stream(StreamName=STREAM_NAME)
    shard_iters = []
    for shard in response['StreamDescription']['Shards']:
        shard_iter_response = kinesis.get_shard_iterator(StreamName=STREAM_NAME,
 ShardId=shard['ShardId'],

 ShardIteratorType='LATEST')
        shard_iters.append(shard_iter_response['ShardIterator'])

    while len(shard_iters) > 0:
        next_shard_iters = []
        for shard_iter in shard_iters:
            response = kinesis.get_records(ShardIterator=shard_iter, Limit=10000)
            for record in response['Records']:
                record_data = record['Data']
                record_data = json.loads(record_data)
                payload_decoded =
 base64.b64decode(record_data['databaseActivityEvents'])
                data_key_decoded = base64.b64decode(record_data['key'])
                data_key_decrypt_result =
 kms.decrypt(CiphertextBlob=data_key_decoded,

 EncryptionContext={'aws:rds:db-id': RESOURCE_ID})
                print (decrypt_decompress(payload_decoded,
 data_key_decrypt_result['Plaintext']))
```

```
            if 'NextShardIterator' in response:
                next_shard_iters.append(response['NextShardIterator'])
        shard_iters = next_shard_iters


if __name__ == '__main__':
    main()
```

Beispiele für IAM-Richtlinien für Datenbankaktivitätsstreams

Jeder Benutzer mit entsprechenden AWS Identity and Access Management (IAM-) Rollenberechtigungen für Datenbank-Aktivitätsstreams kann die Activity Stream-Einstellungen für eine erstellen, starten, beenden und ändern. Diese Aktionen sind im Prüfprotokoll des Streams enthalten. Um DBAs die Einhaltung von Vorschriften zu gewährleisten, empfehlen wir, dass Sie diese Rechte nicht gewähren.

Der Zugriff auf Datenbankaktivitäts-Streams wird mithilfe von IAM-Richtlinien festgelegt. Weitere Informationen zur Amazon RDS-Authentifizierung finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS. Weitere Informationen zum Erstellen von IAM-Richtlinien finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff.

Example Richtlinie zum Zulassen der Konfiguration von Datenbankaktivitäts-Streams

Um Benutzern einen präzisen Zugriff auf die Änderung von Aktivitäts-Streams zu ermöglichen, verwenden Sie die servicespezifischen Operationskontextschlüssel `rds:StartActivityStream` und `rds:StopActivityStream` in einer IAM-Richtlinie. Das folgende IAM-Richtlinienbeispiel ermöglicht es einem Benutzer oder einer Rolle, Aktivitäts-Streams zu konfigurieren.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConfigureActivityStreams",
            "Effect": "Allow",
            "Action": [
                "rds:StartActivityStream",
                "rds:StopActivityStream"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

Example Richtlinie zum Zulassen des Startens von Datenbankaktivitäts-Streams

Das folgende IAM-Richtlinienbeispiel ermöglicht es einem Benutzer oder einer Rolle, Aktivitäts-Streams zu starten.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStartActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
    ]
}
```

Example Richtlinie zum Zulassen des Anhaltens von Datenbankaktivitäts-Streams

Das folgende IAM-Richtlinienbeispiel ermöglicht es einem Benutzer oder einer Rolle, Aktivitäts-Streams zu stoppen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStopActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
```

```
        }
     ]
}
```

Example Richtlinie zum Ablehnen des Startens von Datenbankaktivitäts-Streams

Im folgenden IAM-Richtlinienbeispiel wird ein Benutzer oder eine Rolle daran gehindert, Aktivitäts-Streams zu starten.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStartActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
     ]
}
```

Example Richtlinie zum Ablehnen des Anhaltens von Datenbankaktivitäts-Streams

Im folgenden IAM-Richtlinienbeispiel wird ein Benutzer oder eine Rolle daran gehindert, Aktivitäts-Streams zu stoppen.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStopActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
        }
```

```
    ]
}
```

Überwachung von Bedrohungen mit Amazon GuardDuty RDS Protection

Amazon GuardDuty ist ein Service zur Bedrohungserkennung, der Ihnen hilft, Ihre Konten, Container, Workloads und die Daten in Ihrer AWS Umgebung zu schützen. Mithilfe von Modellen für maschinelles Lernen (ML) und Funktionen zur Erkennung von Anomalien und Bedrohungen werden GuardDuty kontinuierlich verschiedene Protokollquellen und Laufzeitaktivitäten überwacht, um potenzielle Sicherheitsrisiken und böswillige Aktivitäten in Ihrer Umgebung zu identifizieren und zu priorisieren.

GuardDuty RDS Protection analysiert Anmeldeereignisse und erstellt Profile für potenzielle Zugriffsbedrohungen auf Ihre Amazon RDS-Datenbanken. Wenn Sie RDS-Schutz aktivieren, verarbeitet GuardDuty RDS-Anmeldeereignisse aus Ihren RDS-Datenbanken. RDS Protection überwacht diese Ereignisse und erstellt ein Profil für potenzielle interne Bedrohungen oder externe Akteure.

Weitere Informationen zur Aktivierung von GuardDuty RDS Protection finden Sie unter GuardDuty RDS Protection im GuardDuty Amazon-Benutzerhandbuch.

Wenn RDS Protection eine potenzielle Bedrohung erkennt, z. B. ein ungewöhnliches Muster bei erfolgreichen oder fehlgeschlagenen Anmeldeversuchen, GuardDuty generiert RDS Protection ein neues Ergebnis mit Details über die potenziell gefährdete Datenbank. Sie können die Details zu den Ergebnissen im Abschnitt Zusammenfassung der Ergebnisse in der GuardDuty Amazon-Konsole einsehen. Die Ergebnisdetails variieren je nach Ergebnistyp. Die primären Details, Ressourcentyp und Ressourcenrolle, bestimmen, welche Art von Informationen für jedes einzelne Ergebnis verfügbar ist. Weitere Informationen zu den allgemein verfügbaren Details zu Ergebnissen und den Findetypen finden Sie unter Finding details bzw. GuardDuty RDS Protection Finding Types im GuardDuty Amazon-Benutzerhandbuch.

Sie können die RDS-Schutzfunktion für jedes Gerät an jedem Ort ein- oder ausschalten, AWS-Konto in AWS-Region dem diese Funktion verfügbar ist. Wenn der RDS-Schutz nicht aktiviert ist, GuardDuty werden potenziell gefährdete RDS-Datenbanken nicht erkannt und es werden keine Einzelheiten zur Gefährdung bereitgestellt.

Ein vorhandenes GuardDuty Konto kann RDS Protection mit einer 30-tägigen Testphase aktivieren. Für ein neues GuardDuty Konto ist RDS Protection bereits aktiviert und in der 30-tägigen kostenlosen Testphase enthalten. Weitere Informationen finden Sie unter GuardDuty Kostenschätzung im GuardDuty Amazon-Benutzerhandbuch.

Informationen darüber, AWS-Regionen wo RDS Protection noch GuardDuty nicht unterstützt wird, finden Sie unter Verfügbarkeit regionsspezifischer Funktionen im GuardDuty Amazon-Benutzerhandbuch.

Informationen zu den von RDS Protection unterstützten finden Sie unter Unterstützte Amazon Aurora-, Amazon RDS- und Aurora Limitless-Datenbanken im GuardDuty Amazon-Benutzerhandbuch.

Amazon RDS Custom

Amazon RDS Custom automatisiert Aufgaben und Abläufe der Datenbankverwaltung. Custom erlaubt es Ihnen als Datenbankadministrator, auf Ihre Datenbankumgebung und Ihr Betriebssystem zuzugreifen und diese anzupassen. Mit RDS Custom können Sie die Anforderungen von älteren, benutzerdefinierten und verpackten Anwendungen anpassen.

Die neuesten Webinare und Blogs zu RDS Custom finden Sie unter Amazon RDS Custom-resources.

Themen

- Bewältigung der Herausforderung der Datenbankanpassung
- Managementmodell und Vorteile für Amazon RDS Custom
- Amazon RDS Custom Architektur
- Sicherheit in Amazon RDS Custom
- Arbeiten mit RDS Custom for Oracle
- Mit RDS Custom for SQL Server arbeiten

Bewältigung der Herausforderung der Datenbankanpassung

Amazon RDS Custom bringt die Vorteile von Amazon RDS in einen Markt, der aufgrund von Anpassungen, die für Anwendungen von Drittanbietern erforderlich sind, nicht einfach zu einem vollständig verwalteten Service wechseln kann. Amazon RDS Custom spart Verwaltungszeit, ist langlebig und skaliert mit Ihrem Unternehmen.

Wenn Sie möchten, dass die gesamte Datenbank und das Betriebssystem vollständig verwaltet werden AWS, empfehlen wir Amazon RDS. Wenn Sie Administratorrechte für die Datenbank und das zugrunde liegende Betriebssystem benötigen, um abhängige Anwendungen verfügbar zu machen, ist Amazon RDS Custom die bessere Wahl. Wenn Sie die volle Managementverantwortung übernehmen möchten und lediglich einen verwalteten Rechenservice benötigen, ist die beste Option die Selbstverwaltung Ihrer kommerziellen Datenbanken bei Amazon EC2.

Um eine verwaltete Service-Erfahrung zu bieten, erlaubt Amazon RDS keinen Zugriff auf den zugrunde liegenden Host. Amazon RDS schränkt auch den Zugriff auf einige Systemverfahren und Tabellen ein, die erweiterte Berechtigungen erfordern. Für einige Anwendungen müssen Sie jedoch möglicherweise Vorgänge als privilegierter Betriebssystembenutzer ausführen.

Beispielsweise könnte es sein, dass Sie eine der folgenden Aufgaben ausführen müssen:

- Installieren Sie benutzerdefinierte Datenbank- und Betriebssystem-Patches und -Pakete.
- Konfigurieren Sie bestimmte Datenbankeinstellungen.
- Konfigurieren Sie Dateisysteme, um Dateien direkt mit ihren Anwendungen freizugeben.

Bisher mussten Sie, wenn Sie Ihre Anwendung anpassen mussten, Ihre Datenbank lokal oder bei Amazon EC2 bereitstellen. In diesem Fall tragen Sie den größten Teil oder die gesamte Verantwortung für die Datenbankverwaltung, wie in der folgenden Tabelle zusammengefasst.

Funktion	Lokale Verantwortung	EC2 Verantwortung von Amazon	Amazon RDS Verantwortung
Anwendungsoptimierung	Customer	Customer	Customer
Skalierung	Customer	Customer	AWS
Hohe Verfügbarkeit	Customer	Customer	AWS
Datenbank-Backups	Customer	Customer	AWS
Patchen von Datenbanksoftware	Customer	Customer	AWS
Installieren der Datenbanksoftware	Customer	Customer	AWS
Betriebssystem-Patchen	Customer	Customer	AWS
Betriebssysteminstallation	Customer	Customer	AWS
Serverwartung	Customer	AWS	AWS
Hardware-Lebenszyklus	Customer	AWS	AWS

Funktion	Lokale Verantwortung	EC2 Verantwortung von Amazon	Amazon RDS Verantwortung
Strom, Netzwerk und Kühlung	Customer	AWS	AWS

Wenn Sie Datenbanksoftware selbst verwalten, erhalten Sie mehr Kontrolle, sind aber auch anfälliger für Benutzerfehler. Wenn Sie beispielsweise Änderungen manuell vornehmen, können Sie versehentlich Ausfallzeiten der Anwendung verursachen. Möglicherweise verbringen Sie Stunden damit, jede Änderung zu überprüfen, um ein Problem zu identifizieren und zu beheben. Idealerweise möchten Sie einen verwalteten Datenbankdienst, der allgemeine DBA-Aufgaben automatisiert, aber auch privilegierten Zugriff auf die Datenbank und das zugrunde liegende Betriebssystem unterstützt.

Managementmodell und Vorteile für Amazon RDS Custom

Amazon RDS Custom ist ein verwalteter Datenbankdienst für ältere, benutzerdefinierte und gepackte Anwendungen, die Zugriff auf das zugrunde liegende Betriebssystem und die Datenbankumgebung erfordern. RDS Custom automatisiert die Einrichtung, den Betrieb und die Skalierung von Datenbanken und gewährt Ihnen AWS Cloud gleichzeitig Zugriff auf die Datenbank und das zugrunde liegende Betriebssystem. Mit diesem Zugriff können Sie Einstellungen konfigurieren, Patches installieren und native Funktionen aktivieren, um die Anforderungen der abhängigen Anwendung zu erfüllen. Mit RDS Custom können Sie Ihren Datenbank-Workload mit dem AWS Management Console oder dem AWS CLI ausführen.

Derzeit unterstützt RDS Custom nur die Oracle-Datenbank- und Microsoft-SQL-Server-Engines.

Themen

- Modell der geteilten Verantwortung in RDS Custom
- Support-Perimeter und nicht unterstützte Konfigurationen in RDS Custom
- Hauptvorteile von RDS Custom

Modell der geteilten Verantwortung in RDS Custom

Mit RDS Custom verwenden Sie die verwalteten Funktionen von Amazon RDS, aber Sie verwalten den Host und passen das Betriebssystem an, wie Sie es in Amazon tun EC2. Sie übernehmen zusätzliche Aufgaben im Zusammenhang mit der Datenbankverwaltung, die über den Aufwand in

Amazon RDS hinausgehen. Das Ergebnis ist, dass Sie mehr Kontrolle über die Datenbank- und DB-Instance-Verwaltung haben als in Amazon RDS und dennoch von der RDS-Automatisierung profitieren.

Geteilte Verantwortung bedeutet Folgendes:

1. Sie sind für einen Teil des Prozesses verantwortlich, wenn Sie ein Feature von RDS Custom verwenden.

   In RDS Custom für Oracle steuern Sie beispielsweise, welche Oracle-Datenbank-Patches verwendet werden und wann sie auf Ihre DB-Instances angewendet werden sollen.
2. Sie sind dafür verantwortlich, sicherzustellen, dass alle Anpassungen der Features von RDS Custom ordnungsgemäß funktionieren.

   RDS Custom verfügt über eine Automatisierungssoftware, die außerhalb Ihrer DB-Instance ausgeführt wird, um Schutz vor unzulässigen Anpassungen zu bieten. Wenn Ihre zugrunde liegende EC2 Amazon-Instance beeinträchtigt wird, versucht RDS Custom, diese Probleme automatisch zu lösen, indem die Instance entweder neu gestartet oder ersetzt wird. EC2 Die einzige für die Benutzer sichtbare Änderung ist eine neue IP-Adresse. Weitere Informationen finden Sie unter Hostersatz in Amazon RDS Custom.

In der folgenden Tabelle wird das Modell der geteilten Verantwortung für verschiedene Features von RDS Custom beschrieben.

Funktion	EC2 Verantwortung von Amazon	Amazon RDS Verantwortung	RDS-Custom-for-Oracle-Verantwortung	RDS-Custom-for-SQL-Server-Verantwortung
Anwendungsoptimierung	Customer	Customer	Customer	Customer
Skalierung	Customer	AWS	Freigegeben	Freigegeben
Hohe Verfügbarkeit	Customer	AWS	AWS	AWS
Datenbank-Backups	Customer	AWS	Freigegeben	AWS

Funktion	EC2 Verantwortung von Amazon	Amazon RDS Verantwortung	RDS-Custom-for-Oracle-Verantwortung	RDS-Custom-for-SQL-Server-Verantwortung
Patchen von Datenbanksoftware	Customer	AWS	Freigegeben	AWS für RPEV, Kunde für CEV 1
Installieren der Datenbanksoftware	Customer	AWS	Freigegeben	AWS für RPEV, Kunde für CEV 1
Betriebssystem-Patchen	Customer	AWS	Customer	AWS für RPEV, Kunde für CEV 1
Betriebssysteminstallation	Customer	AWS	Freigegeben	AWS
Serverwartung	AWS	AWS	AWS	AWS
Hardware-Lebenszyklus	AWS	AWS	AWS	AWS
Strom, Netzwerk und Kühlung	AWS	AWS	AWS	AWS

[1] Eine benutzerdefinierte Engine-Version (CEV) ist ein binärer Volume-Snapshot einer Datenbankversion und eines Amazon Machine Image (AMI). Eine von RDS bereitgestellte Engine-Version (RPEV) ist die Standardinstallation von Amazon Machine Image (AMI) und Microsoft SQL Server.

Sie können eine benutzerdefinierte RDS DB-Instance mit Microsoft SQL Server erstellen. In diesem Fall.

- Sie können zwischen zwei Lizenzmodellen wählen: License Included (LI) und Bring Your Own Media (BYOM).
- Mit LI müssen Sie SQL Server-Lizenzen nicht separat erwerben. AWS besitzt die Lizenz für die SQL Server-Datenbanksoftware.

- Mit BYOM stellen Sie Ihre eigenen Microsoft SQL Server-Binärdateien und -Lizenzen bereit und installieren diese.

Sie können eine RDS Custom DB-Instance mit Oracle Database erstellen. Führen Sie in diesem Fall folgende Schritte aus:

- Verwalten Sie Ihre eigenen Medien.

  Wenn Sie RDS Custom verwenden, laden Sie Ihre eigenen Datenbankinstallationsdateien und Patches hoch. Aus diesen Dateien erstellen Sie eine benutzerdefinierte Engine-Version (CEV). Anschließend können Sie mit diesem CEV eine RDS Custom DB-Instance erstellen.
- Verwalten Sie Ihre eigenen Lizenzen.

  Sie bringen Ihre eigenen Oracle Database-Lizenzen mit und verwalten Lizenzen selbst.

Support-Perimeter und nicht unterstützte Konfigurationen in RDS Custom

RDS Custom bietet Überwachungsfunktionen, die Support-Perimeter genannt werden. Dieses Feature stellt sicher, dass Ihr Host und Ihre Datenbankumgebung korrekt konfiguriert sind. Wenn Sie eine Änderung vornehmen, die dazu führt, dass sich Ihre DB-Instance außerhalb des Support-Perimeters befindet, ändert RDS Custom den Instance-Status in `unsupported-configuration`, bis Sie die Konfigurationsprobleme manuell beheben. Weitere Informationen finden Sie unter Support-Perimeter in RDS Custom.

Hauptvorteile von RDS Custom

Indem Sie RDS Custom verwenden, können Sie folgende Aktionen ausführen:

- Automatisieren Sie viele der gleichen administrativen Aufgaben wie Amazon RDS, einschließlich der folgenden:
  - Verwaltung des Lebenszyklus von Datenbanken
  - Automatisierte Backups und point-in-time Wiederherstellung (PITR)
  - Überwachung des Zustands von RDS Custom DB-Instances und Beobachtung von Änderungen an der Infrastruktur, dem Betriebssystem und den Datenbankprozessen.
  - Benachrichtigung oder Maßnahmen zur Behebung von Problemen je nach Unterbrechung der DB-Instance
- Installieren Sie Drittanbieter-Anwendungen.

Sie können Software installieren, um benutzerdefinierte Anwendungen und Agents auszuführen. Da Sie privilegierten Zugriff auf den Host haben, können Sie Dateisysteme ändern, um ältere Anwendungen zu unterstützen.

- Installieren Sie benutzerdefinierte Patches.

  Sie können benutzerdefinierte Datenbank-Patches anwenden oder Betriebssystempakete auf Ihre RDS Custom DB-Instanzen ändern.

- Stationieren Sie eine lokale Datenbank, bevor Sie sie in einen vollständig verwalteten Dienst verschieben.

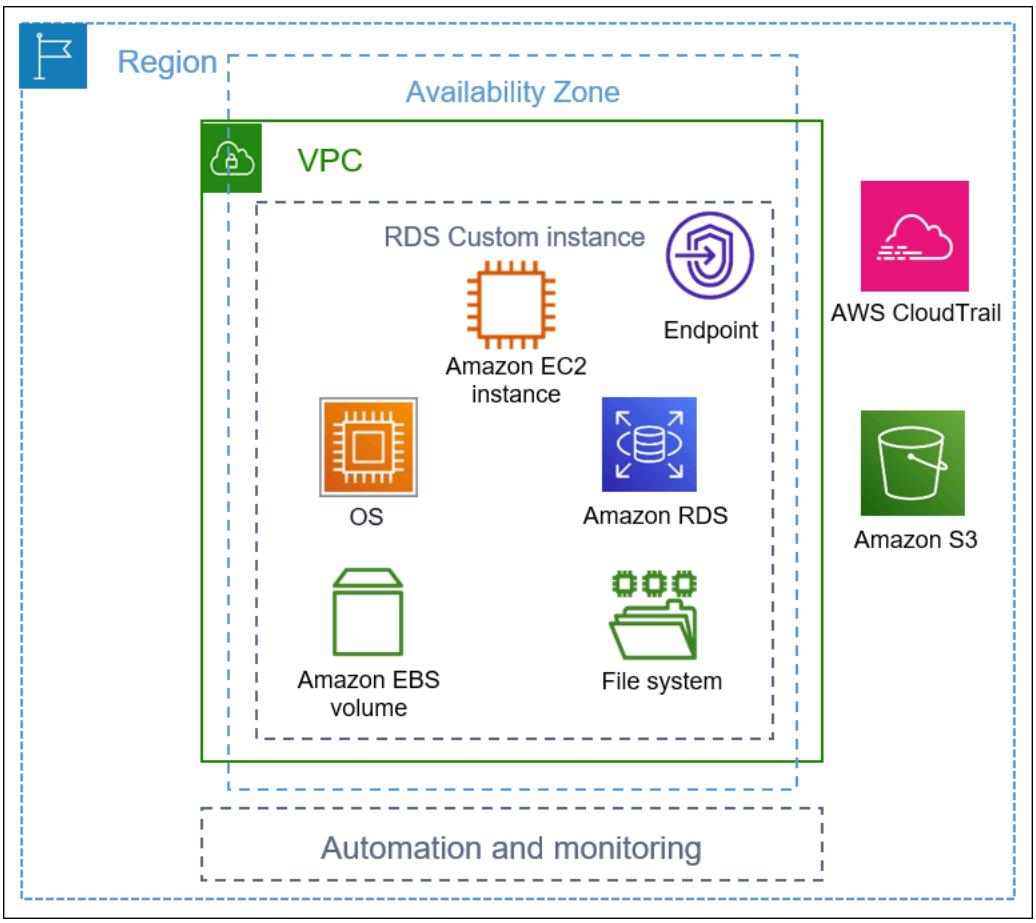
  Wenn Sie Ihre eigene lokale Datenbank verwalten, können Sie die Datenbank unverändert auf RDS Custom stellen. Nachdem Sie sich mit der Cloud-Umgebung vertraut gemacht haben, können Sie Ihre Datenbank auf eine vollständig verwaltete Amazon RDS DB-Instance migrieren.

- Erstellen Sie Ihre eigene Automatisierung.

  Sie können benutzerdefinierte Automatisierungsskripte für Reporting-, Verwaltungs- oder Diagnosetools erstellen, planen und ausführen.

Amazon RDS Custom Architektur

Die Amazon RDS Custom Architecture basiert auf Amazon RDS mit wichtigen Unterschieden. Das folgende Diagramm zeigt die Hauptkomponenten der RDS-Custom-Architektur.



Themen

- VPC
- RDS Kundenspezifische Automatisierung und Überwachung
- Amazon S3
- AWS CloudTrail

VPC

Wie bei Amazon RDS befindet sich Ihre RDS-Custom-DB-Instance in einer Virtual Private Cloud (VPC).



Die RDS-Custom-DB-Instance besteht aus den folgenden Komponenten:

- EC2 Amazon-Instanz
- Instance-Endpunkt
- Auf der EC2 Amazon-Instance installiertes Betriebssystem
- Amazon EBS-Speicher, der zusätzliche Dateisysteme enthält

RDS Kundenspezifische Automatisierung und Überwachung

RDS Custom verfügt über Automatisierungssoftware, die außerhalb der DB-Instance läuft. Diese Software kommuniziert mit Agenten auf der DB-Instance und mit anderen Komponenten innerhalb der gesamten RDS Custom Umgebung.

Die RDS Custom Überwachungs- und Wiederherstellungsfunktionen bieten ähnliche Funktionen wie Amazon RDS. Standardmäßig befindet sich RDS Custom im Vollautomatisierungsmodus. Die Automatisierungssoftware hat folgende Hauptaufgaben:

- Sammeln Sie Metriken und senden Sie Benachrichtigungen
- Automatische Instance-Wiederherstellung

Eine wichtige Aufgabe von RDS Custom Automation besteht darin, auf Probleme mit Ihrer EC2 Amazon-Instance zu reagieren. Aus verschiedenen Gründen kann der Gastgeber beeinträchtigt oder nicht erreichbar werden. RDS Custom löst diese Probleme, indem die Amazon-Instance entweder neu gestartet oder ersetzt wird. EC2

Themen

- Hostersatz in Amazon RDS Custom
- Support-Perimeter in RDS Custom

Hostersatz in Amazon RDS Custom

Wenn der EC2 Amazon-Host beeinträchtigt wird, versucht RDS Custom, ihn neu zu starten. Wenn dieser Versuch fehlschlägt, verwendet RDS Custom dieselbe Stop-and-Start-Funktion, die in Amazon enthalten ist EC2. Die einzige vom Kunden sichtbare Änderung, wenn ein Host ersetzt wird, ist eine neue öffentliche IP-Adresse.

Themen

- Stoppen und Starten des Clusters.
- Auswirkungen des Host-Ersatzes
- Bewährte Methoden für EC2 Amazon-Hosts

Stoppen und Starten des Clusters.

RDS Custom führt automatisch die folgenden Schritte aus, ohne dass ein Benutzereingriff erforderlich ist:

1. Stoppt den EC2 Amazon-Host.

   Die EC2 Instance wird normal heruntergefahren und wird nicht mehr ausgeführt. Amazon EBS-Volumes bleiben der Instance angefügt und die Daten bleiben bestehen. Alle Daten, die in den Instance-Speicher-Volumes gespeichert sind (nicht auf RDS Custom unterstützt) oder im RAM des Hostcomputers, sind verschwunden.

   Weitere Informationen finden Sie unter Stoppen und Starten Ihrer Instance im EC2 Amazon-Benutzerhandbuch.
2. Startet den EC2 Amazon-Host.

   Die EC2 Instance migriert auf eine neue zugrunde liegende Host-Hardware. In einigen Fällen bleibt die RDS Custom DB-Instanz auf dem ursprünglichen Host.

Auswirkungen des Host-Ersatzes

In RDS Custom haben Sie die volle Kontrolle über das Root-Gerätevolume und Amazon EBS-Speicher-Volumes. Das Root-Volume kann wichtige Daten und Konfigurationen enthalten, die Sie nicht verlieren möchten.

RDS Custom for Oracle behält alle Datenbank- und Kundendaten nach dem Vorgang bei, einschließlich Root-Volume-Daten. Es ist kein Benutzereingriff erforderlich. Auf RDS Custom for SQL Server werden Datenbankdaten beibehalten, aber alle Daten auf dem `C:` Laufwerk, einschließlich Betriebssystem- und Kundendaten, gehen verloren.

Nach dem Host-Austauschprozess hat der EC2 Amazon-Host eine neue öffentliche IP-Adresse. Der neue Host behält die in der folgenden Tabelle aufgeführten Daten und Metadaten bei.

Metadaten oder Daten	RDS Custom für Oracle-Aufbewahrung	RDS Custom für die Aufbewahrung von SQL Server
EC2 Instanz-ID	Ja	Nein
EC2 Instanz-Metadaten	Ja	Ja
Daten zum Datenspeichervolumen	Ja	Ja
Stammvolumendaten	Ja	Nein
Private IP-Adressen	Ja	Ja
Elastic-IP-Adressen	Ja	Ja

Bewährte Methoden für EC2 Amazon-Hosts

Die Funktion zum Austausch von EC2 Amazon-Hosts deckt die meisten Szenarien mit EC2 Beeinträchtigungen durch Amazon ab. Wir empfehlen Ihnen, die ACM bewährte Methode für zu befolgen.

- Bevor Sie Ihre Konfiguration oder das Betriebssystem ändern, sichern Sie Ihre Daten. Wenn das Root-Volume oder Betriebssystem beschädigt wird, kann der Host-Ersatz es nicht reparieren. Ihre

einzigen Optionen sind die Wiederherstellung aus einem DB-Snapshot oder eine point-in-time Wiederherstellung.

- Stoppen oder beenden Sie den physischen EC2 Amazon-Host nicht manuell. Beide Aktionen führen dazu, dass die Instanz außerhalb des RDS Custom Supportumfangs platziert wird.
- (RDS Custom for SQL Server) Wenn Sie zusätzliche Volumes an den EC2 Amazon-Host anhängen, konfigurieren Sie sie so, dass sie beim Neustart erneut bereitgestellt werden. Wenn der Host beeinträchtigt ist, stoppt RDS Custom möglicherweise und startet den Host automatisch.

Support-Perimeter in RDS Custom

RDS Custom bietet zusätzliche Überwachungsfunktionen, die Support Perimeter genannt werden. Diese zusätzliche Überwachung stellt sicher, dass Ihre RDS Custom DB-Instance eine unterstützte AWS Infrastruktur, ein Betriebssystem und eine Datenbank verwendet.

Der Support-Perimeter überprüft, ob Ihre DB-Instance die unter Fehlerbehebung bei nicht unterstützten Konfigurationen in RDS Custom für Oracle und Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server aufgeführten Anforderungen erfüllt. Wenn eine dieser Anforderungen nicht erfüllt wird, betrachtet RDS Custom die DB-Instance als außerhalb des Support-Umfangs liegend.

Themen

- Nicht unterstützte Konfigurationen in RDS Custom
- Fehlerbehebung bei nicht unterstützten Konfigurationen

Nicht unterstützte Konfigurationen in RDS Custom

Wenn sich Ihre DB-Instance außerhalb des Support-Perimeters befindet, ändert RDS Custom den Status der DB-Instance in `unsupported-configuration` und sendet Ereignisbenachrichtigungen. Nachdem Sie die Konfigurationsprobleme behoben haben, ändert RDS Custom den Status der DB-Instance wieder in `available`.

Während die DB-Instance den Status `unsupported-configuration` aufweist, ist Folgendes der Fall:

- Ihre Datenbank ist erreichbar. Eine Ausnahme besteht, wenn die DB-Instance den Status `unsupported-configuration` aufweist, weil die Datenbank unerwartet heruntergefahren wird.
- Sie können Ihre DB-Instance nicht ändern.

- Sie können keine DB-Snapshots machen.
- Es werden keine automatischen Backups erstellt.
- Nur für RDS Custom for SQL Server-DB-Instances ersetzt RDS Custom die zugrunde liegende EC2 Amazon-Instance nicht, wenn sie beeinträchtigt wird. Weitere Informationen zum Hostersatz finden Sie unter Hostersatz in Amazon RDS Custom.
- Sie können Ihre DB-Instance löschen, die meisten anderen API-Operationen von RDS Custom sind jedoch nicht verfügbar.
- RDS Custom unterstützt weiterhin point-in-time Recovery (PITR), indem Redo-Log-Dateien archiviert und auf Amazon S3 hochgeladen werden. PITR im Status `unsupported-configuration` unterscheidet sich in folgenden Punkten:
  - Es kann lange dauern, bis PITR das Zurücksetzen auf eine neue RDS-Custom-DB-Instance vollständig abgeschlossen hat. Dies liegt daran, dass Sie weder automatisierte noch manuelle Snapshots erstellen können, während die Instance den Status `unsupported-configuration` aufweist.
  - PITR muss weitere Redo-Logs wiedergeben, beginnend mit dem letzten Snapshot, der erstellt wurde, bevor die Instanz in den Zustand `unsupported-configuration` geht.
  - In einigen Fällen weist die DB-Instance den Status `unsupported-configuration` auf, weil Sie eine Änderung vorgenommen haben, die das Hochladen archivierter Redo-Protokolldateien verhindert hat. Beispiele hierfür sind das Stoppen der EC2 Instance, das Stoppen des RDS Custom-Agenten und das Trennen von EBS-Volumes. In diesen Fällen kann PITR die DB-Instance nicht auf die neueste wiederherstellbare Zeit zurücksetzen.

Fehlerbehebung bei nicht unterstützten Konfigurationen

RDS Custom bietet Anleitungen zur Fehlerbehebung für den Status `unsupported-configuration`. Auch wenn einige Anleitungen sowohl für RDS Custom für Oracle als auch für RDS Custom für SQL Server gelten, hängen andere Anleitungen von Ihrer DB-Engine ab. Engine-spezifische Informationen finden Sie in den folgenden Themen:

- Fehlerbehebung bei nicht unterstützten Konfigurationen in RDS Custom für Oracle
- Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server

Amazon S3

Wenn Sie RDS Custom for Oracle verwenden, laden Sie Installationsmedien in einen vom Benutzer erstellten Amazon-S3-Bucket hoch. RDS Custom for Oracle verwendet die Medien in diesem Bucket, um eine benutzerdefinierte Engine-Version (CEV) zu erstellen. Ein CEV ist ein binärer Volume-Snapshot einer Datenbankversion und Amazon Machine Image (AMI). Aus der CEV können Sie eine RDS-Custom-DB-Instance erstellen. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Engine-Versionen für Amazon RDS Custom für Oracle.

Sowohl für RDS Custom for Oracle als auch für RDS Custom for SQL Server erstellt RDS Custom automatisch einen Amazon-S3-Bucket, dem die Zeichenfolge `do-not-delete-rds-custom-` vorangestellt wird. RDS Custom verwendet den `do-not-delete-rds-custom-`-S3-Bucket zum Speichern der folgenden Dateitypen:

- AWS CloudTrail Logs für den von RDS Custom erstellten Trail
- Unterstützung von Perimeter-Artefakten (siehe Support-Perimeter in RDS Custom)
- Datenbank-Redo-Protokolldateien (nur RDS Custom for Oracle)
- Transaktionsprotokolle (nur RDS Custom for SQL Server)
- Benutzerdefinierte Engine-Versionsartefakte (nur RDS Custom for Oracle)

RDS Custom erstellt den `do-not-delete-rds-custom-`-S3-Bucket beim Erstellen einer der folgenden Ressourcen:

- Ihre erste CEV für RDS Custom for Oracle
- Ihre erste DB-Instance für RDS Custom for SQL Server

RDS Custom erstellt einen Bucket für jede Kombination der folgenden Optionen:

- AWS-Konto ID
- Engine-Typ (entweder RDS Custom for Oracle oder RDS Custom for SQL Server)
- AWS-Region

Wenn Sie beispielsweise RDS Custom for Oracle CEVs in einem einzigen Bucket erstellen AWS-Region, existiert ein einziger `do-not-delete-rds-custom-` Bucket. Wenn Sie mehrere RDS Custom for SQL Server-Instanzen erstellen und diese sich in unterschiedlichen Instanzen befinden AWS-Regionen, ist in jeder AWS-Region Instanz ein `do-not-delete-rds-custom-` Bucket

vorhanden. Wenn Sie eine RDS Custom for Oracle-Instanz und zwei RDS Custom for SQL Server-Instanzen in einer einzigen Instanz erstellen AWS-Region, sind zwei `do-not-delete-rds-custom-` Buckets vorhanden.

AWS CloudTrail

RDS Custom erstellt automatisch einen AWS CloudTrail Trail, dessen Name mit `do-not-delete-rds-custom-` beginnt. Der Umfang der Unterstützung von RDS Custom bestimmt anhand der Ereignisse von CloudTrail , ob sich Ihre Aktionen auf die Automatisierung von RDS Custom auswirken. Weitere Informationen finden Sie unter Fehlerbehebung bei nicht unterstützten Konfigurationen.

RDS Custom erstellt den Trail, wenn Sie Ihre erste DB-Instance erstellen. RDS Custom erstellt einen Trail für jede Kombination der folgenden Optionen:

- AWS-Konto ID
- Engine-Typ (entweder RDS Custom for Oracle oder RDS Custom for SQL Server)
- AWS-Region

Wenn Sie eine benutzerdefinierte RDS-DB-Instance löschen, wird die CloudTrail für diese Instance nicht automatisch entfernt. In diesem Fall werden Ihnen AWS-Konto weiterhin die nicht gelöschten Daten in Rechnung gestellt. CloudTrail RDS Custom ist nicht verantwortlich für das Löschen dieser Ressource. Informationen zum CloudTrail manuellen Entfernen finden Sie unter Löschen eines Pfads im AWS CloudTrail Benutzerhandbuch.

Sicherheit in Amazon RDS Custom

Machen Sie sich mit den Sicherheitsüberlegungen für RDS Custom vertraut.

Weitere Informationen zur Sicherheit von RDS Custom finden Sie in den folgenden Themen.

- Schützen Ihres Amazon-S3-Buckets vor dem Problem des verwirrten Stellvertreters
- Rotieren der Anmeldeinformationen von RDS Custom für Oracle für Compliance-Programme

So verwaltet RDS Custom Aufgaben sicher in Ihrem Namen

RDS Custom verwendet die folgenden Tools und Methoden, um Operationen in Ihrem Namen sicher auszuführen:

AWSServiceRoleForRDSCustom Mit dem Dienst verknüpfte Rolle

Eine serviceverknüpfte Rolle wird vom Service vordefiniert und schließt alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen benötigt. Bei RDS Custom ist `AWSServiceRoleForRDSCustom` eine serviceverknüpfte Rolle, die nach dem Prinzip der geringsten Berechtigung definiert ist. RDS Custom verwendet die Berechtigungen in `AmazonRDSCustomServiceRolePolicy`, d. h. die dieser Rolle zugeordnete Richtlinie, um die meisten Bereitstellungsaufgaben und alle nicht auf dem Host ausgeführten Verwaltungsaufgaben auszuführen. Weitere Informationen finden Sie auf Amazon RDSCustom ServiceRolePolicy.

Bei der Ausführung von Aufgaben auf dem Host verwendet RDS Custom Automation Anmeldeinformationen aus der mit dem Dienst verknüpften Rolle, um Befehle auszuführen mit AWS Systems Manager. Sie können den Befehlsverlauf über den Systems-Manager-Befehlsverlauf und AWS CloudTrail prüfen. Systems Manager stellt mithilfe Ihrer Netzwerkeinrichtung eine Verbindung mit Ihrer DB-Instance von RDS Custom her. Weitere Informationen finden Sie unter Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle.

Temporäre IAM-Anmeldeinformationen

Bei der Bereitstellung oder Löschung von Ressourcen verwendet RDS Custom manchmal temporäre Anmeldeinformationen, die von den Anmeldeinformationen des aufrufenden IAM-Prinzipals abgeleitet werden. Diese IAM-Anmeldeinformationen sind durch die diesem Prinzipal zugeordneten IAM-Richtlinien eingeschränkt und laufen nach Abschluss des Vorgangs ab. Weitere Informationen zu den Berechtigungen, die für IAM-Prinzipale erforderlich sind, welche

RDS Custom verwenden, finden Sie unter Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen.

EC2 Amazon-Instanzprofil

Ein EC2 Instance-Profil ist ein Container für eine IAM-Rolle, mit dem Sie Rolleninformationen an eine EC2 Instance übergeben können. Eine EC2 Instance liegt einer RDS Custom DB-Instance zugrunde. Sie geben ein Instance-Profil an, wenn Sie eine DB-Instance von RDS Custom erstellen. RDS Custom verwendet Anmeldeinformationen für das EC2 Instance-Profil, wenn es hostbasierte Verwaltungsaufgaben wie Backups ausführt. Weitere Informationen finden Sie unter Manuelles Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils.

SSH-Schlüsselpaar

Wenn RDS Custom die EC2 Instance erstellt, die einer DB-Instance zugrunde liegt, erstellt es in Ihrem Namen ein SSH-Schlüsselpaar. Der Schlüssel verwendet das Namenspräfix `do-not-delete-rds-custom-ssh-privatekey-db-` oder. `rds-custom!oracle-do-not-delete-`*`db_resource_id-uuid`*`-ssh-privatekey` AWS Secrets Manager speichert diesen privaten SSH-Schlüssel als Geheimnis in Ihrem AWS-Konto. Amazon RDS speichert diese Anmeldeinformationen nicht, greift nicht auf sie zu und verwendet sie auch nicht. Weitere Informationen finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

SSL-Zertifikate

DB-Instances von RDS unterstützen keine verwalteten SSL-Zertifikate. Wenn Sie SSL bereitstellen möchten, können Sie SSL-Zertifikate in Ihrem eigenen Wallet selbst verwalten und einen SSL-Listener erstellen, um die Verbindungen zwischen der Client-Datenbank oder für die Datenbankreplikation zu sichern. Weitere Informationen finden Sie unter Configuring Transport Layer Security Authentication (Konfiguration der Authentifizierung von Transport Layer Security) in der Oracle-Database-Dokumentation.

Schützen Ihres Amazon-S3-Buckets vor dem Problem des verwirrten Stellvertreters

Wenn Sie eine RDS benutzerdefinierte Engine-Version (CEV) von Amazon Custom for Oracle oder eine RDS Custom for SQL Server-DB-Instance erstellen, erstellt RDS Custom einen Amazon S3 S3-Bucket. Der S3-Bucket speichert Dateien wie CEV Artefakte, Redo-Logs (Transaktions-) Logs, Konfigurationselemente für den Support-Perimeter und AWS CloudTrail Logs.

Sie können diese S3-Buckets sicherer machen, indem Sie die globalen Bedingungskontextschlüssel verwenden, um „confused deputy problem“ zu verhindern. Weitere Informationen finden Sie unter Vermeidung des dienstübergreifenden Confused-Deputy-Problems.

Das folgende Beispiel für RDS Custom for Oracle zeigt die Verwendung der Kontextschlüssel `aws:SourceArn` und der `aws:SourceAccount` globalen Bedingungsschlüssel in einer S3-Bucket-Richtlinie. Stellen Sie bei RDS Custom for Oracle sicher, dass Sie die Amazon-Ressourcennamen (ARNs) für die CEVs und die DB-Instances angeben. Stellen Sie bei RDS Benutzerdefiniert für SQL Server sicher, dass Sie die ARN für die DB-Instances angeben.

```
...
{
  "Sid": "AWSRDSCustomForOracleInstancesObjectLevelAccess",
  "Effect": "Allow",
  "Principal": {
     "Service": "custom.rds.amazonaws.com"
  },
  "Action": [
     "s3:GetObject",
     "s3:GetObjectVersion",
     "s3:DeleteObject",
     "s3:DeleteObjectVersion",
     "s3:GetObjectRetention",
     "s3:BypassGovernanceRetention"
  ],
  "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-east-2-c8a6f7/
RDSCustomForOracle/Instances/*",
  "Condition": {
     "ArnLike": {
        "aws:SourceArn": [
            "arn:aws:rds:us-east-2:123456789012:db:*",
            "arn:aws:rds:us-east-2:123456789012:cev:*/*"
        ]
     },
     "StringEquals": {
        "aws:SourceAccount": "123456789012"
    }
  }
},
...
```

Rotieren der Anmeldeinformationen von RDS Custom für Oracle für Compliance-Programme

Bei einigen Compliance-Programmen müssen die Anmeldeinformationen der Datenbankbenutzer regelmäßig geändert werden, z. B. alle 90 Tage. RDS Custom für Oracle rotiert die Anmeldeinformationen für einige vordefinierte Datenbankbenutzer automatisch.

Themen

- Automatische Rotation der Anmeldeinformationen für vordefinierte Benutzer
- Richtlinien für das Rotieren von Benutzeranmeldeinformationen
- Manuelles Rotieren der Benutzeranmeldeinformationen

Automatische Rotation der Anmeldeinformationen für vordefinierte Benutzer

Wenn Ihre DB-Instance von RDS Custom für Oracle in Amazon RDS gehostet wird, wechseln die Anmeldeinformationen für die folgenden vordefinierten Oracle-Benutzer automatisch alle 30 Tage. Die Anmeldeinformationen der vorherigen Benutzer befinden sich in AWS Secrets Manager.

Datenbankbenutzer	Erstellt von	Unterstützte Engine-Versionen	Hinweise
`SYS`	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2 custom-oracle-se2-cdb	
`SYSTEM`	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2 custom-oracle-se2-CDB	
`RDSADMIN`	RDS	custom-oracle-ee	

Datenbankbenutzer	Erstellt von	Unterstützte Engine-Versionen	Hinweise
		custom-oracle-se2	
`C##RDSADMIN`	RDS	custom-oracle-ee-cdb custom-oracle-se2-CDB	Benutzernamen mit einem `C##` Präfix existieren nur in. CDBs Weitere Informationen finden Sie unter Überblick über CDBs die Architektur von Amazon RDS Custom for Oracle.
`RDS_DATAGUARD`	RDS	custom-oracle-ee	Dieser Benutzer ist nur in Lesereplikaten, Quelldatenbanken für Lesereplikate und Datenbanken vorhanden, die Sie mithilfe von Oracle Data Guard physisch zu RDS Custom migriert haben.
`C##RDS_DATAGUARD`	RDS	custom-oracle-ee-cdb	Dieser Benutzer ist nur in Lesereplikaten, Quelldatenbanken für Lesereplikate und Datenbanken vorhanden, die Sie mithilfe von Oracle Data Guard physisch zu RDS Custom migriert haben. Benutzernamen mit einem `C##` Präfix existieren nur in CDBs. Weitere Informationen finden Sie unter Überblick über CDBs die Architektur von Amazon RDS Custom for Oracle.

Eine Ausnahme von der automatischen Rotation der Anmeldeinformationen ist eine DB-Instance von RDS Custom für Oracle, die Sie manuell als Standby-Datenbank konfiguriert haben. RDS rotiert nur

die Anmeldeinformationen für Lesereplikate, die Sie mit dem CLI-Befehl `create-db-instance-read-replica` oder der API `CreateDBInstanceReadReplica` erstellt haben.

Richtlinien für das Rotieren von Benutzeranmeldeinformationen

Beachten Sie die folgenden Richtlinien, um sicherzustellen, dass Ihre Anmeldeinformationen entsprechend Ihrem Compliance-Programm rotieren:

- Wenn Ihre DB-Instance die Anmeldeinformationen automatisch rotiert, ändern oder löschen Sie keine Secrets, Passwortdateien oder Passwörter für Benutzer, die unter Vordefinierte Oracle-Benutzer aufgeführt sind. Andernfalls platziert RDS Custom Ihre DB-Instance möglicherweise außerhalb des Support-Perimeters, wodurch die automatische Rotation ausgesetzt wird.
- Der RDS-Hauptbenutzer ist nicht vordefiniert, sodass Sie entweder dafür verantwortlich sind, das Passwort manuell zu ändern oder die automatische Rotation in Secrets Manager einzurichten. Weitere Informationen finden Sie unter Rotate AWS Secrets Manager Secrets.

Manuelles Rotieren der Benutzeranmeldeinformationen

Für die folgenden Datenbankkategorien rotiert RDS die Anmeldeinformationen für die Benutzer, die unter Vordefinierte Oracle-Benutzer aufgeführt sind, nicht automatisch:

- Eine Datenbank, die Sie manuell so konfiguriert haben, dass sie als Standby-Datenbank funktioniert.
- Eine On-Premises-Datenbank.
- Eine DB-Instance, die sich außerhalb des Support-Perimeters oder in einem Zustand befindet, in dem die RDS-Custom-Automatisierung nicht ausgeführt werden kann. In diesem Fall rotiert RDS Custom auch keine Schlüssel.

Wenn Ihre Datenbank in eine der vorherigen Kategorien fällt, müssen Sie Ihre Benutzeranmeldeinformationen manuell rotieren.

So rotieren Sie die Benutzeranmeldeinformationen für eine DB-Instance manuell

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Stellen Sie unter Datenbanken sicher, dass RDS derzeit keine Backups Ihrer DB-Instance erstellt oder Vorgänge wie das Konfigurieren von Hochverfügbarkeit ausführt.

3. Wählen Sie auf der Seite mit den Datenbankdetails die Option Konfiguration und notieren Sie sich die Ressourcen-ID für die DB-Instance. Oder Sie können den AWS CLI Befehl verwendendescribe-db-instances.
4. Öffnen Sie die Secrets Manager Manager-Konsole unter https://console.aws.amazon.com/secretsmanager/.
5. Geben Sie im Suchfeld die Ressourcen-ID Ihrer Datenbank ein und suchen Sie mithilfe einer der folgenden Namenskonventionen nach einem Geheimnis:

```
do-not-delete-rds-custom-resource_id-uuid
rds-custom!oracle-do-not-delete-resource_id-uuid
```

Dieses Secret speichert das Passwort für RDSADMIN, SYS und SYSTEM. Die folgenden Beispielschlüssel gelten für die DB-Instance mit der Ressourcen-ID db-ABCDEFG12HIJKLNMNOPQRS3TUVWX und der 123456 UUID:

```
do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-123456
rds-custom!oracle-do-not-delete-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-123456
```

> **Important**
>
> Wenn Ihre DB-Instance ein Lesereplikat ist und die Engine custom-oracle-ee-cdb verwendet, existieren zwei Secrets mit dem Suffix *db-resource_id-uuid*, eines für den Hauptbenutzer und das andere für RDSADMIN ,SYS und SYSTEM. Führen Sie den folgenden Befehl auf dem Host aus, um das richtige Secret zu finden:
>
> ```
> cat /opt/aws/rdscustomagent/config/database_metadata.json | python3 -c
>  "import sys,json; print(json.load(sys.stdin)['dbMonitoringUserPassword'])"
> ```
>
> Das dbMonitoringUserPassword-Attribut gibt das Secret für RDSADMIN, SYS und SYSTEM an.

6. Wenn Ihre DB-Instance in einer Oracle Data Guard-Konfiguration vorhanden ist, suchen Sie mit einer der folgenden Namenskonventionen nach einem Geheimnis:

```
do-not-delete-rds-custom-resource_id-uuid-dg
rds-custom!oracle-do-not-delete-resource_id-uuid-dg
```

Dieses Secret speichert das Passwort für RDS_DATAGUARD. Die folgenden Beispielschlüssel gelten für die DB-Instance mit der DB-Ressourcen-ID **db-ABCDEFG12HIJKLNMNOPQRS3TUVWX** und der UUID 789012:

```
do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-789012-dg
rds-custom!oracle-do-not-delete-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-789012-dg
```

7. Aktualisieren Sie für alle Datenbankbenutzer, die unter Vordefinierte Oracle-Benutzer aufgeführt sind, die Passwörter, indem Sie den Anweisungen unter Ändern eines Geheimnisses folgen. AWS Secrets Manager
8. Wenn Ihre Datenbank eine eigenständige Datenbank oder eine Quelldatenbank in einer Konfiguration von Oracle Data Guard ist:

   a. Starten Sie Ihren Oracle-SQL-Client und melden Sie sich als SYS an.

   b. Führen Sie für jeden Datenbankbenutzer, der unter Vordefinierte Oracle-Benutzer aufgeführt ist, eine SQL-Anweisung in der folgenden Form aus:

   ```
   ALTER USER user-name IDENTIFIED BY pwd-from-secrets-manager ACCOUNT UNLOCK;
   ```

   Wenn das neue Passwort für RDSADMIN, das in Secrets Manager gespeichert ist, beispielsweise pwd-123 lautet, führen Sie die folgende Anweisung aus:

   ```
   ALTER USER RDSADMIN IDENTIFIED BY pwd-123 ACCOUNT UNLOCK;
   ```

9. Wenn Ihre DB-Instance Oracle Database 12c Release 1 (12.1) ausführt und von Oracle Data Guard verwaltet wird, kopieren Sie die Passwortdatei (orapw) manuell von der primären DB-Instance auf jede Standby-DB-Instance.

   Wenn Ihre DB-Instance in Amazon RDS gehostet wird, lautet der Speicherort der Passwortdatei /rdsdbdata/config/orapw. Für Datenbanken, die nicht in Amazon RDS gehostet werden, ist der Standardspeicherort unter Linux und UNIX $ORACLE_HOME/dbs/orapw$ORACLE_SID und unter Windows %ORACLE_HOME%\database\PWD%ORACLE_SID%.ora.

Arbeiten mit RDS Custom for Oracle

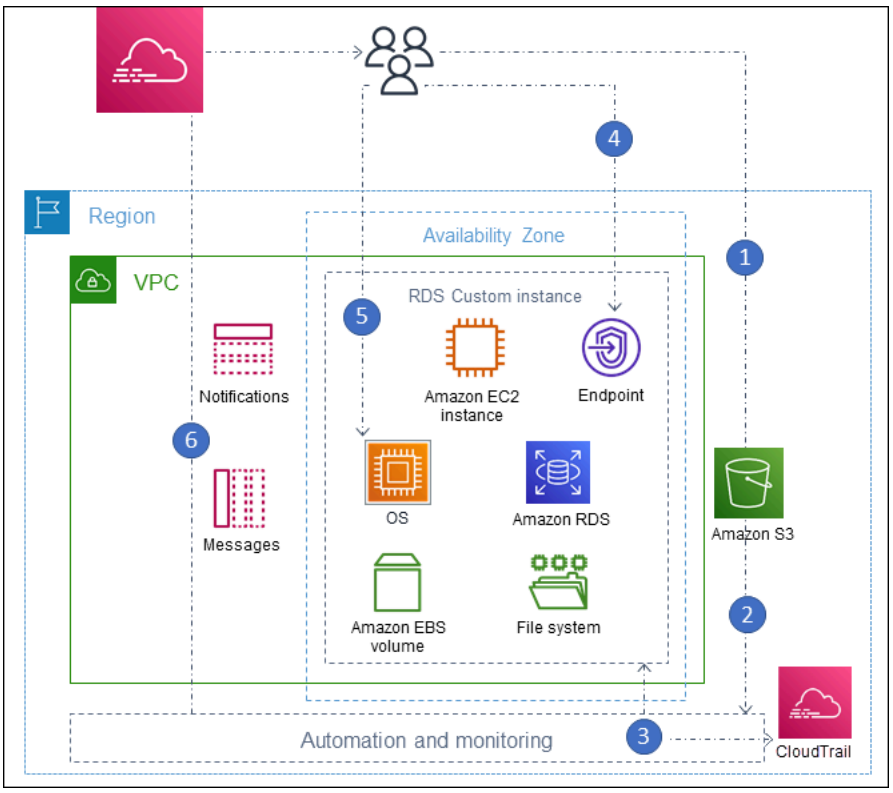
Im Folgenden finden Sie Anweisungen zur Erstellung, Verwaltung und Wartung Ihrer DB-Instance von RDS Custom for Oracle.

Themen

- RDS Benutzerdefiniert für Oracle-Workflow
- Datenbankarchitektur für Amazon RDS Custom für Oracle
- Verfügbarkeit und Unterstützung von Funktionen für RDS Custom for Oracle
- Anforderungen und Einschränkungen für RDS Custom for Oracle
- Einrichten Ihrer Umgebung für Amazon RDS Custom for Oracle
- Arbeiten mit benutzerdefinierten Engine-Versionen für Amazon RDS Custom für Oracle
- Konfigurieren einer DB-Instance für Amazon RDS Custom für Oracle
- Verwaltung einer Amazon RDS Custom for Oracle DB-Instance
- Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Oracle
- Mit Oracle-Repliken für RDS Custom for Oracle arbeiten
- Sicherung und Wiederherstellung einer Amazon RDS Custom for Oracle DB-Instance
- Mit Optionsgruppen in RDS Custom for Oracle arbeiten
- Migration einer lokalen Datenbank zu RDS Custom for Oracle
- Upgrade einer DB-Instance für Amazon RDS Custom for Oracle
- Beheben von DB-Problemen für Amazon RDS Custom für Oracle
- Bekannte Probleme für Amazon RDS Custom for Oracle

RDS Benutzerdefiniert für Oracle-Workflow

Das folgende Diagramm zeigt den typischen Workflow für RDS Custom for Oracle.



Die Schritte sind wie folgt:

1. Laden Sie Ihre Datenbanksoftware in Ihren Amazon S3-Bucket hoch.

   Weitere Informationen finden Sie unter Schritt 3: Hochladen Ihrer Installationsdateien in Amazon S3.
2. Erstellen Sie eine benutzerdefinierte Engine-Version (CEV) von RDS Custom für Oracle von Ihren Medien aus.

   Wählen Sie entweder die CDB-Architektur oder die traditionelle Nicht-CDB-Architektur. Weitere Informationen finden Sie unter Erstellen einer CEV.
3. Erstellen Sie eine DB-Instance von RDS Custom für Oracle aus einer CEV.

   Weitere Informationen finden Sie unter Erstellen einer RDS Custom für Oracle DB-Instance.
4. Verbinden Sie Ihre Anwendung mit dem Endpunkt der DB-Instance.

   Weitere Informationen erhalten Sie unter Herstellen einer Verbindung mit Ihrer RDS Custom DB-Instance über SSH und Mit Session Manager eine Verbindung zu Ihrer RDS Custom DB-Instance herstellen.
5. (Optional) Greifen Sie auf den Host zu, um Ihre Software anzupassen.

6. Überwachen Sie Benachrichtigungen und Nachrichten, die von RDS Custom Automation generiert wurden.

Datenbank-Installationsdateien

Ihre Verantwortung für Medien ist ein wesentlicher Unterschied zwischen Amazon RDS und RDS Custom. Amazon RDS, ein vollständig verwalteter Service, liefert das Amazon Machine Image (AMI) und die Datenbanksoftware. Die Amazon RDS-Datenbanksoftware ist vorinstalliert, Sie müssen also nur ein Datenbankmodul und eine Version auswählen und Ihre Datenbank erstellen.

Für RDS Custom liefern Sie Ihre eigenen Medien. Wenn Sie eine benutzerdefinierte Engine-Version erstellen, installiert RDS Custom die von Ihnen bereitgestellten Medien. RDS Custom Media enthält Ihre Datenbankinstallationsdateien und Patches. Dieses Service-Modell heißt Bringen Sie Ihre eigenen Medien mit (BYOM).

Benutzerdefinierte Engine-Versionen für RDS Custom für Oracle

Eine benutzerdefinierte Engine-Version (Custom Engine Version, CEV) für RDS Custom für Oracle ist ein binärer Volume-Snapshot einer Datenbankversion und eines AMI. Standardmäßig verwendet RDS Custom for Oracle das neueste AMI, das Amazon zur Verfügung EC2 stellt. Sie haben auch die Möglichkeit, ein vorhandenes AMI wiederzuverwenden.

CEV-Manifest

Nachdem Sie die Installationsdateien für die Oracle-Datenbank von Oracle heruntergeladen haben, laden Sie sie in einen Amazon-S3-Bucket hoch. Wenn Sie Ihre CEV erstellen, geben Sie die Dateinamen in einem JSON-Dokument an, das als CEV-Manifest bezeichnet wird. RDS Custom für Oracle verwendet die angegebenen Dateien und das AMI, um Ihre CEV zu erstellen.

RDS Custom für Oracle bietet JSON-Manifestvorlagen mit unseren empfohlenen .zip-Dateien für jede unterstützte Oracle Database Version. Die folgende Vorlage ist beispielsweise für 19.17.0.0.0 RU.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
```

```
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Sie können auch Installationsparameter im JSON-Manifest angeben. Sie können beispielsweise Werte für die Oracle-Basis, das Oracle-Standardverzeichnis sowie die ID und den Namen des UNIX/Linux Benutzers und der Gruppe festlegen. Weitere Informationen finden Sie unter JSON-Felder im CEV-Manifest.

CEV-Namensformat

Benennen Sie Ihre CEV für RDS Custom für Oracle mit einer vom Kunden angegebenen Zeichenfolge. Das Namensformat ist je nach Oracle-Database-Version das folgende:

- `19.customized_string`
- `18.customized_string`
- `12.2.customized_string`
- `12.1.customized_string`

Er darf nur 1-50 alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (-. _) enthalten. Sie können beispielsweise Ihre Rolle `19.my_cev1` nennen.

Oracle-Multitenant-Architektur in RDS Custom für Oracle

Die Oracle-Multitenant-Architektur ermöglicht es einer Oracle-Datenbank, als Container-Datenbank (Container Database, CDB) zu fungieren. Eine CDB umfasst null, eine oder viele vom Kunden

erstellte austauschbare Datenbanken (). PDBs Eine PDB ist eine portable Sammlung von Schemas und Objekten, die einer Anwendung als traditionelle Non-CDB angezeigt wird.

Wenn Sie eine CEV für RDS Custom für Oracle erstellen, geben Sie entweder die CDB- oder Non-CDB-Architektur an. Sie können nur dann eine CDB für RDS Custom-für-Oracle erstellen, wenn die zum Erstellen verwendete CEV die Multitenant-Architektur verwendet. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Engine-Versionen für Amazon RDS Custom für Oracle.

Erstellen einer RDS-Custom-for-Oracle-DB-Instance

Nachdem Sie die CEV erstellt haben, kann sie verwendet werden. Sie können mehrere CEVs, und Sie können mehrere RDS Custom for Oracle DB-Instances aus einem beliebigen CEV erstellen. Sie können auch den Status einer CEV ändern, um sie verfügbar oder inaktiv zu machen.

Sie können Ihre RDS Custom for Oracle DB-Instance entweder mit der Multitenant-Architektur (`custom-oracle-ee-cdb`oder dem `custom-oracle-se2-cdb` Engine-Typ) von Oracle oder mit der herkömmlichen Nicht-CDB-Architektur (`custom-oracle-ee`oder `custom-oracle-se2` dem Engine-Typ) erstellen. Wenn Sie eine Container-Datenbank (CDB) erstellen, enthält diese eine Pluggable Database (PDB) und einen PDB-Seed. Sie können weitere PDBs manuell mit Oracle SQL erstellen.

Um Ihre RDS Custom for Oracle DB-Instance zu erstellen, verwenden Sie den `create-db-instance`-Befehl. Geben Sie in diesem Befehl an, welches CEV verwendet werden soll.
Das Verfahren ähnelt dem Erstellen einer DB-Instance von Amazon RDS. Einige Parameter unterscheiden sich jedoch. Weitere Informationen finden Sie unter Konfigurieren einer DB-Instance für Amazon RDS Custom für Oracle.

Datenbankverbindungen

Wie bei einer DB-Instance von Amazon RDS befindet sich die DB-Instance von RDS Custom in einer Virtual Private Cloud (VPC). Ihre Anwendung stellt mithilfe eines Oracle-Listeners eine Verbindung zur Oracle-Datenbank her.

Wenn es sich bei Ihrer Datenbank um eine CDB handelt, können Sie den Listener `L_RDSCDB_001` verwenden, um eine Verbindung zum CDB-Root und zu einer PDB herzustellen. Wenn Sie eine Nicht-CDB an eine CDB anschließen, müssen Sie die Einstellungen `USE_SID_AS_SERVICE_LISTENER = ON` festlegen, sodass migrierte Anwendungen dieselben Einstellungen beibehalten.

Wenn Sie eine Verbindung mit einer Nicht-CDB herstellen, ist der Hauptbenutzer der Benutzer der Nicht-CDB. Wenn Sie eine Verbindung mit einer CDB herstellen, ist der Hauptbenutzer der Benutzer der PDB. Um eine Verbindung zum CDB-Root herzustellen, melden Sie sich beim Host an, starten Sie einen SQL-Client und erstellen Sie einen Administratorbenutzer mit SQL-Befehlen.

Benutzerdefinierte RDS Anpassung

Sie können auf den RDS Custom Host zugreifen, um Software zu installieren oder anzupassen. Um Konflikte zwischen Ihren Änderungen und der RDS Custom Automation zu vermeiden, können Sie die Automatisierung für einen bestimmten Zeitraum pausieren. Während dieses Zeitraums führt RDS Custom keine Überwachung oder Instanzwiederherstellung durch. Am Ende des Zeitraums nimmt RDS Custom die vollständige Automatisierung wieder auf. Weitere Informationen finden Sie unter Ihre benutzerdefinierte DB-Instance pausieren und wieder aufnehmen RDS.

Datenbankarchitektur für Amazon RDS Custom für Oracle

RDS Custom für Oracle unterstützt sowohl die Multitenant- als auch die Non-Multitenant-Architektur von Oracle.

Themen

- Unterstützte Oracle-Datenbankarchitekturen
- Unterstützte Engine-Typen
- Unterstützte Funktionen in der Oracle-Multitenant-Architektur

Unterstützte Oracle-Datenbankarchitekturen

Die Oracle-Multitenant-Architektur, auch als CDB-Architektur bekannt, ermöglicht es einer Oracle-Datenbank, als Container-Datenbank (CDB) zu fungieren. Eine CDB enthält Pluggable Databases (PDBs). Eine PDB ist eine Sammlung von Schemas und Objekten, die einer Anwendung als herkömmliche Oracle-Datenbank angezeigt wird. Weitere Informationen finden Sie im Abschnitt Einführung in die Multi-Tenant-Architektur im Oracle-Multi-Tenant-Administratorhandbuch.

Die CDB- und Non-CDB-Architekturen schließen sich gegenseitig aus. Wenn eine Oracle-Datenbank keine CDB ist, ist sie eine Non-CDB und kann daher keine PDBs enthalten. In RDS Custom für Oracle unterstützt nur Oracle Database 19c die CDB-Architektur. Wenn Sie also DB-Instances unter Verwendung von früheren Oracle-Database-Versionen erstellen, können Sie nur Non-CDBs erstellen. Weitere Informationen finden Sie unter Überlegungen zur Multi-Tenant-Architektur.

Unterstützte Engine-Typen

Wenn Sie eine Amazon RDS Custom for Oracle CEV- oder DB-Instance erstellen, wählen Sie entweder einen CDB-Engine-Typ oder einen Nicht-CDB-Engine-Typ:

- `custom-oracle-ee-cdb` und `custom-oracle-se2-cdb`

  Diese Engine-Typen spezifizieren die Oracle-Multitenant-Architektur. Diese Option ist nur für Oracle Database 19c verfügbar. Wenn Sie eine RDS-für-Oracle-DB-Instance unter Verwendung der Multi-Tenant-Architektur erstellen, umfasst Ihre CDB folgende Container:

  - CDB-Root (`CDB$ROOT`)
  - PDB-Seed (`PDB$SEED`)
  - Ursprüngliche PDB

Mit dem Oracle-SQL-Befehl `CREATE PLUGGABLE DATABASE` können Sie weitere PDBs erstellen. Zum Erstellen oder Löschen von PDBs können Sie keine RDS-APIs verwenden.

- `custom-oracle-ee` und `custom-oracle-se2`

  Diese Engine-Typen spezifizieren die traditionelle Nicht-CDB-Architektur. Eine Nicht-CDB kann keine Pluggable Databases (PDBs) enthalten.

Weitere Informationen finden Sie unter Überlegungen zur Multi-Tenant-Architektur.

Unterstützte Funktionen in der Oracle-Multitenant-Architektur

Eine CDB-Instance von RDS Custom für Oracle unterstützt die folgenden Funktionen:

- Sicherungen
- Wiederherstellung und point-time-restore (PITR) aus Backups
- Read Replicas
- Unterversion-Upgrades

Verfügbarkeit und Unterstützung von Funktionen für RDS Custom for Oracle

In diesem Thema finden Sie eine Zusammenfassung der Verfügbarkeit und des Supports der Funktionen von RDS Custom for Oracle als Kurzreferenz.

Themen

- AWS-Region und Datenbankversionsunterstützung für RDS Custom for Oracle
- Unterstützung der Datenbankversion für RDS Custom for Oracle
- Editions- und Lizenzunterstützung von RDS Custom für Oracle
- Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle
- Unterstützung von Optionsgruppen für RDS Custom for Oracle

AWS-Region und Datenbankversionsunterstützung für RDS Custom for Oracle

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Verfügbarkeit von Versionen und Regionen von RDS Custom für Oracle finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom.

Unterstützung der Datenbankversion für RDS Custom for Oracle

RDS Custom for Oracle unterstützt die folgenden Oracle-Datenbankversionen:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1)

Editions- und Lizenzunterstützung von RDS Custom für Oracle

RDS Custom for Oracle unterstützt Enterprise Edition (EE) und Standard Edition 2 (SE2) auf dem BYOL-Modell.

Beachten Sie die folgenden Einschränkungen für Standard Edition 2:

- Oracle Data Guard wird nicht unterstützt. Daher können Sie keine Oracle-Read Replicas erstellen.

- Sie können nur DB-Instance-Klassen mit 16 oder weniger V CPUs (bis zu 4xlarge) verwenden.
- Eine CDB-Instance auf Standard Edition 2 unterstützt maximal 3 Tenant-Datenbanken.
- Sie können keine Daten zwischen Enterprise Edition und Standard Edition 2 migrieren.

Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle

RDS Custom für Oracle unterstützt die folgenden DB-Instance-Klassen. Wenn Sie eine DB-Instance auf Standard Edition 2 erstellen, können Sie nur Instanzklassen mit 16 oder weniger V CPUs (bis zu 4x groß) verwenden.

Typ	Größe
db.r7i	db.r7i.large |db.r7i.xlarge |db.r7i.2xlarge |db.r7i.4xlarge |db.r7i.8xlarge |db.r7i.12xlarge |db.r7i.16xlarge |db.r7i.24xlarge |db.r7i.48xlarge
db.r6i	db.r6i.large |db.r6i.xlarge |db.r6i.2xlarge |db.r6i.4xlarge |db.r6i.8xlarge |db.r6i.12xlarge |db.r6i.16xlarge |db.r6i.24xlarge |db.r6i.32xlarge
db.r5b	db.r5b.large |db.r5b.xlarge |db.r5b.2xlarge |db.r5b.4xlarge |db.r5b.8xlarge |db.r5b.12xlarge |db.r5b.16xlarge |db.r5b.24xlarge
db.r5	db.r5.large |db.r5.xlarge |db.r5.2xlarge |db.r5.4xlarge | db.r5.8xlarge |db.r5.12xlarge |db.r5.16xlarge |db.r5.24xlarge
db.x2iec	db.x2iedn.xlarge |db.x2iedn.2xlarge |db.x2iedn.4xlarge | db.x2iedn.8xlarge |db.x2iedn.16xlarge |db.x2iedn.24xlarge | db.x2iedn.32xlarge
db.x2iez	db.x2iezn.2xlarge |db.x2iezn.4xlarge |db.x2iezn.6xlarge | db.x2iezn.8xlarge |db.x2iezn.12xlarge
db.m7i	db.m7i.large |db.m7i.xlarge |db.m7i.2xlarge |db.m7i.4xlarge |db.m7i.8xlarge |db.m7i.12xlarge |db.m7i.16xlarge |db.m7i.24xlarge |db.m7i.48xlarge

Typ	Größe
db.m6i	db.m6i.large | db.m6i.xlarge | db.m6i.2xlarge | db.m6i.4xlarge | db.m6i.8xlarge | db.m6i.12xlarge | db.m6i.16xlarge | db.m6i.24xlarge | db.m6i.32xlarge
db.m5	db.m5.large | db.m5.xlarge | db.m5.2xlarge | db.m5.4xlarge | db.m5.8xlarge | db.m5.12xlarge | db.m5.16xlarge | db.m5.24xlarge
db.t3	db.t3.medium | db.t3.large | db.t3.xlarge | db.t3.2xlarge

Unterstützung von Optionsgruppen für RDS Custom for Oracle

Sie können eine Optionsgruppe angeben, wenn Sie eine RDS Custom for Oracle DB-Instance erstellen oder ändern. Weitere Informationen finden Sie unter Mit Optionsgruppen in RDS Custom for Oracle arbeiten.

Anforderungen und Einschränkungen für RDS Custom for Oracle

In diesem Thema finden Sie eine Zusammenfassung der Funktionsverfügbarkeit und der Anforderungen Amazon RDS Custom für Oracle, um schnell nachschlagen zu können.

Themen

- Allgemeine Anforderungen von RDS Custom für Oracle
- Allgemeine Einschränkungen für RDS Custom für Oracle
- CEV- und AMI-Einschränkungen für RDS Custom for Oracle
- Einstellungen zum Erstellen und Ändern von Workflows werden nicht unterstützt
- DB-Instance-Kontingente für Ihre AWS-Konto

Allgemeine Anforderungen von RDS Custom für Oracle

Stellen Sie sicher, dass die folgenden Anforderungen für Amazon RDS Custom for Oracle erfüllt sind:

- Sie haben Zugriff auf My Oracle Support und Oracle Software Delivery Cloud, um die Liste der unterstützten Installationsdateien und Patches für RDS Custom for Oracle herunterzuladen. Wenn Sie einen unbekannten Patch verwenden, schlägt die Erstellung der benutzerdefinierten Engine-Version (CEV) fehl. Wenden Sie sich in diesem Fall an das RDS Custom Support-Team und bitten Sie es, den fehlenden Patch hinzuzufügen. Weitere Informationen finden Sie unter Schritt 2: Herunterladen Ihrer Datenbankinstallationsdateien und Patches von Oracle Software Delivery Cloud.
- Sie haben Zugriff auf Amazon S3. Sie benötigen diesen Service aus den folgenden Gründen:
  - Sie laden Ihre Oracle-Installationsdateien in S3-Buckets hoch. Sie verwenden die hochgeladenen Installationsdateien, um Ihre RDS-Custom-CEV zu erstellen.
  - RDS Custom für Oracle verwendet Skripts, die aus intern definierten S3-Buckets heruntergeladen wurden, um Aktionen auf Ihren DB-Instances auszuführen. Diese Skripts sind für das Onboarding und die Automatisierung von RDS Custom erforderlich.
  - RDS Custom für Oracle lädt bestimmte Dateien in S3-Buckets hoch, die sich in Ihrem Kundenkonto befinden. Diese Buckets verwenden das folgende Benennungsformat: `do-not-delete-rds-custom-` *`account_id`* `-` *`region`* `-` *`uuid`*. Beispielsweise könnte ein Bucket den Namen `do-not-delete-rds-custom-123456789012-us-east-1-12a3b4` tragen.

  Weitere Informationen erhalten Sie unter Schritt 3: Hochladen Ihrer Installationsdateien in Amazon S3 und Erstellen einer CEV.

- Sie verwenden die unter aufgeführten DB-Instance-KlassenUnterstützung von DB-Instance-Klassen für RDS Custom for Oracle, um Ihre RDS Custom for Oracle DB-Instances zu erstellen.
- Auf Ihren RDS Custom for Oracle DB-Instances wird Oracle Linux 8 (empfohlen) oder Oracle Linux 7 ausgeführt. Wenn Sie Oracle Linux 7 benötigen, wenden Sie sich an Support. Weitere Informationen finden Sie unter Überlegungen zu RDS Custom for Oracle-Datenbank-Upgrades.
- Sie geben die Solid-State-Laufwerke gp2, gp3 oder io1 für den Amazon EBS-Speicher an. Die maximale Speichergröße beträgt 64 TiB.
- Sie haben einen AWS KMS Schlüssel, um eine RDS Custom for Oracle DB-Instance zu erstellen. Weitere Informationen finden Sie unter Schritt 1: Erstellen oder Wiederverwenden eines symmetrischen AWS KMS -Verschlüsselungsschlüssels.
- Sie verfügen über die AWS Identity and Access Management (IAM-) Rolle und das Instance-Profil, die für die Erstellung von RDS Custom für Oracle-DB-Instances erforderlich sind. Weitere Informationen finden Sie unter Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle.
- Der AWS Identity and Access Management (IAM-) Benutzer, der eine benutzerdefinierte CEV- oder RDS-DB-Instance erstellt, verfügt über die erforderlichen Berechtigungen für IAM und Amazon CloudTrail S3.

  Weitere Informationen finden Sie unter Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen.
- Sie stellen Ihre eigene Virtual Private Cloud (VPC) und Ihre Sicherheitsgruppenkonfiguration zur Verfügung. Weitere Informationen finden Sie unter Schritt 6: Konfigurieren Sie Ihre VPC für RDS Custom for Oracle.
- Sie geben eine Netzwerkkonfiguration an, die RDS Custom for Oracle für den Zugriff auf andere verwenden kann. AWS-Services Spezielle Anforderungen finden Sie unter Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle.

Allgemeine Einschränkungen für RDS Custom für Oracle

Die folgenden Einschränkungen gelten für RDS Custom for Oracle:

- Sie können den DB-Instance-Bezeichner einer vorhandenen DB-Instance von RDS Custom für Oracle nicht ändern.
- Sie können die Oracle-Multitenant-Architektur für keine Version außer Oracle Database 19c angeben.

- Sie können nicht mehrere Oracle-Datenbanken auf einer einzigen DB-Instance von RDS Custom für Oracle erstellen.
- Sie können Ihre RDS Custom for Oracle DB-Instance oder die ihr zugrunde liegende EC2 Amazon-Instance nicht stoppen. Die Fakturierung für eine DB-Instance von RDS Custom für Oracle kann nicht gestoppt werden.
- Sie können die automatische Verwaltung gemeinsam genutzter Speicher nicht verwenden, da RDS Custom for Oracle nur die automatische Speicherverwaltung unterstützt. Weitere Informationen finden Sie unter Automatische Speicherverwaltung im Oracle-Database-Administratorhandbuch.
- Achten Sie darauf, `DB_UNIQUE_NAME` für die primäre DB-Instance nicht zu ändern. Eine Änderung des Namens führt dazu, dass jeder Wiederherstellungsvorgang hängen bleibt.
- Sie können nicht mehr als 20 Snapshot-Kopien gleichzeitig in derselben Region erstellen.
- Sie können die `describe-reserved-db-instances` API für RDS Custom für Oracle-DB-Instances nicht verwenden.

Die Einschränkungen beim Ändern einer DB-Instance von RDS Custom für Oracle finden Sie unter Ändern Ihrer RDS Custom for Oracle-DB-Instance. Informationen zu Replikationsbeschränkungen finden Sie unter Allgemeine Einschränkungen für die Replikation mit RDS Custom für Oracle.

CEV- und AMI-Einschränkungen für RDS Custom for Oracle

Die folgenden Einschränkungen gelten für RDS Custom for Oracle CEVs und AMIs:

- Sie können kein eigenes AMI zur Verwendung in einem RDS Custom for Oracle CEV bereitstellen. Sie können entweder das Standard-AMI angeben, das Oracle Linux 8 verwendet, oder ein AMI, das zuvor von einem RDS Custom for Oracle CEV verwendet wurde.

> **Note**
>
> RDS Custom for Oracle veröffentlicht ein neues Standard-AMI, wenn allgemeine Sicherheitslücken und Sicherheitslücken entdeckt werden. Es ist kein fester Zeitplan verfügbar oder garantiert. RDS Custom for Oracle veröffentlicht in der Regel alle 30 Tage ein neues Standard-AMI.

- Sie können eine CEV nicht ändern, um ein anderes AMI zu verwenden.

- Sie können keine CDB-Instance aus einem CEV erstellen, das die `custom-oracle-se2` Engine-Typen `custom-oracle-ee` oder verwendet. Das CEV muss oder verwenden. `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- RDS Custom for Oracle ermöglicht es Ihnen derzeit nicht, das Betriebssystem Ihrer RDS Custom for Oracle-DB-Instance mit RDS-API-Aufrufen zu aktualisieren. Um dieses Problem zu umgehen, können Sie Ihr Betriebssystem manuell mit dem folgenden Befehl aktualisieren:`sudo yum update --security`.

Einstellungen zum Erstellen und Ändern von Workflows werden nicht unterstützt

Wenn Sie eine RDS Custom for Oracle DB-Instance erstellen oder ändern, können Sie Folgendes nicht tun:

- Ändern Sie die Anzahl der CPU-Kerne und -Threads pro Kern in der DB-Instance-Klasse.
- Aktivieren Sie die Speicherskalierung.
- SicherungsaufBEWAHRUNG einstellen auf`0`aus.
- Konfiguration der Kerberos-Authentifizierung
- Geben Sie Ihre eigene DB-Parametergruppe oder Optionsgruppe an.
- Aktivieren Sie Performance Insights.
- Einschalten von automatischen Nebenversions-Upgrades

DB-Instance-Kontingente für Ihre AWS-Konto

Stellen Sie sicher, dass die kombinierte Anzahl der RDS Custom- und Amazon RDS DB-Instances Ihr Kontingentlimit nicht überschreitet. Wenn Ihr Kontingent für Amazon RDS beispielsweise 40 DB-Instances beträgt, können Sie 20 RDS Custom für Oracle DB-Instanzen und 20 Amazon RDS DB-Instances haben.

Einrichten Ihrer Umgebung für Amazon RDS Custom for Oracle

Bevor Sie eine DB-Instance für Amazon RDS Custom für Oracle erstellen, führen Sie die folgenden Schritte aus.

Themen

- Schritt 1: Erstellen oder Wiederverwenden eines symmetrischen AWS KMS - Verschlüsselungsschlüssels
- Schritt 2: Laden Sie das herunter und installieren Sie es AWS CLI
- Schritt 3: Extrahieren Sie die CloudFormation Vorlagen für RDS Custom for Oracle
- Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle
- Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen
- Schritt 6: Konfigurieren Sie Ihre VPC für RDS Custom for Oracle

Schritt 1: Erstellen oder Wiederverwenden eines symmetrischen AWS KMS - Verschlüsselungsschlüssels

Von Kunden verwaltete Schlüssel befinden sich AWS KMS keys in Ihrem AWS Konto, das Sie erstellen, besitzen und verwalten. Ein kundenverwalteter symmetrischer KMS-Verschlüsselungsschlüssel ist für RDS Custom erforderlich. Wenn Sie eine DB-Instance von RDS Custom for Oracle erstellen, geben Sie die KMS-Schlüsselkennung an. Weitere Informationen finden Sie unter Konfigurieren einer DB-Instance für Amazon RDS Custom für Oracle.

Ihnen stehen folgende Optionen zur Verfügung:

- Wenn Sie bereits einen kundenverwalteten KMS-Schlüssel in Ihrem haben AWS-Konto, können Sie ihn mit RDS Custom verwenden. Es sind keine weiteren Maßnahmen erforderlich.
- Wenn Sie bereits einen kundenverwalteten symmetrischen KMS-Verschlüsselungsschlüssel für eine andere RDS-Custom-Engine erstellt haben, können Sie denselben KMS-Schlüssel wiederverwenden. Es sind keine weiteren Maßnahmen erforderlich.
- Wenn Sie keinen vorhandenen kundenverwalteten symmetrischen KMS-Verschlüsselungsschlüssel in Ihrem Konto haben, erstellen Sie einen KMS-Schlüssel, indem Sie den Anweisungen unter Erstellen von Schlüsseln im AWS Key Management Service - Entwicklerhandbuch folgen.
- Wenn Sie eine benutzerdefinierte CEV- oder RDS-DB-Instance erstellen und sich Ihr KMS-Schlüssel in einer anderen befindet AWS-Konto, stellen Sie sicher, dass Sie den AWS CLI

verwenden. Sie können die AWS Konsole nicht mit kontoübergreifenden KMS-Schlüsseln verwenden.

> **Important**
>
> RDS Custom unterstützt keine AWS verwalteten KMS-Schlüssel.

Stellen Sie sicher, dass Ihr symmetrischer Verschlüsselungsschlüssel Zugriff auf die `kms:Decrypt` und `kms:GenerateDataKey` -Operationen der AWS Identity and Access Management (IAM-) Rolle in Ihrem IAM-Instanzprofil gewährt. Wenn Sie einen neuen symmetrischen Verschlüsselungsschlüssel in Ihrem Konto haben, sind keine Änderungen erforderlich. Stellen Sie andernfalls sicher, dass die Richtlinie Ihres symmetrischen Verschlüsselungsschlüssels Zugriff auf diese Operationen erteilt.

Weitere Informationen finden Sie unter Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle.

Weitere Informationen zum Konfigurieren von IAM für RDS Custom for Oracle finden Sie unter Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle.

Schritt 2: Laden Sie das herunter und installieren Sie es AWS CLI

AWS bietet Ihnen eine Befehlszeilenschnittstelle zur Verwendung der benutzerdefinierten Funktionen von RDS. Sie können entweder Version 1 oder Version 2 des AWS CLI nutzen.

Informationen zum Herunterladen und Installieren von finden Sie unter Installation oder Aktualisierung der neuesten Version von. AWS CLI AWS CLI

Überspringen Sie diesen Schritt, wenn einer der folgenden Punkte zutrifft:

- Sie planen, auf RDS Custom nur über den zuzugreifen AWS Management Console.
- Sie haben die Engine AWS CLI für Amazon RDS oder eine andere RDS Custom DB-Engine bereits heruntergeladen.

Schritt 3: Extrahieren Sie die CloudFormation Vorlagen für RDS Custom for Oracle

Um die Einrichtung zu vereinfachen, empfehlen wir dringend, AWS CloudFormation Vorlagen zum Erstellen von CloudFormation Stacks zu verwenden. Wenn Sie planen, IAM und Ihre VPC manuell zu konfigurieren, überspringen Sie diesen Schritt.

Themen

- Schritt 3a: Laden Sie die CloudFormation Vorlagendateien herunter
- Schritt 3b: Extrahieren custom-oracle-iam Sie die Datei .json
- Schritt 3c: Extrahieren Sie custom-vpc.json

Schritt 3a: Laden Sie die CloudFormation Vorlagendateien herunter

Eine CloudFormation Vorlage ist eine Deklaration der AWS Ressourcen, aus denen ein Stapel besteht. Die Vorlage wird als JSON-Datei gespeichert.

Um die CloudFormation Vorlagendateien herunterzuladen

1. Öffnen Sie das Kontextmenü (Rechtsklick) für den Link (custom-oracle-iam.zip) und wählen Sie Link speichern unter.
2. Speichern Sie die Datei auf Ihrem Computer.
3. Wiederholen Sie die vorherigen Schritte für den Link custom-vpc.zip.

   Wenn Sie Ihre VPC für RDS Custom bereits konfiguriert haben, überspringen Sie diesen Schritt.

Schritt 3b: Extrahieren custom-oracle-iam Sie die Datei .json

Öffnen Sie die `custom-oracle-iam.zip` Datei, die Sie heruntergeladen haben, und extrahieren Sie die Datei`custom-oracle-iam.json`. Der Anfang der Datei sieht wie folgt aus.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EncryptionKey": {
      "Type": "String",
      "Default": "*",
      "Description": "KMS Key ARN for encryption of data managed by RDS Custom and by
 DB Instances."
    }
  },
  "Resources": {
    "RDSCustomInstanceServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": { "Fn::Sub": "AWSRDSCustomInstanceRole-${AWS::Region}" },
```

```
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                "Service": "ec2.amazonaws.com"
              }
            }
          ]
        },...
```

Schritt 3c: Extrahieren Sie custom-vpc.json

> **Note**
>
> Wenn Sie bereits eine bestehende VPC für RDS Custom for Oracle konfiguriert haben, überspringen Sie diesen Schritt. Weitere Informationen finden Sie unter Konfigurieren Sie Ihre VPC manuell für RDS Custom for Oracle.

Öffnen Sie die `custom-vpc.zip` Datei, die Sie heruntergeladen haben, und extrahieren Sie die Datei dann. `custom-vpc.json` Der Anfang der Datei sieht wie folgt aus.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "PrivateVpc": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "Private VPC Id to use for RDS Custom DB Instances"
    },
    "PrivateSubnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "Private Subnets to use for RDS Custom DB Instances"
    },
    "RouteTable": {
      "Type": "String",
      "Description": "Route Table that must be associated with the PrivateSubnets and
 used by S3 VPC Endpoint",
      "AllowedPattern": "rtb-[0-9a-z]+"
    }
```

```
  },
  "Resources": {
    "DBSubnetGroup": {
      "Type": "AWS::RDS::DBSubnetGroup",
      "Properties": {
        "DBSubnetGroupName": "rds-custom-private",
        "DBSubnetGroupDescription": "RDS Custom Private Network",
        "SubnetIds": {
          "Ref": "PrivateSubnets"
        }
      }
    },...
```

Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle

Sie verwenden eine IAM-Rolle oder einen IAM-Benutzer (als IAM-Entität bezeichnet), um eine DB-Instance von RDS Custom mit der Konsole oder der AWS CLI zu erstellen. Diese IAM-Entität muss über die erforderlichen Berechtigungen für die Instance-Erstellung verfügen.

Sie können IAM entweder mit CloudFormation oder mit manuellen Schritten konfigurieren.

> **Important**
>
> Wir empfehlen dringend, dass Sie Ihre RDS Custom for Oracle-Umgebung mithilfe von AWS CloudFormation konfigurieren. Diese Methode ist am einfachsten und am wenigsten fehleranfällig.

Themen

- Konfigurieren Sie IAM mit CloudFormation
- Manuelles Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils

Konfigurieren Sie IAM mit CloudFormation

Wenn Sie die CloudFormation Vorlage für IAM verwenden, werden die folgenden erforderlichen Ressourcen erstellt:

- Ein Instanzprofil mit dem Namen `AWSRDSCustomInstanceProfile-`*`region`*
- Eine Servicerolle mit dem Namen `AWSRDSCustomInstanceRole-`*`region`*

- Eine Zugriffsrichtlinie mit dem Namen`AWSRDSCustomIamRolePolicy`, die der Servicerolle zugeordnet ist

Um IAM zu konfigurieren mit CloudFormation

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Starten Sie den Create Stack-Assistenten und wählen Sie Create Stack (Stapel erstellen) aus.
3. Gehen Sie auf der Seite Create stack (Stack erstellen) wie folgt vor:
   a. Wählen Sie unter Prepare template (Vorlage vorbereiten) den Wert Template is ready (Vorlage ist bereit) aus.
   b. Wählen Sie unter Templete source (Vorlagenquelle) den Wert Upload a template file (Vorlagendatei hochladen) aus.
   c. Navigieren Sie für „Datei auswählen“ zu „.json“ und wählen Sie dann auscustom-oracle-iam.
   d. Wählen Sie Weiter aus.
4. Führen Sie auf der Seite Specify DB Details (Festlegen von DB-Detail) die folgenden Schritte aus:
   a. Geben Sie unter Stack name (Stack-Name) **`custom-oracle-iam`** ein.
   b. Wählen Sie Weiter aus.
5. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
6. Führen Sie auf der Seite Review (Überprüfen) custom-oracle-iam die folgenden Schritte aus:
   a. Aktivieren Sie das Kontrollkästchen Ich bestätige, dass AWS CloudFormation ggf. IAM-Ressourcen mit benutzerdefinierten Namen erstellt.
   b. Wählen Sie Absenden aus.

   CloudFormation erstellt die IAM-Rollen, die RDS Custom for Oracle benötigt. Wenn im linken Bereich custom-oracle-iamCREATE_COMPLETE angezeigt wird, fahren Sie mit dem nächsten Schritt fort.
7. Wählen Sie im linken Bereich. custom-oracle-iam Führen Sie im rechten Bereich die folgenden Schritte aus:

a. Wählen Sie Stack-Info aus. Ihr Stack hat eine ID im Format arn:aws:cloudformation: ::stack//. *region account-no* custom-oracle-iam *identifier*

b. Wählen Sie Resources aus. Sie sollten Folgendes sehen:

- Ein AWSRDSCustomInstanceProfile Instanzprofil mit dem Namen - *region*
- Eine Servicerolle mit dem Namen AWSRDSCustomInstanceRole- ***region***

Wenn Sie Ihre RDS-Custom-DB-Instance erstellen, müssen Sie die Instance-Profil-ID angeben.

Manuelles Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils

Die Konfiguration ist am einfachsten, wenn Sie verwenden CloudFormation. Sie können IAM jedoch auch manuell konfigurieren. Gehen Sie für die manuelle Einrichtung wie folgt vor:

- Schritt 1: Erstellen Sie die IAM-Rolle für das AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Schritt 2: Fügen Sie eine Zugriffsrichtlinie hinzu AWSRDSCustom InstanceRoleForRdsCustomInstance.
- Schritt 2: Fügen Sie eine Zugriffsrichtlinie hinzu AWSRDSCustom InstanceRoleForRdsCustomInstance.
- Schritt 4: Hinzufügen AWSRDSCustom InstanceRoleForRdsCustomInstance zu AWSRDSCustom InstanceProfile.

Schritt 1: Erstellen Sie die IAM-Rolle für das AWSRDSCustomInstanceRoleForRdsCustomInstance

In diesem Schritt erstellen Sie die Rolle mithilfe des Namensformats `AWSRDSCustomInstanceRole-`*`region`*. Mithilfe der Vertrauensrichtlinie EC2 kann Amazon die Rolle übernehmen. Im folgenden Beispiel wird davon ausgegangen, dass Sie die Umgebungsvariable `$REGION` auf die AWS-Region festgelegt haben, in der Sie Ihre DB-Instance erstellen möchten.

```
aws iam create-role \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
      "Statement": [
```

```
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
              "Service": "ec2.amazonaws.com"
          }
        }
      ]
    }'
```

Schritt 2: Fügen Sie eine Zugriffsrichtlinie hinzu AWSRDSCustom InstanceRoleForRdsCustomInstance

Wenn Sie eine eingebundene Richtlinie in eine IAM-Rolle einbetten, wird die eingebundene Richtlinie als Teil der Rollezugriffsrichtlinie (Berechtigungsrichtlinie) verwendet. Sie erstellen die `AWSRDSCustomIamRolePolicy` Richtlinie, die es Amazon ermöglicht, Nachrichten EC2 zu senden und zu empfangen und verschiedene Aktionen auszuführen.

Im folgenden Beispiel wird die Zugriffsrichtlinie mit dem Namen `AWSRDSCustomIamRolePolicy` erstellt und fügt es der IAM-Rolle `AWSRDSCustomInstanceRole-`*`region`* hinzu. In diesem Beispiel wird davon ausgegangen, dass Sie die folgenden Umgebungsvariablen festgelegt haben:

`$REGION`

Setzen Sie diese Variable auf die Variable, AWS-Region in der Sie Ihre DB-Instance erstellen möchten.

`$ACCOUNT_ID`

Setzen Sie diese Variable auf Ihre AWS-Konto Nummer.

`$KMS_KEY`

Legen Sie diese Variable auf den Amazon-Ressourcennamen (ARN) des AWS KMS key fest, den Sie für Ihre DB-Instances von RDS Custom verwenden möchten. Um mehr als einen KMS-Schlüssel anzugeben, fügen Sie ihn dem`Resources`Abschnitt der Anweisungs-ID (Sid) 11.

```
aws iam put-role-policy \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --policy-name AWSRDSCustomIamRolePolicy \
  --policy-document '{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:GetManifest",
                "ssm:GetParameter",
                "ssm:GetParameters",
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:PutConfigurePackageResult",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation",
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "2",
            "Effect": "Allow",
            "Action": [
                "ec2messages:AcknowledgeMessage",
                "ec2messages:DeleteMessage",
                "ec2messages:FailMessage",
                "ec2messages:GetEndpoint",
                "ec2messages:GetMessages",
                "ec2messages:SendReply"
            ],
            "Resource": [
                "*"
```

```
            ]
        },
        {
            "Sid": "3",
            "Effect": "Allow",
            "Action": [
                "logs:PutRetentionPolicy",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:'$REGION':'$ACCOUNT_ID':log-group:rds-custom-instance*"
            ]
        },
        {
            "Sid": "4",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::do-not-delete-rds-custom-*/*"
            ]
        },
        {
            "Sid": "5",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "RDSCustomForOracle/Agent"
                    ]
                }
```

```
            }
        },
        {
            "Sid": "6",
            "Effect": "Allow",
            "Action": [
                "events:PutEvents"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "7",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:do-not-delete-
rds-custom-*",
                "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:rds-custom!
oracle-do-not-delete-*"
            ]
        },
        {
           "Sid": "8",
           "Effect": "Allow",
           "Action": [
             "s3:ListBucketVersions"
           ],
           "Resource": [
             "arn:aws:s3:::do-not-delete-rds-custom-*"
           ]
         },
         {
            "Sid": "9",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            }
          },
          {
            "Sid": "10",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*"
            ]
          },
          {
            "Sid": "11",
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKey"
            ],
            "Resource": [
              "arn:aws:kms:'$REGION':'$ACCOUNT_ID':key/'$KMS_KEY'"
            ]
          },
          {
            "Sid": "12",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ec2:CreateAction": [
                        "CreateSnapshots"
                    ]
                }
            }
        },
        {
            "Sid": "13",
            "Effect": "Allow",
            "Action": [
                "sqs:SendMessage",
                "sqs:ReceiveMessage",
```

```
                "sqs:DeleteMessage",
                "sqs:GetQueueUrl"
            ],
            "Resource": "arn:aws:sqs:'$REGION':'$ACCOUNT_Id':do-not-delete-rds-custom-
*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            }
        }
    ]
}'
```

Schritt 3: Erstellen Sie das benutzerdefinierte RDS-Instanzprofil AWSRDSCustom InstanceProfile

Ein Instance-Profil ist ein Container, der eine einzelne IAM-Rolle enthält. RDS Custom verwendet das Instance-Profil, um die Rolle der Instance zu übergeben.

Wenn Sie zum Erstellen einer Rolle die CLI verwenden, erstellen Sie die Rolle und das Instance-Profil als separate Aktionen, deren Namen verschieden sein können. Erstellen Sie Ihr IAM-Instance-Profil wie folgt und verwenden Sie das Namensformat `AWSRDSCustomInstanceProfile-`*`region`*. Im folgenden Beispiel wird davon ausgegangen, dass Sie die Umgebungsvariable `$REGION` auf die Umgebungsvariable gesetzt haben, AWS-Region in der Sie Ihre DB-Instance erstellen möchten.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION
```

Schritt 4: Hinzufügen AWSRDSCustom InstanceRoleForRdsCustomInstance zu AWSRDSCustom InstanceProfile

Fügen Sie Ihre IAM-Rolle dem Instance-Profil hinzu, das Sie zuvor erstellt haben. Im folgenden Beispiel wird davon ausgegangen, dass Sie die Umgebungsvariable `$REGION` auf die Umgebungsvariable gesetzt haben, AWS-Region in der Sie Ihre DB-Instance erstellen möchten.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION \
    --role-name AWSRDSCustomInstanceRole-$REGION
```

Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen

Stellen Sie sicher, dass der IAM-Prinzipal (Benutzer oder Rolle), der die benutzerdefinierte CEV- oder RDS-DB-Instance erstellt, über eine der folgenden Richtlinien verfügt:

- Die Richtlinie `AdministratorAccess`
- Die `AmazonRDSFullAccess` Richtlinie mit den erforderlichen Berechtigungen für Amazon S3 und AWS KMS CEV-Erstellung und DB-Instance-Erstellung

Themen

- Erforderliche IAM-Berechtigungen für Amazon S3 und AWS KMS
- Erforderliche IAM-Berechtigungen zum Erstellen einer CEV
- Erforderliche IAM-Berechtigungen zum Erstellen einer DB-Instance aus einer CEV

Erforderliche IAM-Berechtigungen für Amazon S3 und AWS KMS

Um RDS Custom für Oracle-DB-Instances zu erstellen CEVs , muss Ihr IAM-Principal auf Amazon S3 und AWS KMS zugreifen. Die folgende Beispiel-JSON-Richtlinie gewährt die erforderlichen Berechtigungen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
```

```
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Weitere Informationen zu `kms:CreateGrant`-Berechtigung finden Sie unter AWS KMS key Verwaltung.

Erforderliche IAM-Berechtigungen zum Erstellen einer CEV

Um ein CEV zu erstellen, benötigt Ihr IAM-Principal die folgenden zusätzlichen Berechtigungen:

```
s3:GetObjectAcl
s3:GetObject
s3:GetObjectTagging
s3:ListBucket
mediaimport:CreateDatabaseBinarySnapshot
```

Die folgende JSON-Beispielrichtlinie gewährt die zusätzlichen Berechtigungen, die für den Zugriff auf den Bucket *my-custom-installation-files* und seinen Inhalt erforderlich sind.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToS3MediaBucket",
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectAcl",
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::my-custom-installation-files",
                "arn:aws:s3:::my-custom-installation-files/*"
            ]
        },
        {
            "Sid": "PermissionForByom",
            "Effect": "Allow",
            "Action": [
                "mediaimport:CreateDatabaseBinarySnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

Sie können den Konten von Anrufern ähnliche Berechtigungen für Amazon S3 mit einer S3-Bucket-Richtlinie erteilen.

Erforderliche IAM-Berechtigungen zum Erstellen einer DB-Instance aus einer CEV

Um eine RDS Custom for Oracle DB-Instance aus einem vorhandenen CEV zu erstellen, benötigt der IAM-Principal die folgenden zusätzlichen Berechtigungen.

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
```

Die folgende JSON-Beispielrichtlinie gewährt die Berechtigungen, die für die Validierung einer IAM-Rolle und für die Protokollierung von Informationen in AWS CloudTrail erforderlich sind.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
```

```
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        }
    ]
}
```

Schritt 6: Konfigurieren Sie Ihre VPC für RDS Custom for Oracle

Ihre benutzerdefinierte RDS-DB-Instance befindet sich in einer Virtual Private Cloud (VPC), die auf dem Amazon VPC-Service basiert, genau wie eine EC2 Amazon-Instance oder Amazon RDS-Instance. Sie stellen Ihre eigene VPC zur Verfügung und konfigurieren sie. Im Gegensatz zu RDS Custom für SQL Server erstellt RDS Custom für Oracle keine Zugriffssteuerungsliste oder Sicherheitsgruppen. Sie müssen Ihre eigenen Sicherheitsgruppen, Subnetze und Routing-Tabellen anfügen.

Sie können Ihre Virtual Private Cloud (VPC) entweder manuell CloudFormation oder manuell konfigurieren.

> **Important**
>
> Wir empfehlen dringend, dass Sie Ihre RDS Custom for Oracle-Umgebung mithilfe von AWS CloudFormation konfigurieren. Diese Methode ist am einfachsten und am wenigsten fehleranfällig.

Themen

- Konfigurieren Sie Ihre VPC mit CloudFormation (empfohlen)
- Konfigurieren Sie Ihre VPC manuell für RDS Custom for Oracle

Konfigurieren Sie Ihre VPC mit CloudFormation (empfohlen)

Wenn Sie Ihre VPC bereits für eine andere RDS-Custom-Engine konfiguriert haben und die vorhandene VPC wiederverwenden möchten, überspringen Sie diesen Schritt. In dieser Abbildung wird von Folgendem ausgegangen:

- Sie haben Ihr IAM-Instanzprofil und Ihre Rolle bereits erstellt. CloudFormation
- Sie kennen die ID Ihrer Routing-Tabelle.

  Damit eine DB-Instance privat ist, muss sie sich in einem privaten Subnetz befinden. Damit ein Subnetz privat ist, darf es nicht einer Routing-Tabelle zugeordnet sein, die über ein Standard-Internet-Gateway verfügt. Weitere Informationen finden Sie unter Konfigurieren von Routing-Tabellen im Benutzerhandbuch zu Amazon VPC.

Wenn Sie die CloudFormation Vorlage für Ihre VPC verwenden, werden die folgenden Ressourcen erstellt:

- Eine private VPC
- Eine Subnetzgruppe mit dem Namen `rds-custom-private`
- Die folgenden VPC-Endpunkte, die Ihre DB-Instance für die Kommunikation mit abhängigen Geräten verwendet: AWS-Services
  - `com.amazonaws.region.ec2messages`
  - `com.amazonaws.region.events`
  - `com.amazonaws.region.logs`
  - `com.amazonaws.region.monitoring`
  - `com.amazonaws.region.s3`
  - `com.amazonaws.region.secretsmanager`
  - `com.amazonaws.region.ssm`
  - `com.amazonaws.region.ssmmessages`

  Wenn Sie eine Multi-AZ-Bereitstellung erstellen:

  - `com.amazonaws.region.sqs`

> **Note**
>
> Für eine komplexe Netzwerkkonfiguration mit vorhandenen Konten empfehlen wir, den Zugriff auf abhängige Dienste manuell zu konfigurieren, falls der Zugriff noch nicht besteht. Weitere Informationen finden Sie unter Stellen Sie sicher, dass Ihre VPC auf abhängige zugreifen kann AWS-Services.

So konfigurieren Sie Ihre VPC mit CloudFormation

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Starten Sie den Assistenten zum Erstellen eines Stacks und wählen Sie Stack erstellen und dann Mit neuen Ressourcen (Standard) aus.
3. Gehen Sie auf der Seite Create stack (Stack erstellen) wie folgt vor:
   a. Wählen Sie unter Prepare template (Vorlage vorbereiten) den Wert Template is ready (Vorlage ist bereit) aus.
   b. Wählen Sie unter Templete source (Vorlagenquelle) den Wert Upload a template file (Vorlagendatei hochladen) aus.
   c. Für Datei auswählen, navigieren Sie dahin und wählen `custom-vpc.json` aus.
   d. Wählen Sie Weiter aus.
4. Führen Sie auf der Seite Specify DB Details (Festlegen von DB-Detail) die folgenden Schritte aus:
   a. Geben Sie unter Stack name (Stack-Name) **`custom-vpc`** ein.
   b. Für Parameter wählen Sie die privaten Subnetze aus, die für RDS Custom DB-Instances verwendet werden sollen.
   c. Wählen Sie die private VPC-ID aus, die für RDS Custom DB-Instances verwendet werden soll.
   d. Die Haupt-Routing-Tabelle, die dem privaten Subnetz zugeordnet ist.
   e. Wählen Sie Weiter aus.
5. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
6. Klicken Sie auf der Seite custom-vpc überprüfen auf Absenden.

CloudFormation konfiguriert Ihre private VPC. Wenn custom-vpc im linken Bereich CREATE_COMPLETE anzeigt, fahren Sie mit dem nächsten Schritt fort.

7. (Optional) Überprüfen Sie die Details Ihrer VPC. Wählen Sie im Bereich Stacks die Option custom-vpc aus. Führen Sie im rechten Bereich die folgenden Schritte aus:

   a. Wählen Sie Stack-Info aus. Ihr Stack hat eine ID im Format arn:aws:cloudformation: ::stack/ custom-vpc/. *region account-no identifier*

   b. Wählen Sie Resources aus. Sie sollten eine Subnetzgruppe mit dem Namen rds-custom-privateund mehrere VPC-Endpunkte sehen, die das Benennungsformat vpce- verwenden. ***string*** Jeder Endpunkt entspricht einem AWS-Service , mit dem RDS Custom kommunizieren muss. Weitere Informationen finden Sie unter Stellen Sie sicher, dass Ihre VPC auf abhängige zugreifen kann AWS-Services.

   c. Wählen Sie Parameter aus. Sie sollten die privaten Subnetze, die private VPC und die Routing-Tabelle sehen, die Sie bei der Erstellung des Stacks angegeben haben. Wenn Sie eine DB-Instance erstellen, müssen Sie die VPC-ID und die Subnetzgruppe angeben.

Konfigurieren Sie Ihre VPC manuell für RDS Custom for Oracle

Als Alternative zur Automatisierung der VPC-Erstellung mit AWS CloudFormation können Sie Ihre VPC manuell konfigurieren. Diese Option eignet sich möglicherweise am besten, wenn Sie über ein komplexes Netzwerk-Setup verfügen, das vorhandene Ressourcen nutzt.

Themen

- Stellen Sie sicher, dass Ihre VPC auf abhängige zugreifen kann AWS-Services
- Konfigurieren des Instance-Metadaten-Service

Stellen Sie sicher, dass Ihre VPC auf abhängige zugreifen kann AWS-Services

RDS Custom sendet Kommunikation von Ihrer DB-Instance an andere AWS-Services. Stellen Sie sicher, dass von dem Subnetz aus, in dem Sie Ihre benutzerdefinierten RDS-DB-Instances erstellen, auf die folgenden Dienste zugegriffen werden kann:

- Amazon CloudWatch (`com.amazonaws.`*`region`*`.monitoring`)
- CloudWatch Amazon-Protokolle (`com.amazonaws.`*`region`*`.logs`)
- CloudWatch Amazon-Veranstaltungen (`com.amazonaws.`*`region`*`.events`)

- Amazon EC2 (`com.amazonaws.region.ec2undcom.amazonaws.region.ec2messages`)
- Amazon EventBridge (`com.amazonaws.region.events`)
- Amazon S3 (`com.amazonaws.region.s3`)
- AWS Secrets Manager (`com.amazonaws.region.secretsmanager`)
- AWS Systems Manager (`com.amazonaws.region.ssmundcom.amazonaws.region.ssmmessages`)

Wenn Sie Multi-AZ-Bereitstellungen erstellen

- Amazon Simple Queue Service (`com.amazonaws.region.sqs`)

Wenn RDS Custom nicht mit den erforderlichen Diensten kommunizieren kann, veröffentlicht es die folgenden Ereignisse:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS
 services. Make sure port 443 (HTTPS) allows outbound connections, and try again.
 "Failed to connect to the following services: s3 events"
```

Um `incompatible-network` Fehler zu vermeiden, stellen Sie sicher, dass die VPC-Komponenten, die an der Kommunikation zwischen Ihrer RDS Custom DB-Instance beteiligt sind, die folgenden Anforderungen AWS-Services erfüllen:

- Die DB-Instance kann ausgehende Verbindungen an Port 443 mit anderen AWS-Services herstellen.
- Die VPC lässt eingehende Antworten auf Anfragen zu, die von Ihrer DB-Instance von RDS Custom stammen.
- RDS Custom kann die Domain-Namen von Endpunkten für jeden AWS-Service korrekt auflösen.

Wenn Sie eine VPC bereits für eine andere DB-Engine von RDS Custom konfiguriert haben, können Sie diese VPC wiederverwenden und diesen Prozess überspringen.

Konfigurieren des Instance-Metadaten-Service

Stellen Sie folgendermaßen sicher, dass die EC2-Instance eine Verbindung zu herstellen kann:

- Greifen Sie mithilfe des Instanz-Metadatendienstes Version 2 (IMDSv2) auf den Instanz-Metadatendienst zu.
- Lassen Sie ausgehende Kommunikation über Port 80 (HTTP) zur IMDS-Link-IP-Adresse zu.
- Fordern Sie `http://169.254.169.254` Instanz-Metadaten über den IMDSv2 Link an.

Weitere Informationen finden Sie unter Verwendung IMDSv2 im EC2 Amazon-Benutzerhandbuch.

RDS Custom for Oracle Automation verwendet IMDSv2 standardmäßig per Einstellung `HttpTokens=enabled` auf der zugrunde liegenden EC2 Amazon-Instance. Sie können es jedoch verwenden, IMDSv1 wenn Sie möchten. Weitere Informationen finden Sie unter Konfiguration der Instance-Metadatenoptionen im EC2 Amazon-Benutzerhandbuch.

Arbeiten mit benutzerdefinierten Engine-Versionen für Amazon RDS Custom für Oracle

EINbenutzerdefinierte Engine-Version (CEV)für Amazon RDS Custom for Oracle ist ein binärer Volume-Snapshot eines Datenbankmoduls und eines bestimmten Amazon Machine Image (AMI). RDS Custom für Oracle verwendet standardmäßig das neueste verfügbare AMI, das von RDS Custom verwaltet wird. Sie können jedoch auch ein AMI angeben, das in einer vorherigen CEV verwendet wurde. Sie speichern Ihre Datenbankinstallationsdateien in Amazon S3. RDS Custom verwendet die Installationsdateien und das AMI, um Ihre CEV für Sie zu erstellen.

Themen

- Vorbereiten der Erstellung einer CEV
- Erstellen einer CEV
- Ändern des CEV-Status
- CEVDetails für Amazon RDS Custom for Oracle anzeigen
- Löschen einer CEV

Vorbereiten der Erstellung einer CEV

Um eine CEV zu erstellen, greifen Sie auf die Installationsdateien und Patches zu, die in Ihrem Amazon S3-Bucket für eine der folgenden Releases gespeichert sind:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1)

Sie können beispielsweise den April 2021 RU/RUR für Oracle Database 19c oder eine beliebige gültige Kombination von Installationsdateien und Patches verwenden. Weitere Informationen zu den von RDS Custom fü Oracle unterstützten Versionen und Regionen finden Sie unter RDS Custom mit RDS für Oracle.

Themen

- Schritt 1 (optional): Herunterladen der Manifestvorlagen

- Schritt 2: Herunterladen Ihrer Datenbankinstallationsdateien und Patches von Oracle Software Delivery Cloud
- Schritt 3: Hochladen Ihrer Installationsdateien in Amazon S3
- Schritt 4 (optional): Teilen Sie Ihre Installationsmedien in S3 auf AWS-Konten
- Schritt 5: Vorbereiten des CEV-Manifests
- Schritt 6 (optional): Validieren des CEV-Manifests
- Schritt 7: Hinzufügen der erforderlichen IAM-Berechtigungen

Schritt 1 (optional): Herunterladen der Manifestvorlagen

Ein CEV-Manifest ist ein JSON-Dokument, das die Liste der .zip-Dateien zur Datenbankinstallation für Ihre CEV enthält. Gehen Sie wie folgt vor, um eine CEV zu erstellen:

1. Identifizieren Sie die Installationsdateien der Oracle-Datenbank, die Sie in Ihre CEV aufnehmen möchten.
2. Laden Sie die Installationsdateien herunter.
3. Erstellen Sie ein JSON-Manifest, das die Installationsdateien auflistet.

RDS Custom für Oracle bietet JSON-Manifestvorlagen mit unseren empfohlenen .zip-Dateien für jede unterstützte Oracle Database Version. Die folgende Vorlage ist beispielsweise für 19.17.0.0.0 RU.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
```

```
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Jeder Vorlage ist eine Readme-Datei zugeordnet, die Anweisungen zum Herunterladen der Patches, URLs für die ZIP-Dateien und Dateiprüfsummen enthält. Sie können diese Vorlagen unverändert verwenden oder sie mit Ihren eigenen Patches ändern. Um die Vorlagen zu überprüfen, laden Sie custom-oracle-manifest.zip auf Ihre lokale Festplatte herunter und öffnen Sie sie dann mit einer Dateiarchivierungsanwendung. Weitere Informationen finden Sie unter Schritt 5: Vorbereiten des CEV-Manifests.

Schritt 2: Herunterladen Ihrer Datenbankinstallationsdateien und Patches von Oracle Software Delivery Cloud

Wenn Sie die Installationsdateien identifiziert haben, die Sie für Ihre CEV benötigen, laden Sie sie auf Ihr lokales System herunter. Die Installationsdateien und Patches der Oracle Database werden in der Oracle Software Delivery Cloud gehostet. Für jede CEV ist eine Basisversion, wie Oracle Database 19c oder Oracle Database 12c Release 2 (12.2), und eine optionale Liste mit Patches erforderlich.

So laden Sie die Installationsdateien der Datenbank für Oracle Database herunter

1. Wechseln Sie zuhttps://edelivery.oracle.com/und melden Sie sich an.
2. Geben Sie in das Suchfeld **`Oracle Database Enterprise Edition`** oder ein **`Oracle Database Standard Edition 2`** und wählen Sie Suchen.
3. Wählen Sie eine der folgenden Basisversionen aus:

Datenbankversion	Enterprise Edition	Standard-Edition 2
Oracle Database 19c	DLP: Oracle Database 19c Enterprise Edition 19.3.0.0.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database 19c Standard Edition 2 19.3.0.0.0 (Oracle Datenbank Standard Edition 2)

Datenbankversion	Enterprise Edition	Standard-Edition 2
Oracle Database 18c	DLP: Oracle Database 18c Enterprise Edition 18.0.0.0.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 18.0.0.0.0 (Oracle Database Standard Edition 2)
Oracle Database 12c Release 2 (12.2.0.1)	DLP: Oracle Database 12c Enterprise Edition 12.2.0.1.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 12.2.0.1.0 (Oracle Database Standard Edition 2)
Oracle Database 12c Release 1 (12.1.0.2)	DLP: Oracle Database 12c Enterprise Edition 12.1.0.2.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 12.1.0.2.0 (Oracle Database Standard Edition 2)

4. Klicken Sie auf Weiter.
5. Deaktivieren Sie das Kontrollkästchen Warteschlange herunterladen.
6. Wählen Sie die Option aus, die Ihrer Basisversion entspricht:
   - Oracle Database 19.3.0.0.0 – Long Term Release.
   - Oracle Database 18.0.0.0.0
   - Oracle Database 12.2.0.1.0.
   - Oracle Database 12.1.0.2.0.
7. Klicken Sie auf86-64 LinuxinPlattform/Sprachenaus.
8. Wählen Sie Weiter und unterzeichnen Sie dann die Oracle-Lizenzvereinbarung.
9. Wählen Sie die .zip-Datei aus, die Ihrer Datenbankversion entspricht:

Datenbankversion und Edition	Zip-Dateien	SHA-256-Hash
19c EE und SE2	V982063-01.zip	`BA8329C757133DA313ED3B6D7F86C5AC42CD9970A28BF2E6233F3235233AA8D8`

Datenbankversion und Edition	Zip-Dateien	SHA-256-Hash
18c EE und SE2	V978967-01.zip	C96A4FD768787AF98272008833FE10B17269 1CF84E42816B138C12D4DE63AB96
12.2.0.1 EE und SE2	V839960-01.zip	96ED97D21F15C1AC0CCE3749DA6C3DAC7059 BB60672D76B008103FC754D22DDE
12.1.0.2 EE	V46095-01_1of2.zip V46095-01_2of2.zip	31FDC2AF41687B4E547A3A18F796424D8C1A F36406D2160F65B0AF6A9CD47355 für V46095-01_1of2.zip 03DA14F5E875304B28F0F3BB02AF0EC33227 885B99C9865DF70749D1E220ACCD für V46095-01_2of2.zip
12.1.0.2 SE2	V77388-01_1of2.zip V77388-01_2of2.zip	73873369753230F5A0921F95ACEADB591388 CB06ED72A7F3AEA7BCBCEA2403BC für V77388-01_1of2.zip 2492E1BE1E3E3531DA83D0843C09C08E435A C8CEFD9A00C0DF56BE4F15CEEBF3 für V77388-01_2of2.zip

10. Laden Sie Ihre gewünschten Oracle-Patches von `updates.oracle.com` oder `support.oracle.com` auf Ihr lokales System herunter. Sie finden die URLs für die Patches an den folgenden Orten:

    - Die Readme-Dateien sind in der .zip-Datei, die Sie in Schritt 1 (optional): Herunterladen der Manifestvorlagen heruntergeladen haben, enthalten.
    - Die Patches sind in jedem Versions-Update (RU) in den Versionshinweisen für Amazon Relational Database Service (Amazon RDS) für Oracle aufgelistet.

Schritt 3: Hochladen Ihrer Installationsdateien in Amazon S3

Hochladen Ihrer Oracle-Installations- und Patch-Dateien in Amazon S3 unter Verwendung der AWS CLI aus. Der S3-Bucket, der Ihre Installationsdateien enthält, muss sich in derselben AWS Region wie Ihr CEV befinden.

Für die Beispiele in diesem Abschnitt werden die folgenden Platzhalter verwendet:

- *`install-or-patch-file.zip`*— Oracle-Installationsmediendatei Zum Beispiel ist p32126828_190000_Linux-x86-64.zip ein Patch.
- *`amzn-s3-demo-destination-bucket`* – Ihr Amazon S3-Bucket, der für Ihre hochgeladenen Installationsdateien vorgesehen ist.
- *`123456789012/cev1`* – Ein optionales Präfix in Ihrem Amazon-S3-Bucket.
- *`amzn-s3-demo-source-bucket`* – Ein Amazon-S3-Bucket, in dem Sie Dateien optional inszenieren können.

Themen

- Schritt 3a: Stellen Sie sicher, dass sich Ihr S3-Bucket im richtigen befindet AWS-Region
- Schritt 3b: Sicherstellen, dass Ihre S3-Bucket-Richtlinie über die richtigen Berechtigungen verfügt
- Schritt 3c: Hochladen Ihrer Dateien mit den Befehlen cp oder sync
- Schritt 3d: Auflisten der Dateien in Ihrem S3-Bucket

Schritt 3a: Stellen Sie sicher, dass sich Ihr S3-Bucket im richtigen befindet AWS-Region

Stellen Sie sicher, dass sich Ihr S3-Bucket in der AWS Region befindet, in der Sie den `create-custom-db-engine-version` Befehl ausführen möchten.

```
aws s3api get-bucket-location --bucket amzn-s3-demo-destination-bucket
```

Schritt 3b: Sicherstellen, dass Ihre S3-Bucket-Richtlinie über die richtigen Berechtigungen verfügt

Sie können eine CEV ohne Vorgabe oder anhand einer Quell-CEV erstellen. Wenn Sie vorhaben, ein neues CEV aus dem Quellcode zu erstellen CEVs, stellen Sie sicher, dass Ihre S3-Bucket-Richtlinie über die richtigen Berechtigungen verfügt:

1. Identifizieren Sie den von RDS Custom reservierten S3-Bucket. Der Bucket-Name weist das Format `do-not-delete-rds-custom-`*`account`*`-`*`region`*`-`*`string`* auf. Der Bucket-Name kann

beispielsweise `do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE` lauten.

2. Stellen Sie sicher, dass die folgende Berechtigung an Ihre S3-Bucket-Richtlinie angehängt ist. Ersetzen Sie `do-not-delete-rds-custom-123456789012-us-east-1-abc123EXAMPLE` durch den Namen von Ihrem Bucket.

```
{
    "Sid": "AWSRDSCustomForOracleCustomEngineVersionGetObject",
    "Effect": "Allow",
    "Principal": {
        "Service": "custom.rds.amazonaws.com"
    },
    "Action": [
        "s3:GetObject",
        "s3:GetObjectTagging"
    ],
    "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-
east-1-abc123EXAMPLE/CustomEngineVersions/*"
}, ...
```

Schritt 3c: Hochladen Ihrer Dateien mit den Befehlen cp oder sync

Wählen Sie eine der folgenden Optionen:

- Verwenden von`aws s3 cp`um eine einzelne ZIP-Datei hochzuladen.

  Laden Sie jede ZIP-Installationsdatei separat hoch. Kombinieren Sie die ZIP-Dateien nicht in einer ZIP-Datei.
- Verwenden von`aws s3 sync`um ein Verzeichnis hochzuladen.

Example

Im folgenden Beispiel wird *`install-or-patch-file.zip`* in den *`123456789012/cev1`*-Ordner im RDS Custom Amazon S3-Bucket hochgeladen. Führen Sie ein separates`aws s3`-Befehl für jeden .zip, den Sie hochladen möchten.

Für LinuxmacOS, oderUnix:

```
aws s3 cp install-or-patch-file.zip \
```

```
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Windows:

```
aws s3 cp install-or-patch-file.zip ^
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Example

Das folgende Beispiel lädt die Dateien in Ihrem lokalen *cev1* Ordner in den *123456789012/cev1* Ordner in Ihrem Amazon S3 S3-Bucket hoch.

Für LinuxmacOS, oderUnix:

```
aws s3 sync cev1 \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Windows:

```
aws s3 sync cev1 ^
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Example

Im folgenden Beispiel werden alle Dateien in *amzn-s3-demo-source-bucket* zum *123456789012/cev1*-Ordner in Ihrem Amazon-S3-Bucket hochgeladen.

Für LinuxmacOS, oderUnix:

```
aws s3 sync s3://amzn-s3-demo-source-bucket/ \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Windows:

```
aws s3 sync s3://amzn-s3-demo-source-bucket/ ^
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Schritt 3d: Auflisten der Dateien in Ihrem S3-Bucket

Im folgenden Beispiel wird der Befehl `s3 ls` verwendet, um die Dateien in Ihrem Amazon-S3-Bucket in RDS Custom aufzulisten.

```
aws s3 ls \
    s3://amzn-s3-demo-destination-bucket/123456789012/cev1/
```

Schritt 4 (optional): Teilen Sie Ihre Installationsmedien in S3 auf AWS-Konten

Für die Zwecke dieses Abschnitts ist der Amazon S3-Bucket, der Ihre hochgeladenen Oracle-Installationsdateien enthält, Ihr Medien-Bucket. Ihre Organisation verwendet möglicherweise mehrere AWS-Konten in einem AWS-Region. Wenn ja, möchten Sie vielleicht einen verwenden, um Ihren Medien-Bucket AWS-Konto zu füllen, und einen anderen, AWS-Konto um ihn zu erstellen CEVs. Wenn Sie Ihren Medienbucket nicht teilen möchten, fahren Sie mit dem nächsten Abschnitt fort.

In dieser Abbildung wird von Folgendem ausgegangen:

- Sie können auf das Konto zugreifen, mit dem Ihr Media-Bucket erstellt wurde, und auf ein anderes Konto, in dem Sie das Konto erstellen CEVs möchten.
- Sie beabsichtigen, nur CEVs in einem Konto zu erstellen AWS-Region. Wenn Sie mehrere Regionen verwenden möchten, erstellen Sie in jeder Region einen Medien-Bucket.
- Sie verwenden die CLI. Wenn Sie die Amazon S3-Konsole verwenden, passen Sie die folgenden Schritte an.

So konfigurieren Sie Ihren Media-Bucket für die gemeinsame Nutzung zwischen AWS-Konten

1. Melden Sie sich bei dem an AWS-Konto , der den S3-Bucket enthält, in den Sie Ihre Installationsmedien hochgeladen haben.
2. Beginnen Sie entweder mit einer leeren JSON-Richtlinienvorlage oder einer vorhandenen Richtlinie, die Sie anpassen können.

   Mit dem folgenden Befehl wird eine vorhandene Richtlinie abgerufen und unter gespeichert. *my-policy.json* In diesem Beispiel wird der S3-Bucket benannt*amzn-s3-demo-bucket*, der Ihre Installationsdateien enthält.

   ```
   aws s3api get-bucket-policy \
       --bucket amzn-s3-demo-bucket \
       --query Policy \
       --output text > my-policy.json
   ```

3. Bearbeiten Sie die Media Bucket-Berechtigungen wie folgt:

- Im `Resource`-Element Ihrer Vorlage geben Sie den S3-Bucket an, in den Sie Ihre Oracle Database-Installationsdateien hochgeladen haben.
- Geben Sie im `Principal` Element all das an AWS-Konten , das Sie ARNs für die Erstellung verwenden möchten CEVs. Sie können den Stamm, einen Benutzer oder eine Rolle zur Zulassungsliste des S3-Buckets hinzufügen. Weitere Informationen finden Sie unter IAM-IDs im AWS Identity and Access Management -Benutzerhandbuch.

JSON

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "GrantAccountsAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:user/user-name-with-path",
                    "arn:aws:iam::123456789012:role/role-name-with-path"
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

4. Fügen Sie die Richtlinie zu Ihrem Bucket hinzu.

Im folgenden Beispiel *amzn-s3-demo-bucket* ist dies der Name des S3-Buckets, der Ihre Installationsdateien enthält, sowie *my-policy.json* der Name Ihrer JSON-Datei.

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://my-policy.json
```

5. Melden Sie sich bei einem an, AWS-Konto in dem Sie etwas erstellen möchten CEVs.
6. Stellen Sie sicher, dass dieses Konto auf den Media-Bucket in dem Konto zugreifen kann AWS-Konto , von dem es erstellt wurde.

   ```
   aws s3 ls --query "Buckets[].Name"
   ```

   Weitere Informationen finden Sie unter aws s3 ls in der AWS CLI Befehlsreferenz.
7. Erstellen Sie ein CEV, indem Sie die Schritte unter Erstellen einer CEV ausführen.

Schritt 5: Vorbereiten des CEV-Manifests

Ein CEV-Manifest ist ein JSON-Dokument, das Folgendes enthält:

- (Erforderlich) Die Liste der ZIP-Installationsdateien, die Sie auf Amazon S3 hochgeladen haben. RDS Custom wendet die Patches in der Reihenfolge an, in der sie im Manifest aufgelistet sind.
- (Optional) Installationsparameter, die nicht standardmäßige Werte für die Oracle-Basis, das Oracle-Standardverzeichnis sowie die ID und den Namen des UNIX/Linux Benutzers und der Gruppe festlegen. Beachten Sie, dass Sie die Installationsparameter für eine bestehende CEV oder DB-Instance nicht ändern können. Sie können auch nicht von einer CEV auf eine andere CEV aktualisieren, wenn die Installationsparameter unterschiedliche Einstellungen haben.

Beispiele für CEV-Manifeste finden Sie in den JSON-Vorlagen, die Sie in Schritt 1 (optional): Herunterladen der Manifestvorlagen heruntergeladen haben. Sie können sich die Beispiele auch unter CEV-Manifest-Beispiele ansehen.

Themen

- JSON-Felder im CEV-Manifest
- Erstellen des CEV-Manifests
- CEV-Manifest-Beispiele

JSON-Felder im CEV-Manifest

Die folgende Tabelle beschreibt die JSON-Felder im Manifest.

JSON-Feld	Beschreibung
`MediaImportTemplateVersion`	Version des CEV-Manifests. Das Datum muss im Format `YYYY-MM-DD` angegeben werden.
`databaseInstallationFileNames`	Liste der Installationsdateien für die Datenbank geordnet.
`opatchFileNames`	Geordnete Liste der OPatch Installationsprogramme, die für die Oracle-DB-Engine verwendet werden. Es ist nur ein Wert gültig. Werte für`opatchFileNames` muss mit beginnen`p6880880_` aus.
`psuRuPatchFileNames`	Die PSU- und RU-Patches für diese Datenbank. **Important** Wenn Sie `psuRuPatchFileNames` mit einschließen, ist `opatchFileNames` erforderlich. Werte für`opatchFileNames` muss mit beginnen`p6880880_` aus.
`OtherPatchFileNames`	Die Patches, die nicht in der Liste der PSU- und RU-Patches enthalten sind. RDS Custom wendet diese Patches an, nachdem die PSU- und RU-Patches angewendet wurden. **Important** Wenn Sie `OtherPatchFileNames` mit einschließen, ist `opatchFileNames` erforderlich. Werte für`opatchFileNames` muss mit beginnen`p6880880_` aus.

JSON-Feld	Beschreibung
`installationParameters`	Nicht standardmäßige Einstellungen für die Oracle-Basis, das Oracle-Standardverzeichnis sowie die ID und den Namen des UNIX/Linux Benutzers und der Gruppe. Sie können die folgenden Parameter festlegen: oracleBase Das Verzeichnis, in dem Ihre Oracle-Binärdateien installiert sind. Es ist der Mountingpunkt des Binärvolumes, in dem Ihre Dateien gespeichert sind. Das Oracle-Basisverzeichnis kann mehrere Oracle-Standardverzeichnisse enthalten. Wenn es sich bei `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1` beispielsweise um eines Ihrer Oracle-Standardverzeichnisse handelt, ist `/home/oracle` das Oracle-Basisverzeichnis. Ein benutzerdefiniertes Oracle-Basisverzeichnis ist kein symbolischer Link. Wenn Sie keine Oracle-Basis angeben, ist das Standardverzeichnis `/rdsdbbin` . oracleHome Das Verzeichnis, in dem Ihre Oracle-Datenbankbinärdateien installiert sind. Wenn Sie beispielsweise `/home/oracle/` als Ihre Oracle-Basis angeben, können Sie `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1/` als Ihr Oracle-Standardverzeichnis angeben. Ein benutzerdefiniertes Oracle-Standardverzeichnis ist kein symbolischer Link. Der Oracle-Standardwert wird durch die Umgebungsvariable `$ORACLE_HOME` referenziert. Wenn Sie kein Oracle-Standardverzeichnis angeben, ist das Standardnamensformat `/rdsdbbin/oracle.` *`major-engine-version`* `.custom.r1.` *`engine-edition`* `.1`.

JSON-Feld	Beschreibung
	unixUname Der Name des UNIX-Benutzers, der die Oracle-Software besitzt. RDS Custom geht bei der Ausführung lokaler Datenbankbefehle von diesem Benutzer aus. Wenn Sie sowohl `unixUid` als auch `unixUname` angeben, erstellt RDS Custom den Benutzer, falls er nicht existiert, und weist dem Benutzer dann die UID zu, wenn sie nicht mit der ursprünglichen UID übereinstimmt. Der Standardbenutzername ist `rdsdb`. unixUid Die ID (UID) des UNIX-Benutzers, der die Oracle-Software besitzt. Wenn Sie sowohl `unixUid` als auch `unixUname` angeben, erstellt RDS Custom den Benutzer, falls er nicht existiert, und weist dem Benutzer dann die UID zu, wenn sie nicht mit der ursprünglichen UID übereinstimmt. Die Standard-UID ist `61001`. Dies ist die UID des Benutzers `rdsdb`. unixGroupName Der Name der UNIX-Gruppe. Der UNIX-Benutzer, der die Oracle-Software besitzt, ist Mitglied dieser Gruppe. Der Standardgruppenname lautet `rdsdb`. unixGroupId Die ID der UNIX-Gruppe, zu der der UNIX-Benutzer gehört. Der Standardgruppen-ID lautet `1000`. Dies ist die ID der Gruppe `rdsdb`.

Jede Oracle Database-Version hat eine andere Liste unterstützter Installationsdateien. Achten Sie beim Erstellen Ihres CEV-Manifests darauf, nur Dateien anzugeben, die von RDS Custom for Oracle

unterstützt werden. Andernfalls schlägt die CEV-Erstellung mit einem Fehler fehl. Alle Patches, die in den Versionshinweisen für Amazon Relational Database Service (Amazon RDS) für Oracle aufgelistet sind, werden unterstützt.

Erstellen des CEV-Manifests

So erstellen Sie ein CEV-Manifest

1. Listen Sie alle Installationsdateien auf, die Sie anwenden möchten, in der Reihenfolge, in der sie angewendet werden sollen.
2. Korrelieren Sie die Installationsdateien mit den unter JSON-Felder im CEV-Manifest beschriebenen JSON-Feldern.
3. Führen Sie eine der folgenden Aufgaben aus:
   - Erstellen Sie das CEV-Manifest als JSON-Textdatei.
   - Bearbeiten Sie die CEV-Manifestvorlage, wenn Sie die CEV in der Konsole erstellen. Weitere Informationen finden Sie unter Erstellen einer CEV.

CEV-Manifest-Beispiele

Die folgenden Beispiele zeigen CEV-Manifestdateien für verschiedene Oracle Database-Releases. Wenn Sie ein JSON-Feld in Ihr Manifest einschließen, stellen Sie sicher, dass es leer ist. So ist beispielsweise das folgende CEV-Manifest ungültig, da `otherPatchFileNames` leer ist.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
    ]
}
```

Topics

- Sample CEV manifest for Oracle Database 12c Release 1 (12.1)
- Sample CEV manifest for Oracle Database 12c Release 2 (12.2)
- Sample CEV manifest for Oracle Database 18c
- Sample CEV manifest for Oracle Database 19c

Example Beispiel für ein CEV-Manifests für Oracle Database 12c Release 1 (12.1)

In dem folgenden Beispiel für das PSU vom Juli 2021 für Oracle Database 12c Release 1 (12.1) wendet RDS Custom die Patches in der angegebenen Reihenfolge an. Somit wendet RDS Custom p32768233, dann p32876425, dann p18759211 usw. an. In dem Beispiel werden neue Werte für den UNIX-Benutzer und die UNIX-Gruppe sowie für das Oracle-Standardverzeichnis und die Oracle-Basis festgelegt.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V46095-01_1of2.zip",
        "V46095-01_2of2.zip"
    ],
    "opatchFileNames":[
        "p6880880_121010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32768233_121020_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p32876425_121020_Linux-x86-64.zip",
        "p18759211_121020_Linux-x86-64.zip",
        "p19396455_121020_Linux-x86-64.zip",
        "p20875898_121020_Linux-x86-64.zip",
        "p22037014_121020_Linux-x86-64.zip",
        "p22873635_121020_Linux-x86-64.zip",
        "p23614158_121020_Linux-x86-64.zip",
        "p24701840_121020_Linux-x86-64.zip",
        "p25881255_121020_Linux-x86-64.zip",
        "p27015449_121020_Linux-x86-64.zip",
        "p28125601_121020_Linux-x86-64.zip",
        "p28852325_121020_Linux-x86-64.zip",
        "p29997937_121020_Linux-x86-64.zip",
        "p31335037_121020_Linux-x86-64.zip",
        "p32327201_121020_Linux-x86-64.zip",
```

```
        "p32327208_121020_Generic.zip",
        "p17969866_12102210119_Linux-x86-64.zip",
        "p20394750_12102210119_Linux-x86-64.zip",
        "p24835919_121020_Linux-x86-64.zip",
        "p23262847_12102201020_Linux-x86-64.zip",
        "p21171382_12102201020_Generic.zip",
        "p21091901_12102210720_Linux-x86-64.zip",
        "p33013352_12102210720_Linux-x86-64.zip",
        "p25031502_12102210720_Linux-x86-64.zip",
        "p23711335_12102191015_Generic.zip",
        "p19504946_121020_Linux-x86-64.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.1.0.2",
        "oracleBase": "/home/oracle"
    }
}
```

Example Beispiel für ein CEV-Manifests für Oracle Database 12c Release 2 (12.2)

Im folgenden Beispiel für das PSU vom Oktober 2021 für Oracle Database 12c Release 2 (12.2) wendet RDS Custom p33261817 an, dann p33192662, dann p29213893 und so weiter. In dem Beispiel werden neue Werte für den UNIX-Benutzer und die UNIX-Gruppe sowie für das Oracle-Standardverzeichnis und die Oracle-Basis festgelegt.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V839960-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_122010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p33261817_122010_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p33192662_122010_Linux-x86-64.zip",
        "p29213893_122010_Generic.zip",
```

```
        "p28730253_122010_Linux-x86-64.zip",
        "p26352615_12201211019DBOCT2021RU_Linux-x86-64.zip",
        "p23614158_122010_Linux-x86-64.zip",
        "p24701840_122010_Linux-x86-64.zip",
        "p25173124_122010_Linux-x86-64.zip",
        "p25881255_122010_Linux-x86-64.zip",
        "p27015449_122010_Linux-x86-64.zip",
        "p28125601_122010_Linux-x86-64.zip",
        "p28852325_122010_Linux-x86-64.zip",
        "p29997937_122010_Linux-x86-64.zip",
        "p31335037_122010_Linux-x86-64.zip",
        "p32327201_122010_Linux-x86-64.zip",
        "p32327208_122010_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.2.0.1",
        "oracleBase": "/home/oracle"
    }
}
```

Example Beispiel-CEV-Manifests für Oracle Database 18c

Im folgenden Beispiel für das PSU vom Oktober 2021 für Oracle Database 18c wendet RDS Custom p32126855 an, dann p28730253, dann p27539475 und so weiter. In dem Beispiel werden neue Werte für den UNIX-Benutzer und die UNIX-Gruppe sowie für das Oracle-Standardverzeichnis und die Oracle-Basis festgelegt.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V978967-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_180000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32126855_180000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
```

```
        "p28730253_180000_Linux-x86-64.zip",
        "p27539475_1813000DBRU_Linux-x86-64.zip",
        "p29213893_180000_Generic.zip",
        "p29374604_1813000DBRU_Linux-x86-64.zip",
        "p29782284_180000_Generic.zip",
        "p28125601_180000_Linux-x86-64.zip",
        "p28852325_180000_Linux-x86-64.zip",
        "p29997937_180000_Linux-x86-64.zip",
        "p31335037_180000_Linux-x86-64.zip",
        "p31335142_180000_Generic.zip"
    ]
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/18.0.0.0.ru-2020-10.rur-2020-10.r1",
        "oracleBase": "/home/oracle/"
    }
}
```

Example Beispiel-CEV-Manifests für Oracle Database 19c

Im folgenden Beispiel für Oracle Database 19c wendet RDS Custom p32126828, dann p29213893, dann p29782284 usw. an. In dem Beispiel werden neue Werte für den UNIX-Benutzer und die UNIX-Gruppe sowie für das Oracle-Standardverzeichnis und die Oracle-Basis festgelegt.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p29213893_1910000DBRU_Generic.zip",
        "p29782284_1910000DBRU_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29374604_1910000DBRU_Linux-x86-64.zip",
```

```
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p31335142_190000_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1",
        "oracleBase": "/home/oracle"
    }
}
```

Schritt 6 (optional): Validieren des CEV-Manifests

Stellen Sie optional sicher, dass das Manifest eine gültige JSON-Datei ist, indem Sie die`json.tool`Python-Skript. Wenn Sie beispielsweise in das Verzeichnis wechseln, das ein CEV-Manifest namens enthält`manifest.json`führen Sie den folgenden Befehl aus.

```
python -m json.tool < manifest.json
```

Schritt 7: Hinzufügen der erforderlichen IAM-Berechtigungen

Stellen Sie sicher, dass der IAM-Prinzipal, der die CEV erstellt, über die erforderlichen Richtlinien verfügt, die unter Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen beschrieben sind.

Erstellen einer CEV

Sie können eine CEV mit dem AWS Management Console oder dem erstellen AWS CLI. Geben Sie entweder die Multi-Tenant- oder die Nicht-Multi-Tenant-Architektur an. Weitere Informationen finden Sie unter Überlegungen zur Multi-Tenant-Architektur.

In der Regel CEV dauert das Erstellen von etwa zwei Stunden. Nachdem Sie die erstellt habenCEV, können Sie sie verwenden, um eine RDS benutzerdefinierte DB-Instance zu erstellen oder zu aktualisieren. Weitere Informationen erhalten Sie unter Erstellen einer RDS Custom für Oracle DB-Instance und Eine RDS benutzerdefinierte für Oracle DB-Instance aktualisieren.

> **Note**
>
> Wenn Ihre DB-Instance derzeit Oracle Linux 7.9 verwendet, erstellen Sie eine neueCEV, die die neueste Version verwendetAMI, die Oracle Linux 8 verwendet. Ändern Sie dann Ihre Instance so, dass sie die neue Version verwendetCEV.

Beachten Sie die folgenden Anforderungen und Einschränkungen für die Erstellung einerCEV:

- Der Amazon S3 S3-Bucket, der Ihre Installationsdateien enthält, muss sich in demselben Ordner AWS-Region befinden wie IhrCEV. Andernfalls schlägt der Erstellungsprozess fehl.
- Der CEV Name muss das Format haben*major-engine-version*.*customized_string*, wie in`19.cdb_cev1`.
- Der CEV Name muss 1—50 alphanumerische Zeichen, Unterstriche, Bindestriche oder Punkte enthalten.
- Der CEV Name darf keine aufeinanderfolgenden Punkte enthalten, wie in. `19..cdb_cev1`

Konsole

So erstellen Sie eine CEV

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus.Benutzerdefinierte Engine-Versionenaus.

   Auf der Seite Benutzerdefinierte Engine-Versionen werden alle derzeit CEVs vorhandenen Engine-Versionen angezeigt. Wenn Sie keine erstellt habenCEVs, ist die Seite leer.
3. Klicken Sie auf Erstellen einer benutzerdefinierten Engine-Version.
4. Gehen Sie unter Engine-Optionen wie folgt vor:

   a. Wählen Sie in Engine type (Engine-Typ) Oracle.

   b. Wählen Sie für Architektureinstellungen optional Multitenant Architecture aus, um einen Oracle Multitenant zu erstellenCEV, der die DB-Engine `custom-oracle-ee-cdb` oder verwendet. `custom-oracle-se2-cdb` Sie können eine RDS benutzerdefinierte Lösung für Oracle nur CDB mit einem CEV Multitenant erstellen. Wenn Sie diese Option nicht wählen,

CEV handelt es sich bei Ihrer nicht um eine OptionCDB, die die Engine `custom-oracle-ee` oder `custom-oracle-se2` verwendet.

> **Note**
>
> Die Architektur, für die Sie sich entscheiden, ist ein bleibendes Merkmal IhrerCEV. Sie können Ihre Architektur nicht ändernCEV, um später eine andere zu verwenden.

c. Wählen Sie eine der folgenden Optionen:

- Neu erstellen CEV — Erstellen Sie eine CEV von Grund auf neu. In diesem Fall müssen Sie ein JSON Manifest angeben, das die Datenbank-Binärdateien angibt.
- CEVAus Quelle erstellen — Wählen Sie unter Geben Sie anCEV, dass Sie kopieren möchten, eine vorhandene Datei aus, die als Quelle CEV verwendet werden CEV soll. In diesem Fall können Sie ein neues Amazon Machine Image (AMI) angeben, aber Sie können keine anderen Datenbank-Binärdateien angeben.

d. Wählen Sie unter Engine-Version die Engine-Hauptversion aus.

5. Führen Sie unter Versionsdetails die folgenden Schritte aus:

a. Geben Sie einen gültigen Namen im Feld Benutzerdefinierter Engine-Versionsname ein. So könnten Sie beispielsweise **`19.cdb_cev1`** eingeben.

b. (Optional) Geben Sie eine Beschreibung für Ihre CEV ein.

6. Gehen Sie unter Installationsmedien wie folgt vor:

a. (Optional) Lassen Sie für AMIID entweder das Feld leer, um den neuesten Service zu verwendenAMI, oder geben Sie eine einAMI, mit der Sie zuvor eine CEV erstellt haben. Verwenden Sie eine der folgenden Methoden AMIIDs, um eine gültige Version zu erhalten:

- Wählen Sie in der Konsole im linken Navigationsbereich die Option Benutzerdefinierte Engine-Versionen und wählen Sie den Namen einesCEV. Die von der verwendete AMI ID CEV wird auf der Registerkarte Konfiguration angezeigt.
- Verwenden Sie in der AWS CLI den `describe-db-engine-versions` Befehl. Suchen Sie in der Ausgabe nach `ImageID`.

b. FürS3-Speicherort von ManifestdateienGeben Sie den Speicherort des Amazon S3-Buckets ein, den Sie in angegeben habenSchritt 3: Hochladen Ihrer Installationsdateien in Amazon S3aus. Geben Sie z. B. ei **`s3://my-custom-installation-files/123456789012/cev1/`**.

> **Note**
>
> Der, AWS-Region in dem Sie den erstellen, CEV muss sich in derselben Region wie der S3-Bucket befinden.

c. (CEVNur neu erstellen) Geben Sie für CEVManifest das JSON Manifest ein, in dem Sie es erstellt habenErstellen des CEV-Manifests.

7. Wählen Sie im KMSSchlüsselbereich die Option Schlüssel eingeben aus, ARN um die verfügbaren AWS KMS Schlüssel aufzulisten. Wählen Sie dann Ihren KMS Schlüssel aus der Liste aus.

   Für RDS Benutzerdefiniert ist ein AWS KMS Schlüssel erforderlich. Weitere Informationen finden Sie unter Schritt 1: Erstellen oder Wiederverwenden eines symmetrischen AWS KMS - Verschlüsselungsschlüssels.

8. (Optional) Wählen Sie Neues Tag hinzufügen, um ein Schlüssel-Wert-Paar für Ihr zu erstellen. CEV

9. Klicken Sie auf Erstellen einer benutzerdefinierten Engine-Version.

   Wenn das JSON Manifest ein ungültiges Format hat, zeigt die Konsole Fehler bei der Überprüfung des Manifests an. CEV Beheben Sie die Probleme und versuchen Sie es erneut.

DieBenutzerdefinierte Engine-Versionen-Seite wird angezeigt. Ihr CEV wird mit dem Status Wird erstellt angezeigt. Der Vorgang zum Erstellen einer CEV dauert ungefähr zwei Stunden.

AWS CLI

Um eine mit CEV dem zu erstellen AWS CLI, führen Sie den Befehl create-custom-db-engine-version aus.

Die folgenden Optionen sind erforderlich:

- `--engine`— Geben Sie den Motortyp an. Geben Sie für a CDB entweder `custom-oracle-ee-cdb` oder an`custom-oracle-se2-cdb`. Geben Sie für einen WertCDB, der nicht- ist, entweder `custom-oracle-ee` oder an`custom-oracle-se2`. Sie können CDBs nur aus einer Datei erstellen, die mit `custom-oracle-ee-cdb` oder CEV erstellt wurde`custom-oracle-se2-cdb`. Sie können nicht CDBs nur aus einer Datei erstellen, die mit `custom-oracle-ee` oder CEV erstellt wurde`custom-oracle-se2`.

- `--engine-version` – Geben Sie die Engine-Version an. Das Format ist*major-engine-version*. *customized_string*. Der CEV Name muss 1—50 alphanumerische Zeichen, Unterstriche, Bindestriche oder Punkte enthalten. Der CEV Name darf keine aufeinanderfolgenden Punkte enthalten, wie in. `19..cdb_cev1`
- `--kms-key-id`— Geben Sie eine an AWS KMS key.
- `--manifest` – Geben Sie entweder *manifest_json_string* oder `--manifest file:`*file_name* an. Zeilenumbruchzeichen sind in *manifest_json_string* nicht zulässig. Achten Sie darauf, doppelte Anführungszeichen („) im JSON Code zu umgehen, indem Sie ihnen einen umgekehrten Schrägstrich (\) voranstellen.

  Das folgende Beispiel zeigt das *manifest_json_string* für 19c von Schritt 5: Vorbereiten des CEV-Manifests. Das Beispiel legt neue Werte für die Oracle-Basis, das Oracle-Standardverzeichnis sowie die ID und den Namen des UNIX /Linux-Benutzers und der Gruppe fest. Wenn Sie diese Zeichenfolge kopieren, entfernen Sie alle Zeilenumbruchzeichen, bevor Sie sie in Ihren Befehl einfügen.

  ```
  "{\"mediaImportTemplateVersion\": \"2020-08-14\",
  \"databaseInstallationFileNames\": [\"V982063-01.zip\"],
  \"opatchFileNames\": [\"p6880880_190000_Linux-x86-64.zip\"],
  \"psuRuPatchFileNames\": [\"p32126828_190000_Linux-x86-64.zip\"],
  \"otherPatchFileNames\": [\"p29213893_1910000DBRU_Generic.zip\",
  \"p29782284_1910000DBRU_Generic.zip\",\"p28730253_190000_Linux-
  x86-64.zip\",\"p29374604_1910000DBRU_Linux-x86-64.zip\",
  \"p28852325_190000_Linux-x86-64.zip\",\"p29997937_190000_Linux-x86-64.zip
  \",\"p31335037_190000_Linux-x86-64.zip\",\"p31335142_190000_Generic.zip
  \"]\"installationParameters\":{ \"unixGroupName\":\"dba\",
  \ \"unixUname\":\"oracle\", \ \"oracleHome\":\"/home/oracle/
  oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1\", \ \"oracleBase\":\"/
  home/oracle/\"}}"
  ```

- `--database-installation-files-s3-bucket-name` – Geben Sie denselben Bucket-Namen an, den Sie in Schritt 3: Hochladen Ihrer Installationsdateien in Amazon S3 angegeben haben. Die Region, AWS-Region in der Sie ausführen, `create-custom-db-engine-version` muss dieselbe Region wie Ihr Amazon S3 S3-Bucket sein.

Sie können auch die folgenden Optionen angeben:

- `--description`— Geben Sie eine Beschreibung Ihres anCEV.
- `--database-installation-files-s3-prefix` – Geben Sie den Ordnernamen an, den Sie in Schritt 3: Hochladen Ihrer Installationsdateien in Amazon S3 angegeben haben.
- `--image-id`— Geben Sie eine AMI ID an, die Sie wiederverwenden möchten. Um eine gültige Version zu findenIDs, führen Sie den `describe-db-engine-versions` Befehl aus und suchen Sie dann in der Ausgabe nach`ImageID`. Standardmäßig verwendet RDS Custom for Oracle die neueste verfügbare VersionAMI.

Im folgenden Beispiel wird ein Oracle-Multitenant CEV mit dem Namen `19.cdb_cev1` erstellt. Im Beispiel wird ein vorhandenes wiederverwendet, AMI anstatt das neueste verfügbare zu verwenden. AMI Stellen Sie sicher, dass Ihr Name mit der Versionsnummer der Haupt-Engine CEV beginnt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-se2-cdb \
    --engine-version 19.cdb_cev1 \
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files \
    --database-installation-files-s3-prefix 123456789012/cev1 \
    --kms-key-id my-kms-key \
    --description "test cev" \
    --manifest manifest_string \
    --image-id ami-012a345678901bcde
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-se2-cdb ^
    --engine-version 19.cdb_cev1 ^
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files ^
    --database-installation-files-s3-prefix 123456789012/cev1 ^
    --kms-key-id my-kms-key ^
    --description "test cev" ^
    --manifest manifest_string ^
    --image-id ami-012a345678901bcde
```

Example

Rufen Sie Details zu Ihrem ab, CEV indem Sie den `describe-db-engine-versions` Befehl verwenden.

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-se2-cdb \
    --include-all
```

Die folgende beispielhafte Teilausgabe zeigt die Engine, die Parametergruppen, das Manifest und andere Informationen.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-se2-cdb",
            "EngineVersion": "19.cdb_cev1",
            "DBParameterGroupFamily": "custom-oracle-se2-cdb-19",
            "DBEngineDescription": "Containerized Database for Oracle Custom SE2",
            "DBEngineVersionDescription": "test cev",
            "Image": {
                "ImageId": "ami-012a345678901bcde",
                "Status": "active"
            },
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "MajorEngineVersion": "19",
            "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-custom-
installation-files",
            "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:123456789012:cev:custom-
oracle-se2-cdb/19.cdb_cev1/abcd12e3-4f5g-67h8-i9j0-k1234l56m789",
            "KMSKeyId": "arn:aws:kms:us-
east-1:732027699161:key/1ab2345c-6d78-9ef0-1gh2-3456i7j89k01",
            "CreateTime": "2023-03-07T19:47:58.131000+00:00",
            "TagList": [],
            "SupportsBabelfish": false,
```

```
...
```

Fehler beim Erstellen eines CEV

Wenn der Vorgang zum Erstellen einer CEV fehlschlägt, gibt RDS Custom einen Fehler `RDS-EVENT-0198` mit der Nachricht `Creation failed for custom engine version` *`major-engine-version.cev_name`* aus und enthält Informationen zu dem Fehler. Zum Beispiel druckt das Ereignis fehlende Dateien.

Sie können eine fehlgeschlagene Datei nicht ändernCEV. Sie können es nur löschen und dann erneut versuchen, eine zu erstellen, CEV nachdem Sie die Ursachen des Fehlers behoben haben. Informationen zur Behebung der Gründe für das Fehlschlagen bei der CEV Erstellung finden Sie unterFehlerbehebung bei der Erstellung von benutzerdefinierten Engine-Versionen für RDS Custom for Oracle.

Ändern des CEV-Status

Sie können ein CEV mit dem AWS Management Console oder dem AWS CLIändern. Sie können die CEV-Beschreibung oder ihren Verfügbarkeitsstatus ändern. Ihre CEV hat einen der folgenden Statuswerte:

- `available`— Sie können diesen CEV verwenden, um eine neue RDS Custom DB-Instance zu erstellen oder eine DB-Instance zu aktualisieren. Dies ist der Standardstatus für eine neu erstellte CEV.
- `inactive`— Sie können mit diesem CEV keine RDS Custom Instance erstellen oder aktualisieren. Sie können einen DB-Snapshot nicht wiederherstellen, um eine neue RDS Custom DB-Instance mit diesem CEV zu erstellen.

Sie können den CEV von jedem unterstützten Status in einen anderen unterstützten Status ändern. Sie können den Status ändern, um die versehentliche Verwendung einer CEV zu verhindern oder eine nicht fortgesetzte CEV erneut für die Verwendung berechtigt zu machen. Beispielsweise können Sie den Status Ihrer CEV unter `available` zu `inactive` und von `inactive` zurück zu `available` aus.

Konsole

So ändern Sie eine CEV

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus. Benutzerdefinierte Engine-Versionenaus.
3. Wählen Sie eine CEV aus, deren Beschreibung oder Status Sie ändern möchten.
4. Wählen Sie für Actions (Aktionen) die Option Modify (Ändern) aus.
5. Nehmen Sie eine oder alle der folgenden Änderungen vor:
   - Für Einstellungen für CEV-Status wählen Sie einen neuen Verfügbarkeitsstatus aus.
   - Geben Sie auf der Seite Update description (Beschreibung aktualisieren) eine Beschreibung für die neue Version ein.
6. Klicken Sie auf Ändern der CEV.

   Wenn der CEV verwendet wird, wird die Konsole Sie können den CEV-Status nicht ändern angezeigt. Beheben Sie die Probleme und versuchen Sie es erneut.

DieBenutzerdefinierte Engine-Versionen-Seite wird angezeigt.

AWS CLI

Um ein CEV mit dem zu ändern AWS CLI, führen Sie den Befehl modify-custom-db-engine-version aus. Sie können CEVs zum Ändern finden, indem Sie den Befehl ausführen. describe-db-engine-versions

Die folgenden Optionen sind erforderlich:

- `--engine` *`engine-type`*, wobei der *`Engine-Typ,,, oder`* ist `custom-oracle-ee` `custom-oracle-se2` `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- `--engine-version` *`cev`*, wobei *`cev`* der Name der benutzerdefinierten Engine-Version ist, die Sie ändern möchten
- `--status` *`status`*, wobei *`status`* ist der Verfügbarkeitsstatus, den Sie dem CEV zuweisen möchten

Im folgenden Beispiel wird ein CEV namens `19.my_cev1` von seinem aktuellen Status in `inactive` geändert.

Example

FürLinux, odermacOS: Unix

```
aws rds modify-custom-db-engine-version \
    --engine custom-oracle-se2 \
    --engine-version 19.my_cev1 \
    --status inactive
```

Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-oracle-se2 ^
    --engine-version 19.my_cev1 ^
    --status inactive
```

CEVDetails für Amazon RDS Custom for Oracle anzeigen

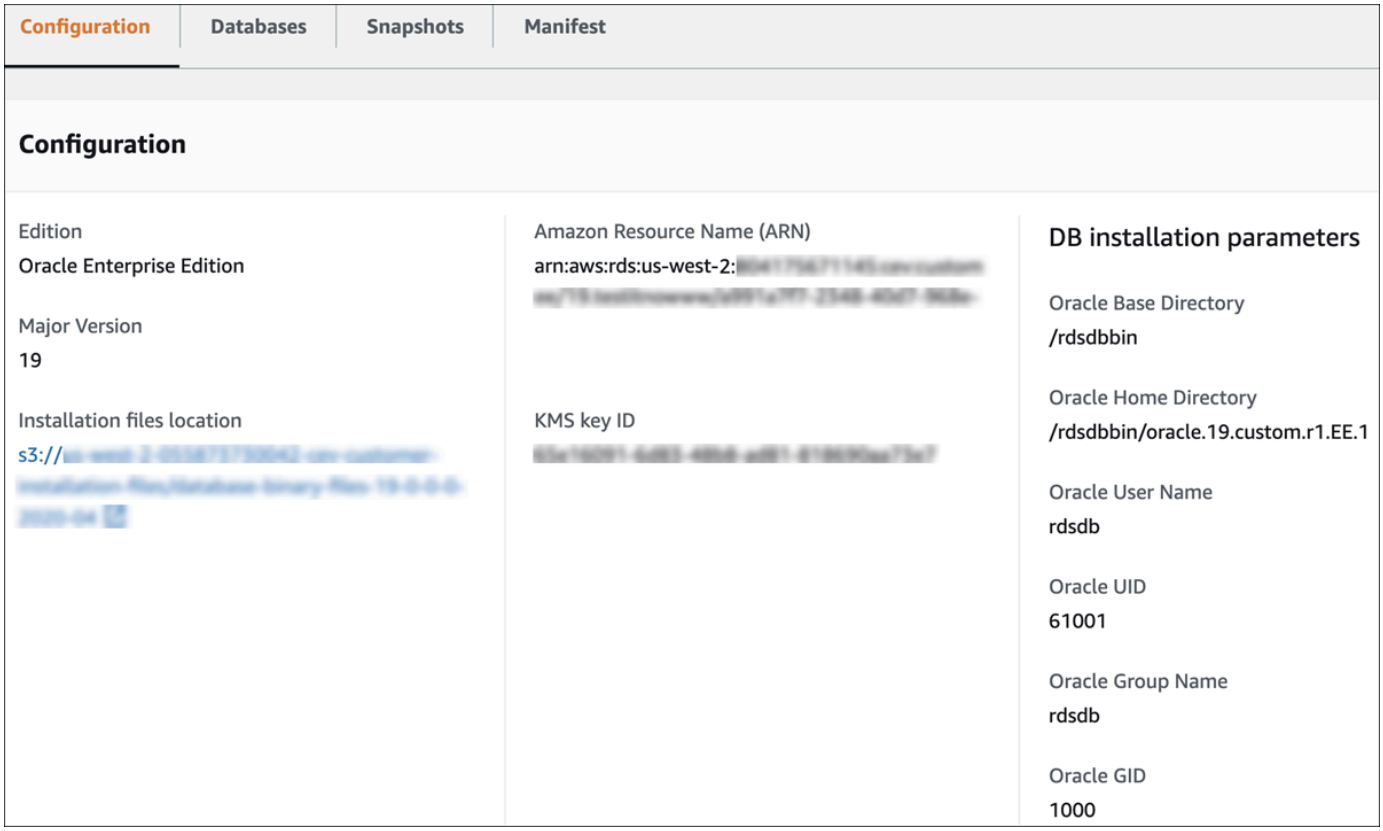
Sie können Details zu Ihrem CEV Manifest und dem Befehl, mit dem Sie Ihr Manifest erstellt haben, anzeigen, CEV indem Sie das AWS Management Console oder das verwenden AWS CLI.

Konsole

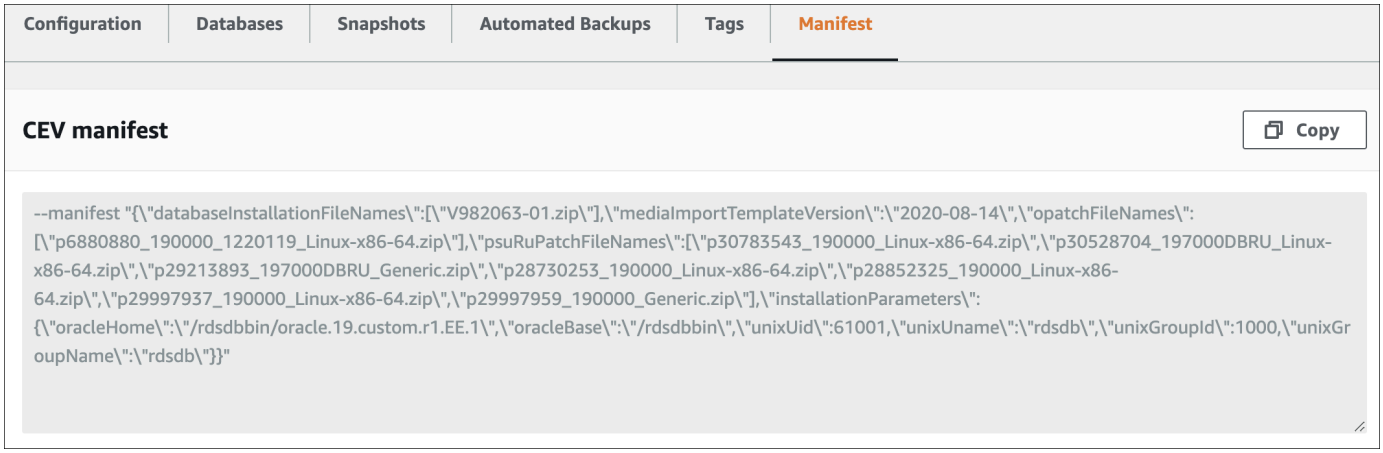
Um CEV Details anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus.Benutzerdefinierte Engine-Versionenaus.

   Auf der Seite Benutzerdefinierte Engine-Versionen werden alle derzeit CEVs vorhandenen Engine-Versionen angezeigt. Wenn Sie keine erstellt habenCEVs, ist die Seite leer.
3. Wählen Sie den Namen der DateiCEV, die Sie sich ansehen möchten.
4. Wählen Sie Configuration (Konfiguration) aus, um die in Ihrem Manifest angegebenen Installationsparameter anzuzeigen.



5. Wählen Sie Manifest aus, um die in der Option `--manifest` des Befehls `create-custom-db-engine-version` angegebenen Installationsparameter anzuzeigen. Sie können diesen Text kopieren, Werte nach Bedarf ersetzen und sie in einem neuen Befehl verwenden.



AWS CLI

Um Details zu a mit CEV dem anzuzeigen AWS CLI, führen Sie den describe-db-engine-versionsBefehl aus.

Die folgenden Optionen sind erforderlich:

- `--engine` *`engine-type`*, wo *`engine-type`* ist `custom-oracle-eecustom-oracle-se2`,`custom-oracle-ee-cdb`, oder `custom-oracle-se2-cdb`
- `--engine-version` *`major-engine-version.customized_string`*

Im folgenden Beispiel wird eine Datei erstellt CDBCEV, die nicht die Enterprise Edition verwendet. Der CEV Name `19.my_cev1` beginnt mit der Versionsnummer der Hauptengine, die erforderlich ist.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Die folgende beispielhafte Teilausgabe zeigt die Engine, die Parametergruppen, das Manifest und andere Informationen.

```
"DBEngineVersions": [
    {
        "Engine": "custom-oracle-ee",
        "MajorEngineVersion": "19",
        "EngineVersion": "19.my_cev1",
        "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-cev-customer-
installation-files",
        "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
        "CustomDBEngineVersionManifest": "{\n\"mediaImportTemplateVersion\":
 \"2020-08-14\",\n\"databaseInstallationFileNames\": [\n\"V982063-01.zip\"\n],
\n\"installationParameters\": {\n\"oracleBase\":\"/tmp\",\n\"oracleHome\":\"/
tmp/Oracle\"\n},\n\"opatchFileNames\": [\n\"p6880880_190000_Linux-x86-64.zip
\"\n],\n\"psuRuPatchFileNames\": [\n\"p32126828_190000_Linux-x86-64.zip
\"\n],\n\"otherPatchFileNames\": [\n\"p29213893_1910000DBRU_Generic.zip\",\n
\"p29782284_1910000DBRU_Generic.zip\",\n\"p28730253_190000_Linux-x86-64.zip\",\n
```

```
\"p29374604_1910000DBRU_Linux-x86-64.zip\",\n\"p28852325_190000_Linux-x86-64.zip\",
\n\"p29997937_190000_Linux-x86-64.zip\",\n\"p31335037_190000_Linux-x86-64.zip\",\n
\"p31335142_190000_Generic.zip\"\n]\n}\n",
        "DBParameterGroupFamily": "custom-oracle-ee-19",
        "DBEngineDescription": "Oracle Database server EE for RDS Custom",
        "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-
ee/19.my_cev1/0a123b45-6c78-901d-23e4-5678f901fg23",
        "DBEngineVersionDescription": "test",
        "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/ab1c2de3-f4g5-6789-h012-
h3ijk4567l89",
        "CreateTime": "2022-11-18T09:17:07.693000+00:00",
        "ValidUpgradeTarget": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.cev.2021-01.09",
            "Description": "test",
            "AutoUpgrade": false,
            "IsMajorVersionUpgrade": false
        }
]
```

Löschen einer CEV

Sie können ein CEV mit dem AWS Management Console oder dem AWS CLI löschen. Dies dauert in der Regel einige Minuten.

Um eine CEV zu löschen, kann sie von keinem der folgenden Optionen verwendet werden:

- Stopp einer RDS-DB-Instance.
- Ein Snapshot einer RDS Custom DB-Instance
- Eine automatisierte Sicherung Ihrer RDS Custom DB-Instance

Konsole

So löschen Sie eine VPC

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus. Benutzerdefinierte Engine-Versionenaus.
3. Wählen Sie eine CEV aus, deren Beschreibung oder Status Sie löschen möchten.

4. Klicken Sie bei Actions auf Delete.

   Die Dialogbox Löschen*cev_name*? wird angezeigt.

5. Geben Sie **delete me** ein und klicken Sie auf Delete (Löschen).

   In derBenutzerdefinierte Engine-Versionen-Seite zeigt das Banner, dass Ihre CEV gelöscht wird.

AWS CLI

Um ein CEV mit dem zu löschen AWS CLI, führen Sie den Befehl delete-custom-db-engine-version aus.

Die folgenden Optionen sind erforderlich:

- `--engine` *`engine-type`*, wobei der *`Engine-Typ,,, oder`* ist `custom-oracle-ee` `custom-oracle-se2` `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- `--engine-version` *`cev`*, wobei*`cev`*ist der Name der zu löschenden benutzerdefinierten Engine-Version

Im folgenden Beispiel wird ein Tresor namens `19.my_cev1` gelöscht.

Example

FürLinux, odermacOS: Unix

```
aws rds delete-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Konfigurieren einer DB-Instance für Amazon RDS Custom für Oracle

Sie können eine RDS Custom DB-Instance erstellen und sich dann über Secure Shell (SSH) mit dieser oder AWS Systems Manager verbinden.

Weitere Informationen zum Herstellen einer Verbindung und Anmeldung bei einer RDS Custom for Oracle-DB-Instance finden Sie in den folgenden Themen.

- Mit Session Manager eine Verbindung zu Ihrer RDS Custom DB-Instance herstellen
- Herstellen einer Verbindung mit Ihrer RDS Custom DB-Instance über SSH
- Anmelden als SYS bei Ihrer Datenbank von RDS Custom für Oracle

Erstellen einer RDS Custom für Oracle DB-Instance

Erstellen Sie eine Amazon RDS Custom for Oracle DB-Instance mit entweder dem AWS Management Console oder dem AWS CLI. Das Verfahren ähnelt dem Verfahren zum Erstellen einer Amazon RDS DB-Instance. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Wenn Sie Installationsparameter in Ihr CEV-Manifest aufgenommen haben, verwendet Ihre DB-Instance die Oracle-Basis, das Oracle-Standardverzeichnis sowie die ID und den Namen des von Ihnen angegebenen UNIX/Linux-Benutzers und der Gruppe. Die `oratab`-Datei, die von Oracle Database während der Installation erstellt wird, verweist auf den tatsächlichen Installationsort und nicht auf einen symbolischen Link. Wenn RDS Custom für Oracle Befehle ausführt, wird es als konfigurierter Betriebssystembenutzer und nicht als Standardbenutzer `rdsdb` ausgeführt. Weitere Informationen finden Sie unter Schritt 5: Vorbereiten des CEV-Manifests.

Sie müssen die Aufgaben im Abschnitt Einrichten Ihrer Umgebung für Amazon RDS Custom for Oracle abschließen, bevor Sie versuchen, eine DB-Instance von RDS Custom zu erstellen oder eine Verbindung mit einer solchen DB-Instance herzustellen.

Konsole

So erstellen Sie eine RDS Custom for Oracle DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.

4. Wählen Sie unter Choose a database creation method (Wählen Sie eine Datenbankerstellungsmethode aus) Standard Create (Standarderstellung) aus.
5. Gehen Sie im Abschnitt Engine-Optionen wie folgt vor:

   a. Wählen Sie in Engine type (Engine-Typ) Oracle.

   b. Für Typ der Datenbankverwaltung, wählenBenutzerdefiniert Amazon RDS Custom aus.

   c. Führen Sie unter Architektureinstellungen einen der folgenden Schritte aus:

      - Wählen Sie Multi-Tenant-Architektur aus, um eine Container-Datenbank (CDB) zu erstellen. Zum Zeitpunkt der Erstellung enthält Ihre CDB einen PDB-Seed und eine anfängliche PDB.

        > **Note**
        >
        > Die Einstellung Multi-Tenant-Architektur wird nur für Oracle Database 19c unterstützt.

      - Deaktivieren Sie Multi-Tenant-Architektur, um eine Nicht-CDB zu erstellen. Eine Nicht-CDB kann nicht enthalten. PDBs

   d. Wählen Sie für Edition Oracle Enterprise Edition oder Oracle Standard Edition 2.

   e. Wählen Sie für Benutzerdefinierte Engine-Version eine vorhandene benutzerdefinierte Engine-Version (CEV) von RDS Custom aus. Eine CEV hat folgendes Format: *`major-engine-version.customized_string`*. Ein Beispiel-Bezeichner ist `19.cdb_cev1`.

      Wenn Sie im vorherigen Schritt die Multitenant-Architektur gewählt haben, können Sie nur ein CEV angeben, das den `custom-oracle-se2-cdb` Engine-Typ `custom-oracle-ee-cdb` oder verwendet. Die Konsole filtert heraus CEVs , welche mit unterschiedlichen Engine-Typen erstellt wurden.

6. Wählen Sie für VorlagenProduktion.
7. Gehen Sie im Abschnitt Settings (Einstellungen) wie folgt vor:

   a. Geben Sie für DB-Instance-Kennung einen eindeutigen Namen für Ihre DB-Instance ein.

   b. Geben Sie im Feld Hauptbenutzername einen Benutzernamen ein. Sie können diesen Wert später von der Konsole abrufen.

      Wenn Sie eine Verbindung mit einer Nicht-CDB herstellen, ist der Hauptbenutzer der Benutzer der Nicht-CDB. Wenn Sie eine Verbindung mit einer CDB herstellen, ist der

Hauptbenutzer der Benutzer der PDB. Um eine Verbindung zum CDB-Root herzustellen, melden Sie sich beim Host an, starten Sie einen SQL-Client und erstellen Sie einen Administratorbenutzer mit SQL-Befehlen.

c. Deaktivieren Sie Passwort automatisch generieren.

8. Wählen Sie eine DB-Instance-Klasse aus.

   Informationen zu unterstützten Klassen finden Sie unter Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle.

9. Gehen Sie im Abschnitt Storage (Speicher) wie folgt vor:

   a. Wählen Sie als Speichertyp einen SSD-Typ aus: io1, gp2 oder gp3. Ihnen stehen folgende zusätzliche Optionen zur Verfügung:

      - Wählen Sie für io1 oder gp3 eine Rate für Bereitgestellte IOPS aus. Die Standardeinstellung ist 1 000 für io1 und 12 000 für gp3.
      - Wählen Sie für gp3 eine Rate für den Speicherdurchsatz aus. Die Standardeinstellung ist 500 MiBps.

   b. Wählen Sie für Zugewiesener Speicher eine Speichergröße aus. Der Standardwert ist 40 GiB.

10. Geben Sie für Konnektivität Ihre Virtual Private Cloud (VPC), DB-Subnetzgruppe und VPC-Sicherheitsgruppe (Firewall) an.

11. FürRDS Benutzerdefinierte Sicherheitwie folgt:

    a. Für IAM-Instance-Profil, wählen Sie das Instanzprofil für Ihre RDS Custom for Oracle DB-Instance aus.

       Das IAM-Instance-Profil muss mit `AWSRDSCustom`, z. B. *`AWSRDSCustomInstanceProfileForRdsCustomInstance`* beginnen.

    b. Wählen Sie unter Verschlüsselung Enter a key ARN aus, um die verfügbaren AWS KMS Schlüssel aufzulisten. Wählen Sie dann Ihren Schlüssel aus der Liste aus.

       Für RDS Custom ist ein AWS KMS Schlüssel erforderlich. Weitere Informationen finden Sie unter Schritt 1: Erstellen oder Wiederverwenden eines symmetrischen AWS KMS -Verschlüsselungsschlüssels.

12. Führen Sie unter Datenbankoptionen folgende Schritte aus:

a. (Optional) Geben Sie für System-ID (SID) einen Wert für die Oracle-SID ein, der auch der Name Ihrer CDB ist. Der SID ist der Name der Oracle-Datenbank-Instance, die Ihre Datenbankdateien verwaltet. In diesem Zusammenhang bezieht sich der Begriff „Oracle-Datenbank-Instance“ ausschließlich auf die System Global Area (SGA) und die Oracle-Hintergrundprozesse. Wenn Sie keinen SID angeben, verwendet das System standardmäßig **`RDSCDB`**.

b. (Optional) Geben Sie für Anfänglicher Datenbankname einen Namen ein. Der Standardwert ist **`ORCL`**. In der Multi-Tenant-Architektur ist der anfängliche Datenbankname der PDB-Name.

> **Note**
>
> SID- und PDB-Name müssen unterschiedlich sein.

c. Wählen Sie für Optionsgruppe eine Optionsgruppe aus, oder akzeptieren Sie die Standardeinstellung.

> **Note**
>
> Die einzige unterstützte Option für RDS Custom for Oracle ist`Timezone`. Weitere Informationen finden Sie unter Oracle-Zeitzone.

d. Wählen Sie unter Aufbewahrungszeitraum für Backups einen Wert aus. Sie können keine 0 Tage auswählen.

e. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die RDS Custom DB-Instance an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances. Die folgenden Einstellungen werden nicht in der Konsole angezeigt und werden nicht unterstützt:

- Prozessorfunktionen
- Automatische Speicherskalierung
- Passwort und Kerberos-AuthentifizierungOption inDatenbankauthentifizierung(nurPasswortauthentifizierungwird unterstützt)
- Performance Insights
- Protokollexporte
- Auto minor version upgrade (Upgrade einer Unterversion automatisch durchführen)

- Löschschutz

13. Wählen Sie Datenbank erstellen aus.

> **Important**
>
> Wenn Sie Mal eine DB-Instance von RDS Custom für Oracle erstellen, wird möglicherweise der folgende Fehler angezeigt: Die serviceverknüpfte Rolle wird gerade erstellt. Bitte versuchen Sie es später erneut. Wenn Sie dies der Fall sind, warten Sie einige Minuten und versuchen Sie dann erneut, die DB-Instance zu erstellen.

DieAnzeigen von Anmeldeinformationenerscheint auf der SchaltflächeDatenbankenangezeigten.

Um den Namen und das Passwort des Hauptbenutzers für die RDS Custom DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen) aus.

Verwenden Sie den angezeigten Benutzernamen und das angezeigte Passwort, um eine Verbindung zu DB-Instance als Hauptbenutzer herzustellen.

> **Important**
>
> Sie können dieses Passwort für den Hauptbenutzer in der Konsole nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern. Um das Passwort für den Hauptbenutzer zu ändern, nachdem die DB-Instance von RDS Custom verfügbar wurde, melden Sie sich bei der Datenbank an und führen Sie einen `ALTER USER`-Befehl aus. Sie können das Passwort nicht über die Option Ändern in der Konsole zurücksetzen.

14. Klicken Sie aufDatenbankenum die Liste der RDS Custom DB-Instanzen anzuzeigen.
15. Wählen Sie die RDS-DB-Instance aus, die Sie soeben erstellt haben.

In der RDS-Konsole werden die Details der neuen DB-Instance angezeigt.

- Die RDS Custom DB-Instance wird mit dem Status creating (Wird erstellt) angezeigt, bis sie erstellt wurde und einsatzbereit ist. Wenn sich der Status in available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und dem dieser zugeteilten Speicher kann es einige Minuten dauern, bis die neue DB-Instance verfügbar ist.

- -Rolle hat den Wert Instanz (RDS Custom).
- RDS Benutzerdefinierter Automatisierungsmodus hat den Wert Vollständige Automatisierung. Diese Einstellung bedeutet, dass die DB-Instance eine automatische Überwachung und Instanzwiederherstellung bietet.

AWS CLI

Sie erstellen eine benutzerdefinierte RDS-DB-Instance mithilfe des create-db-instance AWS CLI Befehls.

Die folgenden Optionen sind erforderlich:

- `--db-instance-identifier`
- `--db-instance-class` (eine Liste der unterstützten Klassen, finden Sie unter Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle.
- `--engine` *`engine-type`*, wo *`engine-type`* ist `custom-oracle-eecustom-oracle-se2`,`custom-oracle-ee-cdb`, oder `custom-oracle-se2-cdb`
- `--engine-version` *`cev`*(wo*`cev`*ist der Name der benutzerdefinierten Engine-Version, die Sie inErstellen einer CEV)
- `--kms-key-id` *`my-kms-key`*
- `--backup-retention-period` *`days`* (wobei *`days`* ein Wert größer `0` ist)
- `--no-auto-minor-version-upgrade`
- `--custom-iam-instance-profile AWSRDSCustomInstanceProfile-`*`us-east-1`* (wobei *`region`* die AWS-Region ist, in der Sie Ihre DB-Instance erstellen)

Im folgenden Beispiel wird eine RDS Custom-DB-Instance mit dem Namen erstell `my-cfo-cdb-instance`. Die Datenbank ist eine CDB mit dem nicht standardmäßigen Namen. *`MYCDB`* Der nicht standardmäßige PDB-Name ist. *`MYPDB`* Die Aufbewahrungszeitraum für Backups beträgt drei Tage.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --engine custom-oracle-ee-cdb \
    --db-instance-identifier my-cfo-cdb-instance \
    --engine-version 19.cdb_cev1 \
```

```
    --db-name MYPDB \
    --db-system-id MYCDB \
    --allocated-storage 250 \
    --db-instance-class db.m5.xlarge \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --port 8200 \
    --kms-key-id my-kms-key \
    --no-auto-minor-version-upgrade \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --engine custom-oracle-ee-cdb ^
    --db-instance-identifier my-cfo-cdb-instance ^
    --engine-version 19.cdb_cev1 ^
    --db-name MYPDB ^
    --db-system-id MYCDB ^
    --allocated-storage 250 ^
    --db-instance-class db.m5.xlarge ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --port 8200 ^
    --kms-key-id my-kms-key ^
    --no-auto-minor-version-upgrade ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Rufen Sie Details zu Ihrer Instance ab, indem Sie die`describe-db-instances`befehl.

Example

```
aws rds describe-db-instances --db-instance-identifier my-cfo-cdb-instance
```

Die folgende Teilausgabe zeigt die Engine, die Parametergruppen und andere Informationen.

```
        {
            "DBInstanceIdentifier": "my-cfo-cdb-instance",
            "DBInstanceClass": "db.m5.xlarge",
            "Engine": "custom-oracle-ee-cdb",
            "DBInstanceStatus": "available",
            "MasterUsername": "admin",
            "DBName": "MYPDB",
            "DBSystemID": "MYCDB",
            "Endpoint": {
                "Address": "my-cfo-cdb-instance.abcdefghijkl.us-
east-1.rds.amazonaws.com",
                "Port": 1521,
                "HostedZoneId": "A1B2CDEFGH34IJ"
            },
            "AllocatedStorage": 100,
            "InstanceCreateTime": "2023-04-12T18:52:16.353000+00:00",
            "PreferredBackupWindow": "08:46-09:16",
            "BackupRetentionPeriod": 7,
            "DBSecurityGroups": [],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-0a1bcd2e",
                    "Status": "active"
                }
            ],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-oracle-ee-cdb-19",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
...
```

Überlegungen zur Multi-Tenant-Architektur

Wenn Sie eine Amazon RDS Custom for Oracle DB-Instance mit der Oracle Multitenant-Architektur (`custom-oracle-ee-cdb`oder dem `custom-oracle-se2-cdb` Engine-Typ) erstellen, ist Ihre Datenbank eine Container-Datenbank (CDB). Wenn Sie die Oracle-Multitenant-Architektur nicht angeben, handelt es sich bei Ihrer Datenbank um eine herkömmliche Nicht-CDB-Datenbank, die den Engine-Typ oder verwendet. `custom-oracle-ee` `custom-oracle-se2` Eine Nicht-CDB

kann keine austauschbaren Datenbanken () enthalten. PDBs Weitere Informationen finden Sie unter Datenbankarchitektur für Amazon RDS Custom für Oracle.

Beachten Sie beim Erstellen einer CDB-Instance von RDS Custom für Oracle Folgendes:

- Sie können eine Multi-Tenant-Datenbank nur von einer Oracle-Database-19c-CEV erstellen.
- Sie können eine CDB-Instance nur erstellen, wenn das CEV den Engine-Typ oder verwendet. `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- Wenn Sie eine CDB-Instanz mit Standard Edition 2 erstellen, kann die CDB maximal 3 enthalten. PDBs
- Standardmäßig erhält Ihre CDB den Namen `RDSCDB`. Dies ist auch der Name der Oracle-System-ID (Oracle SID). Sie können einen anderen Namen wählen.
- Ihre CDB enthält nur eine anfängliche PDB. Der PDB-Name ist standardmäßig `ORCL`. Sie können einen anderen Namen für Ihre anfängliche PDB auswählen, die Oracle-SID und der PDB-Name dürfen jedoch nicht identisch sein.
- RDS Custom for Oracle bietet kein Angebot APIs für. PDBs Verwenden Sie den Oracle SQL-Befehl PDBs, um weitere zu erstellen`CREATE PLUGGABLE DATABASE`. RDS Custom for Oracle schränkt die Anzahl der Dateien PDBs , die Sie erstellen können, nicht ein. Im Allgemeinen sind Sie wie bei einer lokalen Bereitstellung für die Erstellung und Verwaltung PDBs verantwortlich.
- Sie können RDS nicht APIs zum Erstellen, Ändern und Löschen verwenden PDBs: Sie müssen Oracle-SQL-Anweisungen verwenden. Wenn Sie eine PDB mit Oracle SQL erstellen, empfehlen wir, dass Sie anschließend einen manuellen Snapshot erstellen, falls Sie eine point-in-time Wiederherstellung (PITR) durchführen müssen.
- PDBs Mit Amazon RDS können Sie bestehende Dateien nicht umbenennen APIs. Sie können die CDB auch nicht mit dem `modify-db-instance`-Befehl umbenennen.
- Der offene Modus für das CDB-Root ist `READ WRITE` in der primären und `MOUNTED` in einer gemounteten Standby-Datenbank. RDS Custom for Oracle versucht, PDBs beim Öffnen der CDB alle zu öffnen. Wenn RDS Custom for Oracle nicht alle öffnen kann PDBs, wird das Ereignis `tenant database shutdown` ausgelöst.

RDS Benutzerdefinierte serviceverknüpfte Rolle

Eine serviceverknüpfte Rolle gewährt Amazon RDS Custom Zugriff auf Ressourcen in Ihrem AWS-Konto. Dadurch wird das Einrichten eines RDS Custom vereinfacht, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. RDS Custom definiert die Berechtigungen seiner

serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann nur RDS Custom die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Wenn Sie eine RDS Custom DB-Instance erstellen, werden sowohl die mit Amazon RDS als auch RDS Custom Service verknüpften Rollen erstellt (falls sie noch nicht vorhanden sind) und verwendet. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon RDS.

Wenn Sie zum ersten Mal eine RDS Custom for Oracle DB-Instance erstellen, wird möglicherweise der folgende Fehler angezeigt:Die serviceverknüpfte Rolle wird gerade erstellt. Bitte versuchen Sie es später erneut. Wenn Sie dies der Fall sind, warten Sie einige Minuten und versuchen Sie dann erneut, die DB-Instance zu erstellen.

Installieren zusätzlicher Softwarekomponenten auf Ihrer DB-Instance von RDS Custom für Oracle

In einer neu erstellten DB-Instance umfasst Ihre Datenbankumgebung Oracle-Binärdateien, eine Datenbank und einen Datenbank-Listener. Möglicherweise möchten Sie zusätzliche Software auf dem Host-Betriebssystem der DB-Instance installieren. Vielleicht möchten Sie beispielsweise Oracle Application Express (APEX), den Oracle Enterprise Manager (OEM) Agent oder den Guardium S-TAP Agent installieren. Richtlinien und allgemeine Anweisungen finden Sie im ausführlichen AWS Blogbeitrag Installieren zusätzlicher Softwarekomponenten auf Amazon RDS Custom for Oracle.

Mit Session Manager eine Verbindung zu Ihrer RDS Custom DB-Instance herstellen

Nachdem Sie Ihre RDS benutzerdefinierte DB-Instance erstellt haben, können Sie mithilfe von eine Verbindung zu ihr herstellen AWS Systems Manager Session Manager. Dies ist die bevorzugte Methode, wenn Ihre DB-Instance nicht öffentlich zugänglich ist.

Mit Session Manager können Sie über eine browserbasierte Shell oder über die auf EC2 Amazon-Instances zugreifen. AWS CLI Weitere Informationen erhalten Sie unter AWS Systems Manager Session Manager.

Konsole

Herstellen einer Verbindung mit Ihrer Instance mithilfe von Session Manager

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte DB-Instance aus, zu der Sie eine Verbindung herstellen möchten.
3. Wählen Sie Konfiguration.
4. Notieren Sie die Ressource-ID für die DB-Instance. Die Ressourcen-ID kann beispielsweise `db-ABCDEFGHIJKLMNOPQRS0123456` sein.
5. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
6. Wählen Sie im Navigationsbereich Instances aus.
7. Suchen Sie nach dem Namen Ihrer EC2 Instance und klicken Sie dann auf die damit verknüpfte Instance-ID. Die Instance-ID kann beispielsweise `i-abcdefghijklm01234` sein.
8. Wählen Sie Connect aus.
9. Klicken Sie auf Session Manager.
10. Wählen Sie Connect aus.

    Es öffnet sich ein Fenster für Ihre Sitzung.

AWS CLI

Sie können mit dem eine Verbindung zu Ihrer RDS benutzerdefinierten DB-Instance herstellen AWS CLI. Für diese Technik ist das Session Manager-Plugin für die AWS CLI. Informationen zum Installieren des Plugins finden Sie unter Installieren Sie das Session Manager-Plugin für das AWS CLI.

Um die DB-Ressourcen-ID Ihrer RDS benutzerdefinierten DB-Instance zu finden, verwenden Sie`aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Die folgende Beispielausgabe zeigt die Ressourcen-ID für Ihre RDS benutzerdefinierte Instance. Das Präfix lautet `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Um die EC2 Instance-ID Ihrer DB-Instance zu ermitteln, verwenden Sie`aws ec2 describe-instances`. Im folgenden Beispiel wird verwendet`db-ABCDEFGHIJKLMNOPQRS0123456`für die Ressourcen-ID.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Die folgende Beispielausgabe zeigt die EC2 Instance-ID.

```
i-abcdefghijklm01234
```

Verwenden Sie den `aws ssm start-session` Befehl und geben Sie die EC2 Instanz-ID im `--target` Parameter an.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Ein erfolgreiches Ergebnis sieht wie folgt aus.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Herstellen einer Verbindung mit Ihrer RDS Custom DB-Instance über SSH

Das Secure Shell Protocol (SSH) ist ein Netzwerkprotokoll, das verschlüsselte Kommunikation über ein nicht gesichertes Netzwerk unterstützt. Nachdem Sie Ihre DB-Instance von RDS Custom erstellt haben, können Sie sich über einen SSH-Client mit dieser Instance verbinden. Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Ihrer Linux-Instance mithilfe von SSH.

Ihre SSH-Verbindungsmethode hängt davon ab, ob Ihre DB-Instance privat ist, was bedeutet, dass sie keine Verbindungen aus dem öffentlichen Internet akzeptiert. In diesem Fall müssen Sie SSH-Tunneling verwenden, um das SSH-Dienstprogramm mit Ihrer Instance zu verbinden. Diese Methode transportiert Daten mit einem dedizierten Datenstrom (Tunnel) innerhalb einer bestehenden SSH-Sitzung. Sie können das SSH-Tunneling mit AWS Systems Manager konfigurieren.

> **Note**
>
> Für den Zugriff auf private Instances werden verschiedene Strategien unterstützt. Informationen dazu, wie Sie einen SSH-Client mithilfe von Bastion-Hosts mit privaten Instances verbinden, finden Sie unter Linux Bastion Hosts in AWS. Informationen zur

Konfiguration der Port-Weiterleitung finden Sie unter Port-Weiterleitung mit Using AWS Systems Manager Session Manager.

Wenn sich Ihre DB-Instance in einem öffentlichen Subnetz befindet und die Einstellung „Öffentlich verfügbar“ hat, ist kein SSH-Tunneling erforderlich. Sie können sich mit SSH genauso verbinden wie mit einer öffentlichen EC2 Amazon-Instance.

Führen Sie die folgenden Schritte aus, um einen SSH-Client mit Ihrer DB-Instance zu verbinden:

1. Schritt 1: Konfigurieren Ihrer DB-Instance, um SSH-Verbindungen zuzulassen
2. Schritt 2: Rufen Sie Ihren geheimen SSH-Schlüssel und EC2 Ihre Instanz-ID ab
3. Schritt 3: Stellen Sie mithilfe des SSH-Dienstprogramms eine Verbindung zu Ihrer EC2 Instance her

Schritt 1: Konfigurieren Ihrer DB-Instance, um SSH-Verbindungen zuzulassen

Gehen Sie wie folgt vor, um sicherzustellen, dass Ihre DB-Instance SSH-Verbindungen akzeptieren kann:

- Stellen Sie sicher, dass Ihre Sicherheitsgruppe der DB-Instance eingehende Verbindungen an Port 22 für TCP zulässt.

  Informationen zum Konfigurieren der Sicherheitsgruppe für Ihre DB-Instance finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
- Wenn Sie nicht vorhaben, SSH-Tunneling zu verwenden, stellen Sie sicher, dass sich Ihre DB-Instance in einem öffentlichen Subnetz befindet und öffentlich zugänglich ist.

  In der Konsole ist das relevante Feld öffentlich zugänglich auf der Registerkarte Konnektivität und Sicherheit auf der Seite mit den Datenbankdetails. Führen Sie den folgenden Befehl aus, um Ihre Einstellungen in der CLI zu überprüfen:

```
aws rds describe-db-instances \
--query 'DBInstances[*].
{DBInstanceIdentifier:DBInstanceIdentifier,PubliclyAccessible:PubliclyAccessible}' \
--output table
```

Informationen zum Ändern der Barrierefreiheitseinstellungen für Ihre DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Schritt 2: Rufen Sie Ihren geheimen SSH-Schlüssel und EC2 Ihre Instanz-ID ab

Zum Herstellen einer Verbindung mit der DB-Instance über SSH benötigen Sie das SSH-Schlüsselpaar, das der Instance zugeordnet ist. RDS Custom erstellt das SSH-Schlüsselpaar in Ihrem Namen unter Verwendung der Namenskonvention `do-not-delete-rds-custom-ssh-privatekey-`*`resource_id`*`-`*`uuid`* oder`rds-custom!oracle-do-not-delete-`*`resource_id`*`-`*`uuid`*`-ssh-privatekey`. AWS Secrets Manager speichert Ihren privaten SSH-Schlüssel als Geheimnis.

Rufen Sie Ihren geheimen SSH-Schlüssel mit einem AWS Management Console oder dem ab. AWS CLI Wenn Ihre Instance über ein öffentliches DNS verfügt und Sie nicht beabsichtigen, SSH-Tunneling zu verwenden, rufen Sie auch den DNS-Namen ab. Sie geben den DNS-Namen für öffentliche Verbindungen an.

Konsole

So rufen Sie den geheimen SSH-Schlüssel ab

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die RDS Custom DB-Instance aus, die Sie anhalten möchten.
3. Wählen Sie Konfiguration.
4. Beachten Sie dieRessourcen-IDWert. Die Ressourcen-ID der DB-Instance kann beispielsweise `db-ABCDEFGHIJKLMNOPQRS0123456` sein.
5. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
6. Wählen Sie im Navigationsbereich Instances aus.
7. Suchen Sie den Namen Ihrer EC2 Instance und wählen Sie die damit verknüpfte Instance-ID aus. Die EC2 Instanz-ID könnte beispielsweise lauten`i-abcdefghijklm01234`.
8. In -Details, finden Sie Schlüsselpaarname. Der Paarname enthält die Ressourcen-ID der DB-Instance. Der Name des Paares könnte beispielsweise `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c` oder lauten`rds-`

`custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey`.

9. Wenn Ihre EC2 Instance öffentlich ist, notieren Sie sich den öffentlichen IPv4 DNS. Im Beispiel kann die DNS-Adresse (Public Domain Name System) `ec2-12-345-678-901.us-east-2.compute.amazonaws.com` sein.
10. Öffnen Sie die AWS Secrets Manager Konsole unter https://console.aws.amazon.com/secretsmanager/.
11. Wählen Sie das Geheimnis aus, das den gleichen Namen wie Ihr key pair hat.
12. Wählen Sie Retrieve secret value (Secret-Wert abrufen) aus.
13. Kopieren Sie den privaten SSH-Schlüssel in eine Textdatei und speichern Sie die Datei anschließend mit der `.pem`-Erweiterung. Speichern Sie die Datei beispielsweise als `/tmp/do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem` oder`/tmp/rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey.pem`.

AWS CLI

Um den privaten SSH-Schlüssel abzurufen und in einer PEM-Datei zu speichern, können Sie die AWS CLI verwenden.

1. Suchen Sie die DB-Ressourcen-ID Ihrer DB-Instance von RDS Custom mit `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Die folgende Beispielausgabe zeigt die Ressourcen-ID für Ihre RDS Custom Instance. Das Präfix lautet `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

2. Finden Sie die EC2 Instance-ID Ihrer DB-Instance mithilfe von`aws ec2 describe-instances`. Im folgenden Beispiel wird verwendet`db-ABCDEFGHIJKLMNOPQRS0123456`für die Ressourcen-ID.

```
aws ec2 describe-instances \
```

```
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Die folgende Beispielausgabe zeigt die EC2 Instance-ID.

```
i-abcdefghijklm01234
```

3. Um den Schlüsselnamen zu finden, geben Sie die EC2 Instanz-ID an. Das folgende Beispiel beschreibt die EC2 Instanz`i-0bdc4219e66944afa`.

```
aws ec2 describe-instances \
    --instance-ids i-0bdc4219e66944afa \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

Die folgende Beispielausgabe zeigt den Schlüsselnamen, der das Benennungsformat `do-not-delete-rds-custom-ssh-privatekey-`*`resource_id`*`-`*`uuid`* oder verwendet`rds-custom!oracle-do-not-delete-`*`resource_id`*`-`*`uuid`*`-ssh-privatekey`.

```
do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c
rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey
```

4. Speichern Sie den privaten Schlüssel in einer PEM-Datei, die nach dem Schlüssel benannt ist, unter Verwendung von `aws secretsmanager`.

Im folgenden Beispiel wird der Schlüssel `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c` in einer Datei in Ihrem `/tmp` Verzeichnis gespeichert.

```
aws secretsmanager get-secret-value \
    --secret-id do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c \
    --query SecretString \
    --output text >/tmp/do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

Im folgenden Beispiel wird der Schlüssel `rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey` in einer Datei in Ihrem `/tmp` Verzeichnis gespeichert.

```
aws secretsmanager get-secret-value \
    --secret-id rds-custom!oracle-do-not-delete-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey \
    --query SecretString \
    --output text >/tmp/rds-custom!oracle-do-not-delete-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey.pem
```

Schritt 3: Stellen Sie mithilfe des SSH-Dienstprogramms eine Verbindung zu Ihrer EC2 Instance her

Ihre Verbindungsmethode hängt davon ab, ob Sie eine Verbindung mit einer privaten DB-Instance oder mit einer öffentlichen Instance herstellen. Für eine private Verbindung müssen Sie SSH-Tunneling über AWS Systems Manager konfigurieren.

So stellen Sie mithilfe des EC2 SSH-Dienstprogramms eine Verbindung zu einer Instanz her

1. Ändern Sie für private Verbindungen Ihre SSH-Konfigurationsdatei, um Befehle an AWS Systems Manager Session Manager weiterzuleiten. Für öffentliche Verbindungen fahren Sie mit Schritt 2 fort.

   Fügen Sie die folgenden Zeilen zu `~/.ssh/config` hinzu. Diese Zeilen leiten SSH-Befehle für Hosts weiter, deren Namen mit `i-` oder `mi-` beginnen.

   ```
   Host i-* mi-*
       ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-
   StartSSHSession --parameters 'portNumber=%p'"
   ```

2. Wechseln Sie in das Verzeichnis, das Ihre PEM-Datei enthält. Benutzen Sie`chmod` und legen Sie die Berechtigungen auf `400` fest.

   Im folgenden Beispiel wird das `/tmp` Verzeichnis geändert und die Berechtigungen für die PEM-Datei festgelegt. `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem`

   ```
   cd /tmp
   chmod 400 do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
   ```

Im folgenden Beispiel wird das `/tmp` Verzeichnis geändert und die Berechtigungen für die PEM-Datei festgelegt. `rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-privatekey.pem`

```
cd /tmp
chmod 400 rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-
privatekey.pem
```

3. Führen Sie das SSH-Hilfsprogramm aus und geben Sie die PEM-Datei und entweder den öffentlichen DNS-Namen (für öffentliche Verbindungen) oder die EC2 Instanz-ID (für private Verbindungen) an. Melden Sie sich als Benutzer `ec2-user` an.

   Das folgende Beispiel stellt mithilfe des DNS-Namens eine Verbindung zu einer öffentlichen Instance her. `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`

   ```
   # .pem file using naming prefix do-not-delete
   ssh -i \
     "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
     ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com

   # .pem file using naming prefix rds-custom!oracle-do-not-delete
   ssh -i \
     "rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-
   privatekey.pem" \
     ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com
   ```

   Im folgenden Beispiel wird über die Instanz-ID eine Verbindung zu einer privaten EC2 Instance hergestellt`i-0bdc4219e66944afa`.

   ```
   # .pem file using naming prefix do-not-delete
   ssh -i \
     "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
     ec2-user@i-0bdc4219e66944afa

   # .pem file using naming prefix rds-custom!oracle-do-not-delete
   ssh -i \
     "rds-custom!oracle-do-not-delete-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c-ssh-
   privatekey.pem" \
   ```

```
ec2-user@i-0bdc4219e66944afa
```

Anmelden als SYS bei Ihrer Datenbank von RDS Custom für Oracle

Nachdem Sie Ihre DB-Instance von RDS Custom erstellt haben, können Sie sich bei Ihrer Oracle-Datenbank als Benutzer SYS anmelden, wodurch Sie SYSDBA-Berechtigungen erhalten. Ihnen stehen folgende Anmeldeoptionen zur Verfügung:

- Rufen Sie das SYS-Passwort von Secrets Manager ab und geben Sie dieses Passwort in Ihrem SQL-Client an.
- Verwenden Sie die Betriebssystemauthentifizierung, um sich bei Ihrer Datenbank anzumelden. In diesem Fall benötigen Sie kein Passwort.

Suchen des SYS-Passworts für Ihre Datenbank von RDS Custom für Oracle

Sie können sich bei Ihrer Oracle-Datenbank als SYS oder SYSTEM oder durch Angabe des Hauptbenutzernamens in einem API-Aufruf anmelden. Das Passwort für SYS und SYSTEM ist in Secrets Manager gespeichert.

Das Geheimnis verwendet das Benennungsformat `do-not-delete-rds-custom-`*`resource_id`*`-`*`uuid`* oder`rds-custom!oracle-do-not-delete-`*`resource_id`*`-`*`uuid`*. Sie finden das Passwort über die AWS Management Console.

Konsole

So finden Sie das SYS-Passwort für Ihre Datenbank in Secrets Manager

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Führen Sie in der RDS-Konsole folgende Schritte aus:
   a. Wählen Sie im Navigationsbereich Datenbanken aus.
   b. Wählen Sie den Namen Ihrer DB-Instance von RDS Custom für Oracle aus.
   c. Wählen Sie Konfiguration.
   d. Kopieren Sie den Wert unter Ressourcen-ID. Ihre Ressourcen-ID könnte beispielsweise db-ABC12 CDE3 FGH4 I5 lauten JKLMNO6 PQR7.

3. Öffnen Sie die Secrets Manager Manager-Konsole unter https://console.aws.amazon.com/secretsmanager/.
4. Führen Sie in der Secrets-Manager-Konsole folgende Schritte aus:
   a. Wählen Sie im linken Navigationsbereich Secrets aus.
   b. Filtern Sie die Secrets nach der Ressourcen-ID, die Sie in Schritt 2.d kopiert haben.
   c. Wählen Sie das Geheimnis, das das Benennungsformat do-not-delete-rds-custom-***resource_id*** - ***uuid*** oder rds-custom verwendet! oracle-do-not-delete- -. ***resource_id*** ***uuid*** Das *resource_id* ist die Ressourcen-ID, die Sie in Schritt 2.d kopiert haben.

      Wenn Ihre Ressourcen-ID beispielsweise db- ABC12 CDE3 FGH4 I5 JKLMNO6 PQR7 und Ihre UUID 1234ab ist, heißt Ihr Secret -custom-db- I5 do-not-delete-rds-1234ab oder rds-custom! ABC12 CDE3 FGH4 JKLMNO6 PQR7 oracle-do-not-delete-db- I5 -1234ab. ABC12 CDE3 FGH4 JKLMNO6 PQR7
   d. Wählen Sie im Bereich Secret-Wert die Option Secret-Wert abrufen aus.
   e. Kopieren Sie im Feld Schlüssel/Wert den Wert für Passwort.
5. Installieren Sie SQL*Plus auf Ihrer DB-Instance und melden Sie sich bei Ihrer Datenbank als SYS an. Weitere Informationen finden Sie unter Schritt 3: Verbinden Ihres SQL-Clients mit einer Oracle-DB-Instance.

Anmelden bei Ihrer Datenbank von RDS Custom für Oracle-Datenbank mit der Betriebssystemauthentifizierung

Der Betriebssystembenutzer `rdsdb` besitzt die Oracle-Datenbankbinärdateien. Sie können zum Benutzer `rdsdb` wechseln und sich ohne Passwort bei Ihrer Datenbank von RDS Custom für Oracle anmelden.

1. Connect zu Ihrer DB-Instance mit her AWS Systems Manager. Weitere Informationen finden Sie unter Mit Session Manager eine Verbindung zu Ihrer RDS Custom DB-Instance herstellen.
2. Wechseln Sie zum Benutzer `rdsdb`.

```
sudo su - rdsdb
```

3. Melden Sie sich mit der Betriebssystemauthentifizierung bei Ihrer Datenbank an. Sie können den `sql` Alias `sqlplus / as sysdba` oder verwenden.

```
$ sqlplus / as sysdba
```

```
SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 12 20:11:08 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.10.0.0.0
```

Alternativ können Sie den `sql` Alias verwenden:

```
$ sql
```

Verwaltung einer Amazon RDS Custom for Oracle DB-Instance

Amazon RDS Custom unterstützt einen Teil der üblichen Verwaltungsaufgaben für Amazon RDS DB-Instances. Im Folgenden finden Sie Anweisungen für die unterstützten Verwaltungsaufgaben von RDS Custom for Oracle mithilfe von AWS Management Console und. AWS CLI

Themen

- Arbeiten mit Container-Datenbanken (CDBs) in RDS Custom for Oracle
- Arbeiten mit Hochverfügbarkeitsfunktionen für RDS Custom for Oracle
- Anpassung Ihrer RDS benutzerdefinierten Umgebung
- Ändern Ihrer RDS Custom for Oracle-DB-Instance
- Den Zeichensatz einer RDS Custom for Oracle DB-Instance ändern
- Einstellung des LANG Werts NLS _ in RDS Benutzerdefiniert für Oracle
- Unterstützung für transparente Datenverschlüsselung in SQL Server
- Markieren von Ressourcen für RDS Custom for Oracle
- Löschen einer RDS benutzerdefinierten für Oracle-DB-Instance

Arbeiten mit Container-Datenbanken (CDBs) in RDS Custom for Oracle

Sie können Ihre RDS Custom for Oracle-DB-Instance entweder mit der Oracle-Multitenant-Architektur (`custom-oracle-ee-cdb`oder dem `custom-oracle-se2-cdb` Engine-Typ) oder mit der herkömmlichen CDB Nicht-Architektur (`custom-oracle-ee`oder dem `custom-oracle-se2` Engine-Typ) erstellen. Wenn Sie eine Container-Datenbank (CDB) erstellen, enthält sie eine austauschbare Datenbank (PDB) und einen Seed. PDB Sie können weitere PDBs manuell mit Oracle erstellen. SQL

PDBund CDB Namen

Wenn Sie eine RDS Custom for CDB Oracle-Instanz erstellen, geben Sie einen Namen für die Initiale anPDB. Standardmäßig PDB ist Ihre Initiale benannt`ORCL`. Sie können einen anderen Namen wählen.

Standardmäßig CDB ist Ihr Name`RDSCDB`. Sie können einen anderen Namen wählen. Der CDB Name ist auch der Name Ihres Oracle-Systembezeichners (SID), der den Speicher und die Prozesse, die Ihren verwalten, eindeutig identifiziertCDB. Weitere Informationen über Oracle finden Sie unter Oracle SID System Identifier (SID) in Oracle Database Concepts.

Sie können bestehende Dateien nicht PDBs mit Amazon umbenennen RDSAPIs. Sie können das auch nicht CDB mit dem `modify-db-instance` Befehl umbenennen.

PDBVerwaltung

Im Modell der gemeinsamen Verantwortung von RDS Custom for Oracle sind Sie für die Verwaltung PDBs und Erstellung aller zusätzlichen Aufgaben verantwortlichPDBs. RDSBenutzerdefiniert schränkt die Anzahl von nicht einPDBs. Sie können manuell etwas erstellen, ändern und löschen, PDBs indem Sie eine Verbindung zum CDB Stammverzeichnis herstellen und eine SQL Anweisung ausführen. Erstellen Sie PDBs auf einem EBS Amazon-Datenvolume, um zu verhindern, dass die DB-Instance den Support-Perimeter verlässt.

Gehen Sie wie folgt vorPDBs, um Ihr CDBs oder zu ändern:

1. Unterbrechen Sie die Automatisierung, um Störungen durch RDS benutzerdefinierte Aktionen zu vermeiden.
2. Ändern Sie Ihr CDB oderPDBs.
3. Sichern Sie alle geändertenPDBs.
4. Setzen Sie die RDS benutzerdefinierte Automatisierung fort.

Automatische Wiederherstellung des CDB Stammverzeichnisses

RDSBenutzerdefiniert hält das CDB Stammverzeichnis auf die gleiche Weise geöffnet wie ein nicht CDB geöffnetes Stammverzeichnis. Wenn sich der Status des CDB Stammverzeichnisses ändert, versucht die Überwachungs- und Wiederherstellungsautomatisierung, das CDB Stammverzeichnis in den gewünschten Zustand zurückzuversetzen. Sie erhalten RDS Ereignisbenachrichtigungen, wenn der Root heruntergefahren (`RDS-EVENT-0004`) oder neu gestartet CDB wird (`RDS-EVENT-0006`), ähnlich wie bei der CDB Nicht-Architektur. RDSBenutzerdefinierte Versuche, den `READ WRITE` All-In-Modus PDBs beim Start der DB-Instance zu öffnen. Wenn einige nicht geöffnet werden PDBs können, veröffentlicht RDS Custom das folgende Ereignis:`tenant database shutdown`.

Arbeiten mit Hochverfügbarkeitsfunktionen für RDS Custom for Oracle

RDS Custom for Oracle bietet integrierte Hochverfügbarkeit durch Multi-AZ-Bereitstellungen. Alternativ können Sie Oracle Data Guard je nach Ihren Anwendungsfällen als vom Kunden verwaltete Option verwenden.

Multi-AZ-Bereitstellungen (vollständig verwaltet)

Bei Multi-AZ-Bereitstellungen für RDS Custom for Oracle stellt Amazon RDS automatisch ein synchrones Standby-Replikat in einer anderen Availability Zone (AZ) bereit und verwaltet es. Die primäre DB-Instance wird aus Gründen der Datenredundanz synchron auf ein Standby-Replikat AZs repliziert. Die Multi-AZ-Bereitstellung wird sowohl in der Enterprise Edition als auch in der Standard Edition 2 unterstützt. Details dazu finden Sie unter the section called “Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Oracle”.

Oracle Data Guard (vom Kunden verwaltet)

Alternativ können Sie eine hohe Verfügbarkeit erreichen, indem Sie Oracle Data Guard manuell so konfigurieren, dass Daten zwischen RDS Custom for Oracle DB-Instances repliziert werden. Die primäre DB-Instance synchronisiert Daten automatisch mit den Standby-Instanzen. Oracle Data Guard wird nur in der Enterprise Edition unterstützt.

Sie können Ihre Hochverfügbarkeitsumgebung folgendermaßen konfigurieren:

- Konfigurieren Sie Standby-Instanzen AZs auf verschiedenen Ebenen, um bei AZ-Ausfällen widerstandsfähig zu sein.
- Versetzen Sie Ihre Standby-Datenbanken in den bereitgestellten oder schreibgeschützten Modus. Für den Nur-Lese-Modus ist eine Oracle Active Data Guard-Lizenz erforderlich.
- Scheitern oder wechseln Sie ohne Datenverlust von der Primärdatenbank zu einer Standby-Datenbank.
- Migrieren Sie Daten, indem Sie Hochverfügbarkeit für Ihre lokale Instance konfigurieren und dann ein Failover durchführen oder zur Standby-Datenbank von RDS Custom for Oracle wechseln.

Informationen zur Konfiguration von Oracle Data Guard für Hochverfügbarkeit finden Sie im AWS Blog Build high availability for RDS Custom for Oracle using Read Replicas. Sie können folgende Aufgaben ausführen:

- Verwenden Sie einen VPN (Virtual Private Network) -Tunnel, um Daten während der Übertragung für Ihre Instanzen mit hoher Verfügbarkeit zu verschlüsseln. Die Verschlüsselung bei der Übertragung wird von RDS Custom for Oracle nicht automatisch konfiguriert.
- Konfigurieren Sie Oracle Fast-Failover Observer (FSFO), um Ihre Hochverfügbarkeitsinstanzen zu überwachen.
- Erlauben Sie dem Beobachter, ein automatisches Failover durchzuführen, wenn die erforderlichen Bedingungen erfüllt sind.

Anpassung Ihrer RDS benutzerdefinierten Umgebung

RDSCustom for Oracle umfasst integrierte Funktionen, mit denen Sie Ihre DB-Instance-Umgebung anpassen können, ohne die Automatisierung unterbrechen zu müssen. Sie können RDS APIs Ihre Umgebung beispielsweise wie folgt anpassen:

- Erstellen Sie DB-Snapshots und stellen Sie sie wieder her, um eine Klonumgebung zu erstellen.
- Erstellen Sie Lesereplikate.
- Ändern Sie die Speichereinstellungen.
- Ändern Sie dasCEV, um Versionsupdates anzuwenden

Für einige Anpassungen, wie z. B. das Ändern des Zeichensatzes, können Sie den RDS APIs nicht verwenden. In diesen Fällen müssen Sie die Umgebung manuell ändern, indem Sie als Root-Benutzer auf Ihre EC2 Amazon-Instance zugreifen oder sich als bei Ihrer Oracle-Datenbank anmeldenSYSDBA.

Um Ihre Instance manuell anzupassen, müssen Sie die RDS benutzerdefinierte Automatisierung anhalten und wieder aufnehmen. Diese Pause stellt sicher, dass Ihre Anpassungen die RDS benutzerdefinierte Automatisierung nicht beeinträchtigen. Auf diese Weise vermeiden Sie, dass der Support-Perimeter unterbrochen wird, wodurch die Instance in den `unsupported-configuration` Status versetzt wird, bis Sie die zugrunde liegenden Probleme behoben haben. Das Anhalten und Wiederaufnehmen sind die einzigen unterstützten Automatisierungsaufgaben, wenn Sie eine RDS Custom for Oracle-DB-Instance ändern.

Allgemeine Schritte zum Anpassen Ihrer benutzerdefinierten Umgebung RDS

Gehen Sie wie folgt vor, um Ihre RDS benutzerdefinierte DB-Instance anzupassen:

1. Halten Sie die RDS benutzerdefinierte Automatisierung mithilfe der Konsole oder für einen bestimmten Zeitraum anCLI.
2. Identifizieren Sie Ihre zugrunde liegende EC2 Amazon-Instance.
3. Stellen Sie mithilfe von SSH Schlüsseln Connect zu Ihrer zugrunde liegenden EC2 Amazon-Instance her oder AWS Systems Manager.
4. Überprüfen Sie Ihre aktuellen Konfigurationseinstellungen auf Datenbank- oder Betriebssystemebene.

Sie können Ihre Änderungen überprüfen, indem Sie die ursprüngliche Konfiguration mit der geänderten Konfiguration vergleichen. Verwenden Sie je nach Art der Anpassung Betriebssystemtools oder Datenbankabfragen.

5. Passen Sie Ihre RDS Custom for Oracle-DB-Instance nach Bedarf an.
6. Starten Sie Ihre Instance oder Datenbank neu, falls erforderlich.

> **Note**
>
> In einem lokalen Oracle CDB können Sie einen bestimmten Öffnungsmodus für die PDBs Verwendung eines integrierten Befehls oder nach einem Start-Trigger beibehalten. Dieser Mechanismus versetzt PDBs den Computer in einen bestimmten Zustand, wenn der Computer CDB neu gestartet wird. Beim Öffnen verwirft CDB die RDS benutzerdefinierte Automatisierung alle vom Benutzer angegebenen beibehaltenen Status und versucht, alle zu öffnen. PDBs Wenn RDS Custom nicht alle öffnen kannPDBs, wird das folgende Ereignis ausgelöst:. `The following PDBs failed to open: list-of-PDBs`

7. Überprüfen Sie Ihre neuen Konfigurationseinstellungen, indem Sie sie mit den vorherigen Einstellungen vergleichen.
8. Setzen RDS Sie die benutzerdefinierte Automatisierung auf eine der folgenden Arten fort:
   - Nehmen Sie die Automatisierung manuell fort.
   - Warten Sie, bis der Pausezeitraum endet. In diesem Fall nimmt RDS Custom die Überwachung und die Instanzwiederherstellung automatisch wieder auf.
9. Überprüfen Sie das RDS Custom Automation Framework

   Wenn Sie die vorherigen Schritte korrekt ausgeführt haben, startet RDS Custom ein automatisiertes Backup. Der Status der Instance in der Konsole lautet Verfügbar.

Bewährte Methoden und step-by-step Anleitungen finden Sie in den AWS Blogbeiträgen Vornehmen von Konfigurationsänderungen an einer Amazon RDS Custom for Oracle-Instance: Teil 1 und Neuerstellen einer Amazon RDS Custom for Oracle-Datenbank: Teil 2.

Ihre benutzerdefinierte DB-Instance pausieren und wieder aufnehmen RDS

Sie können die Automatisierung für Ihre DB-Instance über die Konsole oder anhalten und wieder aufnehmen. CLI

Konsole

Um die RDS benutzerdefinierte Automatisierung anzuhalten oder fortzusetzen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte DB-Instance aus, die Sie ändern möchten.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Wählen Sie für den RDSbenutzerdefinierten Automatisierungsmodus eine der folgenden Optionen aus:

   - Pausiert unterbricht die Überwachung und Instance-Wiederherstellung für die RDS benutzerdefinierte DB-Instance. Geben Sie die Pausedauer ein, für die Sie (in Minuten) die Dauer des Automatisierungsmodus möchten. Der Standardwert beträgt 60 Minuten. Der Maximalwert beträgt 1440 Minuten.
   - Vollständige Automatisierung nimmt die Automatisierung wieder auf.
5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.

   Eine Meldung weist darauf hin, dass RDS Custom die Änderungen sofort anwendet.
6. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern. Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

   Auf der RDS Konsole werden die Details für die Änderung angezeigt. Wenn Sie die Automatisierung angehalten haben, gibt der Status Ihrer RDS benutzerdefinierten DB-Instance an, dass die Automatisierung angehalten wurde.
7. (Optional) Wählen Sie im Navigationsbereich Datenbanken und dann Ihre RDS benutzerdefinierte DB-Instance aus.

   Im Übersichtsbereich gibt der RDSbenutzerdefinierte Automatisierungsmodus den Automatisierungsstatus an. Wenn die Automatisierung angehalten ist, lautet der Wert Angehalten. Die Automatisierung wird fortgesetzt in ***num*** Minuten.

AWS CLI

Verwenden Sie den `modify-db-instance` AWS CLI Befehl, um die RDS benutzerdefinierte Automatisierung anzuhalten oder fortzusetzen. Identifizieren Sie die DB-Instance mit dem erforderlichen Parameter `--db-instance-identifier`. Steuern Sie den Automatisierungsmodus mit den folgenden Parametern:

- `--automation-mode` gibt den Pausestatus der DB-Instance an. Gültige Werte sind `all-paused`, was die Automatisierung anhält, und `full`, was es wieder aufnimmt.
- `--resume-full-automation-mode-minutes` gibt die Dauer der Pause an. Der Standardwert beträgt 60 Minuten.

> **Note**
>
> Unabhängig davon, ob Sie `--no-apply-immediately` oder angeben`--apply-immediately`, wendet RDS Benutzerdefiniert Änderungen so schnell wie möglich asynchron an.

`ResumeFullAutomationModeTime`Gibt in der Befehlsantwort die Wiederaufnahmezeit als UTC Zeitstempel an. Wenn der Automatisierungsmodus `all-paused` ist, können Sie `modify-db-instance` verwenden, um den Automatisierungsmodus fortzusetzen oder den Pausezeitraum zu verlängern. Es werden keine anderen `modify-db-instance`-Optionen unterstützt.

Das folgende Beispiel unterbricht die Automatisierung für `my-custom-instance` für 90 Minuten.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
```

```
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

Im folgenden Beispiel wird die Pausedauer um weitere 30 Minuten verlängert. Die 30 Minuten werden zur Originalzeit hinzugefügt, die in `ResumeFullAutomationModeTime` angezeigt wird.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

Im folgenden Beispiel wird die vollständige Automatisierung für `my-custom-instance` wieder aufgenommen.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

In der folgenden partiellen Beispielausgabe ist der ausstehende `AutomationMode`-Wert `full`.

```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
```

```
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```


Ändern Ihrer RDS Custom for Oracle-DB-Instance

Das Ändern einer RDS Custom for Oracle-DB-Instance ähnelt dem Ändern einer RDS Amazon-DB-Instance. Sie können Einstellungen wie die folgenden ändern:

- DB-Instance-Klasse
- Speicherzuweisung und -typ
- Aufbewahrungszeitraum für Backups
- Löschschutz
- Option group
- CEV(siehe Eine RDS benutzerdefinierte für Oracle DB-Instance aktualisieren)
- Port

Themen

- Anforderungen und Einschränkungen bei der Änderung Ihres DB-Instance-Speichers
- Anforderungen und Einschränkungen bei der Änderung Ihrer DB-Instance-Klasse

- So erstellt RDS Custom Ihre DB-Instance, wenn Sie die Instance-Klasse ändern
- Ändern Sie Ihre RDS benutzerdefinierte Oracle-DB-Instance

Anforderungen und Einschränkungen bei der Änderung Ihres DB-Instance-Speichers

Beachten Sie die folgenden Anforderungen und Einschränkungen, wenn Sie den Speicher für eine RDS Custom for Oracle-DB-Instance ändern:

- Der zugewiesene Mindestspeicher für RDS Custom for Oracle beträgt 40 GiB und der Höchstspeicher 64 TiB.
- Wie bei Amazon RDS können Sie den zugewiesenen Speicherplatz nicht verringern. Dies ist eine Beschränkung der EBS Amazon-Volumina.
- Die automatische Speicherskalierung wird für RDS benutzerdefinierte DB-Instances nicht unterstützt.
- Alle Speichervolumes, die Sie manuell an Ihre RDS benutzerdefinierte DB-Instance anhängen, befinden sich außerhalb des Support-Bereichs.

  Weitere Informationen finden Sie unter Support-Perimeter in RDS Custom.
- Magnetischer (Standard-) EBS Amazon-Speicher wird für RDS Custom nicht unterstützt. Sie können nur die Speichertypen io1, gp2 oder gp3 SSD wählen.

Weitere Informationen zum EBS Amazon-Speicher finden Sie unterAmazon RDS-DB-Instance-Speicher. Allgemeine Informationen zur Änderung des Speichers finden Sie unter Arbeiten mit Speicher für Amazon RDS DB-Instances.

Anforderungen und Einschränkungen bei der Änderung Ihrer DB-Instance-Klasse

Beachten Sie die folgenden Anforderungen und Einschränkungen, wenn Sie die Instance-Klasse für eine RDS Custom for Oracle-DB-Instance ändern:

- Ihre DB-Instance muss sich im Status `available` befinden.
- Ihre DB-Instance muss über mindestens 100 MiB freien Speicherplatz auf dem Root-Volume, Datenvolume und Binärvolume verfügen.
- Sie können Ihrer RDS Custom for Oracle-DB-Instance nur eine einzige Elastic IP (EIP) zuweisen, wenn Sie die standardmäßige elastic network interface (ENI) verwenden. Wenn Sie Ihrer DB-Instance mehrere ENIs hinzufügen, schlägt der Änderungsvorgang fehl.
- Alle RDS Custom for Oracle-Tags müssen vorhanden sein.

- Wenn Sie RDS Custom für die Oracle-Replikation verwenden, beachten Sie die folgenden Anforderungen und Einschränkungen:
  - Für primäre DB-Instances und Lesereplikate können Sie die Instance-Klasse jeweils nur für eine DB-Instance ändern.
  - Wenn Ihre RDS Custom for Oracle-DB-Instance über eine lokale Primär- oder Replikatdatenbank verfügt, stellen Sie sicher, dass Sie die privaten IP-Adressen auf der lokalen DB-Instance nach Abschluss der Änderung manuell aktualisieren. Diese Aktion ist erforderlich, um die Oracle-Funktionalität aufrechtzuerhalten. DataGuard RDSCustom for Oracle veröffentlicht ein Ereignis, wenn die Änderung erfolgreich ist.
  - Sie können Ihre DB-Instance-Klasse RDS Custom for Oracle nicht ändern, wenn die primären DB-Instances oder Read Replica-DB-Instances konfiguriert sind FSFO (Fast-Start Failover).

So erstellt RDS Custom Ihre DB-Instance, wenn Sie die Instance-Klasse ändern

Wenn Sie Ihre Instance-Klasse ändern, erstellt RDS Custom Ihre DB-Instance wie folgt:

- Erzeugt die EC2 Amazon-Instance.
- Erstellt das Root-Volumes aus dem letzten Snapshot. RDSCustom for Oracle speichert keine Informationen, die dem Root-Volume nach dem letzten DB-Snapshot hinzugefügt wurden.
- Erzeugt CloudWatch Amazon-Alarme.
- Erstellt ein EC2 SSH Amazon-Schlüsselpaar, wenn Sie das ursprüngliche key pair gelöscht haben. Andernfalls behält RDS Custom for Oracle das ursprüngliche key pair bei.
- Erstellt neue Ressourcen mithilfe der Tags, die Ihrer DB-Instance angefügt sind, wenn Sie die Änderung einleiten. RDSCustom überträgt keine Tags auf die neuen Ressourcen, wenn sie direkt an die zugrunde liegenden Ressourcen angehängt sind.
- Überträgt die Binär- und Datenvolumes mit den neuesten Änderungen an die neue DB-Instance.
- Überträgt die elastische IP-Adresse (EIP). Wenn die DB-Instance öffentlich zugänglich ist, fügt RDS Custom der neuen DB-Instance vorübergehend eine öffentliche IP-Adresse hinzu, bevor die übertragen wirdEIP. Wenn die DB-Instance nicht öffentlich zugänglich ist, erstellt RDS Custom keine öffentlichen IP-Adressen.

Ändern Sie Ihre RDS benutzerdefinierte Oracle-DB-Instance

Sie können die DB-Instance-Klasse oder den Speicher mithilfe der Konsole AWS CLI, oder ändern RDSAPI.

Konsole

Um eine RDS benutzerdefinierte DB-Instance für Oracle zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. (Optional) Wählen Sie in der Instance-Konfiguration einen Wert für die DB-Instance-Klasse aus. Informationen zu unterstützten Klassen finden Sie unter Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle.
6. (Optional) Nehmen Sie im Speicher nach Bedarf die folgenden Änderungen vor:
   a. Geben Sie einen neuen Wert für Allocated storage (Zugewiesener Speicherplatz) ein. Er muss größer als der aktuelle Wert und zwischen 40 GiB und 64 TiB sein.
   b. Ändern Sie den Wert für Speichertyp in General Purpose SSD (gp2), General Purpose SSD (gp3) oder Provisioned IOPS (io1).
   c. Wenn Sie Provisioned IOPS (io1) oder General Purpose SSD (gp3) verwenden, können Sie den Wert Provisioned ändern. IOPS
7. (Optional) Nehmen Sie unter Zusätzliche Konfiguration nach Bedarf die folgenden Änderungen vor:
   - Wählen Sie für Optionsgruppe eine neue Optionsgruppe aus. Weitere Informationen finden Sie unter Mit Optionsgruppen in RDS Custom for Oracle arbeiten.
8. Klicken Sie auf Continue (Fortfahren).
9. Wählen Sie During the next scheduled maintenance window (Während des nächsten geplanten Wartungsfensters) oder Sofort aus.
10. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Verwenden Sie den modify-db-instance AWS CLI Befehl, um den Speicher für eine RDS Custom for Oracle-DB-Instance zu ändern. Stellen Sie die folgenden Parameter nach Bedarf ein:

- `--db-instance-class` – Eine neue Instancde-Klasse. Informationen zu unterstützten Klassen finden Sie unter Unterstützung von DB-Instance-Klassen für RDS Custom for Oracle.
- `--allocated-storage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes. Er muss größer als der aktuelle Wert und zwischen 40 und 65.536 GiB sein.
- `--storage-type` – Der Speichertyp: gp2, gp3 oder io1.
- `--iops`— IOPS Für die DB-Instance bereitgestellt, wenn die Speichertypen io1 oder gp3 verwendet werden.
- `--apply-immediately` – Verwenden Sie `--apply-immediately`, um die Speicheränderungen sofort anzuwenden.

  Oder verwenden Sie `--no-apply-immediately` (Standardeinstellung), um die Änderungen während des nächsten Wartungsfensters anzuwenden.

Im folgenden Beispiel wird die DB-Instance-Klasse von my-cfo-instance in db.m5.16xlarge geändert. Der Befehl ändert außerdem die Speichergröße auf 1 TiB, den Speichertyp auf io1, Provisioned auf 3000 und die Optionsgruppe IOPS auf cfo-ee-19-mt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cfo-instance \
    --db-instance-class db.m5.16xlarge \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 1024 \
    --option-group cfo-ee-19-mt \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cfo-instance ^
    --db-instance-class db.m5.16xlarge ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 1024 ^
```

```
    --option-group cfo-ee-19-mt ^
    --apply-immediately
```

Den Zeichensatz einer RDS Custom for Oracle DB-Instance ändern

RDSCustom for Oracle verwendet standardmäßig den ZeichensatzUS7ASCII. Möglicherweise möchten Sie verschiedene Zeichensätze angeben, um die Anforderungen an Sprache oder Multibyte-Zeichen zu erfüllen. Wenn Sie RDS Custom for Oracle verwenden, können Sie die Automatisierung anhalten und dann den Zeichensatz Ihrer Datenbank manuell ändern.

Für das Ändern des Zeichensatzes einer RDS Custom for Oracle-DB-Instance gelten die folgenden Anforderungen:

- Sie können das Zeichen nur in einer neu bereitgestellten RDS benutzerdefinierten Instance ändern, die über eine leere Datenbank oder eine Startdatenbank ohne Anwendungsdaten verfügt. In allen anderen Szenarien ändern Sie den Zeichensatz mithilfe von DMU (Database Migration Assistant for Unicode).
- Sie können nur zu einem Zeichensatz wechseln, der von RDS for Oracle unterstützt wird. Weitere Informationen finden Sie unter Unterstützte DB-Zeichensätze.

Um den Zeichensatz einer RDS Custom for Oracle DB-Instance zu ändern

1. Unterbrechen Sie die RDS benutzerdefinierte Automatisierung. Weitere Informationen finden Sie unter Ihre benutzerdefinierte DB-Instance pausieren und wieder aufnehmen RDS.
2. Melden Sie sich bei Ihrer Datenbank als Benutzer mit SYSDBA-Berechtigungen an.
3. Starten Sie die Datenbank im eingeschränkten Modus neu, ändern Sie den Zeichensatz und starten Sie die Datenbank dann im normalen Modus erneut.

   Führen Sie das folgende Skript in Ihrem SQL Client aus:

   ```
   SHUTDOWN IMMEDIATE;
   STARTUP RESTRICT;
   ALTER DATABASE CHARACTER SET INTERNAL_CONVERT AL32UTF8;
   SHUTDOWN IMMEDIATE;
   STARTUP;
   SELECT VALUE FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET';
   ```

   Stellen Sie sicher, dass die Ausgabe den richtigen Zeichensatz anzeigt:

```
VALUE
--------
AL32UTF8
```

4. Setzen Sie die RDS benutzerdefinierte Automatisierung fort. Weitere Informationen finden Sie unter Ihre benutzerdefinierte DB-Instance pausieren und wieder aufnehmen RDS.

Einstellung des LANG Werts NLS _ in RDS Benutzerdefiniert für Oracle

Ein Gebietsschema ist eine Reihe von Informationen, die sprachlichen und kulturellen Anforderungen für eine bestimmte Sprache und ein bestimmtes Land entsprechen. Um das Gebietsschemaverhalten für Oracle-Software festzulegen, legen Sie die Umgebungsvariable NLS_LANG auf Ihrem Client-Host fest. Mit dieser Variablen werden die Sprache, die Region und der Zeichensatz definiert, die von der Clientanwendung in einer Datenbanksitzung verwendet werden.

Für RDS Benutzerdefiniert für Oracle können Sie nur die Sprache in der NLS_LANG Variablen festlegen: das Gebiet und die Standardwerte für die Zeichenverwendung. Die Sprache wird für Oracle-Datenbanknachrichten, Sortierung, Tagesnamen und Monatsnamen verwendet. Jede unterstützte Sprache hat einen eindeutigen Namen, beispielsweise Amerikanisch, Französisch oder Deutsch. Der Standardwert ist Amerikanisch, wenn keine Sprache angegeben ist.

Nachdem Sie Ihre RDS Custom for Oracle-Datenbank erstellt haben, können Sie NLS_LANG auf Ihrem Client-Host eine andere Sprache als Englisch einstellen. Um eine Liste der von Oracle Database unterstützten Sprachen zu sehen, melden Sie sich bei Ihrer RDS Custom for Oracle-Datenbank an und führen Sie die folgende Abfrage aus:

```
SELECT VALUE FROM V$NLS_VALID_VALUES WHERE PARAMETER='LANGUAGE' ORDER BY VALUE;
```

Sie können NLS_LANG in der Host-Befehlszeile festlegen. Im folgenden Beispiel wird die Sprache für Ihre Client-Anwendung mithilfe der Z-Shell unter Linux auf Deutsch festgelegt.

```
export NLS_LANG=German
```

Ihre Anwendung liest den Wert NLS_LANG beim Start und übermittelt ihn beim Herstellen einer Verbindung an die Datenbank.

Weitere Informationen finden Sie unter Choosing a Locale with the NLS _ LANG Environment Variable im Oracle Database Globalization Support Guide.

Unterstützung für transparente Datenverschlüsselung in SQL Server

RDSCustom unterstützt Transparent Data Encryption (TDE) für RDS benutzerdefinierte Oracle-DB-Instances.

Sie können die TDE Verwendung einer Option in einer benutzerdefinierten Optionsgruppe jedoch nicht aktivieren, wie dies RDS bei Oracle der Fall ist. Sie schalten TDE manuell ein. Weitere Informationen zum Verwenden von Oracle Transparent Data Encryption finden Sie unter Sichern gespeicherter Daten mit Transparent Data Encryption.

Markieren von Ressourcen für RDS Custom for Oracle

Sie können RDS Custom Ressourcen wie bei Amazon RDS-Ressourcen kennzeichnen, jedoch mit einigen wichtigen Unterschieden:

- Erstellen oder modifizieren Sie die`AWSRDSCustom`-Tag, das für die RDS Custom Automatisierung erforderlich ist. Wenn Sie dies tun, könnten Sie die Automatisierung unterbrechen.
- Das `Name` Tag wird zu benutzerdefinierten RDS-Ressourcen mit dem Präfixwert `do-not-delete-rds-custom` oder hinzugefügt`rds-custom!oracle-do-not-delete`. Vom Kunden übergebene Werte für den Schlüssel werden überschrieben.
- Tags, die während der Erstellung zu RDS Custom DB-Instanzen hinzugefügt wurden, werden an alle anderen verwandten RDS Custom Ressourcen weitergegeben.
- Tags werden nicht propagiert, wenn Sie sie nach der Erstellung der DB-Instance zu RDS Custom Ressourcen hinzufügen.

Allgemeine Informationen zum Markieren von Ressourcen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.

Löschen einer RDS benutzerdefinierten für Oracle-DB-Instance

Gehen Sie wie folgt vor, um eine RDS benutzerdefinierte DB-Instance zu löschen:

- Geben Sie den Namen der Instance an.
- Aktivieren oder deaktivieren Sie die Option, einen endgültigen DB-Snapshot der Instance zu erstellen.
- Wählen oder deaktivieren Sie die Option zum Speichern automatisierter Sicherungen.

Sie können eine RDS benutzerdefinierte DB-Instance mit der Konsole oder dem löschenCLI. Die zum Löschen einer DB-Instance erforderliche Zeit kann je nach Aufbewahrungszeitraum für Backups (d. h. wie viele Backups gelöscht werden sollen), wie viele Daten gelöscht werden und ob ein endgültiger Snapshot erstellt wird, variieren.

Konsole

Um eine RDS benutzerdefinierte DB-Instance zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte DB-Instance aus, die Sie löschen möchten. RDSBenutzerdefinierte DB-Instances zeigen die Rolle Instance (RDSCustom).
3. Klicken Sie bei Actions auf Delete.
4. Wählen Sie Retain automated backups (Automatisierte Sicherungen aufbewahren), um automatisierte Sicherungen aufzubewahren.
5. Geben Sie **`delete me`** in das Feld ein.
6. Wählen Sie Löschen.

AWS CLI

Sie löschen eine RDS benutzerdefinierte DB-Instance mit dem delete-db-instance AWS CLI Befehl. Identifizieren Sie die DB-Instance mit dem erforderlichen Parameter `--db-instance-identifier`. Die übrigen Parameter sind dieselben wie für eine RDS Amazon-DB-Instance, mit den folgenden Ausnahmen:

- `--skip-final-snapshot` ist erforderlich.
- `--no-skip-final-snapshot` wird nicht unterstützt.
- `--final-db-snapshot-identifier` wird nicht unterstützt.

Das folgende Beispiel löscht die benannte RDS `my-custom-instance` benutzerdefinierte DB-Instance und behält automatische Backups bei.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier my-custom-instance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

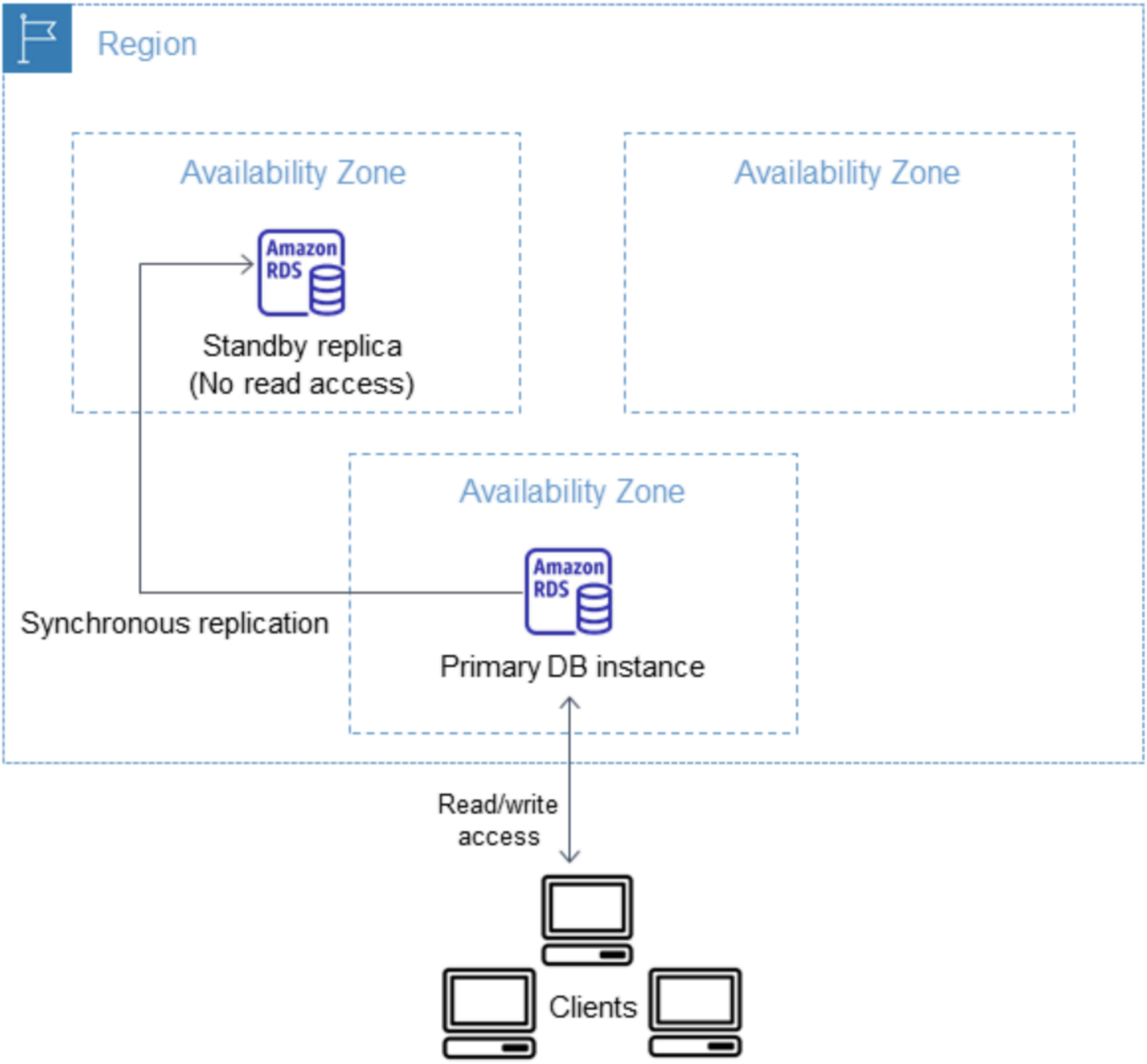
Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Oracle

In einer Multi-AZ-DB-Instance-Bereitstellung für RDS Custom for Oracle stellt Amazon RDS automatisch ein synchrones Standby-Replikat in einer anderen Availability Zone (AZ) bereit und verwaltet es. Die primäre DB-Instance wird über die Availability Zone synchron auf ein Standby-Replikat repliziert, um Datenredundanz zu erzielen.

> **Important**
>
> Eine Multi-AZ-Bereitstellung für RDS Custom for Oracle unterscheidet sich von Multi-AZ für RDS for Oracle. Im Gegensatz zu Multi-AZ für RDS for Oracle müssen Sie die Voraussetzungen für RDS Custom for Oracle einrichten, bevor Sie Ihre Multi-AZ-DB-Instance erstellen, da RDS Custom in Ihrem eigenen Konto ausgeführt wird, wofür Berechtigungen erforderlich sind. Außerdem müssen Sie die Multi-AZ-Instance auf dem CEV mit dem neuesten vom Service bereitgestellten AMI erstellen, das die Multi-AZ-Bereitstellungen unterstützt.
> Wenn Sie die Voraussetzungen nicht erfüllen, kann Ihre Multi-AZ-DB-Instance möglicherweise nicht ausgeführt werden oder wird automatisch in eine Single-AZ-DB-Instance geändert. Weitere Informationen über Voraussetzungen finden Sie unter the section called “Voraussetzungen”.

Wenn Sie eine DB-Instance mit hoher Verfügbarkeit ausführen, kann dies die Verfügbarkeit bei geplanten Systemwartungen verbessern. Im Falle einer geplanten Datenbankwartung oder einer ungeplanten Serviceunterbrechung wechselt Amazon RDS automatisch zur up-to-date Standby-DB-Instance. Mit dieser Funktion können Datenbankoperationen schnell ohne manuellen Eingriff fortgesetzt werden. Die Primär- und Standby-Instances verwenden denselben Endpunkt, dessen physische Netzwerkadresse im Rahmen des Failover-Prozesses auf das Standby-Replikat umgestellt wird. Sie müssen Ihre Anwendung nicht neu konfigurieren, wenn ein Failover auftritt.

Das folgende Diagramm zeigt die Multi-AZ-Architektur für RDS Custom for Oracle:



Sie können eine RDS Custom for Oracle Multi-AZ-Bereitstellung erstellen, indem Sie bei der Erstellung einer benutzerdefinierten RDS-DB-Instance Multi-AZ angeben. RDS Custom für Oracle-DB-Instances, die vor dem 30. Juni 2025 erstellt wurden, können nicht in Multi-AZ-Bereitstellungen konvertiert werden, indem die DB-Instance geändert und die Multi-AZ-Option angegeben wird. Sie müssen eine neue RDS Custom for Oracle DB-Instance erstellen, indem Sie die Multi-AZ-Option angeben. Neue DB-Instances, die mit dem vom Service bereitgestellten AMI erstellt wurden, das nach dem 30. Juni 2025 verfügbar ist, unterstützen die Multi-AZ-Option. Anschließend müssen Sie

Ihre bestehende RDS Custom for Oracle-Datenbank auf die neu erstellte DB-Instance migrieren. Details hierzu finden Sie unter the section called “Migration für CEVs vor dem 30.06.2025”.

Die Amazon RDS-Konsole zeigt die Availability Zone der Standby-Replik (die sekundäre AZ). Sie können auch den CLI-Befehl `describe-db-instances` oder die API-Operation `DescribeDBInstances` verwenden, um die sekundäre AZ zu suchen.

RDS Custom for Oracle DB-Instances mit Multi-AZ-Bereitstellung kann im Vergleich zu einer Single-AZ-Bereitstellung eine höhere Schreib- und Commit-Latenz aufweisen. Dieser Anstieg kann aufgrund der synchronen Datenreplikation zwischen DB-Instances entstehen. Es kann zu einer Änderung der Latenz kommen, wenn bei Ihrer Bereitstellung ein Failover auf das Standby-Replikat erfolgt, obwohl es auf Netzwerkkonnektivität mit niedriger Latenz zwischen Availability Zones ausgelegt AWS ist.

Themen

- Verfügbarkeit von Regionen und Versionen
- Einschränkungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Oracle
- Eine benutzerdefinierte RDS-Instanz für Oracle mit einer Multi-AZ-Bereitstellung erstellen
- Voraussetzungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Oracle
- Änderung einer RDS Custom for Oracle Single-AZ-Bereitstellung zu einer Multi-AZ-Bereitstellung
- Änderung einer Multi-AZ-Bereitstellung von RDS Custom for Oracle zu einer Single-AZ-Bereitstellung
- Failover-Prozess für eine Multi-AZ-Bereitstellung von RDS Custom for Oracle
- Bereitstellung von RDS Custom für Oracle mit AWS CloudFormation

Verfügbarkeit von Regionen und Versionen

Multi-AZ-Bereitstellungen für RDS Custom for Oracle werden für die folgenden Oracle Database-Editionen unterstützt:

- Oracle Database 12c Release 1 (12.1)
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 19c
- Enterprise Edition (EE)
- Standard Edition (2) SE2

> **Note**
>
> Multi-AZ-Bereitstellungen für RDS Custom for Oracle werden auf Oracle Database 18c nicht unterstützt.

Multi-AZ-Bereitstellungen für RDS Custom for Oracle sind in allen Regionen verfügbar, in denen RDS Custom for Oracle verfügbar ist. Weitere Informationen zur regionalen Verfügbarkeit von Multi-AZ-Bereitstellungen für RDS Custom for Oracle finden Sie unter. the section called “Unterstützte Regionen und DB-Engines für RDS Custom for Oracle”

Einschränkungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Oracle

Multi-AZ-Bereitstellungen mit RDS Custom for Oracle haben die folgenden Einschränkungen:

- Regionsübergreifende Multi-AZ-Bereitstellungen werden nicht unterstützt.
- Sie können die Standby-DB-Instance nicht so konfigurieren, dass sie Datenbank-Leseaktivitäten akzeptiert.
- Wenn Sie eine Custom Engine Version (CEV) mit einer Multi-AZ-Bereitstellung verwenden, verwendet Ihre Standby-DB-Instance ebenfalls dieselbe CEV. Die Standby-DB-Instance kann kein anderes CEV verwenden.
- Sie können keine Read Replica auf einer Multi-AZ-Bereitstellungs-Instance erstellen und Sie können auch nicht die primäre Instance der Read Replica so ändern, dass sie eine Multi-AZ-Bereitstellung hat.

Eine benutzerdefinierte RDS-Instanz für Oracle mit einer Multi-AZ-Bereitstellung erstellen

Um eine RDS Custom for Oracle DB-Instance mit einer Multi-AZ-Bereitstellung zu erstellen, folgen Sie den Schritten unter the section called “Einrichten Ihrer RDS Custom for Oracle-Umgebung” So richten Sie Ihre Umgebung mit den Voraussetzungen ein.

> **Important**
>
> Um die Einrichtung zu vereinfachen, empfehlen wir, die neueste AWS CloudFormation Vorlagendatei zu verwenden, die in den Anweisungen zur Netzwerkeinrichtung enthalten

ist. Weitere Informationen finden Sie unter the section called “Bereitstellen mit AWS CloudFormation”.

Sie können eine RDS Custom for Oracle-Instance mit einer Multi-AZ-Bereitstellung erstellen, indem Sie bei der Erstellung der Datenbank-Instance in der Amazon RDS-Konsole die Multi-AZ-Option wählen. Alternativ können Sie den `--multi-az` Parameter im Amazon `create-db-instance` RDS-Befehl in der angeben AWS CLI.

Voraussetzungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Oracle

> **Note**
>
> Wenn Sie Ihre RDS-Instanz Custom for Oracle vor dem 30. Juni 2025 erstellt haben, können Sie sie nicht von einer Single-AZ- zu einer Multi-AZ-Bereitstellung ändern. Ihre zugrunde liegende Custom Engine Version (CEV) wurde mit einem älteren, vom Service bereitgestellten AMI erstellt, das keine Unterstützung für Multi-AZ-Bereitstellungen bietet. Sie müssen Ihre Datenbank mithilfe eines CEV, das Sie nach dem 30. Juni 2025 mit einem vom Service bereitgestellten AMI erstellen, auf eine neue Instance migrieren.
> Siehe the section called “Migration für CEVs vor dem 30.06.2025”.

RDS Custom for Oracle erfordert bestimmte Voraussetzungen für die Umstellung von einer Single-AZ- auf eine Multi-AZ-Bereitstellung. Unvollständige Voraussetzungen führen zu einem Fehler beim Multi-AZ-Setup. Verwenden Sie entweder die manuelle Einrichtung oder die neueste AWS CloudFormation Vorlagendatei, die in den Anweisungen zur Netzwerkeinrichtung enthalten ist. Weitere Informationen finden Sie unter the section called “Schritt 3: Extrahieren Sie die CloudFormation Vorlagen für RDS Custom for Oracle”.

Um die Voraussetzungen manuell zu erfüllen, folgen Sie den Schritten unter the section called “Von Single-AZ zu Multi-AZ” und beachten Sie dabei Folgendes:

- Stellen Sie sicher, dass Ihre RDS Custom for Oracle-DB-Instance ein CEV verwenden muss, das nach dem 30. Juni 2025 mit dem neuesten vom Service bereitgestellten AMI erstellt wurde.
- Aktualisieren Sie die Regeln für eingehenden und ausgehenden Datenverkehr der Amazon-RDS-Sicherheitsgruppe, um Port 1120 zuzulassen.
- Erstellen Sie neue Amazon SQS VPC-Endpoints, die es der Amazon RDS RDS Custom for Oracle DB-Instance ermöglichen, mit Amazon SQS zu kommunizieren.

- Aktualisieren Sie die Amazon SQS SQS-Berechtigungen in der Instance-Profilrolle.

> **Important**
>
> Starten Sie Ihre primäre Multi-AZ-DB-Instance nicht manuell neu, indem Sie sich bei der Instance anmelden, ohne RDS Custom for Oracle Automation anzuhalten.

Migrationsschritte für DB-Instances, die vor dem 30. Juni 2025 CEVs erstellt wurden

Wenn Sie Ihre RDS Custom for Oracle DB-Instance vor dem 30. Juni 2025 erstellt haben, können Sie sie nicht direkt auf eine Multi-AZ-Bereitstellung umstellen, da das zugrunde liegende CEV mit einem älteren, vom Service bereitgestellten AMI erstellt wurde, das keine Multi-AZ-Unterstützung bietet. Sie müssen Ihre Datenbank mithilfe eines CEV, das Sie nach dem 30. Juni 2025 erstellt haben, auf eine neue Instance migrieren. Gehen Sie für die Migration wie folgt vor.

1. Erstellen Sie CEV aus dem Quellcode mit dem neuesten vom Dienst bereitgestellten AMI (nach dem 30. Juni 2025)
2. Starten Sie eine neue DB-Instance mit dem neuen CEV
3. Migrieren Sie Ihre Datenbank mit einer der folgenden Methoden von der vorhandenen DB-Instance, die die Multi-AZ-Bereitstellung nicht unterstützt, auf die neu erstellte Instance:
   - Physische Migration von Oracle-Datenbanken zu Amazon RDS Custom mithilfe von Data Guard
   - Physische Migration von Oracle-Datenbanken zu Amazon RDS Custom mithilfe von RMAN-Duplizierung

Änderung einer RDS Custom for Oracle Single-AZ-Bereitstellung zu einer Multi-AZ-Bereitstellung

Sie können eine bestehende Multi-AZ-kompatible RDS Custom for Oracle-Instance von einer Single-AZ-Bereitstellung zu einer Multi-AZ-Bereitstellung ändern. Wenn Sie die DB-Instance ändern, führt Amazon RDS mehrere Aktionen aus:

- Es wird ein Snapshot der primären DB-Instance aufgenommen.
- Erstellt neue Volumes für das Standby-Replikat aus dem Snapshot. Diese Volumes werden im Hintergrund initialisiert, und die maximale Volume-Leistung wird erreicht, nachdem die Daten vollständig initialisiert wurden.

- Aktiviert die synchrone Replikation auf Blockebene zwischen der primären und der Standby-DB-Instance.

> **Important**
>
> Wir empfehlen Ihnen, Ihre RDS Custom for Oracle-DB-Instance in Zeiten hoher Aktivität nicht von einer Single-AZ- zu einer Multi-AZ-Bereitstellung auf einer Produktions-DB-Instance zu ändern.

AWS verwendet einen Snapshot, um die Standby-Instance zu erstellen, um Ausfallzeiten bei der Umstellung von Single-AZ auf Multi-AZ zu vermeiden. Die Leistung kann jedoch während und nach der Konvertierung zu Multi-AZ beeinträchtigt werden. Diese Auswirkung kann bei Workloads erheblich sein, die empfindlich auf Schreiblatenz reagieren. Diese Funktion ermöglicht zwar die schnelle Wiederherstellung großer Volumes aus Snapshots, kann jedoch aufgrund der synchronen Replikation zu einer erhöhten Latenz von I/O Vorgängen führen. Diese Latenz kann sich auf die Leistung Ihrer Datenbank auswirken.

Konfiguration der Voraussetzungen für die Änderung einer Single-AZ- zu einer Multi-AZ-Bereitstellung mit CloudFormation

Gehen Sie wie folgt vorthe section called “Schritt 3: Extrahieren Sie die CloudFormation Vorlagen für RDS Custom for Oracle”, um Ihr VPC- und IAM-Profil erneut einzurichten, um den SQS-VPC-Endpunkt und die SQS-Berechtigung zum IAM-Profil hinzuzufügen.

Konfigurieren der Voraussetzungen für die manuelle Änderung einer Single-AZ- in eine Multi-AZ-Bereitstellung

Wenn Sie Voraussetzungen manuell konfigurieren möchten, führen Sie die folgenden Schritte aus.

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/
2. Wählen Sie Endpunkt aus. DieErstellen eines Endpunktswird angezeigt.
3. Wählen Sie unter Servicekategorie die Option AWS -Services aus.
4. Suchen Sie unter Services nach`SQS`.
5. Wählen Sie in VPC die VPC aus, in der Ihre RDS Custom for Oracle DB-Instance bereitgestellt wird.

6. Wählen Sie unter Subnetze die Subnetze aus, in denen Ihre RDS Custom for Oracle DB-Instance bereitgestellt wird.
7. Wählen Sie unter Sicherheitsgruppen die Sicherheitsgruppe aus, in der Ihre RDS Custom for Oracle DB-Instance bereitgestellt wird.
8. Wählen Sie für Policy die Option Benutzerdefiniert aus.
9. Ersetzen Sie in Ihrer benutzerdefinierten Richtlinie die Werte `AWS partitionRegion`,`accountId`, und `IAM-Instance-role` durch Ihre eigenen Werte.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            },
            "Action": [
                "SQS:SendMessage",
                "SQS:ReceiveMessage",
                "SQS:DeleteMessage",
                "SQS:GetQueueUrl"
            ],
            "Resource": "arn:aws:sqs:us-east-1:111122223333:do-not-delete-rds-
custom-*",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/{IAM-Instance-role}"
            }
        }
    ]
}
```

Aktualisieren Sie das Instance-Profil mit der Berechtigung, auf Amazon SQS zuzugreifen. Ersetzen Sie die `AWS partitionRegion`, und `accountId` durch Ihre eigenen Werte.

```
{
    "Sid": "13",
    "Effect": "Allow",
    "Action": [
        "SQS:SendMessage",
        "SQS:ReceiveMessage",
        "SQS:DeleteMessage",
        "SQS:GetQueueUrl"
    ],
    "Resource": [
        {
            "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
        }
    ],
    "Condition": {
        "StringLike": {
            "aws:ResourceTag/AWSRDSCustom": "custom-oracle"
        }
    }
}
```

Aktualisieren Sie die Regeln für eingehenden und ausgehenden Datenverkehr der Amazon-RDS-Sicherheitsgruppe, um Port 1120 zuzulassen.

- Wählen Sie unter Sicherheitsgruppen die Gruppe aus, in der Ihre RDS Custom for Oracle DB-Instance bereitgestellt wird.
- Erstellen Sie für Inbound Rules eine benutzerdefinierte TCP-Regel, um den Port `1120` aus der Quellgruppe zuzulassen.
- Erstellen Sie für ausgehende Regeln eine benutzerdefinierte TCP-Regel, um den Port zur `1120` Zielgruppe zuzulassen.

Ändern Sie mithilfe der RDS-Konsole oder der RDS-API AWS CLI

Nachdem Sie die Voraussetzungen erfüllt haben, können Sie eine RDS Custom for Oracle DB-Instance mithilfe der Amazon RDS-Konsole oder der Amazon RDS-API von einer Single-AZ- zu einer Multi-AZ-Bereitstellung ändern. AWS CLI

Konsole

Um eine bestehende RDS-Bereitstellung Custom for Oracle Single-AZ auf Multi-AZ zu ändern

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der Amazon RDS-Konsole Databases (Datenbanken) aus. Der Bereich Databases (Datenbanken) wird angezeigt.
3. Wählen Sie die DB-Instance RDS Custom for Oracle, die Sie ändern möchten.
4. Wählen Sie unter Aktionen die Option In Multi-AZ-Bereitstellung konvertieren aus.
5. Damit Änderungen sofort übernommen werden, wählen Sie die Option Sofort anwenden auf der Seite Bestätigung aus. Die Auswahl dieser Option verursacht keine Ausfallzeiten, kann jedoch zur Beeinträchtigung der Leistung führen. Sie können die Aktualisierung auch im nächsten Wartungsfenster übernehmen. Weitere Informationen finden Sie unter the section called “Änderungen planen”.
6. Wählen Sie auf der Seite Bestätigung die Option Konvertieren in Multi-AZ aus.

AWS CLI

Um mithilfe von in eine Multi-AZ-DB-Instance-Bereitstellung zu konvertieren AWS CLI, rufen Sie den modify-db-instanceBefehl auf und legen Sie die `--multi-az` Option fest. Geben Sie die DB-Instance-Kennung und die Werte für andere Optionen an, die geändert werden sollen. Informationen zu den jeweiligen Optionen finden Sie unter the section called “Verfügbare Einstellungen”.

Example Beispiel

Mit dem folgenden Code wird `mycustomdbinstance` geändert, indem die Option `--multi-az` hinzugefügt wird. Die Änderungen werden während des nächsten Wartungsfensters (mit) übernommen `--no-apply-immediately`. Verwenden Sie `--apply-immediately`, damit Änderungen sofort angewendet werden. Weitere Informationen finden Sie unter the section called “Änderungen planen”.

Für Linux, macOS oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --multi-az \
```

```
    [--no-apply-immediately | --apply-immediately]
```

Für Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --multi-az ^
    [--no-apply-immediately | --apply-immediately]
```

RDS-API

Um mit der Amazon RDS-API in eine Multi-AZ-DB-Instance-Bereitstellung zu konvertieren, rufen Sie den DBInstance Vorgang Modify auf und setzen Sie den `MultiAZ` Parameter auf true.

Änderung einer Multi-AZ-Bereitstellung von RDS Custom for Oracle zu einer Single-AZ-Bereitstellung

Sie können eine bestehende RDS-Custom for Oracle-DB-Instance von einer Multi-AZ- zu einer Single-AZ-Bereitstellung ändern.

Konsole

So ändern Sie eine RDS Custom for Oracle DB-Instance von einer Multi-AZ- zu einer Single-AZ-Bereitstellung

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der Amazon RDS-Konsole Databases (Datenbanken) aus. Der Bereich Databases (Datenbanken) wird angezeigt.
3. Wählen Sie die DB-Instance RDS Custom for Oracle, die Sie ändern möchten.
4. Wählen Sie für die Multi-AZ-Bereitstellung die Option Nein aus.
5. Damit Änderungen sofort übernommen werden, wählen Sie die Option Sofort anwenden auf der Seite Bestätigung aus. Die Auswahl dieser Option verursacht keine Ausfallzeiten, kann jedoch zur Beeinträchtigung der Leistung führen. Sie können die Aktualisierung auch im nächsten Wartungsfenster übernehmen. Weitere Informationen finden Sie unter the section called “Änderungen planen”.
6. Klicken Sie auf der Seite Bestätigung auf DB-Instance ändern.

AWS CLI

Um eine Multi-AZ-Bereitstellung mithilfe von in eine Single-AZ-Bereitstellung zu ändern AWS CLI, rufen Sie den modify-db-instanceBefehl auf und fügen Sie die Option hinzu. `--no-multi-az` Geben Sie die DB-Instance-Kennung und die Werte für andere Optionen an, die geändert werden sollen. Informationen zu den jeweiligen Optionen finden Sie unter the section called “Verfügbare Einstellungen”.

Example Beispiel

Mit dem folgenden Code wird `mycustomdbinstance` geändert, indem die Option `--no-multi-az` hinzugefügt wird. Die Änderungen werden während des nächsten Wartungsfensters (mit) übernommen `--no-apply-immediately`. Verwenden Sie `--apply-immediately`, damit Änderungen sofort angewendet werden. Weitere Informationen finden Sie unter the section called “Änderungen planen”.

Für Linux, macOS oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --no-multi-az \
    [--no-apply-immediately | --apply-immediately]
```

Für Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --no-multi-az ^
    [--no-apply-immediately | --apply-immediately]
```

RDS-API

Um eine Multi-AZ-Bereitstellung mithilfe der Amazon RDS-API in eine Single-AZ-Bereitstellung umzuwandeln, rufen Sie den DBInstance Vorgang Modify auf und setzen Sie den `MultiAZ` Parameter auf. `false`

Failover-Prozess für eine Multi-AZ-Bereitstellung von RDS Custom for Oracle

Wenn ein geplanter oder ungeplanter Ausfall Ihrer DB-Instance durch einen Infrastrukturdefekt resultiert, wechselt Amazon RDS automatisch zu einem Standby-Replikat in einer anderen Availability

Zone, wenn Sie Multi-AZ aktiviert haben. Die Dauer, bis der Failover-Prozess abgeschlossen ist, hängt von der Datenbankaktivität sowie von anderen Bedingungen zu dem Zeitpunkt ab, an dem die primäre DB-Instance ausgefallen ist. Der Failover-Prozess dauert normalerweise 60-120 Sekunden. Diese Failover-Dauer kann sich verlängern, wenn umfangreiche Transaktionen oder zeitintensive Wiederherstellungsprozesse durchgeführt werden. Wenn der Failover abgeschlossen ist, kann es länger dauern, bis die Amazon RDS-Konsole die neue Availability Zone anzeigt.

> **Note**
>
> Sie können ein Failover manuell erzwingen, wenn Sie Ihren primären EC2 Host beenden und starten, solange Ihre DB-Instance verfügbar ist.

Amazon RDS führt den Failover-Prozess automatisch durch, sodass der Datenbankbetrieb so schnell wie möglich und ohne Verwaltungseingriff wieder aufgenommen werden kann. Die primäre DB-Instance schaltet automatisch auf das Standby-Replikat um, wenn eine der in der folgenden Tabelle beschriebenen Bedingungen eintritt: Sie können diese Failover-Gründe im Amazon RDS-Ereignisprotokoll einsehen.

Failover-Grund	Beschreibung
Das Betriebssystem, das der RDS-Datenbank-Instance zugrunde liegt, wird offline gepatcht.	Ein Failover wurde während des Wartungsfensters für einen Betriebssystem-Patch oder ein Sicherheitsupdate ausgelöst. Weitere Informationen finden Sie unter the section called “Warten einer DB-Instance”.
Der primäre Host der RDS-Multi-AZ-Instance ist fehlerhaft.	Die Multi-AZ-DB-Instance-Bereitstellung hat eine beeinträchtigte primäre DB-Instance erkannt und ein Failover eingeleitet.
Der primäre Host der RDS-Multi-AZ-Instance ist aufgrund des Verlusts der Netzwerkverbindung nicht erreichbar.	Die RDS-Überwachung hat einen Fehler bei der Erreichbarkeit des Netzwerks für die primäre DB-Instance festgestellt und ein Failover ausgelöst.
Die RDS-Instance wurde vom Kunden geändert.	Eine Änderung der RDS-DB-Instance hat ein Failover ausgelöst. Weitere Informationen finden Sie unter the section called “Ändern Ihrer DB-Instance”.

Failover-Grund	Beschreibung
Das Speichervolumen, das dem primären Host der RDS-Multi-AZ-Instance zugrunde liegt, ist ausgefallen.	Die Multi-AZ-DB-Instance-Bereitstellung hat ein Speicherproblem der primären DB-Instance erkannt und ein Failover eingeleitet.
Die primäre RDS-Multi-AZ-Instance ist beschäftigt und reagiert nicht.	Die primäre DB-Instance reagiert nicht. Wir empfehlen Ihnen, wie folgt vorzugehen: Untersuchen Sie das Ereignis und die CloudWatch Protokolle auf übermäßige CPU-, Arbeitsspeicher- oder Auslagerungsspeicherauslastung. Weitere Informationen erhalten Sie unter the section called “Mit RDS Amazon-Event-Benachrichtigungen arbeiten” und the section called “Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird”. Prüfen Sie Ihren Workload, um festzustellen, ob Sie die angemessene DB-Instance-Klasse verwenden. Weitere Informationen finden Sie unter the section called “DB-Instance-Klassen”.

Um festzustellen, ob Ihre Multi-AZ-DB-Instance erfolgreich ausgeführt wurde, können Sie Folgendes tun:

- Sie können Benachrichtigungen per E-Mail oder per SMS für DB-Ereignisse abonnieren, bei denen ein Failover ausgelöst wird. Weitere Informationen über -Ereignisse finden Sie unter the section called “Mit RDS Amazon-Event-Benachrichtigungen arbeiten”.
- Sie können Ihre DB-Ereignisse über die Amazon-RDS-Konsole oder mittels API-Operationen anzeigen.
- Zeigen Sie den aktuellen Status Ihrer Multi-AZ-DB-Instance-Bereitstellung von RDS Custom for Oracle an, indem Sie die Amazon RDS-Konsole, CLI oder API-Operationen verwenden.

Time-to-Live-Einstellungen (TTL) bei Anwendungen, die eine Multi-AZ-Bereitstellung von RDS Custom for Oracle verwenden

Bei dem Failover-Prozess wird der DNS-Datensatz (Domain Name System) der DB-Instance so geändert, dass er auf die Standby-DB-Instance verweist. Als Ergebnis müssen alle bestehenden

Verbindungen zu Ihrer DB-Instance neu hergestellt werden. Stellen Sie sicher, dass der Wert für die DNS-Cache-Konfiguration time-to-live (TTL) niedrig ist, und stellen Sie sicher, dass Ihre Anwendung DNS nicht über einen längeren Zeitraum zwischenspeichert. Ein hoher TTL-Wert kann verhindern, dass Ihre Anwendung nach einem Failover schnell wieder eine Verbindung mit der DB-Instance herstellt.

Bereitstellung von RDS Custom für Oracle mit AWS CloudFormation

Automatisieren Sie Ihre Bereitstellung von RDS Custom for Oracle mithilfe der bereitgestellten AWS CloudFormation Vorlage. Erfüllen Sie die folgenden Voraussetzungen, bevor Sie die Ressourcen bereitstellen.

Voraussetzungen

Laden Sie die erforderlichen Oracle-Dateien herunter

Sie benötigen spezielle Oracle-Installationsdateien, bevor Sie die AWS CloudFormation Vorlage erstellen können. Laden Sie diese Dateien herunter, bevor Sie sie bereitstellen.

1. Navigieren Sie zu Oracle Database 19c (19.3)
2. Suchen Sie die Datei und laden Sie sie herunter `LINUX.X64_193000_db_home.zip`
3. Benennen Sie die Datei um in `V982063-01.zip`
4. Laden Sie die verbleibenden Patches herunter und wählen Sie Plattform oder Sprache als `Linux x86-64`

Neuestes OPatch Hilfsprogramm

Patch 6880880

PSU-Patches vom Januar 2023

PSU- und RU-Patches für Datenbanken

- Patch 34765931
- Aufnäher 34786990

Zusätzliche erforderliche Patches

- Patch 35099667

- Patch 35099674
- Aufnäher 28730253
- Aufnäher 29213893
- Patch 35012866

PSU-Patches für April 2023

PSU- und RU-Patches für Datenbanken

- Patch 35042068
- Aufnäher 35050341

Zusätzliche erforderliche Patches

- Patch 28730253
- Aufnäher 29213893
- Aufnäher 33125873
- Aufnäher 35220732
- Aufnäher 35239280

Einrichtung Amazon S3 S3-Buckets

1. Erstellen Sie einen Amazon S3 S3-Bucket in Ihrem AWS-Konto oder wählen Sie einen vorhandenen Bucket aus.
2. Erstellen Sie eine Ordnerstruktur im Bucket, die dem Beispiel unten ähnelt.

```
<bucket-name>/
### oracle_cev/
    ### V982063-01.zip
    ### p6880880_190000_Linux-x86-64.zip
    ### p34765931_190000_Linux-x86-64.zip
    ### p34786990_190000_Linux-x86-64.zip
    ### p35099667_190000_Linux-x86-64.zip
    ### p35099674_190000_Generic.zip
    ### p28730253_190000_Linux-x86-64.zip
    ### p29213893_1918000DBRU_Generic.zip
    ### p35012866_1918000DBRU_Linux-x86-64.zip
```

```
### p35042068_190000_Linux-x86-64.zip
### p35050341_190000_Linux-x86-64.zip
### p29213893_1919000DBRU_Generic.zip
### p33125873_1919000DBRU_Linux-x86-64.zip
### p35220732_190000_Linux-x86-64.zip
### p35239280_190000_Generic.zip
```

3. Laden Sie alle Oracle-Dateien, die Sie zuvor heruntergeladen haben, in die entsprechenden Ordner hoch.

Stellen Sie RDS Custom für Oracle bereit mit AWS CloudFormation

Schritt 1: Bereiten Sie die AWS CloudFormation Vorlage vor

Bevor Sie RDS Custom for Oracle bereitstellen können, müssen Sie die AWS CloudFormation Vorlage herunterladen und konfigurieren, die die erforderlichen Voraussetzungen schafft.

Kopieren und speichern Sie die Vorlage

1. Gehen Sie zu Deployment RDS Custom for Oracle mit einzelnen und mehreren Availability Zones
2. Kopieren Sie die Vorlage in Ihr bevorzugtes Format (YAML oder JSON)
3. Speichern Sie die Datei im YAML- oder JSON-Format. Beispiel: `rds-custom-oracle-prereqs.yaml`

Starten Sie den Stack in der Konsole AWS

1. Öffnen Sie die AWS Konsole und navigieren Sie zu AWS CloudFormation
2. Wählen Sie Stack erstellen > Mit neuen Ressourcen (Standard)
3. Wählen Sie „Bestehende Vorlage auswählen“
4. Wählen Sie „Eine Vorlagendatei hochladen“ > „Datei auswählen“
5. Wählen Sie die Vorlagendatei aus, die Sie zuvor heruntergeladen haben
6. Behalten Sie die Standardparameterwerte bei
7. Wählen Sie Weiter, um den Stapel zu erstellen

Alternative: AWS CLI verwenden

Anstatt die Konsole zu verwenden, können Sie den Stack mit dem folgenden Befehl erstellen AWS CLI:

```
aws cloudformation create-stack \
  --stack-name rds-custom-oracle \
  --template-body file://rds-custom-oracle-prereqs.yaml \
  --capabilities CAPABILITY_NAMED_IAM
```

Schritt 2: Custom Engine-Versionen (CEVs) und Amazon RDS-Instances erstellen

Kopieren und speichern Sie die Vorlage

1. Gehen Sie zu Deployment RDS Custom for Oracle mit einzelnen und mehreren Availability Zones
2. Kopieren Sie die Vorlage in Ihr bevorzugtes Format (YAML oder JSON)
3. Aktualisieren Sie bei Bedarf die folgenden Parameter in der Vorlage:
   - `BucketName`
   - `CEVS3Prefix`
   - Master-Passwort für die Datenbank (ersetzen Sie **********)
4. Speichern Sie die Datei im YAML- oder JSON-Format

Schritt 3: Bereitstellen mit der Konsole AWS

1. Öffnen Sie die AWS Konsole und navigieren Sie zu AWS CloudFormation
2. Wählen Sie Stack erstellen > Mit neuen Ressourcen (Standard)
3. Wählen Sie „Bestehende Vorlage auswählen“
4. Wählen Sie „Eine Vorlagendatei hochladen“ > „Datei auswählen“
5. Wählen Sie die Vorlagendatei aus, die Sie zuvor heruntergeladen haben
6. Belassen Sie die Parameter als Standardwerte
7. Füllen Sie die Parameter wie folgt aus:

```
BucketName: rds-custom-id
CEVS3Prefix: oracle_cev
CEVCreation: Yes
```

8. Überprüfen Sie die Konfiguration und wählen Sie Weiter, um den Stack zu erstellen

Optional: Bereitstellen mit AWS CLI

```
aws cloudformation create-stack \
```

```
  --stack-name rds-custom-oracle \
  --template-body file://rds-custom-oracle.yaml \
  --parameters \
    ParameterKey=BucketName,ParameterValue=rds-custom-id \
    ParameterKey=CEVS3Prefix,ParameterValue=oracle_cev \
    ParameterKey=CEVCreation,ParameterValue=Yes \
  --capabilities CAPABILITY_NAMED_IAM
```

Bereitstellungsressourcen wurden erstellt

Die -Vorlage erstellt die folgenden Ressourcen:

- Amazon VPC mit öffentlichen und privaten Subnetzen
- Sicherheitsgruppen
- Amazon VPC-Endpunkte
- IAM-Rollen und -Richtlinien
- AWS KMS Schlüssel für die Verschlüsselung
- Benutzerdefinierte Engine-Versionen (CEVs)
- RDS Custom für Oracle-Instances sowohl für Single-AZ- als auch Multi-AZ-Konfigurationen

Überwachen Sie Ihren Bereitstellungsfortschritt

Nachdem Sie den AWS CloudFormation Stack erstellt haben, sollten Sie dessen Fortschritt überwachen, um eine erfolgreiche Bereitstellung sicherzustellen. Der Bereitstellungsprozess umfasst die Erstellung von Custom Engine Versions (CEVs) und RDS-Instances.

So überwachen Sie den Bereitstellungsfortschritt:

1. Öffnen Sie die AWS CloudFormation Konsole.
2. Wählen Sie Ihren Stack-Namen.
3. Wählen Sie den Tab Ereignisse, um den Fortschritt zu sehen und etwaige Fehler zu identifizieren.

> **Note**
>
> Die CEV-Erstellung dauert in der Regel 2-3 Stunden. Nachdem die CEV-Erstellung erfolgreich abgeschlossen wurde, beginnt Amazon RDS automatisch mit der Erstellung der Amazon RDS-Instance.

Nach der Bereitstellung

Führen Sie nach Abschluss der Stack-Erstellung die folgenden Überprüfungs- und Konfigurationsschritte nach der Bereitstellung durch:

1. Navigieren Sie auf der Amazon RDS-Konsolenseite zu Custom Engine versions, um die CEV-Erstellung zu überprüfen.
2. Bestätigen Sie, dass Amazon RDS-Instances erstellt wurden und verfügbar sind
3. Testen Sie die Konnektivität zu den Amazon RDS-Instances
4. Richten Sie nach Bedarf Überwachungs- und Backup-Strategien ein

Bereinigen

Führen Sie den folgenden AWS CLI Befehl aus, um alle Ressourcen zu entfernen:

```
aws cloudformation delete-stack --stack-name rds-custom-oracle
```

Fehlerbehebung

Wenn bei der Bereitstellung Probleme auftreten, verwenden Sie die folgenden Lösungen, um häufig auftretende Probleme zu lösen.

Die CEV-Erstellung schlägt fehl

- Stellen Sie sicher, dass alle erforderlichen Patches auf Amazon S3 hochgeladen wurden
- Überprüfen Sie die IAM-Berechtigungen
- Stellen Sie sicher, dass die Patch-Versionen korrekt sind. Eine Liste der erforderlichen Patches finden Sie in. the section called “Voraussetzungen”

Die Erstellung der Amazon RDS-Instance schlägt fehl

- Überprüfen Sie die VPC/subnet Konfigurationen
- Überprüfen Sie die Sicherheitsgruppenregeln
- Bestätigen Sie, dass CEV verfügbar ist

Mit Oracle-Repliken für RDS Custom for Oracle arbeiten

Sie können Oracle-Repliken für RDS benutzerdefinierte Oracle-DB-Instances erstellen, auf denen Oracle Enterprise Edition ausgeführt wird. Sowohl Container-Datenbanken (CDBs) als auch Nicht-Datenbanken CDBs werden unterstützt. Standard Edition 2 unterstützt Oracle Data Guard nicht.

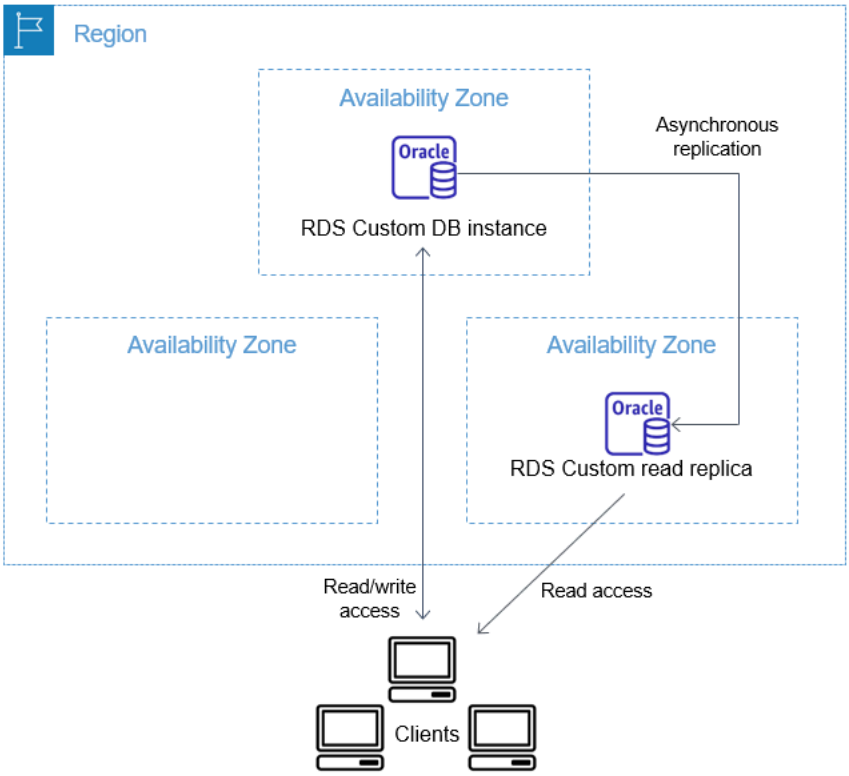
Das Erstellen eines RDS benutzerdefinierten Replikats für Oracle ähnelt dem Erstellen eines Replikats RDS für Oracle, weist jedoch wichtige Unterschiede auf. Allgemeine Informationen zum Erstellen und Verwalten von Oracle-Replikaten finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten und Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle.

Themen

- Überblick über die Replikation „RDSBenutzerdefiniert für Oracle“
- Richtlinien und Einschränkungen für Replikate von RDS Custom für Oracle
- Heraufstufen eines RDS Custom for Oracle-Replikats zu einer eigenständigen DB-Instance
- Konfiguration eines VPN Tunnels zwischen Primär- und Replikatinstanzen von RDS Custom for Oracle

Überblick über die Replikation „RDSBenutzerdefiniert für Oracle“

Die Architektur von RDS Custom for Oracle Replication entspricht der Architektur von Custom RDS for Oracle Replication. Eine primäre DB-Instance repliziert asynchron auf ein oder mehrere Oracle-Replikate.



Maximale Anzahl von Replikaten

Wie RDS bei Oracle können Sie bis zu fünf verwaltete Oracle-Repliken Ihrer primären RDS Custom for Oracle-DB-Instance erstellen. Sie können auch eigene manuell konfigurierte (externe) Oracle-Replikate erstellen. Externe Replikate werden nicht auf Ihr DB-Instance-Limit angerechnet. Sie liegen auch außerhalb des RDS Custom-Supportbereichs. Weitere Informationen über die Unterstützung finden Sie unter Support-Perimeter in RDS Custom.

Benennungskonvention für Replikate

Oracle-Replikatnamen basieren auf dem eindeutigen Datenbanknamen. Das Format ist *DB_UNIQUE_NAME_X*, mit Buchstaben, die nacheinander angehängt werden. Zum Beispiel, wenn der eindeutige Name Ihrer Datenbank `ORCL` ist, heißen die ersten beiden Repliken `ORCL_A` und `ORCL_B`. Die ersten sechs Buchstaben A—F sind für Benutzerdefiniert reserviert. RDS RDS Custom kopiert Datenbankparameter von Ihrer primären DB-Instance in die Replikate. Weitere Informationen finden Sie unter DB_ UNIQUE _ NAME in der Oracle-Dokumentation.

Backup-Aufbewahrung von Replikaten

Standardmäßig verwenden RDS benutzerdefinierte Oracle-Repliken den gleichen Aufbewahrungszeitraum für Backups wie Ihre primäre DB-Instance. Sie können den Aufbewahrungszeitraum für Backups auf 1—35 Tage ändern. RDS Custom unterstützt das Sichern,

Wiederherstellen und point-in-time Wiederherstellen (PITR). Weitere Informationen zum Sichern und Wiederherstellen von RDS benutzerdefinierten DB-Instances finden Sie unterSicherung und Wiederherstellung einer Amazon RDS Custom for Oracle DB-Instance.

> **Note**
>
> Beim Erstellen eines Oracle-Replikats unterbricht RDS Custom vorübergehend die Bereinigung von Redo-Log-Dateien. Auf diese Weise stellt RDS Custom sicher, dass es diese Logs auf das neue Oracle-Replikat anwenden kann, sobald es verfügbar ist.

Hochstufung von Replikaten

Sie können verwaltete Oracle-Replikate in RDS Custom for Oracle mithilfe der Konsole, des `promote-read-replica` AWS CLI Befehls oder heraufstufen. `PromoteReadReplica` API Wenn Sie Ihre primäre DB-Instance löschen und alle Replikate fehlerfrei sind, stuft RDS Custom for Oracle Ihre verwalteten Replikate automatisch zu eigenständigen Instances herauf. Wenn ein Replikat die Automatisierung angehalten hat oder sich außerhalb des Support-Bereichs befindet, müssen Sie das Replikat reparieren, bevor Custom es automatisch heraufstufen kann. RDS Sie können externe Oracle-Replikate nur manuell heraufstufen.

Richtlinien und Einschränkungen für Replikate von RDS Custom für Oracle

Wenn Sie RDS Custom für Oracle-Replikate, werden nicht alle RDS Oracle-Replikat-Optionen unterstützt.

Themen

- Richtlinien und Einschränkungen für die Replikation mit RDS Custom für Oracle
- Allgemeine Einschränkungen für die Replikation mit RDS Custom für Oracle
- Netzwerkanforderungen und -einschränkungen für die Replikation mit RDS Custom für Oracle
- Externe Replikatseinschränkungen für RDS Custom für Oracle

Richtlinien und Einschränkungen für die Replikation mit RDS Custom für Oracle

Beachten Sie bei der Verwendung von RDS Custom für Oracle folgende Richtlinien:

- Sie können RDS Custom für die Oracle-Replikation nur in der Oracle Enterprise Edition verwenden. Standard Edition 2 wird nicht unterstützt.

- Wir empfehlen dringend, einen VPN-Tunnel zu implementieren, um die Kommunikation zwischen Ihrer Primär- und Standby-Instanz zu verschlüsseln. Weitere Informationen finden Sie unter Konfiguration eines VPN Tunnels zwischen Primär- und Replikatinstanzen von RDS Custom for Oracle.
- Ändern Sie nicht den `RDS_DATAGUARD`-Benutzer. Dieser Benutzer ist für RDS Custom für Oracle Automation reserviert. Das Ändern des Benutzers kann zu unerwünschten Ergebnissen führen, z. B. keine Oracle-Replikate für Ihre RDS Custom für Oracle DB-Instance erstellen zu können.
- Ändern Sie das Benutzerpasswort für die Replikation nicht. Es ist erforderlich, um die Konfiguration von Oracle Data Guard auf dem Host zu verwalten. Wenn Sie das Kennwort ändern, kann RDS Custom für Oracle Ihr Oracle-Replikat außerhalb des Support-Umfangs platzieren. Weitere Informationen finden Sie unter Support-Perimeter in RDS Custom.

  Das Passwort wird gespeichert AWS Secrets Manager und mit der DB-Ressourcen-ID gekennzeichnet. Jedes Oracle-Replikat hat sein eigenes Geheimnis in Secrets Manager. Das Geheimnis verwendet eines der folgenden Benennungsformate.

  ```
  do-not-delete-rds-custom-db-DB_resource_id-uuid-dg
  rds-custom!oracle-do-not-delete-DB_resource_id-uuid-dg
  ```

- Ändern Sie `DB_UNIQUE_NAME` für die primäre DB-Instance nicht. Eine Änderung des Namens führt dazu, dass jeder Wiederherstellungsvorgang hängen bleibt.
- Geben Sie die Klausel `STANDBYS=NONE` nicht in einem `CREATE PLUGGABLE DATABASE`-Befehl in einer RDS-Custom-CDB an. Auf diese Weise enthält Ihre Standby-CDB bei einem Failover alle Daten. PDBs

Allgemeine Einschränkungen für die Replikation mit RDS Custom für Oracle

RDS Custom für Oracle-Replikate haben folgende Beschränkungen:

- Sie können Replikate von RDS Custom für Oracle nicht im schreibgeschützten Modus erstellen. Sie können den Modus der aufgespielten Replikate jedoch manuell in schreibgeschützt und von schreibgeschützt auf aufgespielt ändern. Weitere Informationen finden Sie in der Dokumentation zum Befehl create-db-instance-read AWS CLI -replica.
- Replikate von RDS Custom für Oracle können nicht regionsübergreifend erstellt werden.
- Sie können den Wert von Oracle Data Guard `CommunicationTimeout`-Parameter nicht ändern. Dieser Parameter ist für DB-Instances von RDS Custom für Oracle auf 15 Sekunden festgelegt.

Netzwerkanforderungen und -einschränkungen für die Replikation mit RDS Custom für Oracle

Stellen Sie sicher, dass Ihre Netzwerkkonfiguration RDS Custom für Oracle-Replikate unterstützt. Berücksichtigen Sie dabei Folgendes:

- Stellen Sie sicher, dass Sie Port 1140 für die eingehende und ausgehende Kommunikation innerhalb Ihrer Virtual Private Cloud (VPC) für die primäre DB-Instance und deren Replikate aktivieren. Dies ist für die Kommunikation von Oracle Data Guard mit den Lesereplikaten erforderlich.
- RDS Custom für Oracle validiert das Netzwerk beim Erstellen eines Oracle-Replikats. Wenn die primäre DB-Instance und das neue Replikat keine Verbindung über das Netzwerk herstellen können, erstellt RDS Custom für Oracle das Replikat nicht und platziert es in den `INCOMPATIBLE_NETWORK`-Zustand.
- Verwenden Sie für externe Oracle-Replikate, z. B. solche, die Sie auf Amazon EC2 oder vor Ort erstellen, einen anderen Port und Listener für die Oracle Data Guard-Replikation. Der Versuch, Port 1140 zu verwenden, kann zu Konflikten mit der benutzerdefinierten RDS Automatisierung führen.
- Die `/rdsdbdata/config/tnsnames.ora`-Datei enthält Netzwerkdienstnamen, die den Adressen des Listener-Protokolls zugeordnet sind. Beachten Sie die folgenden Anforderungen und Empfehlungen:
  - Einträge in `tnsnames.ora` mit dem Präfix `rds_custom_` sind für RDS Custom reserviert, wenn Oracle-Replikat-Operationen verarbeitet werden.

    Beim Erstellen von manuellen Einträgen in `tnsnames.ora`, benutzen Sie dieses Präfix nicht.
  - In einigen Fällen möchten Sie möglicherweise manuell umschalten oder ausfallen oder Failover-Technologien wie Fast-Start Failover (FSFO) verwenden. Wenn ja, stellen Sie sicher, dass Sie manuell synchronisieren `tnsnames.ora`-Einträge von der primären DB-Instance zu allen Standby-Instances. Diese Empfehlung gilt sowohl für Oracle-Replikate, die von RDS Custom verwaltet werden, als auch für externe Oracle-Replikate.

    RDS Customer Automatisierungsaktualisierungen `tnsnames.ora`-Einträge nur auf der primären DB-Instance. Stellen Sie sicher, dass Sie auch synchronisieren, wenn Sie ein Oracle-Replikat hinzufügen oder entfernen.

    Wenn Sie die `tnsnames.ora`-Dateien nicht synchronisieren und manuell umschalten oder ausfallen lassen, ist Oracle Data Guard auf der primären DB-Instance möglicherweise nicht in der Lage, mit den Oracle-Replikaten zu kommunizieren.

Externe Replikatseinschränkungen für RDS Custom für Oracle

RDS Custom für externe Oracle-Replikate, die lokale Replikate enthalten, gelten folgende Einschränkungen:

- RDS Custom für Oracle erkennt keine Instance-Rollenänderungen bei manuellem Failover, wie z. B. FSFO, für externe Oracle-Replikate.

  RDS Custom für Oracle erkennt aber Änderungen für verwaltete Replikate. Die Rollenänderung wird im Ereignisprotokoll vermerkt. Sie können den neuen Status auch mithilfe des Befehls anzeigen. describe-db-instances AWS CLI
- RDS Custom für Oracle erkennt keine hohe Replikationsverzögerung für externe Oracle-Replikate.

  RDS Custom für Oracle erkennt aber Verzögerungen für verwaltete Replikate. Eine hohe Replikationsverzögerung führt zum Event `Replication has stopped`. Sie können den Replikationsstatus auch mithilfe des describe-db-instances AWS CLI Befehls anzeigen, aber es kann zu Verzögerungen bei der Aktualisierung kommen.
- RDS Custom für Oracle stuft externe Oracle-Replikate nicht automatisch hoch, wenn Sie Ihre primäre DB-Instance löschen.

  Die automatische Heraufstufungsfunktion ist nur für verwaltete Oracle-Replikate verfügbar. Informationen zur manuellen Hochstufung von Oracle-Replikaten finden Sie im Whitepaper Hochverfügbarkeit mit Data Guard auf Amazon RDS Custom für Oracle aktivieren.

Heraufstufen eines RDS Custom for Oracle-Replikats zu einer eigenständigen DB-Instance

Genau wie bei Oracle können Sie ein RDS Custom RDS for Oracle-Replikat zu einer eigenständigen DB-Instance heraufstufen. Wenn Sie ein Oracle-Replikat hochstufen, startet RDS Custom for Oracle die DB-Instance neu, bevor sie verfügbar wird. Weitere Informationen über das Hochstufen von Oracle-Replikaten finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Beim Hochstufen eines Replikats gelten die folgenden Richtlinien:

- Initiieren Sie kein Failover, während RDS Custom for Oracle Ihr Replikat hochstuft. Andernfalls könnte der Promotion-Workflow hängen bleiben.
- Wechseln Sie nicht zu Ihrer primären DB-Instance, während RDS Custom for Oracle Ihr Oracle-Replikat bewirbt. Andernfalls könnte der Promotion-Workflow hängen bleiben.

- Fahren Sie Ihre primäre DB-Instance nicht herunter, während RDS Custom for Oracle Ihr Oracle-Replikat bewirbt. Andernfalls könnte der Promotion-Workflow hängen bleiben.
- Versuchen Sie nicht, die Replikation mit Ihrer neu hochgestuften DB-Instance als Ziel neu zu starten. Nachdem RDS Custom for Oracle Ihr Oracle-Replikat hochgestuft hat, wird es zu einer eigenständigen DB-Instance und hat nicht mehr die Replikatrolle.

Beachten Sie die folgenden Einschränkungen bei der Promotion „RDSBenutzerdefiniert für Oracle-Replikate“:

- Sie können ein Replikat nicht hochstufen, solange RDS Custom for Oracle es sichert.
- Sie können den Aufbewahrungszeitraum für Backups nicht auf `0` ändern, wenn Sie Ihr Oracle-Replikat hochstufen.
- Sie können Ihr Replikat nicht hochstufen, wenn es sich nicht in einem fehlerfreien Zustand befindet.

  Wenn Sie ein Problem `delete-db-instance` auf der primären DB-Instance ausgeben, überprüft RDS Custom for Oracle, ob jedes verwaltete Oracle-Replikat fehlerfrei ist und für eine Heraufstufung verfügbar ist. Ein Replikat ist möglicherweise nicht für eine Hochstufung berechtigt, da die Automatisierung angehalten wurde oder sich außerhalb des Supportimeters befindet. In solchen Fällen veröffentlicht RDS Custom for Oracle eine Veranstaltung, in der das Problem erklärt wird, sodass Sie Ihr Oracle-Replikat manuell reparieren können.

Die folgenden Schritte zeigen den allgemeinen Vorgang für das Hochstufen eines Oracle-Replikats zu einer DB-Instance:

1. Halten Sie das Schreiben aller Schreibtransaktionen an, damit die primäre DB-Instance.
2. Warten Sie, bis RDS Custom for Oracle alle Updates auf Ihr Oracle-Replikat angewendet hat.
3. Bewerben Sie Ihr Oracle-Replikat, indem Sie in der RDS Amazon-Konsole die Option Promote, den AWS CLI Befehl `promote-read-replica`oder den `PromoteReadReplica`RDSAPIAmazon-Vorgang auswählen.

Das Hochstufen einer Oracle-Replica kann einige Minuten in Anspruch nehmen. Während des Vorgangs stoppt RDS Custom for Oracle die Replikation und startet Ihr Replikat neu. Wenn der Neustart abgeschlossen ist, ist das Oracle-Replikat als eigenständige DB-Instanz verfügbar. Hinweise zur Problembehandlung bei der Replikaterhöhung finden Sie unter. Behebung von Fehlerbehebung bei der Heraufstufung von Replikaten für RDS

Konsole

So stufen Sie ein RDS Custom for Oracle-Replikat zu einer eigenständigen DB-Instance herauf

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.

   Der Bereich Databases (Datenbanken) wird angezeigt. Jedes Oracle-Replikat zeigt Replica (Replikat) in der Spalte Role (Rolle) an.
3. Wählen Sie das Replikat „RDSBenutzerdefiniert für Oracle“ aus, das Sie bewerben möchten.
4. Wählen Sie für Actions (Aktionen)Promote (Hochstufen) aus.
5. Geben Sie auf der Seite Oracle-Replikat hochstufen den Aufbewahrungszeitraum und das Backup-Fenster für die neu hochgestufte DB-Instance an. Sie können diesen Wert nicht auf 0 setzen.
6. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Promote Oracle replica (Oracle-Replikat hochstufen).

AWS CLI

Verwenden Sie den Befehl, um Ihr RDS Custom for Oracle-Replikat zu einer eigenständigen DB-Instance hochzustufen. AWS CLI `promote-read-replica`

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds promote-read-replica \
--db-instance-identifier my-custom-read-replica \
--backup-retention-period 2 \
--preferred-backup-window 23:00-24:00
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds promote-read-replica ^
--db-instance-identifier my-custom-read-replica ^
--backup-retention-period 2 ^
--preferred-backup-window 23:00-24:00
```

RDS API

Um Ihr RDS Custom for Oracle-Replikat zu einer eigenständigen DB-Instance hochzustufen, rufen Sie den RDS API `PromoteReadReplica`Amazon-Vorgang mit dem erforderlichen Parameter `DBInstanceIdentifier` auf.

Konfiguration eines VPN Tunnels zwischen Primär- und Replikatinstanzen von RDS Custom for Oracle

Ein VPN Tunnel ist eine verschlüsselte Verbindung zwischen zwei oder mehr Geräten über ein Netzwerk. Um ein Höchstmaß an Sicherheit für Ihre Oracle Data Guard-Instances in RDS Custom for Oracle zu gewährleisten, empfehlen wir Ihnen dringend, einen VPN Tunnel zur Verschlüsselung der Kommunikation zwischen Ihren Primär- und Standby-Instances zu implementieren. Der Tunnel dient als Schutz für sensible Daten, die das Netzwerk zwischen den Instanzen durchqueren. Diese Konfiguration ist zwar optional, wir empfehlen sie jedoch als bewährte Methode, um Datensicherheit und die Einhaltung gesetzlicher Vorschriften zu gewährleisten.

Stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Sie haben Root-Zugriff auf die Primär- und Standby-Hosts.
- Sie verfügen über das technische Fachwissen, um den `ipsec` Befehl auszuführen.

Um in RDS Custom for VPN Oracle einen Tunnel zwischen einem Primär- und einem Replikat zu konfigurieren

1. Fügen Sie die Sicherheitsgruppen sowohl für die primäre Instance als auch für die Standby-Instance mithilfe der folgenden Regeln zur Zulassungsliste hinzu:

```
ACTION FLOW SOURCE PROTO PORT

ALLOW ingress this-SG 50 (ESP) all (N/A)
ALLOW egress this-SG 50 (ESP) all (N/A)

ALLOW ingress this-SG 17 (UDP) 500 (IKE)
ALLOW egress this-SG 17 (UDP) 500 (IKE)
```

2. Wechseln Sie zum Root-Benutzer.

```
$ sudo su - root
```

3. Führen Sie die folgenden Befehle sowohl auf der primären Instanz als auch auf der Standby-Instanz aus, um die Network Security Services (NSS) -Datenbank unter dem Benutzer `root` zu initialisieren.

```
ipsec initnss --nssdir /etc/ipsec.d
```

4. Generieren Sie RSA Schlüssel wie folgt:

   a. Generieren Sie die Schlüssel auf der primären Instanz je nach Betriebssystemversion mit einem der folgenden `ipsec` Befehle.

```
ipsec newhostkey --nssdir /etc/ipsec.d       ## for Oracle Linux Version 8
ipsec newhostkey --output /etc/ipsec.secrets ## for Oracle Linux version 7.9
```

   b. Besorgen Sie sich den öffentlichen Schlüssel, den Sie zum Erstellen der Konfiguration benötigen. Im folgenden Beispiel `left` bezieht sich die primäre Instanz `left` im `ipsec` Sprachgebrauch auf das Gerät, das Sie gerade konfigurieren, und `right` auf das Gerät am anderen Ende des Tunnels.

```
ipsec showhostkey --left --ckaid ckaid-returned-in-last-statement
```

   c. Generieren Sie auf der Standby-Instanz Schlüssel für die Standby-Instanz.

```
ipsec newhostkey --nssdir /etc/ipsec.d       ## for Oracle Linux Version 8
ipsec newhostkey --output /etc/ipsec.secrets ## for Oracle Linux version 7.9
```

   d. Besorgen Sie sich den öffentlichen Schlüssel für die Standby-Instanz, den Sie zum Erstellen der Konfiguration benötigen. Im folgenden Beispiel liegt die Standby-Instanz `right` daran, dass sie sich auf das Gerät am anderen Ende des Tunnels bezieht.

```
ipsec showhostkey --right --ckaid ckaid-returned-in-last-statement
```

5. Generieren Sie die Konfiguration auf der Grundlage der RSA Schlüssel, die Sie erhalten haben. Die Konfiguration ist sowohl für die primäre Instance als auch für die Standby-Instance identisch. Sie finden die Adresse der primären Instanz und die IPv4 Adresse der Standby-Instanz IPv4 in der AWS Konsole.

   Speichern Sie sowohl auf der primären Instance als auch auf der Standby-Instance die folgende Konfiguration in der Datei`/etc/ipsec.d/custom-fb-tunnel.conf`.

```
conn custom-db-tunnel
 type=transport
 auto=add
 authby=rsasig
 left=IPV4-for-primary
 leftrsasigkey=RSA-key-generated-on-primary
 right=IPV4-for-standby
 rightrsasigkey=RSA-key-generated-on-standby
```

6. Starten Sie den `ipsec` Daemon sowohl auf der primären Instance als auch auf der Standby-Instance auf beiden Hosts.

```
ipsec setup start
```

7. Starten Sie den Tunnel entweder auf der primären Instance oder der Standby-Instance. Die Ausgabe sollte in etwa folgendermaßen aussehen:

```
[root@ip-172-31-6-81 ~]# ipsec auto --up custom-db-tunnel
181 "custom-db-tunnel" #1: initiating IKEv2 connection
181 "custom-db-tunnel" #1: sent IKE_SA_INIT request to 172.31.32.196:500
182 "custom-db-tunnel" #1: sent IKE_AUTH request {cipher=AES_GCM_16_256 integ=n/a
 prf=HMAC_SHA2_512 group=DH19}
003 "custom-db-tunnel" #1: initiator established IKE SA; authenticated peer '3584-
bit PKCS#1 1.5 RSA with SHA1' signature using preloaded certificate '172.31.32.196'
004 "custom-db-tunnel" #2: initiator established Child SA using #1; IPsec transport
 [172.31.6.81-172.31.6.81:0-65535 0] -> [172.31.32.196-172.31.32.196:0-65535 0]
 {ESP/ESN=>0xda9c4815 <0xb742ca42 xfrm=AES_GCM_16_256-NONE DPD=passive}
[root@ip-172-31-6-81 ~]#
```

Sicherung und Wiederherstellung einer Amazon RDS Custom for Oracle DB-Instance

Wie Amazon RDS erstellt und speichert RDS Custom automatische Backups Ihrer RDS Custom for Oracle-DB-Instance während des Backup-Fensters Ihrer DB-Instance. Sie können Ihre DB-Instance auch sichern, indem Sie manuell einen DB-Snapshot erstellen.

Das Verfahren ist identisch mit dem Erstellen eines Snapshots einer Amazon RDS DB-Instance. Der erste Snapshot einer RDS benutzerdefinierten DB-Instance enthält die Daten für die vollständige DB-Instance. Nachfolgende Snapshots sind inkrementell.

Stellen Sie DB-Snapshots mit entweder dem AWS Management Console oder dem AWS CLI wieder her.

Themen

- Einen RDS benutzerdefinierten Snapshot für Oracle erstellen
- Wiederherstellung aus einem RDS Custom for Oracle DB-Snapshot
- Wiederherstellung einer RDS Custom for Oracle-Instanz auf einen bestimmten Zeitpunkt
- Löschen eines RDS Custom for Oracle-Snapshots
- Automatisierte Backups von RDS Custom for Oracle löschen

Einen RDS benutzerdefinierten Snapshot für Oracle erstellen

RDSCustom for Oracle erstellt einen Speicher-Volume-Snapshot Ihrer DB-Instance und sichert dabei die gesamte DB-Instance und nicht nur einzelne Datenbanken. Wenn Ihre DB-Instance eine Container-Datenbank (CDB) enthält, umfasst der Snapshot der Instance den Stamm CDB und allesPDBs.

Wenn Sie einen RDS Custom for Oracle-Snapshot erstellen, geben Sie an, welche RDS benutzerdefinierte DB-Instance gesichert werden soll. Geben Sie dann dem DB-Snapshot einen Namen, sodass über diesen eine Wiederherstellung zu einem späteren Zeitpunkt möglich ist.

Wenn Sie einen Snapshot erstellen, erstellt RDS Custom for Oracle einen EBS Amazon-Snapshot für jedes Volume, das an die DB-Instance angehängt ist. RDSCustom for Oracle verwendet den EBS Snapshot des Root-Volumes, um ein neues Amazon Machine Image (AMI) zu registrieren. Damit Snapshots einfach mit einer bestimmten DB-Instance verknüpft werden können, werden sie mit `DBSnapshotIdentifier`, `DbiResourceId` und `VolumeType` getaggt.

Das Erstellen eines DB-Snapshots führt zu einer kurzen I/O-Suspendierung. Diese Suspendierung kann je nach Größe und Klasse Ihrer DB-Instance einige Sekunden bis einige Minuten dauern. Die Erstellungszeit für Snapshots variiert je nach Größe Ihrer Datenbank. Da der Snapshot das gesamte Speichervolume umfasst, wirkt sich die Größe von Dateien, wie z. B. temporäre Dateien, auch auf die Zeit aus, die zum Erstellen des Snapshots benötigt wird. Weitere Informationen zum Erstellen von Snapshots finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Erstellen Sie mit der Konsole oder dem einen RDS benutzerdefinierten Snapshot für Oracle AWS CLI.

Konsole

Um einen RDS benutzerdefinierten Snapshot zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie in der Liste der RDS benutzerdefinierten DB-Instances die Instance aus, für die Sie einen Snapshot erstellen möchten.
4. Wählen Sie für Aktionen die Option Take snapshot (Snapshot aufnehmen).

   Das Fenster Take DB Snapshot (DB-Snapshot erstellen) wird angezeigt.
5. Geben Sie den Namen des Snapshots in das Feld Snapshot name (Snapshot-Name) ein.
6. Wählen Sie Take Snapshot (Snapshot erstellen) aus.

AWS CLI

Sie erstellen einen Snapshot einer RDS benutzerdefinierten DB-Instance mithilfe des create-db-snapshot AWS CLI Befehls.

Folgende Optionen stehen Ihnen zur Verfügung:

- `--db-instance-identifier`— Identifiziert, welche RDS benutzerdefinierte DB-Instance Sie sichern werden
- `--db-snapshot-identifier`— Benennt Ihren RDS benutzerdefinierten Snapshot, damit Sie ihn später wiederherstellen können

In diesem Beispiel erstellen Sie einen DB-Snapshot mit dem Namen *my-custom-snapshot* für eine RDS benutzerdefinierte DB-Instance namens*my-custom-instance*.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Wiederherstellung aus einem RDS Custom for Oracle DB-Snapshot

Wenn Sie eine RDS benutzerdefinierte für Oracle-DB-Instance wiederherstellen, geben Sie den Namen des DB-Snapshots und einen Namen für die neue Instance an. Sie können keine Wiederherstellung von einem Snapshot auf eine bestehende RDS benutzerdefinierte DB-Instance durchführen. Bei der Wiederherstellung wird eine neue RDS Custom for Oracle-DB-Instance erstellt.

Der Wiederherstellungsprozess unterscheidet sich in folgenden Punkten von der Wiederherstellung in AmazonRDS:

- Vor der Wiederherstellung eines Snapshots erstellt RDS Custom for Oracle eine Sicherungskopie vorhandener Konfigurationsdateien. Diese Dateien sind auf der wiederhergestellten Instanz im Verzeichnis `/rdsdbdata/config/backup` verfügbar. RDSCustom for Oracle stellt den DB-Snapshot mit Standardparametern wieder her und überschreibt die vorherigen Datenbankkonfigurationsdateien mit vorhandenen. Daher behält die wiederhergestellte Instanz keine benutzerdefinierten Parameter und Änderungen an Datenbankkonfigurationsdateien bei.
- Die wiederhergestellte Datenbank hat denselben Namen wie im Snapshot. Wenn Sie möchten, können Sie einen anderen Namen eingeben. (Für RDS Benutzerdefiniert für Oracle ist die Standardeinstellung`ORCL`.)

Konsole

Um eine RDS benutzerdefinierte DB-Instance aus einem DB-Snapshot wiederherzustellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot für die Wiederherstellung aus.
4. Wählen Sie in Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen) aus.
5. Geben Sie auf der Seite DB-Instance wiederherstellen als DB-Instance-ID den Namen für Ihre wiederhergestellte RDS benutzerdefinierte DB-Instance ein.
6. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

AWS CLI

Sie stellen einen RDS benutzerdefinierten DB-Snapshot mithilfe des Befehls restore-db-instance-from AWS CLI -db-snapshot wieder her.

Wenn der Snapshot, von dem Sie wiederherstellen, für eine private DB-Instance bestimmt ist, geben Sie beide die richtigen `db-subnet-group-name` und `no-publicly-accessible` an. Andernfalls ist die DB-Instance standardmäßig öffentlich zugänglich. Die folgenden Optionen sind erforderlich:

- `db-snapshot-identifier` — Identifiziert den Snapshot, aus dem wiederhergestellt werden soll
- `db-instance-identifier`— Gibt den Namen der RDS benutzerdefinierten DB-Instance an, die aus dem DB-Snapshot erstellt werden soll
- `custom-iam-instance-profile`— Gibt das Instance-Profil an, das der zugrunde liegenden EC2 Amazon-Instance einer RDS Custom DB-Instance zugeordnet ist.

Der folgende Code stellt den Snapshot mit dem Namen `my-custom-snapshot` zu `my-custom-instance` her.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
```

```
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Wiederherstellung einer RDS Custom for Oracle-Instanz auf einen bestimmten Zeitpunkt

Sie können eine DB-Instance zu einem bestimmten Zeitpunkt (PITR) wiederherstellen und so eine neue DB-Instance erstellen. Zur Unterstützung PITR muss für Ihre DB-Instances die Backup-Aufbewahrung auf einen Wert ungleich Null gesetzt sein.

Die letzte wiederherstellbare Zeit für eine RDS Custom for Oracle-DB-Instance hängt von mehreren Faktoren ab, liegt aber in der Regel innerhalb von 5 Minuten nach der aktuellen Uhrzeit. Um die letzte wiederherstellbare Zeit für eine DB-Instance zu ermitteln, verwenden Sie den AWS CLI describe-db-instancesBefehl und sehen Sie sich den Wert an, der im `LatestRestorableTime` Feld für die DB-Instance zurückgegeben wurde. Um den letzten wiederherstellbaren Zeitpunkt für jede DB-Instance in der RDS Amazon-Konsole zu sehen, wählen Sie Automatisierte Backups.

Sie können die Backup auf jeden beliebigen Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups vornehmen. Um den frühesten wiederherstellbaren Zeitpunkt für jede DB-Instance zu sehen, wählen Sie in der RDS Amazon-Konsole Automatisierte Backups aus.

Allgemeine Informationen zu PITR finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Themen

- PITRÜberlegungen zu RDS Custom for Oracle

PITRÜberlegungen zu RDS Custom for Oracle

In RDS Custom for Oracle PITR unterscheidet sich in folgenden wichtigen Punkten von PITR AmazonRDS:

- Die wiederhergestellte Datenbank hat denselben Namen wie in der Quell-DB-Instance. Wenn Sie möchten, können Sie einen anderen Namen eingeben. Der Standardwert ist `ORCL`.
- `AWSRDSCustomIamRolePolicy` benötigt neue Berechtigungen. Weitere Informationen finden Sie unter Schritt 2: Fügen Sie eine Zugriffsrichtlinie hinzu AWSRDSCustom InstanceRoleForRdsCustomInstance.
- Für alle RDS Custom for Oracle-DB-Instances muss die Backup-Aufbewahrung auf einen Wert ungleich Null gesetzt sein.
- Wenn Sie die Zeitzone des Betriebssystems oder der DB-Instance ändern, funktioniert das PITR möglicherweise nicht. Weitere Informationen zum Ändern von Zeitzonen finden Sie unter Oracle-Zeitzone.
- Wenn Sie die Automatisierung auf einstellen`ALL_PAUSED`, pausiert RDS Custom den Upload archivierter Redo-Log-Dateien, einschließlich Logs, die vor dem letzten wiederherstellbaren Zeitpunkt () erstellt wurden. LRT Es wird empfohlen, die Automatisierung für einen kurzen Zeitraum anzuhalten.

  Gehen Sie zur Veranschaulichung davon aus, dass Sie vor 10 LRT Minuten waren. - Pausieren einer Automatisierung Während der Pause lädt RDS Custom keine archivierten Redo-Logs hoch. Wenn Ihre DB-Instance abstürzt, können Sie nur zu einer Zeit LRT vor der Pause zurückkehren. Wenn Sie die Automatisierung fortsetzen, setzt RDS Custom das Hochladen von Protokollen fort. Die LRT Fortschritte. Es gelten PITR die normalen Regeln.
- In RDS Benutzerdefiniert können Sie manuell eine beliebige Anzahl von Stunden angeben, um archivierte Redo-Logs aufzubewahren, bevor RDS Custom sie nach dem Upload löscht. Geben Sie die Anzahl der Stunden wie folgt an:

  1. Erstellen Sie eine Textdatei mit dem Namen `/opt/aws/rdscustomagent/config/redo_logs_custom_configuration.json`.
  2. Fügen Sie ein JSON Objekt im folgenden Format hinzu:. `{"archivedLogRetentionHours" : "num_of_hours"}` Die Zahl muss eine Ganzzahl im Bereich von 1–840 sein.
- Gehen Sie davon aus, dass Sie ein CDB Nicht-Objekt in eine Container-Datenbank (CDB) einfügen PDB und es dann versuchenPITR. Der Vorgang ist nur erfolgreich, wenn Sie zuvor eine Sicherungskopie von erstellt haben. PDB Nachdem Sie eine erstellt oder geändert habenPDB, empfehlen wir, sie immer zu sichern.
- Es wird empfohlen, die -Initialisierungsparameter für die Datenbank nicht anzupassen. Das Ändern der folgenden Parameter wirkt sich beispielsweise auf Folgendes ausPITR:

  - `CONTROL_FILE_RECORD_KEEP_TIME` wirkt sich auf die Regeln zum Hochladen und Löschen von Protokollen aus.
  - `LOG_ARCHIVE_DEST_`*n* unterstützt mehrere Destinationen nicht.
  - `ARCHIVE_LAG_TARGET`wirkt sich auf den letzten wiederherstellbaren Zeitpunkt aus. `ARCHIVE_LAG_TARGET`ist auf eingestellt, `300` weil das Wiederherstellungspunktziel (RPO) 5 Minuten beträgt. Um dieses Ziel zu erreichen, RDS wechselt das Online-Redo-Log alle 5 Minuten und speichert es in einem Amazon S3 S3-Bucket. Wenn die Häufigkeit des Protokollwechsels ein Leistungsproblem für Ihre RDS Custom for Oracle-Datenbank verursacht, können Sie Ihre DB-Instance und Ihren Speicher auf eine Instanz mit höherem IOPS Durchsatz skalieren. Falls für Ihren Wiederherstellungsplan erforderlich, können Sie die Einstellung des `ARCHIVE_LAG_TARGET` Initialisierungsparameters auf einen Wert zwischen 60 und 7200 anpassen.
- Wenn Sie die Datenbankinitialisierungsparameter anpassen, empfehlen wir dringend, dass Sie nur die folgenden Einstellungen anpassen:
  - `COMPATIBLE`
  - `MAX_STRING_SIZE`
  - `DB_FILES`
  - `UNDO_TABLESPACE`
  - `ENABLE_PLUGGABLE_DATABASE`
  - `CONTROL_FILES`
  - `AUDIT_TRAIL`
  - `AUDIT_TRAIL_DEST`

  Für alle anderen Initialisierungsparameter stellt RDS Custom die Standardwerte wieder her. Wenn Sie einen Parameter ändern, der nicht in der vorherigen Liste enthalten ist, kann dies negative Auswirkungen auf die Ergebnisse haben PITR und zu unvorhersehbaren Ergebnissen führen. Beispiel, `CONTROL_FILE_RECORD_KEEP_TIME` wirkt sich auf die Regeln zum Hochladen und Löschen von Protokollen aus.

Sie können eine RDS benutzerdefinierte DB-Instance mit dem AWS Management Console, dem oder dem auf einen AWS CLI bestimmten Zeitpunkt zurücksetzen RDSAPI.

Konsole

So stellen Sie eine RDS benutzerdefinierte DB-Instance zu einem bestimmten Zeitpunkt wieder her

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
3. Wählen Sie die RDS benutzerdefinierte DB-Instance aus, die Sie wiederherstellen möchten.
4. Wählen Sie unter Aktionen die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

   Anschließend wird das Fenster Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) angezeigt.
5. Wählen Sie Späteste Wiederherstellungszeit, um auf den spätesten möglichen Zeitpunkt wiederherzustellen oder wählen Sie Benutzerdefiniert, um eine Zeit auszuwählen.

   Geben Sie bei der Auswahl von Custom das Datum und die Uhrzeit ein, zu der Sie den Instance-Cluster wiederherstellen möchten.

   Die Zeiten werden in Ihrer lokalen Zeitzone angezeigt, was durch einen Offset von der koordinierten Weltzeit (UTC) gekennzeichnet ist. Zum Beispiel ist UTC -5 Eastern Standard Time/Central Daylight Time.
6. Geben Sie als DB-Instance-ID den Namen der wiederhergestellten RDS benutzerdefinierten DB-Ziel-Instance ein. Der Name muss eindeutig sein.
7. Wählen Sie bei Bedarf andere Optionen aus, z. B. DB-Instance-Class.
8. Wählen Sie Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

AWS CLI

Sie stellen eine DB-Instance zu einem bestimmten Zeitpunkt wieder her, indem Sie den point-in-time AWS CLI Befehl restore-db-instance-to- verwenden, um eine neue RDS benutzerdefinierte DB-Instance zu erstellen.

Verwenden Sie eine der folgenden Optionen, um die Sicherung anzugeben, von der wiederhergestellt werden soll:

- `--source-db-instance-identifier` *`mysourcedbinstance`*

- `--source-dbi-resource-id` *dbinstanceresourceID*
- `--source-db-instance-automated-backups-arn` *backupARN*

Die Option `custom-iam-instance-profile` ist erforderlich.

Der folgende Befehl stellt `my-custom-db-instance` auf eine neue DB-Instance namens `my-restored-custom-db-instance` wieder her, und zwar zum angegebenen Zeitpunkt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Löschen eines RDS Custom for Oracle-Snapshots

Sie können von RDS Custom for Oracle verwaltete DB-Snapshots löschen, wenn Sie sie nicht mehr benötigen. Das Löschverfahren ist für Amazon- RDS und RDS Custom DB-Instances identisch.

Die EBS Amazon-Snapshots für die Binär- und Root-Volumes verbleiben länger in Ihrem Konto, da sie möglicherweise mit einigen Instances verknüpft sind, die in Ihrem Konto ausgeführt werden, oder mit anderen RDS Custom for Oracle-Snapshots. Diese EBS Snapshots werden automatisch gelöscht, wenn sie keinen Bezug mehr zu vorhandenen RDS Custom for Oracle-Ressourcen (DB-Instances oder Backups) haben.

Konsole

Um einen Snapshot einer RDS benutzerdefinierten DB-Instance zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot aus, den Sie löschen möchten.
4. Wählen Sie unter Actions (Aktionen) die Option Delete Snapshot (Snapshot löschen) aus.
5. Wählen Sie auf der Bestätigungsseite die Option Delete (Löschen) aus.

AWS CLI

Verwenden Sie den AWS CLI Befehl, um einen RDS benutzerdefinierten Snapshot zu löschen delete-db-snapshot.

Die folgenden Optionen sind erforderlich:

- `--db-snapshot-identifier` — Der zu löschende Snapshot

Das folgende Beispiel löscht den Cluster-Snapshot `my-custom-snapshot`.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Automatisierte Backups von RDS Custom for Oracle löschen

Sie können gespeicherte automatische Backups für RDS Custom for Oracle löschen, wenn sie nicht mehr benötigt werden. Das Verfahren entspricht dem Verfahren zum Löschen von RDS Amazon-Backups.

Konsole

So löschen Sie eine aufbewahrte automatisierte Backup:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
3. Wählen Sie Retained (Aufbewahrt).
4. Wählen Sie die aufbewahrte automatisierte Sicherung, die Sie löschen möchten.
5. Klicken Sie bei Actions auf Delete.
6. Geben Sie auf der Bestätigungsseite **`delete me`** und wählen Sie Löschen aus.

AWS CLI

Sie können ein gespeicherter automatisierter Backup mit dem AWS CLI Befehl delete-db-instance-automated-backup löschen.

Zum Löschen einer aufbewahrten automatisierten Sicherung werden die folgenden Optionen verwendet.

- `--dbi-resource-id`— Die Ressourcen-ID für die RDS benutzerdefinierte Quell-DB-Instance.

  Sie können die Ressourcen-ID für die Quell-DB-Instance eines gespeicherten automatisierten Backups mithilfe des AWS CLI Befehls describe-db-instance-automated-backups ermitteln.

Im folgenden Beispiel wird das aufbewahrte automatisierte Backup mit der Quell-DB-Instance-Ressourcenkennung `custom-db-123ABCEXAMPLE` gelöscht.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance-automated-backup ^
```

```
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Mit Optionsgruppen in RDS Custom for Oracle arbeiten

RDSCustom verwendet Optionsgruppen, um zusätzliche Funktionen zu aktivieren und zu konfigurieren. Eine Optionsgruppe spezifiziert Funktionen, sogenannte Optionen, die für eine RDS Custom for Oracle DB-Instance verfügbar sind. Optionen können Einstellungen enthalten, die angeben, wie die Option funktioniert. Wenn Sie eine RDS Custom for Oracle DB-Instance einer Optionsgruppe zuordnen, werden die angegebenen Optionen und Optionseinstellungen für diese Instance aktiviert. Allgemeine Informationen zu Optionsgruppen bei Amazon RDS finden Sie unterArbeiten mit Optionsgruppen.

Themen

- Überblick über Optionsgruppen in RDS Custom for Oracle
- Oracle-Zeitzone

Überblick über Optionsgruppen in RDS Custom for Oracle

Damit diese Optionen für Ihre Oracle-Datenbank aktiviert werden, fügen Sie diese einer Optionsgruppe hinzu und ordnen anschließend die Optionsgruppe Ihrer DB-Instance zu. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.

Themen

- Zusammenfassung der Optionen von RDS Custom for Oracle
- Grundlegende Schritte zum Hinzufügen einer Option zu einer RDS Custom for Oracle DB-Instance
- Eine Optionsgruppe für in RDS Custom for Oracle erstellen
- Zuordnen einer Optionsgruppe zu einer RDS Custom for Oracle-DB-Instance

Zusammenfassung der Optionen von RDS Custom for Oracle

RDSCustom for Oracle unterstützt die folgenden Optionen für eine DB-Instance.

Option	Options-ID	Beschreibung
Oracle-Zeitzone	`Timezone`	Die Zeitzone, die von Ihrer RDS Custom for Oracle-DB-Instance verwendet wird.

Grundlegende Schritte zum Hinzufügen einer Option zu einer RDS Custom for Oracle DB-Instance

Das allgemeine Verfahren zum Hinzufügen einer Option zu Ihrer RDS Custom for Oracle DB-Instance ist wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie die Option zur Optionsgruppe hinzu.
3. Ordnen Sie die Optionsgruppe Ihrer DB-Instance zu, wenn Sie sie erstellen oder ändern.

Eine Optionsgruppe für in RDS Custom for Oracle erstellen

Sie können eine neue Optionsgruppe erstellen, die ihre Einstellungen von der Standardoptionsgruppe ableitet. Fügen Sie der neuen Optionsgruppe dann eine oder mehrere Optionen hinzu. Wenn Sie bereits über eine vorhandene Optionsgruppe verfügen, können Sie diese Optionsgruppe auch mit allen Optionen in eine neue Optionsgruppe kopieren. Informationen zum Kopieren einer Optionsgruppe finden Sie unterKopieren einer Optionsgruppe.

Die Standardoptionsgruppen für RDS Custom for Oracle sind die folgenden:

- `default:custom-oracle-ee`
- `default:custom-oracle-se2`
- `default:custom-oracle-ee-cdb`
- `default:custom-oracle-se2-cdb`

Wenn Sie eine Optionsgruppe erstellen, werden die Einstellungen von der Standardoptionsgruppe abgeleitet. Nachdem Sie die `TIME_ZONE` Option hinzugefügt haben, können Sie die Optionsgruppe Ihrer DB-Instance zuordnen.

Konsole

Eine Möglichkeit zur Erstellung einer Optionsgruppe besteht in der Verwendung der AWS Management Console.

So erstellen Sie eine neue Optionsgruppe mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Fenster Create option group (Optionsgruppe erstellen) Folgendes aus:

   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihres AWS Kontos eindeutig ist. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.

   b. Bei Description (Beschreibung) geben Sie eine kurze Beschreibung der Optionsgruppe ein. Die Beschreibung ist nur zur Information.

   c. Wählen Sie für Engine eine der folgenden Optionen RDS Benutzerdefiniert für Oracle-DB-Engines:

      - custom-oracle-ee
      - custom-oracle-se2
      - custom-oracle-ee-cdb
      - custom-oracle-se2 CDB

   d. Wählen Sie für Major-Engine-Version eine Haupt-Engine-Version aus, die von RDS Custom for Oracle unterstützt wird. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.
5. Klicken Sie zum Fortfahren auf Create (Erstellen). Wählen Sie zum Abbrechen der Operation Cancel (Abbrechen) aus.

AWS CLI

Verwenden Sie den AWS CLI `create-option-group`Befehl mit den folgenden erforderlichen Parametern, um eine Optionsgruppe zu erstellen.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

Das folgende Beispiel erstellt eine Optionsgruppe namens `testoptiongroup`, die der Oracle Enterprise Edition DB-Engine zugeordnet ist. Die Beschreibung ist in Anführungszeichen gesetzt.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name custom-oracle-ee-cdb \
    --major-engine-version 19 \
    --option-group-description "Test option group for a Custom Oracle CDB"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name custom-oracle-ee-cdb ^
    --major-engine-version 19 ^
    --option-group-description "Test option group for a Custom Oracle CDB"
```

RDS API

Um eine Optionsgruppe zu erstellen, rufen Sie den RDS API `CreateOptionGroup`Amazon-Vorgang auf.

Zuordnen einer Optionsgruppe zu einer RDS Custom for Oracle-DB-Instance

Sie können Ihre Optionsgruppe einer neuen oder vorhandenen DB-Instance zuordnen:

- Wenden Sie für eine neue DB-Instance die Optionsgruppe an, wenn Sie die Instance erstellen. Weitere Informationen finden Sie unter Erstellen einer RDS Custom für Oracle DB-Instance.
- Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern Ihrer RDS Custom for Oracle-DB-Instance.

Oracle-Zeitzone

Verwenden Sie die Zeitzonenoption, um die von Ihrer RDS Custom for Oracle DB-Instance verwendete Systemzeitzone zu ändern. Gründe, die Zeitzone einer DB-Instance zu ändern, sind beispielsweise Kompatibilitätsanforderungen der Umgebung an einem Standort oder eine veraltete Anwendung. Mit der Zeitzonenoption wird die Zeitzone auf der Ebene des Hosts geändert. Wenn Sie die Zeitzone ändern, wirkt sich dies auf alle Datumsspalten und -werte aus, u. a. auf `SYSDATE` und `SYSTIMESTAMP`.

Themen

- Einstellungen für die Zeitzonenoption in RDS Benutzerdefiniert für Oracle
- Verfügbare Zeitzonen in RDS Custom for Oracle
- Überlegungen zur Einstellung der Zeitzone in RDS Custom for Oracle
- Einschränkungen für die Zeitzoneneinstellung in RDS Custom for Oracle
- Hinzufügen der Zeitzonenoption zu einer Optionsgruppe
- Entfernen der Zeitzonenoption

Einstellungen für die Zeitzonenoption in RDS Benutzerdefiniert für Oracle

Amazon RDS unterstützt die folgenden Einstellungen für die Zeitzonenoption.

Optionseinstellung	Zulässige Werte	Beschreibung
`TIME_ZONE`	Eine der verfügbaren Zeitzonen. Eine vollständige Liste finden Sie unter Verfügbare Zeitzonen in RDS Custom for Oracle.	Die neue Zeitzone für Ihre DB-Instance.

Verfügbare Zeitzonen in RDS Custom for Oracle

Sie können die folgenden Werte für die Zeitzonenoption verwenden:

Bereich	Zeitzone
Afrika	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
Amerika	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York,

Bereich	Zeitzone
	America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asien	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damascus, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Jakarta, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantic/Azores, Atlantic/Cape_Verde
Australien	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brasilien	Brasilien/, Brasilien/Ost DeNoronha
Kanada	Canada/Newfoundland, Canada/Saskatchewan
Etc	Etc/ GMT -3
Europa	Europe/Amsterdam, Europe/Athens, Europe/Berlin, Europe/Dublin, Europe/Helsinki, Europe/Kaliningrad, Europe/London, Europe/Madrid, Europe/Moscow, Europe/Paris, Europe/Prague, Europe/Rome, Europe/Sarajevo
Pazifik	Pacific/Apia, Pacific/Auckland, Pacific/Chatham, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Kiritimati, Pacific/Marquesas, Pacific/Samoa, Pacific/Tongatapu, Pacific/Wake
USA	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific
UTC	UTC

Überlegungen zur Einstellung der Zeitzone in RDS Custom for Oracle

Wenn Sie die Zeitzone für Ihre DB-Instance festlegen möchten, sollten Sie Folgendes berücksichtigen:

- Wenn Sie die Zeitzonenoption hinzufügen, entsteht während des automatischen Neustarts Ihrer DB-Instance ein kurzzeitiger Nutzungsausfall.
- Wenn Sie die Zeitzone versehentlich falsch eingestellt haben, müssen Sie die DB-Instance auf ihre vorherige Zeitzoneneinstellung zurücksetzen. Aus diesem Grund empfehlen wir Ihnen dringend, eine der folgenden Strategien zu verwenden, bevor Sie Ihrer Instance die Zeitzonenoption hinzufügen:
  - Wenn Ihre RDS Custom for Oracle-DB-Instance die Standardoptionsgruppe verwendet, erstellen Sie einen Snapshot Ihrer DB-Instance. Weitere Informationen finden Sie unter Einen RDS benutzerdefinierten Snapshot für Oracle erstellen.
  - Wenn Ihre DB-Instance derzeit eine nicht standardmäßige Optionsgruppe verwendet, erstellen Sie einen Snapshot Ihrer DB-Instance und erstellen Sie dann eine neue Optionsgruppe mit der Zeitzonenoption.
- Wir empfehlen dringend, dass Sie Ihre DB-Instance manuell sichern, nachdem Sie die `Timezone` Option angewendet haben.
- Wir empfehlen Ihnen dringend, die Zeitzonenoption auf einer Test-DB-Instance zu testen, bevor Sie sie zu einer Produktions-DB-Instance hinzufügen. Bei dem Hinzufügen der Zeitzonenoption können Probleme in Zusammenhang mit Tabellen auftreten, die die Systemzeit verwenden, um Datums- bzw. Uhrzeitangaben einzufügen. Analysieren Sie Ihre Daten und Anwendungen, um die Auswirkungen einer Änderung der Zeitzone zu einzuschätzen.

Einschränkungen für die Zeitzoneneinstellung in RDS Custom for Oracle

Es gelten die folgenden Einschränkungen:

- Sie können Ihre Zeitzone nicht direkt auf Ihrem Host ändern, ohne sie außerhalb des Support-Bereichs zu verschieben. Um die Zeitzone Ihrer Datenbank zu ändern, müssen Sie eine Optionsgruppe erstellen.
- Da es sich bei der Zeitzonenoption um eine persistente Option (aber keine permanente Option) handelt, können Sie Folgendes nicht tun:
  - Entfernen Sie die Option aus einer Optionsgruppe, nachdem Sie die Option hinzugefügt haben.
  - Ändern Sie die Zeitzoneneinstellung für die Option zu einer anderen Zeitzone.
- Sie können Ihrer RDS Custom for Oracle-DB-Instance nicht mehrere Optionsgruppen zuordnen.
- Sie können die Zeitzone für eine Person PDBs innerhalb einer nicht festlegenCDB.

Hinzufügen der Zeitzonenoption zu einer Optionsgruppe

Die Standardoptionsgruppen für RDS Custom for Oracle sind die folgenden:

- `default:custom-oracle-ee`
- `default:custom-oracle-se2`
- `default:custom-oracle-ee-cdb`
- `default:custom-oracle-se2-cdb`

Wenn Sie eine Optionsgruppe erstellen, werden die Einstellungen von der Standardoptionsgruppe abgeleitet. Allgemeine Informationen zu Optionsgruppen bei Amazon RDS finden Sie unterArbeiten mit Optionsgruppen.

Konsole

So fügen Sie die Zeitzonenoption zu einer Optionsgruppe hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus, die Sie ändern möchten, und wählen Sie dann Add option (Option hinzufügen).
4. Führen Sie im Fenster Add option (Option hinzufügen) die folgenden Schritte aus:
   a. Wählen Sie Timezone.
   b. Wählen Sie in den Optionseinstellungen eine Zeitzone aus.
   c. Um die Option für alle zugehörigen RDS Custom for Oracle DB-Instances zu aktivieren, sobald Sie sie hinzufügen, wählen Sie für Apply Immediately die Option Ja. Wenn Sie Nein (Standardeinstellung) wählen, wird die Option für alle zugehörigen DB-Instances im nächsten Wartungsfenster aktiviert.
   d.

> **Important**
>
> Wenn Sie einer bestehenden Optionsgruppe, die bereits mit einer oder mehreren DB-Instances verknüpft ist, die Zeitzonenoption hinzufügen, werden alle DB-Instances neu gestartet und es entsteht ein kurzzeitiger Nutzungsausfall.

5. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Add option (Option hinzufügen) aus.
6. Erstellen Sie eine RDS Sicherungskopie von Custom für Oracle-DB-Instances, deren Zeitzonen aktualisiert wurden. Weitere Informationen finden Sie unter Einen RDS benutzerdefinierten Snapshot für Oracle erstellen.

AWS CLI

Im folgenden Beispiel wird der Befehl AWS CLI add-option-to-option-group verwendet, um die `Timezone` Option und die `TIME_ZONE` Optionseinstellung zu einer Optionsgruppe namens `testoptiongroup` hinzuzufügen. Als Zeitzone wird festgeleg `America/Los_Angeles`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "testoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "testoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

Entfernen der Zeitzonenoption

Die Zeitzonenoption ist eine persistente Option, aber keine permanente Option. Wenn Sie die Option einer Optionsgruppe hinzugefügt haben, kann sie nicht mehr aus der Gruppe entfernt werden. So trennen Sie die alte Optionsgruppe von Ihrer DB-Instance:

1. Erstellen Sie eine neue Optionsgruppe mit einer aktualisierten `Timezone` Option.
2. Ordnen Sie die neue Optionsgruppe Ihrer DB-Instance zu, wenn Sie die Instance ändern.

Migration einer lokalen Datenbank zu RDS Custom for Oracle

Bevor Sie eine lokale Oracle-Datenbank zu RDS Custom for Oracle migrieren, sollten Sie die folgenden Faktoren berücksichtigen:

- Die Länge der Ausfallzeiten, die für die Anwendung akzeptabel sind
- Die Größe der Quelldatenbank
- Netzwerkkonnektivität
- Die Notwendigkeit eines Fallback-Plans
- Die Quell- und Ziel-Oracle-Datenbankversion und die Betriebssystemtypen der DB-Instance
- Verfügbare Replikationstools wie AWS Database Migration Service GoldenGate, Oracle oder Replikationstools von Drittanbietern

Basierend auf diesen Faktoren können Sie zwischen einer physischen Migration, einer logischen Migration oder einer Kombination wählen. Wenn Sie sich für eine physische Migration entscheiden, können Sie die folgenden Methoden verwenden:

RMANVervielfältigung

Für die aktive Datenbankduplizierung ist kein Backup Ihrer Quelldatenbank erforderlich. Die Live-Quelldatenbank wird auf den Zielhost dupliziert, indem Datenbankdateien über das Netzwerk auf die Zusatz-Instance kopiert werden. Der RMAN `DUPLICATE` Befehl kopiert die erforderlichen Dateien als Image-Kopien oder Backup-Sets. Informationen zu dieser Technik finden Sie im AWS Blogbeitrag Physische Migration von Oracle-Datenbanken zu Amazon RDS Custom mithilfe von RMAN Duplikation.

Oracle Data Guard

Bei dieser Methode sichern Sie eine primäre On-Premises Datenbank und kopieren die Backups in einen Amazon-S3-Bucket. Anschließend kopieren Sie die Backups in Ihre RDS Custom for Oracle-Standby-DB-Instance. Nachdem Sie die erforderliche Konfiguration durchgeführt haben, wechseln Sie manuell von Ihrer Primärdatenbank zu Ihrer RDS Custom for Oracle-Standby-Datenbank. Informationen zu dieser Technik finden Sie im AWS Blogbeitrag Physische Migration von Oracle-Datenbanken zu Amazon RDS Custom mit Data Guard.

Allgemeine Informationen zum logischen Import von Daten in Oracle RDS finden Sie unterDaten in Oracle auf Amazon importieren RDS.

Upgrade einer DB-Instance für Amazon RDS Custom for Oracle

Sie können eine Amazon RDS Custom DB-Instance aktualisieren, indem Sie sie so ändern, dass sie eine neue benutzerdefinierte Engine-Version (CEV) verwendet. Allgemeine Informationen zu Upgrades finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Themen

- Überblick über Upgrades in RDS Custom for Oracle
- Anforderungen für RDS Custom for Oracle-Upgrades
- Überlegungen zu RDS Custom for Oracle-Datenbank-Upgrades
- Überlegungen zu RDS kundenspezifischen Upgrades für Oracle OS
- Gültige CEV Upgrade-Ziele für RDS benutzerdefinierte Oracle-DB-Instances anzeigen
- Eine RDS benutzerdefinierte für Oracle DB-Instance aktualisieren
- Ausstehende Datenbank-Upgrades für RDS benutzerdefinierte DB-Instances anzeigen
- Behebung eines Upgrade-Fehlers für eine RDS Custom for Oracle DB-Instance

Überblick über Upgrades in RDS Custom for Oracle

Mit RDS Custom for Oracle können Sie entweder Ihre Oracle-Datenbank oder Ihr DB-Instance-Betriebssystem (OS) patchen, indem Sie eine neue Instanz erstellen CEVs und dann Ihre Instance so ändern, dass sie die neue Version verwendetCEV.

Themen

- CEVUpgrade-Optionen
- Patchen ohne CEVs
- Allgemeine Schritte zum Patchen Ihrer DB-Instance mit einem CEV

CEVUpgrade-Optionen

Wenn Sie eine CEV für ein Upgrade erstellen, stehen Ihnen die folgenden Optionen zur Verfügung, die sich gegenseitig ausschließen:

Nur Datenbank

Verwenden Sie das Amazon Machine Image (AMI), das derzeit von Ihrer DB-Instance verwendet wird, wieder, geben Sie jedoch andere Datenbank-Binärdateien an. RDSCustom weist ein neues

Binärvolume zu und hängt es dann an die bestehende Amazon-Instance an. EC2 RDSCustom ersetzt das gesamte Datenbank-Volume durch ein neues Volume, das Ihre Zieldatenbankversion verwendet.

Nur OS

Verwenden Sie die Datenbank-Binärdateien, die derzeit von Ihrer DB-Instance verwendet werden, erneut, geben Sie jedoch eine andere AMI an. RDSCustom weist eine neue EC2 Amazon-Instance zu und hängt dann das bestehende Binär-Volume an die neue Instance an. Das bestehende Datenbank-Volume wird beibehalten.

Wenn Sie sowohl das Betriebssystem als auch die Datenbank aktualisieren möchten, müssen Sie sie zweimal aktualisieren. CEV Sie können entweder das Betriebssystem und dann die Datenbank oder die Datenbank und dann das Betriebssystem aktualisieren.

> **Warning**
>
> Wenn Sie Ihr Betriebssystem patchen, verlieren Sie Ihre Root-Volume-Daten und alle vorhandenen Betriebssystemanpassungen. Daher empfehlen wir Ihnen dringend, das Root-Volume nicht für Installationen oder zum Speichern von permanenten Daten oder Dateien zu verwenden. Außerdem sollten Sie Ihre Daten vor dem Upgrade sichern.

Patchen ohne CEVs

Wir empfehlen dringend, dass Sie Ihre RDS Custom for Oracle-DB-Instance mithilfe von CEVs aktualisieren. RDSCustom for Oracle Automation synchronisiert die Patch-Metadaten mit der Datenbank-Binärdatei auf Ihrer DB-Instance.

Unter besonderen Umständen unterstützt RDS Custom das direkte Anwenden eines „einmaligen“ Datenbank-Patches auf die zugrunde liegende EC2 Amazon-Instance mithilfe des OPatch Dienstprogramms. Ein gültiger Anwendungsfall könnte ein Datenbank-Patch sein, den Sie sofort anwenden möchten, aber das RDS Custom-Team aktualisiert die CEV Funktion, was zu Verzögerungen führt. Führen Sie die folgenden Schritte aus, um einen manuellen Datenbank-Patch anzuwenden:

1. Unterbrechen Sie die RDS benutzerdefinierte Automatisierung.
2. Wenden Sie Ihren Patch auf die Datenbank-Binärdateien auf der EC2 Amazon-Instance an.

3. Setzen Sie die RDS benutzerdefinierte Automatisierung fort.

Ein Nachteil der vorherigen Technik besteht darin, dass Sie den Patch manuell auf jede Instance anwenden müssen, die Sie aktualisieren möchten. Wenn Sie dagegen eine neue erstellenCEV, können Sie mehrere DB-Instances mit derselben erstellen oder aktualisierenCEV.

Allgemeine Schritte zum Patchen Ihrer DB-Instance mit einem CEV

Unabhängig davon, ob Sie das Betriebssystem oder Ihre Datenbank patchen, führen Sie die folgenden grundlegenden Schritte aus:

1. Erstellen Sie eineCEV, die eines der folgenden Elemente enthält, je nachdem, ob Sie die Datenbank oder das Betriebssystem patchen:
   - Die Oracle Database RU, die Sie auf Ihre DB-Instance anwenden möchten
   - Ein anderes AMI — entweder das neueste verfügbare oder eines, das Sie angeben — und ein vorhandenes, das als Quelle verwendet werden CEV soll

   Führen Sie die Schritte unter Erstellen einer CEV aus.
2. (Optional für Datenbank-Patches) Überprüfen Sie die verfügbaren Engine-Versions-Upgrades, indem Sie den Befehl `describe-db-engine-versions` ausführen.
3. Starten Sie den Patch-Vorgang, indem Sie `modify-db-instance` ausführen.

   Der Status der Instance, die gepatcht wird, unterscheidet sich wie folgt:
   - Während die Datenbank gepatcht RDS wird, ändert sich der Status der DB-Instance in Upgrading.
   - Während RDS das Betriebssystem gepatcht wird, ändert sich der Status der DB-Instance in Modifying.

   Wenn die DB-Instance den Status Verfügbar hat, ist das Patchen abgeschlossen.
4. Vergewissern Sie sich, dass Ihre DB-Instance die neue Version verwendet, CEV indem Sie Folgendes ausführen`describe-db-instances`.

Anforderungen für RDS Custom for Oracle-Upgrades

Wenn Sie Ihre RDS Custom for Oracle-DB-Instance auf ein Ziel aktualisierenCEV, stellen Sie sicher, dass Sie die folgenden Anforderungen erfüllen:

- Das ZielCEV, auf das Sie ein Upgrade durchführen, muss vorhanden sein.
- Sie müssen entweder das Betriebssystem oder die Datenbank in einem einzigen Vorgang aktualisieren. Ein Upgrade sowohl des Betriebssystems als auch der Datenbank in einem einzigen API Aufruf wird nicht unterstützt.
- Das Ziel CEV muss die Installationsparametereinstellungen verwenden, die im Manifest der aktuellen Version enthalten sindCEV. Sie können beispielsweise eine Datenbank, die das Standard-Oracle-Standardverzeichnis verwendet, nicht auf eine Datenbank aktualisierenCEV, die ein anderes Oracle-Standardverzeichnis verwendet.
- Für Datenbank-Upgrades CEV muss das Ziel eine neue Datenbanknebenversion verwenden, keine neue Hauptversion. Sie können beispielsweise kein Upgrade von einer Oracle-Datenbank 12c CEV auf eine Oracle-Datenbank CEV 19c durchführen. Sie können jedoch ein Upgrade von Version 21.0.0.0.ru-2023-04.rur-2023-04.r1 auf Version 21.0.0.0.ru-2023-07.rur-2023-07.r1 durchführen.
- Für Betriebssystem-Upgrades CEV muss das Ziel eine andere, AMI aber dieselbe Hauptversion verwenden.

Überlegungen zu RDS Custom for Oracle-Datenbank-Upgrades

Wenn Sie planen, Ihre Datenbank zu aktualisieren, sollten Sie Folgendes berücksichtigen:

- Die derzeit unterstützte Betriebssystemversion (OS) ist Oracle Linux 8. Um weiterhin die neuesten Sicherheitsupdates und Patches von RDS Custom for Oracle zu erhalten, aktualisieren Sie Ihre DB-Instances auf Oracle Linux 8, indem Sie ein CEV angeben, das auf diesem Betriebssystem basiert. Oracle Database 12c Release 1 (12.1), Oracle Database Release 2 (12.2) und Oracle Database 19c sind die einzigen Versionen, die Oracle Linux 8 unterstützen. Um auf das neueste Oracle Linux 8 AMI zu migrieren, aktualisieren Sie Ihr Betriebssystem auf das neueste AMI. Weitere Informationen finden Sie unter Eine RDS benutzerdefinierte für Oracle DB-Instance aktualisieren.

  Der Support für Oracle Linux 7.9 wurde am 31. Dezember 2024 eingestellt. Um Oracle Linux 7 nach dem Ende des Support weiter ausführen zu können, erwerben Sie eine Oracle Extended Support-Lizenz. Sie sind für Sicherheitsupdates verantwortlich und müssen Ihre RDS Custom for Oracle-Instances manuell patchen. Weitere Informationen finden Sie unter Lifetime Support Policy: Deckung für Oracle Open Source Service Offerings.
- Wenn Sie ein Upgrade für die Datenbank-Binärdateien in der primären DB-Instance durchführen, aktualisiert RDS Custom für Oracle Ihre Lesereplikate automatisch. Wenn Sie das Betriebssystem aktualisieren, müssen Sie die Read Replicas jedoch manuell aktualisieren.

- Wenn Sie eine Container-Datenbank (CDB) auf eine neue Datenbankversion aktualisieren, überprüft RDS Custom for Oracle, ob alle geöffnet PDBs sind oder geöffnet werden könnten. Wenn diese Bedingungen nicht erfüllt sind, stoppt RDS Custom die Prüfung und setzt die Datenbank in ihren ursprünglichen Zustand zurück, ohne das Upgrade zu versuchen. Wenn die Bedingungen erfüllt sind, patcht RDS Custom zuerst den CDB-Stamm und dann alle anderen PDBs (einschließlichPDB$SEED) parallel.

  Nach Abschluss des Patchens versucht RDS Custom, alle zu öffnen. PDBs Wenn eines davon PDBs nicht geöffnet werden kann, erhalten Sie das folgende Ereignis:`The following PDBs failed to open: list-of-PDBs`. Wenn RDS Custom das CDB-Root oder ein anderes nicht patchen kann PDBs, wird die Instance in den `PATCH_DB_FAILED` Status versetzt.
- Möglicherweise möchten Sie gleichzeitig ein Hauptversions-Upgrade und eine Konvertierung von Nicht-CDB zu CDB durchführen. In diesem Fall empfehlen wir folgende Vorgehensweise:
  1. Erstellen Sie eine neue RDS Custom for Oracle DB-Instance, die die Oracle-Multitenant-Architektur verwendet.
  2. Schließen Sie eine Nicht-CDB in Ihr CDB-Root an und erstellen Sie sie als PDB. Stellen Sie sicher, dass die Nicht-CDB dieselbe Hauptversion wie Ihre CDB hat.
  3. Konvertieren Sie Ihre PDB, indem Sie das `noncdb_to_pdb.sql` Oracle SQL-Skript ausführen.
  4. Validieren Sie Ihre CDB-Instance.
  5. Führen Sie ein Upgrade für Ihre CDB-Instance aus.

Überlegungen zu RDS kundenspezifischen Upgrades für Oracle OS

Wenn Sie ein Betriebssystem-Upgrade planen, sollten Sie Folgendes berücksichtigen:

- Sie können keine eigenen Daten AMI zur Verwendung in einem RDS Custom for Oracle bereitstellenCEV. Sie können entweder den Standard angebenAMI, der Oracle Linux 8 verwendet, oder einenAMI, der zuvor von einem RDS Custom for Oracle verwendet wurdeCEV.

> **Note**
>
> RDSCustom for Oracle veröffentlicht einen neuen Standard, AMI wenn allgemeine Sicherheitslücken und Sicherheitslücken entdeckt werden. Es ist kein fester Zeitplan verfügbar oder garantiert. RDS Custom for Oracle veröffentlicht in der Regel AMI alle 30 Tage einen neuen Standard.

- Wenn Sie das Betriebssystem in Ihrer primären DB-Instance aktualisieren, müssen Sie die zugehörigen Read Replicas manuell aktualisieren.
- Reservieren Sie ausreichend EC2 Amazon-Rechenkapazität für Ihren Instance-Typ in Ihrer AZ, bevor Sie mit dem Patchen des Betriebssystems beginnen.

  Wenn Sie eine Kapazitätsreservierung erstellen, geben Sie die AZ, die Anzahl der Instances und die Instance-Attribute (einschließlich des Instance-Typs) an. Wenn Ihre DB-Instance beispielsweise den zugrunde liegenden EC2 Instance-Typ r5.large verwendet, empfehlen wir Ihnen, EC2 Kapazität für r5.large in Ihrer AZ zu reservieren. Während des Betriebssystem-Patches erstellt RDS Custom einen neuen Host vom Typ db.r5.large, der hängen bleiben kann, wenn der AZ die Kapazität für diesen Instance-Typ fehlt. EC2 Wenn Sie EC2 Kapazität reservieren, verringern Sie das Risiko, dass Patches aufgrund von Kapazitätsengpässen blockiert werden. Weitere Informationen finden Sie unter Kapazitätsreservierungen auf Abruf im EC2Amazon-Benutzerhandbuch.
- Erstellen Sie eine Sicherungskopie Ihrer DB-Instance, bevor Sie ihr Betriebssystem aktualisieren. Durch das Upgrade werden Ihre Root-Volume-Daten und alle vorhandenen Betriebssystemanpassungen entfernt.
- Im Modell der geteilten Verantwortung sind Sie dafür verantwortlich, Ihr Betriebssystem auf dem neuesten Stand zu halten. RDS Custom for Oracle schreibt nicht vor, welche Patches Sie auf Ihr Betriebssystem anwenden. Wenn Ihr RDS Custom for Oracle funktionsfähig ist, können Sie die AMI damit verbundenen Funktionen CEV unbegrenzt verwenden.

Gültige CEV Upgrade-Ziele für RDS benutzerdefinierte Oracle-DB-Instances anzeigen

Sie können die vorhandenen Versionen CEVs auf der Seite Benutzerdefinierte Engine-Versionen im sehen AWS Management Console.

Sie können den describe-db-engine-versions AWS CLI Befehl auch verwenden, um nach einer gültigen Version CEVs zu suchen, die Sie verwenden können, wenn Sie Ihre DB-Instances aktualisieren, wie im folgenden Beispiel gezeigt. In diesem Beispiel wird vorausgesetzt, dass Sie eine DB-Instance mit der Engine-Version `19.my_cev1` erstellt haben und dass die Upgrade-Versionen `19.my_cev2` und `19.my_cev` vorhanden sind.

```
aws rds describe-db-engine-versions --engine custom-oracle-ee --engine-version
 19.my_cev1
```

Die Ausgabe sieht in etwa folgendermaßen aus. Das `ImageId` Feld zeigt die AMI ID.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.my_cev1",
            ...
            "Image": {
                "ImageId": "ami-2345",
                "Status": "active"
            },
            "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-
oracle-ee/19.my_cev1/12a34b5c-67d8-90e1-2f34-gh56ijk78lm9"
            "ValidUpgradeTarget": [
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev2",
                    "Description": "19.my_cev2 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev3",
                    "Description": "19.my_cev3 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                }
            ]
            ...
```

Eine RDS benutzerdefinierte für Oracle DB-Instance aktualisieren

Um Ihre RDS Custom for Oracle DB-Instance zu aktualisieren, ändern Sie sie so, dass sie eine neue verwendetCEV. Diese CEV kann entweder neue Datenbank-Binärdateien oder eine neue AMI enthalten. Um beispielsweise Ihre Oracle Linux 7.9-DB-Instance auf Oracle Linux 8 zu aktualisieren, geben Sie die neueste Version anAMI, die Oracle Linux 8 verwendet. Um die Datenbank und das Betriebssystem zu aktualisieren, müssen Sie zwei separate Upgrades durchführen.

> **Note**
>
> Wenn Sie die Datenbank aktualisieren, aktualisiert RDS Custom automatisch Read Replicas, nachdem es die primäre DB-Instance aktualisiert hat. Wenn Sie das Betriebssystem aktualisieren, müssen Sie die Lesereplikate manuell aktualisieren.

Bevor Sie beginnen, überprüfen Sie Anforderungen für RDS Custom for Oracle-Upgrades und Überlegungen zu RDS Custom for Oracle-Datenbank-Upgrades.

Konsole

Um eine RDS Custom for Oracle-DB-Instance zu aktualisieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte Oracle-DB-Instance aus, für die Sie ein Upgrade durchführen möchten.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Wählen Sie für die DB-Engine-Version eine neue ausCEV. Gehen Sie wie folgt vor:

   - Wenn Sie die Datenbank patchen, stellen Sie sicher, dass die Datenbank-Binärdateien CEV angibt, die sich von denen unterscheiden, die von Ihrer DB-Instance verwendet werden, und AMI dass Sie keine angeben, die sich von den AMI aktuell von Ihrer DB-Instance verwendeten unterscheidet.
   - Wenn Sie das Betriebssystem patchen, stellen Sie sicher, dass das eine CEV spezifiziert, das sich von dem AMI aktuell von Ihrer DB-Instance verwendeten unterscheidet, und AMI dass es keine anderen Datenbank-Binärdateien spezifiziert.

   > **Warning**
   >
   > Wenn Sie Ihr Betriebssystem patchen, verlieren Sie Ihre Root-Volume-Daten und alle vorhandenen Betriebssystemanpassungen.

5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.

   Wählen Sie Apply immediately (Sofort anwenden), um die Änderungen sofort anzuwenden.

6. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern. Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Die folgenden Beispiele zeigen mögliche Upgrade-Szenarien. In den Beispielen wird davon ausgegangen, dass Sie eine RDS benutzerdefinierte DB-Instance für Oracle mit den folgenden Merkmalen erstellt haben:

- DB-Instance mit dem Namen `my-custom-instance`
- CEVbenannt `19.my_cev1`
- Oracle Database 19c verwendet die Nicht-Architektur CDB
- Oracle Linux 8 unter Verwendung AMI `ami-1234`

Der neueste bereitgestellte Service AMI ist`ami-2345`. Sie können es finden, AMIs indem Sie den CLI Befehl `describe-db-engine-versions` ausführen.

Themen

- Betriebssystem-Upgrade
- Aktualisieren der Datenbank

Betriebssystem-Upgrade

In diesem Beispiel möchten Sie ein Upgrade `ami-1234` auf `ami-2345` den neuesten bereitgestellten Dienst durchführenAMI. Da Sie das Betriebssystem aktualisieren, `ami-2345` müssen die Datenbankbinärdateien für `ami-1234` und identisch sein. Sie erstellen einen neuen CEV Namen auf `19.my_cev1` der `19.my_cev2` Grundlage von.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev2 \
    --description "Non-CDB CEV based on ami-2345" \
```

```
    --kms-key-id key-name \
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 \
    --image-id ami-2345
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev2 ^
    --description "Non-CDB CEV based on ami-2345" ^
    --kms-key-id key-name ^
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 ^
    --image-id ami-2345
```

Um eine RDS benutzerdefinierte DB-Instance zu aktualisieren, verwenden Sie den modify-db-instance AWS CLI Befehl mit den folgenden Parametern:

- `--db-instance-identifier`— Geben Sie die RDS benutzerdefinierte Oracle-DB-Instance an, die aktualisiert werden soll.
- `--engine-version`— Geben Sie die anCEV, die die neue hatAMI.
- `--no-apply-immediately` | `--apply-immediately` – Geben Sie an, ob das Upgrade sofort durchgeführt oder bis zum geplanten Wartungsfenster gewartet werden soll.

Das folgende Beispiel aktualisiert `my-custom-instance` zur Version `19.my_cev2`. Nur das Betriebssystem wird aktualisiert.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev2 \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev2 ^
    --apply-immediately
```

Aktualisieren der Datenbank

In diesem Beispiel möchten Sie den Oracle-Patch p35042068 auf Ihre RDS Oracle-DB-Instance anwenden. Da Sie Ihr Betriebssystem in Betriebssystem-Upgrade aktualisiert haben, verwendet Ihre DB-Instance derzeit `19.my_cev2`, basierend auf `ami-2345`. Sie erstellen einen neuen CEV Namen`19.my_cev3`, der auch verwendet`ami-2345`, aber Sie geben ein neues JSON Manifest in der `$MANIFEST` Umgebungsvariablen an. Somit unterscheiden sich nur die Datenbank-Binärdateien in Ihrer neuen Version CEV und in derCEV, die Ihre Instanz gerade verwendet.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev3 \
    --description "Non-CDB CEV with p35042068 based on ami-2345" \
    --kms-key-id key-name \
    --image-id ami-2345 \
    --manifest $MANIFEST
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev3 ^
    --description "Non-CDB CEV with p35042068 based on ami-2345" ^
    --kms-key-id key-name ^
    --image-id ami-2345 ^
    --manifest $MANIFEST
```

Im folgenden Beispiel wird `my-custom-instance` auf die Engine-Version `19.my_cev3` aktualisiert. Nur die Datenbank wird aktualisiert.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev3 \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev3 ^
    --apply-immediately
```

Ausstehende Datenbank-Upgrades für RDS benutzerdefinierte DB-Instances anzeigen

Mit dem describe-pending-maintenance-actions AWS CLI Befehl describe-db-instancesoder können Sie ausstehende Datenbank-Upgrades für Ihre Amazon RDS Custom DB-Instances einsehen.

Dieser Ansatz funktioniert jedoch nicht, wenn Sie die `--apply-immediately`-Option genutzt haben oder wenn das Upgrade ausgeführt wird.

Der folgende `describe-db-instances`-Befehl zeigt ausstehende Datenbank-Upgrades für `my-custom-instance` an.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
        }
    ]
}
```

Behebung eines Upgrade-Fehlers für eine RDS Custom for Oracle DB-Instance

Wenn ein Upgrade einer RDS benutzerdefinierten DB-Instance fehlschlägt, wird ein RDS Ereignis generiert und der DB-Instance-Status lautet`upgrade-failed`.

Sie können diesen Status mithilfe des describe-db-instances AWS CLI Befehls anzeigen, wie im folgenden Beispiel gezeigt.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
            "DBInstanceStatus": "upgrade-failed"
        }
    ]
}
```

Nach einem Upgrade-Fehler werden alle Datenbankaktionen blockiert, außer dass die DB-Instance geändert wird, um die folgenden Aufgaben auszuführen:

- Wiederholen des gleichen Upgrades
- Anhalten und Wiederaufnehmen RDS der benutzerdefinierten Automatisierung
- Point-in-time Wiederherstellung () PITR
- Löschen einer DB-Instance

> **Note**
>
> Wenn die Automatisierung für die RDS benutzerdefinierte DB-Instance angehalten wurde, können Sie das Upgrade erst wiederholen, wenn Sie die Automatisierung fortsetzen.

Bei einem Upgrade-Fehler gelten für eine RDS -verwaltete Read Replica dieselben Aktionen wie für die primäre.

Weitere Informationen finden Sie unter Fehlerbehebung bei Upgrades für RDS Custom for Oracle.

Beheben von DB-Problemen für Amazon RDS Custom für Oracle

Das Modell der gemeinsamen Verantwortung von RDS Custom bietet Zugriff auf Betriebssystem-Shell-Ebene und Zugriff auf Datenbankadministratoren. RDS Custom führt Ressourcen in Ihrem Konto aus, im Gegensatz zu Amazon RDS, das Ressourcen in einem Systemkonto ausführt. Mit einem größeren Zugang kommt eine größere Verantwortung mit sich. In den folgenden Abschnitten können Sie lernen, wie Sie Probleme mit Amazon-RDS-Custom DB-Instances beheben können.

> **Note**
>
> In diesem Abschnitt wird die Fehlerbehebung für RDS Custom für Oracle beschrieben. Informationen zur Fehlerbehebung für RDS Custom für SQL Server finden Sie unter Beheben von DB-Problemen für Amazon RDS Custom for SDL Server.

Themen

- Anzeigen von benutzerdefinierten RDS-Ereignissen
- Abonnieren von benutzerdefinierten RDS-Ereignissen
- Fehlerbehebung bei der Erstellung von benutzerdefinierten Engine-Versionen für RDS Custom for Oracle
- Fehlerbehebung bei nicht unterstützten Konfigurationen in RDS Custom für Oracle
- Fehlerbehebung bei Upgrades für RDS Custom for Oracle
- Behebung von Fehlerbehebung bei der Heraufstufung von Replikaten für RDS

Anzeigen von benutzerdefinierten RDS-Ereignissen

Das Verfahren zum Anzeigen von Ereignissen ist für RDS Custom- und Amazon RDS DB-Instances gleich. Weitere Informationen finden Sie unter RDSAmazon-Ereignisse anzeigen.

Verwenden Sie den `describe-events` Befehl AWS CLI, um die benutzerdefinierte RDS-Ereignisbenachrichtigung mit dem anzuzeigen. RDS Custom führt mehrere neue Ereignisse ein. Die Ereigniskategorien sind die gleichen wie für Amazon RDS. Eine Liste der Ereignisse finden Sie unter Amazon RDS-Ereigniskategorien und Ereignisnachrichten .

Im folgenden Beispiel werden Details zu den Ereignissen abgerufen, die für die angegebene RDS Custom DB-Instance aufgetreten sind.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Abonnieren von benutzerdefinierten RDS-Ereignissen

Das Verfahren zum Abonnieren von Ereignissen ist für RDS Custom- und Amazon RDS DB-Instances gleich. Weitere Informationen finden Sie unter RDSAmazon-Veranstaltungsbenachrichtigung abonnieren.

Verwenden Sie den Befehl `create-event-subscription`, um Ereignisbenachrichtigungen von RDS Custom zu abonnieren. Nutzen Sie die folgenden erforderlichen Parameter:

- `--subscription-name`
- `--sns-topic-arn`

Im folgenden Beispiel wird ein Abonnement für Backup- und Wiederherstellungsereignisse für eine RDS Custom DB-Instance im aktuellen AWS -Konto. Sie können auch anfordern, dass Benachrichtigungen an ein bestimmtes Amazon-SNS-Thema (Amazon Simple Notification Service) gesendet werden, die `--sns-topic-arn` bestimmt.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Fehlerbehebung bei der Erstellung von benutzerdefinierten Engine-Versionen für RDS Custom for Oracle

Wenn die CEV-Erstellung fehlschlägt, gibt es Probleme mit RDS Custom `RDS-EVENT-0198`mit der Mitteilung`Creation failed for custom engine version` *`major-engine-version.cev_name`* und enthält Details über den Fehler. Zum Beispiel druckt das Ereignis fehlende Dateien.

Die Erstellung von CEV schlägt möglicherweise aufgrund der folgenden Probleme fehl:

- Der Amazon S3 S3-Bucket, der Ihre Installationsdateien enthält, befindet sich nicht in derselben AWS Region wie Ihr CEV.

- Wenn Sie die CEV-Erstellung in einem AWS-Region zum ersten Mal anfordern, erstellt RDS Custom einen S3-Bucket zum Speichern von benutzerdefinierten RDS-Ressourcen (wie CEV-Artefakten, AWS CloudTrail Protokollen und Transaktionsprotokollen).

  Die CEV-Erstellung schlägt fehl, wenn RDS Custom den S3-Bucket nicht erstellen kann. Entweder hat der Anrufer keine S3-Berechtigungen wie unter Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen oder die Anzahl der S3-Buckets hat das Limit erreicht.
- Der Anrufer hat keine Berechtigung, Dateien aus Ihrem S3-Bucket abzurufen, der die Installationsmediendateien enthält. Diese Berechtigungen sind unter Schritt 7: Hinzufügen der erforderlichen IAM-Berechtigungen beschrieben.
- Ihre IAM-Richtlinie hat eine `aws:SourceIp`-Bedingung. Befolgen Sie unbedingt die Empfehlungen unter AWS verweigert Zugriff auf AWS basierend auf der Quell-IP im AWS Identity and Access Management -Benutzerhandbuch. Stellen Sie außerdem sicher, dass der Aufrufer über die S3-Berechtigungen verfügt, die unter Schritt 5: Erteilen Sie Ihrem IAM-Benutzer oder Ihrer IAM-Rolle die erforderlichen Berechtigungen beschrieben sind.
- Die im CEV-Manifest aufgeführten Installationsmediendateien befinden sich nicht in Ihrem S3-Bucket.
- Die SHA-256-Prüfsummen der Installationsdateien sind RDS Custom nicht bekannt.

  Vergewissern Sie sich, dass die SHA-256-Prüfsummen der bereitgestellten Dateien mit der SHA-256-Prüfsumme auf der Oracle-Website übereinstimmen. Wenn die Prüfsummen übereinstimmen, wenden Sie sich an AWS -Support und geben Sie den fehlgeschlagenen CEV-Namen, den Dateinamen und die Prüfsumme an.
- Die OPatch Version ist nicht mit Ihren Patch-Dateien kompatibel. Möglicherweise wird die folgende Meldung angezeigt: `OPatch is lower than minimum required version. Check that the version meets the requirements for all patches, and try again`. Um einen Oracle-Patch anzuwenden, müssen Sie eine kompatible Version des OPatch Dienstprogramms verwenden. Sie finden die erforderliche Version des OPatch-Dienstprogramms in der Readme-Datei für den Patch. Laden Sie das neueste OPatch Tool von My Oracle Support herunter, und versuchen Sie erneut, Ihr CEV zu erstellen.
- Die im CEV-Manifest angegebenen Patches sind in der falschen Reihenfolge.

Sie können RDS-Ereignisse entweder auf der RDS-Konsole (wählen Sie im Navigationsbereich Ereignisse aus) oder mithilfe des `describe-events` AWS CLI Befehls anzeigen. Die

Standardsitzungsdauer beträgt 60 Minuten. Wenn keine Ereignisse zurückgegeben werden, geben Sie eine längere Dauer an, wie im folgenden Beispiel gezeigt.

```
aws rds describe-events --duration 360
```

Derzeit ist der MediaImport Service, der Dateien aus Amazon S3 importiert, um sie zu erstellen, CEVs nicht integriert AWS CloudTrail. Wenn Sie also die Datenprotokollierung für Amazon RDS einschalten CloudTrail, werden Anrufe an den MediaImport Service, wie z. B. das `CreateCustomDbEngineVersion` Ereignis, nicht protokolliert.

Möglicherweise sehen Sie jedoch Aufrufe vom API-Gateway, das auf Ihren Amazon S3-Bucket zugreift. Diese Anrufe stammen vom MediaImport Service für das `CreateCustomDbEngineVersion` Ereignis.

Fehlerbehebung bei nicht unterstützten Konfigurationen in RDS Custom für Oracle

Im Modell der geteilten Verantwortung sind Sie dafür verantwortlich, Konfigurationsprobleme zu beheben, die Ihre DB-Instance von RDS Custom für Oracle in den Status `unsupported-configuration` versetzen. Wenn das Problem in der AWS Infrastruktur liegt, verwenden Sie die Konsole oder die, AWS CLI um es zu beheben. Wenn das Problem mit dem Betriebssystem oder der Datenbankkonfiguration zusammenhängt, melden Sie sich beim Host an, um das Problem zu beheben.

> **Note**
>
> In diesem Abschnitt wird erläutert, wie Sie nicht unterstützte Konfigurationen in RDS Custom für Oracle beheben. Weitere Informationen zu RDS Custom für SQL Server finden Sie unter Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server.

Die folgenden Tabellen enthalten Beschreibungen der Benachrichtigungen und Ereignisse, die vom Support-Perimeter gesendet werden, und deren Behebung. Diese Benachrichtigungen und der Support-Umfang können sich ändern. Hintergrundinformationen zum Support-Perimeter finden Sie unter Support-Perimeter in RDS Custom. Beschreibungen der Ereignisse finden Sie unter Amazon RDS-Ereigniskategorien und Ereignisnachrichten .

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-00000	Manuelle, nicht unterstüt zte Konfigura tion	Der Status der benutzerdefinierte n RDS-DB-In stance ist aus folgenden Gründen auf [Nicht unterstüt zte Konfiguration] gesetzt:. *reason*	Um dieses Problem zu beheben, erstellen Sie einen Support Fall.

AWS Ressourcen (Infrastruktur)

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O1001	Amazon Elastic Block Store (Amazon EBS)-Volu mes	Die folgenden EBS-Volumes wurden der Instanz hinzugefügt:. EC2 *ec2_id volume_id* Um das Problem zu beheben, trennen Sie die angegeben en Volumes von der Instance.	RDS Custom erstellt neben dem Root-Volume, das aus dem Amazon Machine Image (AMI) erstellt wurde, zwei Typen von EBS-Volumes und ordnet sie der EC2 Instance zu: • Das Binärvolume, auf dem sich die Binärdateien der Datenbanksoftware befinden • Die Datenvolumes, auf denen sich die Datenbank dateien befinden Wenn Sie Ihre DB-Instance erstellen, konfigurieren die von Ihnen angegebenen Speicherkonfigurationen die Datenvolumes. Der Support-Umfang überwacht Folgendes: • Die ersten EBS-Volumes, die mit der DB-Instan ce erstellt wurden, sind weiterhin der Instance zugeordnet.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion

			• Die anfänglichen EBS-Volumes haben immer noch die gleichen Konfigurationen wie ursprünglich festgelegt: Speichertyp, Größe, bereitgestellte IOPS und Speicherdurchsatz.
			• An die DB-Instance sind keine zusätzlichen EBS-Volumes angehängt.
			
			Verwenden Sie den folgenden CLI-Befehl, um den Volume-Typ der EBS-Volume-Details mit den Details der RDS Custom for Oracle-DB-Instance zu vergleichen:
			
			`aws rds describe-db-instances \`
			`--db-instance-identifier db-instance-name | grep StorageType`

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O1002	Amazon Elastic Block Store (Amazon EBS)-Volu mes	Das EBS-Volum e *volume_id* wurde von Instance [] getrennt. EC2 *ec2_id* Sie können das ursprüngliche Volume nicht von dieser Instance trennen. Um das Problem zu beheben, stellen Sie erneut *volume_id* eine Verbindung zu her. *ec2_id*	RDS Custom erstellt neben dem Root-Volume, das aus dem Amazon Machine Image (AMI) erstellt wurde, zwei Typen von EBS-Volumes und ordnet sie der EC2 Instance zu: • Das Binärvolume, auf dem sich die Binärdateien der Datenbanksoftware befinden • Die Datenvolumes, auf denen sich die Datenbank dateien befinden Wenn Sie Ihre DB-Instance erstellen, konfigurieren die von Ihnen angegebenen Speicherkonfigurationen die Datenvolumes. Der Support-Umfang überwacht Folgendes: • Die ersten EBS-Volumes, die mit der DB-Instan ce erstellt wurden, sind weiterhin der Instance zugeordnet. • Die anfänglichen EBS-Volumes haben immer noch die gleichen Konfigurationen wie ursprünglich festgelegt: Speichertyp, Größe, bereitgestellte IOPS und Speicherdurchsatz. • An die DB-Instance sind keine zusätzlichen EBS-Volumes angehängt. Verwenden Sie den folgenden CLI-Befehl, um den Volume-Typ der EBS-Volume-Details mit den Details der RDS Custom for Oracle-DB-Instance zu vergleich en: `aws rds describe-db-instances \`

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
			`--db-instance-identifier db-instance-name | grep StorageType`

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O1003	Amazon Elastic Block Store (Amazon EBS)-Volu mes	Das der EC2 Instance *volume_id* zugeordnete ursprüngliche EBS-Volume *ec2_id* wurde wie folgt geändert: Größe [*X*] auf [], Typ [*Y*] auf [*N*] oder IOPS [*M*] auf []. *J K* Um das Problem zu beheben, machen Sie die Änderung rückgängig.	RDS Custom erstellt neben dem Root-Volume, das aus dem Amazon Machine Image (AMI) erstellt wurde, zwei Typen von EBS-Volumes und ordnet sie der EC2 Instance zu: • Das Binärvolume, auf dem sich die Binärdateien der Datenbanksoftware befinden • Die Datenvolumes, auf denen sich die Datenbank dateien befinden Wenn Sie Ihre DB-Instance erstellen, konfigurieren die von Ihnen angegebenen Speicherkonfigurationen die Datenvolumes. Der Support-Umfang überwacht Folgendes: • Die ersten EBS-Volumes, die mit der DB-Instan ce erstellt wurden, sind weiterhin der Instance zugeordnet. • Die anfänglichen EBS-Volumes haben immer noch die gleichen Konfigurationen wie ursprünglich festgelegt: Speichertyp, Größe, bereitgestellte IOPS und Speicherdurchsatz. • An die DB-Instance sind keine zusätzlichen EBS-Volumes angehängt. Verwenden Sie den folgenden CLI-Befehl, um den Volume-Typ der EBS-Volume-Details mit den Details der RDS Custom for Oracle-DB-Instance zu vergleich en: `aws rds describe-db-instances \`

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
			`--db-instance-identifier db-instance-name | grep StorageType`
SP-O1004	Status der EC2 Amazon-Instanz	Bei der automatis ierten Wiederher stellung befand sich die EC2 Instance [*ec2_id*] in einem beeinträc htigten Zustand. Informationen zur Behebung des Problems finden Sie unter Behebung von Fehlern bei der Instanzwiederherst ellung.	Um den Status einer DB-Instance zu überprüfen, verwenden Sie die Konsole oder führen Sie den folgenden AWS CLI Befehl aus: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep DBInstanceStatus`
SP-O1005	EC2 Amazon-Instance-Attribute	EC2 Instanz [*ec2_id*] wurde wie folgt geändert: Attribut [*att1*] wurde von [*val-old*] auf [*val-new*] geändert, Attribut [*att2*] wurde von [*val-old*] auf [*val-new*] geändert. Um das Problem zu beheben, kehren Sie zum ursprüngl ichen Wert zurück.	

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O1006	Status der EC2 Amazon-Instanz	EC2 Instanz [*ec2_id*] wurde beendet oder kann nicht gefunden werden. Um das Problem zu beheben, löschen Sie die RDS Custom DB-Instan ce.	Der Support-Perimeter überwacht Benachrichtigungen über Statusänderungen der EC2 Instance. Die EC2 Instanz muss immer laufen. Um Ihre DB-Instance zu löschen 1. Um den Status einer DB-Instance zu überprüfen, verwenden Sie die Konsole oder führen Sie den folgenden AWS CLI Befehl aus: `aws rds describe-db-instances \ --db-instance-identifier db-instan ce-name |grep DBInstanceStatus` 2. Löschen Sie Ihre RDS Custom for Oracle DB-Instance.
SP-O1007	Status der EC2 Amazon-Instanz	EC2 Instanz [*ec2_id*] wurde gestoppt. Um das Problem zu beheben, starten Sie die Instanz.	Der Support-Perimeter überwacht Benachrichtigungen über Statusänderungen der EC2 Instanz. Die EC2 Instanz muss immer laufen. Um Ihre DB-Instance neu zu starten 1. Um den Status einer DB-Instance zu überprüfen, verwenden Sie die Konsole oder führen Sie den folgenden AWS CLI Befehl aus: `aws rds describe-db-instances \ --db-instance-identifier db-instan ce-name |grep DBInstanceStatus` 2. Starten Sie Ihre DB-Instance. 3. Stellen Sie die Binär- und Datenvolumen neu ein.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion

SP-10	Amazon SQS SQS-Geneh migung	Für Amazon SQS fehlen Berechtig ungen. Überprüfe n Sie die Berechtig ungen für das IAM-Instanzprofil, die VPC-Endpu nktrichtlinie und die abhängige n Dienstver bindungen, und versuchen Sie es dann erneut.	Sie können dieses Problem lösen, indem Sie sicherste llen, dass das dem Host zugeordnete IAM-Profil über die folgenden Berechtigungen verfügt:
			`"SQS:SendMessage"`
			`"SQS:ReceiveMessage"`
			`"SQS:DeleteMessage"`
			`"SQS:GetQueueUrl"`

SP-10	Amazon-Simple-Queue-Service (Amazon SQS)	Die SQS-Warte schlange [%s] wurde gelöscht und konnte nicht wiederhergestellt werden. Um dieses Problem zu beheben, erstellen Sie die Warteschl ange neu.	Erstellen Sie die Amazon SQS SQS-Warteschlange neu.

Betriebssystem

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O2001	Status des RDS-	Der RDS Custom Agent wird auf der EC2 Instanz	Bei RDS Custom für Oracle geht die DB-Instance außerhalb des Supportumfangs, wenn der RDS Custom Agent stoppt. Der Agent veröffentlicht die

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion

	Custom-Agenten	[] nicht ausgeführ t. *ec2_id* Stellen Sie sicher, dass der Agent auf [*ec2_id*] läuft.	`IamAlive` Metrik CloudWatch alle 30 Sekunden auf Amazon. Ein Alarm wird ausgelöst, wenn die Metrik 30 Sekunden lang nicht veröffentlicht wurde. Der Support-Umfang überwacht auch alle 30 Minuten den Prozessstatus des RDS Custom Agents auf dem Host.
			
			Um den RDS Custom Agent neu zu starten
			
			1. Melden Sie sich bei Ihrem Host an und stellen Sie sicher, dass der RDS-Custom-Agent ausgeführt wird.
			2. Führen Sie den folgenden Befehl aus, um den Status des Agenten zu ermitteln.
			
			`service rdscustomagent status`
			
			3. Verwenden Sie den folgenden Befehl, um den Agenten zu starten.
			
			`service rdscustomagent start`
			
			Wenn der RDS Custom Agent wieder läuft, wird die `IamAlive` Metrik auf Amazon CloudWatch veröffent licht und der Alarm wechselt in den `OK` Status. Dieser Switch informiert den Support-Umfang darüber, dass der Agent ausgeführt wird.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-02002	AWS Systems Manager Status des Agenten (SSM-Agent)	Der Systems Manager Manager-Agent auf EC2 Instanz [*ec2_id*] ist nicht erreichba r. Stellen Sie sicher, dass Sie die Netzwerk-, Agenten- und IAM-Berechtigungen korrekt konfiguriert haben.	Der SSM-Agent muss immer ausgeführt werden. Der RDS Custom Agent ist dafür verantwortlich, sicherzustellen, dass der Systems Manager-Agent ausgeführt wird. Wenn der SSM-Agent beendet und anschließend neu gestartet wurde, veröffentlicht der RDS Custom Agent eine Metrik für. CloudWatch Der RDS Custom Agent hat einen Alarm für die Metrik, die ausgelöst werden soll, wenn in jeder der letzten drei Minuten ein Neustart stattgefunden hat. Der Support-Perimeter überwacht außerdem alle 30 Minuten den Prozessstatus des SSM-Agenten auf dem Host. Weitere Informationen finden Sie unter Fehlerbeh ebung bei SSM-Agent.
SP-02003	AWS Systems Manager Status des Agenten (SSM-Agent)	Der Systems Manager Manager-Agent auf EC2 Instanz [*ec2_id*] ist mehrfach abgestürzt. Weitere Informationen finden Sie in der Dokumentation zur Fehlerbehebung für den SSM-Agenten.	Weitere Informationen finden Sie unter Fehlerbeh ebung bei SSM-Agent.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O 2004	Zeitzone des Betriebss ystems	Die Zeitzone auf EC2 Instanz [*ec2_id*] wurde geändert. Um dieses Problem zu beheben, setzen Sie die Zeitzone auf die vorherige Einstellung von [] *previous-time-zone* zurück. Verwenden Sie dann eine RDS-Optio nsgruppe, um die Zeitzone zu ändern.	Die RDS-Automatisierung hat festgestellt, dass die Zeitzone auf dem Host ohne Verwendung einer Optionsgruppe geändert wurde. Diese Änderung auf Hostebene kann zu Fehlern bei der RDS-Autom atisierung führen, sodass die EC2 Instance in den `unsupported-configuration` Status versetzt wird. Um die Zeitzoneneinstellung zu korrigieren 1. Melden Sie sich bei Ihrem EC2 Host an und überprüfen Sie die Zeitzone des Betriebssystems wie folgt: `timedatectl` 2. Pausieren Sie RDS Custom Automatisierung. Weitere Informationen finden Sie unter Ihre benutzerdefinierte DB-Instance pausieren und wieder aufnehmen RDS. 3. Stoppen Sie die DB-Instance. 4. Macht die Änderung der Zeitzone auf dem Betriebssystem rückgängig. 5. Starten Sie die &db;-Instance. 6. Fortsetzen Sie RDS Custom Automatisierung fort Ihre DB-Instance ist innerhalb von 30 Minuten verfügbar. Um zu verhindern, dass Sie sich in future außerhalb des Perimeters bewegen, ändern Sie Ihre Zeitzone über eine Optionsgruppe. Weitere Informati onen finden Sie unter Oracle-Zeitzone.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O 2005	sudo-Konfigur ationen	Den Sudo-Konf igurationen auf EC2 Instanz [*ec2_id*] fehlen die erforderlichen Berechtigungen. Um dieses Problem zu beheben, machen Sie die letzten Änderunge n an den Sudo-Konfigurationen rückgängig.	Der Support-Perimeter überprüft, ob bestimmte Betriebssystembenutzer bestimmte Befehle auf dem Host ausführen dürfen. Es überwacht `sudo` Konfigura tionen und vergleicht sie mit dem unterstützten Status. Wenn die `sudo` Konfigurationen nicht unterstützt werden, versucht RDS Custom, sie zu überschreiben und zum vorherigen unterstützten Status zurückzuk ehren. Wenn der Versuch erfolgreich ist, sendet RDS Custom die folgende Benachrichtigung: RDS Custom hat Ihre Konfiguration erfolgreich überschrieben. Wenn das Überschreiben nicht erfolgreich ist, bleibt Ihre DB-Instance im Konfigurationsstatus „Nicht unterstützt“. Um dieses Problem zu beheben, machen Sie entweder die Änderungen in der `sudoers.d/` Datei rückgängig oder korrigieren Sie die Berechtig ungen. Um Änderungen an den Konfigurationen zu untersuch en **sudo** 1. Loggen Sie sich bei Ihrem Host ein. 2. Führen Sie den folgenden Befehl aus. `visudo -c -f /etc/sudoers.d/ individua l_sudo_files` 3. Ändern Sie die `sudo` Konfigurationen nach Bedarf. Nachdem der Support-Perimeter festgestellt hat, dass die `sudo` Konfigurationen unterstützt werden, ist Ihre

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
			RDS Custom for Oracle DB-Instance innerhalb von 30 Minuten verfügbar.
SP-O 2006	Barrieref reiheit im S3-Bucket	RDS Custom Automation kann keine Dateien aus dem S3-Bucket auf EC2 Instanz [*ec2_id*] herunterl aden. Überprüfen Sie Ihre Netzwerkk onfiguration und stellen Sie sicher, dass die Instance Verbindungen zu und von S3 zulässt.	
SP-20	Version der Softwarel ösung mit hoher Verfügbar keit	Die HA-Lösung Ihrer Instanz unterscheidet sich von der erwartete n Version. Um dieses Problem zu beheben, erstellen Sie einen AWS -Support Fall.	Erstellen Sie einen AWS -Support Fall.

Datenbank

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O-300	Verzögeru ngsziel für	Der Parameter ARCHIVE_L	Der Support-Perimeter überwacht den `ARCHIVE_L AG_TARGET` Datenbankparameter, um sicherzus

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
	Datenbank archiv	AG_TARGET für EC2 Instanz [] liegt außerhalb des empfohlenen Bereichs. *ec2_id value_range* Um das Problem zu beheben, setzen Sie den Parameter auf einen Wert innerhalb von value_range.	tellen, dass der letzte wiederherstellbare Zeitpunkt der DB-Instance innerhalb angemessener Grenzen liegt. Um das Verzögerungsziel für archivierte Redo-Logs zu ändern 1. Loggen Sie sich bei Ihrem EC2 Host ein 2. Connect zu Ihrer RDS Custom for Oracle DB-Instan ce her 3. Ändern Sie den `ARCHIVE_LAG_TARGET` Parameter auf einen Wert zwischen 60 und 7200. Verwenden Sie beispielsweise die folgende SQL-Anweisung. `ALTER SYSTEM SET ARCHIVE_LAG_TARGET=300 SCOPE=BOTH;` Ihre DB-Instance ist innerhalb von 30 Minuten verfügbar.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion

SP-O3002	Oracle-Da ta-Guard-Rolle	Die Datenbankrolle [*role_name*] wird für Oracle Data Guard auf EC2 Instance [] nicht unterstützt. *ec2_id* Um das Problem zu beheben, setzen Sie den Parameter DATABASE_ ROLE entweder auf PRIMARY oder PHYSICAL STANDBY.	Der Support-Perimeter überwacht die aktuelle Datenbankrolle alle 15 Sekunden und sendet eine CloudWatch Benachrichtigung, wenn sich die Datenbankrolle geändert hat. Oracle Data Guard `DATABASE_ROLE` -Parameter muss entweder `PRIMARY` oder `PHYSICAL STANDBY` sein.
			
			Um Ihre Oracle Data Guard-Datenbankrolle auf einen unterstützten Wert zurückzusetzen
			
			1. Überprüfen Sie die Oracle Data Guard-Rolle, indem Sie die folgende Anweisung ausführen:
			
			`SELECT DATABASE_ROLE FROM V$DATABASE;`
			
			2. Wenn Ihre DB-Instance eigenständig ist, verwenden Sie eine der folgenden Anweisungen, um sie wieder in die `PRIMARY` Rolle umzuwandeln:
			
			`ALTER DATABASE COMMIT TO SWITCHOVER PRIMARY;`
			`ALTER DATABASE ACTIVATE STANDBY DATABASE;`
			
			Wenn es sich bei Ihrer DB-Instance um ein Replikat handelt, verwenden Sie die folgende Anweisung , um ihr wieder die `PHYSICAL STANDBY` Rolle zuzuweisen:
			
			`ALTER DATABASE CONVERT TO PHYSICAL STANDBY;`
			
			Nachdem der Support-Perimeter feststellt, dass die Datenbankrolle unterstützt wird, wird die DB-Instance von RDS Custom für Oracle innerhalb von 15 Sekunden verfügbar.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O3003	Datenbank -Zustand	Der SMON-Proz ess der Oracle-Da tenbank befindet sich in einem Zombie-Status. Um das Problem zu beheben, stellen Sie die Datenbank auf EC2 Instanz [*ec2_id*] manuell wieder her, öffnen Sie die Datenbank und erstellen Sie dann sofort eine Sicherungskopie. Wenn Sie weitere Hilfe benötigen, wenden Sie sich an Support.	Der Support-Umfang überwacht den Status der DB-Instance. Es überwacht auch, wie viele Neustarts während der vorherigen Stunde und des vorherigen Tages stattgefunden haben. Sie werden benachric htigt, wenn sich die Instanz in einem Zustand befindet, in dem sie noch existiert, aber Sie können nicht damit interagieren. Damit der Support-Perimeter den Status Ihrer Instanz auswertet 1. Melden Sie sich bei Ihrem Host an und ermitteln Sie den Datenbankstatus. `ps -eo pid,state,command | grep smon` 2. Falls erforderlich, starten Sie Ihre DB-Instance neu. Wenn der Neustart fehlschlägt, fahren Sie mit dem nächsten Schritt fort. 3. Falls erforderlich, starten Sie Ihren EC2 Host neu. Nach dem Neustart Ihrer DB-Instance erkennt der RDS Custom Agent, dass Ihre DB-Instance nicht mehr reagiert. Anschließend wird der Support-Perimeter benachrichtigt, damit der Status der DB-Instance neu bewertet wird.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion

SP-O3004	Datenbank -Protokol lmodus	Der Datenbank protokollmodus auf EC2 Instanz [*ec2_id*] wurde auf [] geändert. *value_b* Um das Problem zu beheben, setzen Sie den Protokollmodus auf [*value_a*].	Um den Protokollmodus Ihrer DB-Instance zu ändern **ARCHIVELOG**
			1. Melden Sie sich bei Ihrem EC2 Host an.
			2. Connect zu Ihrer Datenbank her und führen Sie die folgende Anweisung aus:
			`SELECT LOG_MODE FROM V$DATABASE;`
			Oder Sie können den folgenden Befehl in SQL*Plus ausführen:
			`ARCHIVE LOG LIST`
			3. Führen Sie den folgenden SQL*Plus-Befehl aus, um ein konsistentes Herunterfahren einzuleiten.
			`SHUTDOWN IMMEDIATE`
			Der RDS Custom Agent startet Ihre DB-Instance automatisch neu und setzt den Protokollmodus auf `ARCHIVELOG` Ihre DB-Instance ist innerhalb von 30 Minuten verfügbar.

Ereign ID	Konfigura tion	RDS-Ereignisnachri cht	Aktion
SP-O3005	Oracle-Ho mepfad	Das Oracle-Standardver zeichnis auf EC2 Instanz [*ec2_id*] wurde geändert in*new_path*. Um das Problem zu beheben, setzen Sie die Einstellung auf zurück*old_path*.	

SP-O3006	Eindeutig er Datenbank name	Der eindeutig e Datenbank name auf EC2 Instanz [*ec2_id*] wurde geändert in*new_value* . Um das Problem zu beheben, setzen Sie den Namen auf zurück*old_value* .	Um den eindeutigen Datenbanknamen für Ihre DB-Instance zu ändern
			1. Melden Sie sich bei Ihrem EC2 Host an.
			2. Connect der Datenbank her und führen Sie die folgende Anweisung aus:
			`SELECT DB_UNIQUE_NAME FROM V$DATABASE;`
			3. Geben Sie mit dem Befehl den eindeutigen Namen der ursprünglichen Datenbank an`ALTER SYSTEM SET DB_UNIQUE_NAME` .
			4. Führen Sie die folgende SQL-Anweisung aus, um ein konsistentes Herunterfahren einzuleiten.
			`SHUTDOWN IMMEDIATE;`
			Der RDS Custom Agent startet Ihre DB-Instance automatisch neu und setzt den Protokollmodus auf`ARCHIVELOG` . Ihre DB-Instance ist innerhalb von 30 Minuten verfügbar.

Fehlerbehebung bei Upgrades für RDS Custom for Oracle

Ihr Upgrade einer RDS Custom for Oracle-Instanz schlägt möglicherweise fehl. Im Folgenden finden Sie einige wichtige Techniken, Dateien und Befehle, die Sie bei Upgrades von RDS Custom DB für Oracle DB-Instances verwenden können:

- Untersuchen Sie die Upgrade-Ausgabeprotokolldateien im `/tmp`-Verzeichnis auf Ihrer DB-Instance. Die Namen der Protokolle hängen von Ihrer DB-Engine-Version ab. Es können beispielsweise Protokolle angezeigt werden, die die Zeichenfolgen `catupgrd` oder `catup` enthalten.
- Untersuchen Sie die Datei `alert.log` im Verzeichnis `/rdsdbdata/log/trace`.
- Führen Sie folgenden `grep`-Befehl im `root`-Verzeichnis zur Verfolgung des Upgrade-Betriebssystemprozesses aus. Dieser Befehl zeigt an, wo die Protokolldateien geschrieben werden, und bestimmt den Status des Upgrade-Prozesses.

  ```
  ps -aux | grep upg
  ```

  Das folgende Beispiel zeigt die Beispielausgabe.

  ```
  root     18884  0.0  0.0 235428  8172 ?        S<   17:03   0:00 /usr/bin/
  sudo -u rdsdb /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-
  UPGRADE/2.upgrade.sh
  rdsdb    18886  0.0  0.0 153968 12164 ?        S<   17:03   0:00 /usr/bin/perl -T -
  w /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-UPGRADE/2.upgrade.sh
  rdsdb    18887  0.0  0.0 113196  3032 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
  oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
  rdsdb    18900  0.0  0.0 113196  1812 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
  oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
  rdsdb    18901  0.1  0.0 167652 20620 ?        S<   17:03   0:07 /rdsdbbin/oracle/
  perl/bin/perl catctl.pl -n 4 -d /rdsdbbin/oracle/rdbms/admin -l /tmp catupgrd.sql
  root     29944  0.0  0.0 112724  2316 pts/0    S+   18:43   0:00 grep --color=auto
   upg
  ```

- Führen Sie die folgende SQL-Abfrage aus, um den aktuellen Status der Komponenten zu überprüfen, um die Datenbankversion und die auf der DB-Instance installierten Optionen zu finden.

  ```
  SET LINESIZE 180
  COLUMN COMP_ID FORMAT A15
  COLUMN COMP_NAME FORMAT A40 TRUNC
  COLUMN STATUS FORMAT A15 TRUNC
  ```

```
SELECT COMP_ID, COMP_NAME, VERSION, STATUS FROM DBA_REGISTRY ORDER BY 1;
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
COMP_NAME                                STATUS               PROCEDURE
---------------------------------------- --------------------
 ------------------------------------------------
Oracle Database Catalog Views            VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATALOG
Oracle Database Packages and Types       VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATPROC
Oracle Text                              VALID                VALIDATE_CONTEXT
Oracle XML Database                      VALID                DBMS_REGXDB.VALIDATEXDB

4 rows selected.
```

- Führen Sie die folgende SQL-Abfrage aus, um nach ungültigen Objekten zu suchen, die den Upgradeprozess beeinträchtigen könnten.

```
SET PAGES 1000 LINES 2000
COL OBJECT FOR A40
SELECT SUBSTR(OWNER,1,12) OWNER,
       SUBSTR(OBJECT_NAME,1,30) OBJECT,
       SUBSTR(OBJECT_TYPE,1,30) TYPE, STATUS,
       CREATED
FROM   DBA_OBJECTS
WHERE  STATUS <>'VALID'
AND    OWNER IN ('SYS','SYSTEM','RDSADMIN','XDB');
```

Behebung von Fehlerbehebung bei der Heraufstufung von Replikaten für RDS

Sie können verwaltete Oracle-Repliken in RDS Custom for Oracle mithilfe der Konsole, des `promote-read-replica` AWS CLI Befehls oder `PromoteReadReplica` der API hochstufen. Wenn Sie Ihre primäre DB-Instance löschen und alle Replikate fehlerfrei sind, stuft RDS Custom for Oracle Ihre verwalteten Replikate automatisch zu eigenständigen Instances herauf. Wenn ein Replikat die Automatisierung unterbrochen hat oder sich außerhalb des Supportumfangs befindet, müssen Sie das Replikat reparieren, bevor RDS Custom es automatisch heraufstufen kann. Weitere Informationen finden Sie unter Heraufstufen eines RDS Custom for Oracle-Replikats zu einer eigenständigen DB-Instance.

Der Workflow zur Replikat-Heraufstufung kann in der folgenden Situation hängen bleiben:

- Die primäre DB-Instance hat den Status `STORAGE_FULL`.
- Die Primärdatenbank kann nicht alle ihre Online-Redo-Logs archivieren.
- Zwischen den archivierten Redo-Log-Dateien auf dem Oracle-Replikat und der Primärdatenbank besteht eine Lücke.

Um auf den festgefahrenen Workflow zu reagieren

1. Synchronisieren Sie die Redo-Log-Lücke auf Ihrer Oracle Replica DB-Instance.
2. Erzwingen Sie die Heraufstufung Ihres gelesenen Replikats auf das neueste angewandte Redo-Protokoll. Führen Sie die folgenden Befehle in SQL*Plus aus:

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
SHUTDOWN IMMEDIATE
STARTUP
```

3. Kontaktieren Sie sie Support und bitten Sie sie, Ihre DB-Instance in `available` den Status zu versetzen.

Bekannte Probleme für Amazon RDS Custom for Oracle

Beachten Sie bei der Arbeit mit RDS Custom for Oracle die folgenden Probleme für DB-Instances:

- Die Größenänderung der Root- oder DBBIN-Volumes wird nicht unterstützt.

  > **Warning**
  >
  > Wir empfehlen dringend, die Größe der Root- oder DBBIN-Volumes nicht manuell zu ändern. Wir empfehlen, dass Sie alle Konfigurationen auf dem Datenvolume speichern, das nach dem Patchen erhalten bleibt, und dass Sie die Größe des Volumes nur mithilfe des Skalenspeichers ändern. RDS API

- Einige RDS APIs können blockiert werden, wenn sich eine Datenbank-Instance auf einer älteren Version befindetAMI, z. B. auf einer, AMI die Oracle Linux 7 verwendet. Um dieses Problem zu beheben, patchen Sie Ihre DB-Instance AMI mithilfe von Betriebssystem-Patching auf den neuesten Stand. Weitere Informationen finden Sie unter CEVUpgrade-Optionen.
- Bevor Sie RDS Operationen ausführen, stellen Sie sicher, dass Sie AWS-Konto über genügend Datenverarbeitungs- und Speicherkontingent verfügen.
- Wenn sich die Datenbank im Erstellungsstatus befindet und Sie sich aktiv bei der Datenbank oder beim EC2 Amazon-Host anmelden und Befehle ausführen, wird die Datenbankerstellung möglicherweise nicht abgeschlossen.
- Das Multiplexing von Kontrolldateien wird derzeit aufgrund eines Read Replica-Problems nicht unterstützt. Bevor Sie ein Read Replica erstellen, stellen Sie sicher, dass Sie im `CONTROL_FILES` Initialisierungsparameter der Quelldatenbank nur einen Dateinamen angeben.
- Sie können den Datenbankmodus nicht von `PHYSICAL STANDBY` (bereitgestellt oder schreibgeschützt) auf `SNAPSHOT STANDBY` (Konvertierung in Lesen/Schreiben) ändern.
- Wenn eine Teil einer AWS Organisation mit einer Dienststeuerungsrichtlinie (SCP) AWS-Konto ist und die einen Bedingungsschlüssel SCP enthält, schlägt die Erstellung einer RDS Custom for Oracle-DB-Instance möglicherweise fehl und es wird folgender Fehler angezeigt:

```
You can't create the DB instance because of incompatible resources.
The IAM instance profile role [AWSRDSCustomInstanceRole1-us-east-1] is missing the
 following permissions:
EFFECT [Allow] on ACTION(S) [ssm:DescribeAssociation, ssm:DescribeDocument,
 ssm:GetConnectionStatus,
  ssm:GetDeployablePatchSnapshotForInstance, ssmmessages:OpenControlChannel,
 ssm:GetParameters,
```

```
 ssm:ListInstanceAssociations, ssm:PutConfigurePackageResult,
ssmmessages:CreateControlChannel,
 ssm:GetParameter, ssm:UpdateAssociationStatus, ssm:GetManifest,
ssmmessages:CreateDataChannel,
 ssm:PutInventory, ssm:UpdateInstanceInformation, ssm:DescribeInstanceInformation,
 ssmmessages:OpenDataChannel, ssm:GetDocument, ssm:ListAssociations,
ssm:PutComplianceItems,
 ssm:UpdateInstanceAssociationStatus] for RESOURCE(S) [], EFFECT [Allow] on
ACTION(S) [ec2messages:DeleteMessage,
 ec2messages:FailMessage, ec2messages:GetEndpoint, ec2messages:AcknowledgeMessage,
ec2messages:GetMessages,
 ec2messages:SendReply] for RESOURCE(S) [], EFFECT [Allow] on ACTION(S)
[logs:CreateLogStream,
 logs:DescribeLogStreams, logs:PutRetentionPolicy, logs:PutLogEvents]
```

Um dieses Problem zu beheben, erstellen Sie ein Ticket mit Support.

Bekannte Probleme mit Datenbankbenutzerkonten

Beachten Sie die folgenden Probleme:

- Entfernen Sie keine Datenbankbenutzerkonten, die mit der Zeichenfolge beginnen`RDS`, wie z. B. `RDSADMIN` und`RDS_DATAGUARD`. RDS Custom for Oracle verwendet das `RDS` Konto für die Automatisierung. Wenn Sie dieses Benutzerkonto entfernen, versetzt RDS Custom die Instanz in den Konfigurationsstatus „Nicht unterstützt“.
- Sie können den Master-Benutzernamen für Ihre RDS Custom for Oracle-DB-Instance nicht mit dem `ModifyDBInstance` API ändern.
- RDSCustom for Oracle rotiert die Anmeldeinformationen für Benutzerkonten auf allen DB-Instances. Weitere Informationen finden Sie unter Rotieren der Anmeldeinformationen von RDS Custom für Oracle für Compliance-Programme. Wenn Sie eine lokale Primär-/Standby-Konfiguration verwenden, kann sich die Rotation der Anmeldeinformationen auf die folgenden Ressourcen auswirken:
  - Manuell erstellter benutzerdefinierter Standby-Modus RDS für Oracle-Instanzen

    Um dieses Problem zu beheben, löschen Sie die manuellen Standby-Datenbanken und erstellen Sie dann mithilfe eines API Aufrufs eine Oracle-Read Replica. Verwalten Sie die Geheimnisse für die manuellen Standby-Datenbanken manuell, sodass sie mit der Quell-DB-Instance übereinstimmen.

- Manuell erstellte regionsübergreifende Read Replicas

  Um dieses Problem zu beheben, sollten Sie die Geheimnisse manuell so aufbewahren, dass sie mit der primären DB-Instance übereinstimmen.

Bekannte Probleme mit Parameter- und Konfigurationsdateien

- Sie müssen die `crontab` Datei nach der Skalierung von Berechnungen, Betriebssystem-Upgrades und anderen Vorgängen, bei denen RDS Custom das Root-Volume ersetzt, konfigurieren. Wir empfehlen Ihnen dringend, eine Sicherungskopie von zu erstellen`crontab`.
- Beachten Sie bei der Konfiguration der `listener.ora` Datei die folgenden Richtlinien:
  - Stellen Sie sicher, dass sich jeder Eintrag in der Datei in einer einzigen Zeile befindet. Dieser Ansatz vermeidet Probleme mit Einrückungen bei der Instanzerstellung.
  - Stellen Sie sicher, `GLOBAL_DBNAME` dass dies dem Wert von `SID_NAME` entspricht.
  - Stellen Sie sicher, dass der Wert für der Benennungskonvention `LISTENER` entsprichtL_*dbname*_001.
  - Stellen Sie sicher, dass die `listener.ora` Datei eine Verbindung zum Datenbanknamen unterhält. RDS Custom verwendet diese Verbindung, um den Datenbankstart zu überprüfen. Wenn Sie diese Datei falsch ändern, schlagen Operationen wie Skalierung oder Patchen möglicherweise fehl.

    Das folgende Beispiel zeigt eine`listener.ora`, die korrekt konfiguriert ist:

    ```
    ADR_BASE_L_ORCL_001=/rdsdbdata/log/
    USE_SID_AS_SERVICE_L_ORCL_001=ON
    SID_LIST_L_ORCL_001=(SID_LIST = (SID_DESC = (SID_NAME = ORCL)(GLOBAL_DBNAME = ORCL)
     (ORACLE_HOME = /rdsdbbin/oracle.19.custom.r1.EE.1)))
    SUBSCRIBE_FOR_NODE_DOWN_EVENT_L_ORCL_001=OFF
    L_ORCL_001=(DESCRIPTION_LIST = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT =
     XXXX)(HOST = x.x.x.x))) (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(PORT = XXXX)
    (HOST = 127.0.0.1))))
    ```

- Kommentare werden in einer Serverparameterdatei oder Initialisierungsparameterdatei nicht unterstützt.
- Sie müssen die folgenden Initialisierungsparameter in der Serverparameterdatei () `/rdsdbdata/config/oracle_pfile` deklarieren:
  - `MEMORY_MAX_TARGET`

- MEMORY_TARGET
- PGA_AGGREGATE_TARGET
- PROCESSES
- SGA_TARGET
- USE_LARGE_PAGES

Wenn die oben genannten Parameter nicht deklariert sind`/rdsdbdata/config/oracle_pfile`, schlagen die Erstellung von Read Replica und die Skalierung der Berechnung möglicherweise fehl.

- Sie können die symbolischen Links für Konfigurationsdateien wie die Serverparameterdatei, die Auditdateien, `listener.oratnsnames.ora`, oder `sqlnet.ora` nicht löschen. Sie können auch die Verzeichnisstruktur für diese Dateien nicht ändern. RDS Die benutzerdefinierte Automatisierung erwartet, dass diese Dateien in einer bestimmten Verzeichnisstruktur existieren.

  Verwenden Sie die folgende Syntax, um eine Serverparameterdatei aus einer Initialisierungsparameterdatei zu erstellen.

```
CREATE SPFILE='/rdsdbdata/admin/$ORACLE_SID/pfile/spfile$ORACLE_SID.ora'
    FROM PFILE='/rdsdbdata/config/oracle_pfile';
```

Mit RDS Custom for SQL Server arbeiten

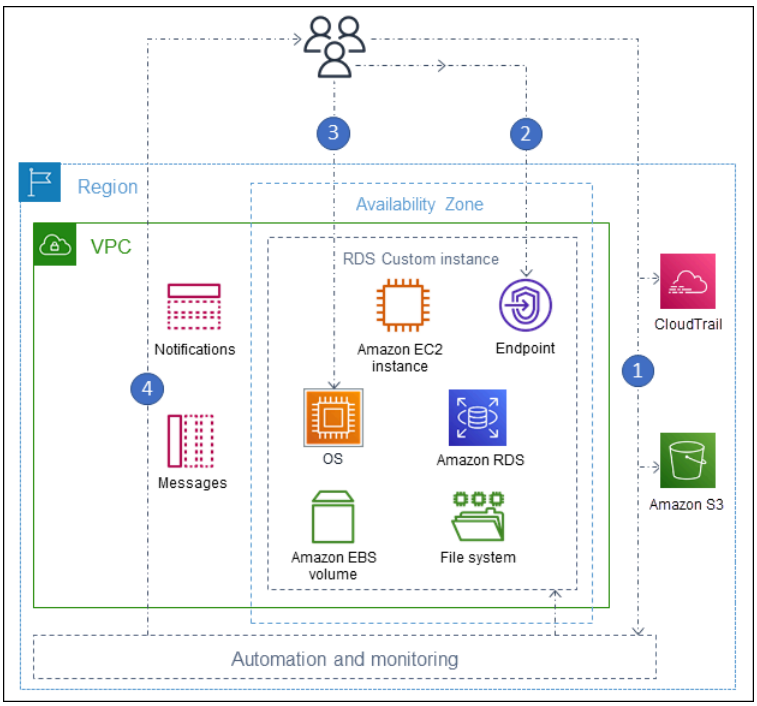
Im Folgenden finden Sie Anweisungen zum Erstellen, Verwalten und Verwalten Ihrer RDS Custom for SQL Server-DB-Instances.

Themen

- RDSBenutzerdefinierter Arbeitsablauf für SQL Server
- Anforderungen und Einschränkungen für Amazon RDS Custom for SQL Server
- Einrichten Ihrer Umgebung für Amazon RDS Custom for SQL Server
- Bringen Sie Ihre eigenen Medien mit RDS Custom for SQL Server
- Arbeiten mit benutzerdefinierten Engine-Versionen für RDS Custom für SQL Server
- Eine DB-Instance für Amazon RDS Custom for SQL Server erstellen und eine Verbindung zu ihr herstellen
- Verwaltung einer Amazon RDS Custom for SQL Server-DB-Instance
- Arbeiten mit Microsoft Active Directory mit RDS Custom for SQL Server
- Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Server SQL
- Sicherung und Wiederherstellung einer Amazon RDS Custom for SQL Server-DB-Instance
- Kopieren eines Amazon RDS Custom for SQL Server-DB-Snapshots
- Migration einer lokalen Datenbank zu Amazon RDS Custom for SQL Server
- Upgrade einer Amazon RDS Custom for SQL Server-DB-Instance
- Beheben von DB-Problemen für Amazon RDS Custom for SDL Server

RDSBenutzerdefinierter Arbeitsablauf für SQL Server

Das folgende Diagramm zeigt den typischen Arbeitsablauf für RDS Custom for SQL Server.



Die Schritte sind wie folgt:

1. Erstellen Sie eine RDS benutzerdefinierte SQL Server-DB-Instance aus einer Engine-Version, die von RDS Custom angeboten wird.

   Weitere Informationen finden Sie unter Eine RDS benutzerdefinierte SQL Server-DB-Instance erstellen.
2. Connect Ihre Anwendung mit dem RDS Custom DB-Instance-Endpunkt.

   Weitere Informationen erhalten Sie unter Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit AWS Systems Manager und Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit RDP.
3. (Optional) Greifen Sie auf den Host zu, um Ihre Software anzupassen.
4. Überwachen Sie Benachrichtigungen und Nachrichten, die von RDS Custom Automation generiert wurden.

Eine DB-Instance für RDS Custom erstellen

Sie erstellen Ihre RDS benutzerdefinierte DB-Instance mit dem `create-db-instance` Befehl. Das Verfahren ähnelt dem Erstellen einer RDS Amazon-Instance. Einige Parameter unterscheiden sich

jedoch. Weitere Informationen finden Sie unter Eine DB-Instance für Amazon RDS Custom for SQL Server erstellen und eine Verbindung zu ihr herstellen.

Datenbankverbindungen

Wie eine RDS Amazon-DB-Instance befindet sich Ihre RDS Custom for SQL Server-DB-Instance in einerVPC. Ihre Anwendung stellt über einen Client wie SQL Server Management Suite (SSMS) eine Verbindung zur RDS benutzerdefinierten Instance her, genau wie RDS bei SQL Server.

RDSBenutzerdefinierte Anpassung

Sie können auf den RDS benutzerdefinierten Host zugreifen, um Software zu installieren oder anzupassen. Um Konflikte zwischen Ihren Änderungen und der RDS benutzerdefinierten Automatisierung zu vermeiden, können Sie die Automatisierung für einen bestimmten Zeitraum unterbrechen. Während dieses Zeitraums führt RDS Custom keine Überwachung oder Instanzwiederherstellung durch. Am Ende des Zeitraums nimmt RDS Custom die vollständige Automatisierung wieder auf. Weitere Informationen finden Sie unter Anhalten und Wiederaufnehmen RDS der benutzerdefinierten Automatisierung.

Anforderungen und Einschränkungen für Amazon RDS Custom for SQL Server

Im Folgenden finden Sie eine Zusammenfassung der benutzerdefinierten Anforderungen und Einschränkungen von Amazon RDS Custom for SQL Server zur schnellen Referenz. Anforderungen und Einschränkungen erscheinen auch in den entsprechenden Abschnitten.

Themen

- Verfügbarkeit von Regionen und Versionen
- Allgemeine Anforderungen von RDS Custom for SQL Server
- Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server
- Einschränkungen RDS Custom for SQL Server
- Zeichensätze und Sortierungen für RDS benutzerdefinierte SQL Server-DB-Instances festlegen
- Lokale Zeitzone für RDS benutzerdefinierte SQL Server-DB-Instances
- Verwenden eines Service-Hauptschlüssels mit RDS Custom for SQL Server
- Unterstützung für Change Data Capture (CDC) mit RDS Custom for SQL Server

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Hinweise zur Versions- und Regionsverfügbarkeit von Amazon RDS with Amazon RDS Custom for SQL Server finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for SQL Server.

Allgemeine Anforderungen von RDS Custom for SQL Server

Befolgen Sie diese Anforderungen für Amazon RDS Custom for SQL Server:

- Verwenden Sie die Instance-Klassen, die in Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server angezeigt werden. Die einzigen unterstützten Speichertypen sind Solid-State-Laufwerke (SSD) der Typen gp2, gp3, io1 und io2 Block Express. Das maximale Speicherlimit für io1, gp2 und gp3 beträgt 16 TiB, während io2 64 TiB unterstützt.
- Stellen Sie sicher, dass Sie über einen symmetrischen AWS KMS Verschlüsselungsschlüssel verfügen, um eine benutzerdefinierte RDS-DB-Instance zu erstellen. Weitere Informationen finden Sie unter Stellen Sie sicher, dass Sie über einen symmetrischen AWS KMS Verschlüsselungsschlüssel verfügen.

- Stellen Sie sicher, dass Sie ein AWS Identity and Access Management (IAM-) Rollen- und Instanzprofil erstellen. Weitere Informationen erhalten Sie unter Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils und Automatisierte Erstellung von Instanzprofilen mit dem AWS Management Console.
- Stellen Sie sicher, dass Sie eine Netzwerkkonfiguration angeben, mit der RDS Custom auf andere AWS-Services zugreifen kann. Spezielle Anforderungen finden Sie unter Schritt 2: Konfiguration von Netzwerk, Instanzprofil und Verschlüsselung.
- Die kombinierte Anzahl von RDS Custom- und Amazon RDS DB-Instances darf Ihr Kontingentlimit nicht überschreiten. Wenn Ihr Kontingent beispielsweise 40 DB-Instanzen beträgt, können Sie 20 RDS Custom für SQL Server DB-Instanzen und 20 Amazon RDS DB-Instanzen haben.
- RDS Custom erstellt automatisch einen AWS CloudTrail Trail, dessen Name mit beginntdo-not-delete-rds-custom-. Der Umfang der Unterstützung von RDS Custom bestimmt anhand der Ereignisse von CloudTrail , ob sich Ihre Aktionen auf die Automatisierung von RDS Custom auswirken. RDS Custom erstellt den Trail, wenn Sie Ihre erste DB-Instance erstellen. Um ein bereits vorhandenes zu verwenden CloudTrail, wenden Sie sich an den AWS Support. Weitere Informationen finden Sie unter AWS CloudTrail.

Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server

Prüfen Sie mithilfe des Befehls describe-orderable-db-instance-options, ob die DB-Instance-Klasse in Ihrer Region unterstützt wird.

RDSCustom for SQL Server unterstützt die in der folgenden Tabelle aufgeführten DB-Instance-Klassen:

SQLServer-Edition	RDSBenutzerdefinierter Support
Enterprise Edition	db.r5.large–db.r5.24xlarge db.r5b.xlarge — db.r5b.24xlarge db.m5.large–db.m5.24xlarge db.r6i.xlarge — db.r6i.32xlarge db.m6i.xlarge — db.m6i.32xlarge

SQLServer-Edition	RDSBenutzerdefinierter Support
	db.x2iedn.xlarge–db.x2iedn.32xlarge
Standard Edition	db.r5.large–db.r5.24xlarge db.r5b.large–db.r5b.8xlarge db.m5.large–db.m5.24xlarge db.r6i.large — db.r6i.8xlarge db.m6i.large — db.m6i.8xlarge db.x2iedn.xlarge — db.x2iedn.8xlarge
Developer Edition	db.r5.large–db.r5.24xlarge db.r5b.xlarge — db.r5b.24xlarge db.m5.large–db.m5.24xlarge db.r6i.xlarge — db.r6i.32xlarge db.m6i.xlarge — db.m6i.32xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge
Web Edition	db.r5.large–db.r5.4xlarge db.m5.large–db.m5.4xlarge db.r6i.large–db.r6i.4xlarge db.m6i.large — db.m6i.4xlarge db.r5b.large–db.r5b.4xlarge

Die folgenden Empfehlungen gelten für db.x2iedn-Klassentypen:

- Bei der Erstellung ist der lokale Speicher ein unformatiertes und nicht zugewiesenes Gerät. Bevor Sie eine DB-Instance mit dieser Instance-Klasse verwenden, müssen Sie den lokalen Speicher mounten und formatieren. Konfigurieren Sie `tempdb` es anschließend, um eine optimale Leistung zu gewährleisten. Weitere Informationen finden Sie unter Optimieren der Tempdb-Leistung in Amazon RDS Custom for SQL Server mithilfe von lokalem Instance-Speicher.
- Wenn Sie DB-Instance-Operationen wie Scale Compute, Instance-Ersatz, Snapshot-Wiederherstellung oder point-in-time Wiederherstellung () ausführen, kehrt der lokale Speicher in seinen Rohzustand und den Status „Nicht zugewiesen“ zurück. PITR In diesen Situationen müssen Sie das Laufwerk erneut bereitstellen, neu formatieren und konfigurieren und die Funktionalität wiederherstellen. `tempdb`
- Für Multi-AZ-Instances empfehlen wir, die Konfiguration auf einer Standby-DB-Instance durchzuführen. Auf diese Weise kann das System bei einem Failover problemlos weiterbetrieben werden, da die Konfiguration bereits auf der Standby-Instance vorhanden ist.

Einschränkungen RDS Custom for SQL Server

Die folgenden Einschränkungen gelten für die Verwendung von MSDTC auf RDS for SQL Server:

- Sie können keine Lesereplikate in Amazon RDS for RDS Custom für SQL Server DB-Instances erstellen. Sie können Hochverfügbarkeit jedoch automatisch mit einer Multi-AZ-Bereitstellung konfigurieren. Weitere Informationen finden Sie unter Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Server SQL.
- Sie können die DB-Instance-ID einer vorhandenen RDS Custom for SQL Server-DB-Instance nicht ändern.
- Für eine RDS Custom for SQL Server-DB-Instance, die nicht mit einer Custom Engine Version (CEV) erstellt wurde, kann nicht garantiert werden, dass Änderungen am Microsoft Windows-Betriebssystem bestehen bleiben. Beispielsweise gehen diese Änderungen verloren, wenn Sie einen Snapshot- oder point-in-time Wiederherstellungsvorgang initiieren. Wenn die DB-Instance von RDS Custom für SQL Server mit einer CEV erstellt wurde, werden diese Änderungen beibehalten.
- Nicht alle Optionen werden unterstützt. Wenn Sie beispielsweise eine RDS-Custom for SQL Server DB-Instance erstellen, können Sie Folgendes nicht tun:
  - Ändern Sie die Anzahl der CPU-Kerne und -Threads pro Kern in der DB-Instance-Klasse.
  - Aktivieren Sie die Speicherskalierung.
  - Geben Sie Ihre eigene DB-Parametergruppe, Optionsgruppe oder Zeichensatz an.

  - Aktivieren Sie Performance Insights.
  - Einschalten von automatischen Nebenversions-Upgrades
- Der maximale DB-Instance-Speicher beträgt 64 TiB.
- Sie können RDS Proxy nicht mit RDS Custom for SQL Server verwenden.
- Sie können die `describe-reserved-db-instances` API für RDS Custom für SQL Server-DB-Instances nicht verwenden.

Zeichensätze und Sortierungen für RDS benutzerdefinierte SQL Server-DB-Instances festlegen

Übersicht

Mit RDS Benutzerdefiniert für SQL Server-DB-Instances können Sie den Zeichensatz und die Sortierungseinstellungen konfigurieren, die bestimmen, wie Daten gespeichert und sortiert werden. Zeichensätze definieren, welche Zeichen zulässig sind, während Sortierungen die Regeln für das Sortieren und Vergleichen von Daten festlegen. Es ist nicht wichtig, die geeigneten Zeichensätze und Sortierungen für Anwendungen festzulegen, die mit mehrsprachigen Daten arbeiten oder spezielle Sortieranforderungen haben. Beispielsweise müssen Sie möglicherweise mit Zeichen mit Akzenten umgehen und sprachspezifische Sortierregeln definieren oder die Datenintegrität über verschiedene Gebietsschemas hinweg sicherstellen. Die folgenden Abschnitte enthalten Informationen zur Zeichensatz- und Sortierungsunterstützung für Ihre RDS Custom for Server-DB-Instances. SQL

RDSCustom for SQL Server unterstützt eine Vielzahl von Serversortierungen, sowohl in traditioneller Kodierung als auch in der UTF -8-Kodierung, für die Gebietsschemas SQL _Latin, Japanisch, Deutsch und Arabisch. Die Standard-Serversortierung ist `SQL_Latin1_General_CP1_CI_AS`, Sie können jedoch eine andere unterstützte Sortierung auswählen, die verwendet werden soll. Sie können eine Sortierung mit demselben Verfahren auswählen, das for Server verwendet. RDS SQL Weitere Informationen finden Sie unter Verwaltung von Sortierungen und Zeichensätzen für Amazon RDS for Microsoft SQL Server.

Überlegungen

Bei der Arbeit mit Serversortierungen auf RDS Custom for SQL Server gelten die folgenden Anforderungen und Einschränkungen:

- Sie können die Serversortierung festlegen, wenn Sie eine RDS Custom for SQL Server-DB-Instance erstellen. Sie können die Sortierung auf Serverebene nicht ändern, nachdem die DB-Instance erstellt wurde.
- Sie können die Sortierung auf Serverebene bei der Wiederherstellung aus einem DB-Snapshot oder während einer Point-in-Time-Wiederherstellung () PITR nicht ändern.
- Wenn Sie eine DB-Instance aus einer RDS Custom for SQL Server-Instance erstellenCEV, erbt die DB-Instance nicht die Server-Sortierung von. CEV Stattdessen wird die Standard-Serversortierung von `SQL_Latin1_General_CP1_CI_AS` verwendet. Wenn Sie auf einer RDS Custom for Server eine nicht standardmäßige Serversortierung konfiguriert haben CEV und dieselbe SQL Serversortierung für eine neue DB-Instance verwenden möchten, achten Sie darauf, dieselbe Sortierung auszuwählen, wenn Sie die DB-Instance aus dem erstellen. CEV

  > **Note**
  >
  > Wenn sich die Sortierung, die Sie beim Erstellen der DB-Instance auswählen, von der Sortierung der unterscheidetCEV, werden die Microsoft SQL Server-Systemdatenbanken auf der neuen RDS Custom for SQL Server-DB-Instance neu erstellt, sodass sie die aktualisierte Sortierung verwenden. Der Neuaufbau wird nur für die neue RDS Custom for SQL Server-DB-Instance durchgeführt und hat keine Auswirkungen auf die DB-Instance selbst. CEV Alle vorherigen Änderungen, die Sie an den Systemdatenbanken auf der CEV vorgenommen haben, werden in der neuen RDS Custom for SQL Server-DB-Instance nicht beibehalten, sobald die Systemdatenbanken neu erstellt wurden. Zu einigen Änderungen gehören beispielsweise benutzerdefinierte Objekte in der `master` Datenbank, geplante Jobs in der `msdb` Datenbank oder Änderungen der Standard-Datenbankeinstellungen in der `model` Datenbank auf IhremCEV. Sie können Ihre Änderungen manuell neu erstellen, sobald die neue RDS Custom for SQL Server-DB-Instance erstellt wurde.

- Wenn Sie eine DB-Instance aus einer RDS Custom for SQL Server-Engine-Version (CEV) erstellen und eine andere Sortierung als die auswählen, stellen Sie sicherCEV, dass Ihr für die CEV Erstellung verwendetes Golden Image (AMI) die folgenden Anforderungen erfüllt, damit die Microsoft SQL Server-Systemdatenbanken auf der neuen DB-Instance neu erstellt werden können:
  - Stellen SQL Sie für Server 2022 sicher, dass sich die `setup.exe` Datei im folgenden Pfad befindet: `C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap \SQL2022\setup.exe`

  - Stellen SQL Sie für Server 2019 sicher, dass sich die `setup.exe` Datei im folgenden Pfad befindet: `C:\Program Files\Microsoft SQL Server\150\Setup Bootstrap \SQL2019\setup.exe`
  - Kopien der Daten- und Protokollvorlagen für die Datenbanken `master`, `model` und `msdb` müssen an ihren Standardspeicherorten vorhanden sein. Weitere Informationen finden Sie in der öffentlichen Dokumentation von Microsoft unter Neuerstellen von Systemdatenbanken.
  - Stellen Sie sicher, dass Ihre SQL Server Database Engine `NT Service\MSSQLSERVER` oder `NT AUTHORITY\NETWORK SERVICE` als Dienstkonto verwendet. Kein anderes Konto verfügt über die erforderlichen Berechtigungen auf dem `C:\`-Laufwerk, wenn eine nicht standardmäßige Serversortierung für die DB-Instance konfiguriert wird.
- Wenn die für eine neue DB-Instance ausgewählte Serversortierung mit der auf Ihrer konfigurierten übereinstimmtCEV, werden die Microsoft SQL Server-Systemdatenbanken auf der neuen RDS Custom for SQL Server-DB-Instance nicht neu erstellt. Alle vorherigen Änderungen, die Sie an den Systemdatenbanken auf der vorgenommen haben, CEV werden automatisch in der neuen RDS Custom for SQL Server-DB-Instance übernommen.

Unterstützte Kollationen

Sie können Ihre Sortierung auf einen der in der folgenden Tabelle gelisteten Werte einstellen.

Kollation	Beschreibung
Arabic_100_BIN	Arabic-100, binary sort
Arabic_100_BIN2	Arabic-100, binary code point comparison sort
Arabic_100_CI_AI	Arabic-100, case-insensitive, accent-insensitive , kanatype-insensitive, width-insensitive
Arabic_100_CI_AI_KS	Arabic-100, case-insensitive, accent-insensitive , kanatype-sensitive, width-insensitive
Arabic_100_CI_AI_KS_SC	Arabic-100, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters

Kollation	Beschreibung
Arabic_100_CI_AI_KS_SC_UTF8	Arabic-100, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CI_AI_KS_WS	Arabic-100, case-insensitive, accent-insensitive , kanatype-sensitive, width-sensitive
Arabic_100_CI_AI_KS_WS_SC	Arabic-100, case-insensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Arabic_100_CI_AI_KS_WS_SC_UTF8	Arabic-100, case-insensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Arabic_100_CI_AI_SC	Arabic-100, case-insensitive, accent-insensitive , kanatype-insensitive, width-insensitive, supplementary characters
Arabic_100_CI_AI_SC_UTF8	Arabic-100, case-insensitive, accent-insensitive , kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CI_AI_WS	Arabic-100, case-insensitive, accent-insensitive , kanatype-insensitive, width-sensitive
Arabic_100_CI_AI_WS_SC	Arabic-100, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Arabic_100_CI_AI_WS_SC_UTF8	Arabic-100, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Arabic_100_CI_AS	Arabic-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive

Kollation	Beschreibung
Arabic_100_CI_AS_KS	Arabic-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Arabic_100_CI_AS_KS_SC	Arabic-100, case-insensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters
Arabic_100_CI_AS_KS_SC_UTF8	Arabic-100, case-insensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CI_AS_KS_WS	Arabic-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Arabic_100_CI_AS_KS_WS_SC	Arabic-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplemen tary characters
Arabic_100_CI_AS_KS_WS_SC_UTF8	Arabic-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplemen tary characters, UTF8
Arabic_100_CI_AS_SC	Arabic-100, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive, supplementary characters
Arabic_100_CI_AS_SC_UTF8	Arabic-100, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CI_AS_WS	Arabic-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Arabic_100_CI_AS_WS_SC	Arabic-100, case-insensitive, accent-se nsitive, kanatype-insensitive, width-sensitive, supplementary characters

Kollation	Beschreibung
Arabic_100_CI_AS_WS_SC_UTF8	Arabic-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Arabic_100_CS_AI	Arabic-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Arabic_100_CS_AI_KS	Arabic-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Arabic_100_CS_AI_KS_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters
Arabic_100_CS_AI_KS_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CS_AI_KS_WS	Arabic-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Arabic_100_CS_AI_KS_WS_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters
Arabic_100_CS_AI_KS_WS_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Arabic_100_CS_AI_SC	Arabic-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters
Arabic_100_CS_AI_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8

Kollation	Beschreibung
Arabic_100_CS_AI_WS	Arabic-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Arabic_100_CS_AI_WS_SC	Arabic-100, case-sensitive, accent-insensitive , kanatype-insensitive, width-sensitive, supplementary characters
Arabic_100_CS_AI_WS_SC_UTF8	Arabic-100, case-sensitive, accent-insensitive , kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Arabic_100_CS_AS	Arabic-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Arabic_100_CS_AS_KS	Arabic-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Arabic_100_CS_AS_KS_SC	Arabic-100, case-sensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters
Arabic_100_CS_AS_KS_SC_UTF8	Arabic-100, case-sensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CS_AS_KS_WS	Arabic-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Arabic_100_CS_AS_KS_WS_SC	Arabic-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplemen tary characters
Arabic_100_CS_AS_KS_WS_SC_UTF8	Arabic-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplemen tary characters, UTF8

Kollation	Beschreibung
Arabic_100_CS_AS_SC	Arabic-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters
Arabic_100_CS_AS_SC_UTF8	Arabic-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Arabic_100_CS_AS_WS	Arabic-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Arabic_100_CS_AS_WS_SC	Arabic-100, case-sensitive, accent-se nsitive, kanatype-insensitive, width-sensitive, supplementary characters
Arabic_100_CS_AS_WS_SC_UTF8	Arabic-100, case-sensitive, accent-se nsitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Arabic_BIN	Arabic, binary sort
Arabic_BIN2	Arabic, binary code point comparison sort
Arabic_CI_AI	Arabic, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Arabic_CI_AI_KS	Arabic, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Arabic_CI_AI_KS_WS	Arabic, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Arabic_CI_AI_WS	Arabic, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Arabic_CI_AS	Arabic, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive

Kollation	Beschreibung
Arabic_CI_AS_KS	Arabic, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Arabic_CI_AS_KS_WS	Arabic, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Arabic_CI_AS_WS	Arabic, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Arabic_CS_AI	Arabic, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Arabic_CS_AI_KS	Arabic, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Arabic_CS_AI_KS_WS	Arabic, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Arabic_CS_AI_WS	Arabic, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Arabic_CS_AS	Arabic, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Arabic_CS_AS_KS	Arabic, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Arabic_CS_AS_KS_WS	Arabic, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Arabic_CS_AS_WS	Arabic, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Chinese_PRC_BIN2	Chinese-PRC, binary code point comparison sort
Chinese_PRC_CI_AS	Chinese-PRC, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive

Kollation	Beschreibung
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Danish_Norwegian_CI_AS	Danish-Norwegian, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Finnish_Swedish_CI_AS	Finnish-Swedish, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
French_CI_AS	French, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_100_BIN	German-PhoneBook-100, binary sort
German_PhoneBook_100_BIN2	German-PhoneBook-100, binary code point comparison sort
German_PhoneBook_100_CI_AI	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_100_CI_AI_KS	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
German_PhoneBook_100_CI_AI_KS_SC	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters
German_PhoneBook_100_CI_AI_KS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AI_KS_WS	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive

Kollation	Beschreibung
German_PhoneBook_100_CI_AI_KS_WS_SC	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters
German_PhoneBook_100_CI_AI_KS_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AI_SC	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters
German_PhoneBook_100_CI_AI_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AI_WS	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
German_PhoneBook_100_CI_AI_WS_SC	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters
German_PhoneBook_100_CI_AI_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AS	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_100_CI_AS_KS	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive

Kollation	Beschreibung
German_PhoneBook_100_CI_AS_KS_SC	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters
German_PhoneBook_100_CI_AS_KS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AS_KS_WS	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
German_PhoneBook_100_CI_AS_KS_WS_SC	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters
German_PhoneBook_100_CI_AS_KS_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AS_SC	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters
German_PhoneBook_100_CI_AS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CI_AS_WS	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
German_PhoneBook_100_CI_AS_WS_SC	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters

Kollation	Beschreibung
German_PhoneBook_100_CI_AS_WS_SC_UTF8	German-PhoneBook-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_100_CS_AI	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_100_CS_AI_KS	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
German_PhoneBook_100_CS_AI_KS_SC	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters
German_PhoneBook_100_CS_AI_KS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CS_AI_KS_WS	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
German_PhoneBook_100_CS_AI_KS_WS_SC	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters
German_PhoneBook_100_CS_AI_KS_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_100_CS_AI_SC	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters

Kollation	Beschreibung
German_PhoneBook_100_CS_AI_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CS_AI_WS	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
German_PhoneBook_100_CS_AI_WS_SC	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters
German_PhoneBook_100_CS_AI_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_100_CS_AS	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_100_CS_AS_KS	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
German_PhoneBook_100_CS_AS_KS_SC	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters
German_PhoneBook_100_CS_AS_KS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
German_PhoneBook_100_CS_AS_KS_WS	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive

Kollation	Beschreibung
German_PhoneBook_100_CS_AS_KS_WS_SC	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters
German_PhoneBook_100_CS_AS_KS_WS_SC_UTF8	German-PhoneBook-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
German_PhoneBook_BIN	German-PhoneBook, binary sort
German_PhoneBook_BIN2	German-PhoneBook, binary code point comparison sort
German_PhoneBook_CI_AI	German-PhoneBook, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_CI_AI_KS	German-PhoneBook, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
German_PhoneBook_CI_AI_KS_WS	German-PhoneBook, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
German_PhoneBook_CI_AI_WS	German-PhoneBook, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
German_PhoneBook_CI_AS	German-PhoneBook, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
German_PhoneBook_CI_AS_KS	German-PhoneBook, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
German_PhoneBook_CI_AS_KS_WS	German-PhoneBook, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive

Kollation	Beschreibung
German_PhoneBook_CI_AS_WS	German-PhoneBook, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
German_PhoneBook_CS_AI	German-PhoneBook, case-sensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive
German_PhoneBook_CS_AI_KS	German-PhoneBook, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive
German_PhoneBook_CS_AI_KS_WS	German-PhoneBook, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive
German_PhoneBook_CS_AI_WS	German-PhoneBook, case-sensitive, accent-in sensitive, kanatype-insensitive, width-sensitive
German_PhoneBook_CS_AS	German-PhoneBook, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
German_PhoneBook_CS_AS_KS	German-PhoneBook, case-sensitive, accent-se nsitive, kanatype-sensitive, width-insensitive
German_PhoneBook_CS_AS_KS_WS	German-PhoneBook, case-sensitive, accent-se nsitive, kanatype-sensitive, width-sensitive
German_PhoneBook_CS_AS_WS	German-PhoneBook, case-sensitive, accent-se nsitive, kanatype-insensitive, width-sensitive
Hebrew_BIN	Hebrew, binary sort
Hebrew_CI_AS	Hebrew, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_90_BIN	Japanese-90, binary sort
Japanese_90_BIN2	Japanese-90, binary code point comparison sort

Kollation	Beschreibung
Japanese_90_CI_AI	Japanese-90, case-insensitive, accent-in sensitive, kanatype-insensitive, width-ins ensitive
Japanese_90_CI_AI_KS	Japanese-90, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive
Japanese_90_CI_AI_KS_SC	Japanese-90, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_90_CI_AI_KS_SC_UTF8	Japanese-90, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_90_CI_AI_KS_WS	Japanese-90, case-insensitive, accent-in sensitive, kanatype-sensitive, width-sensitive
Japanese_90_CI_AI_KS_WS_SC	Japanese-90, case-insensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_90_CI_AI_KS_WS_SC_UTF8	Japanese-90, case-insensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CI_AI_SC	Japanese-90, case-insensitive, accent-in sensitive, kanatype-insensitive, width-ins ensitive, supplementary characters
Japanese_90_CI_AI_SC_UTF8	Japanese-90, case-insensitive, accent-in sensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, UTF8
Japanese_90_CI_AI_WS	Japanese-90, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive

Kollation	Beschreibung
Japanese_90_CI_AI_WS_SC	Japanese-90, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_90_CI_AI_WS_SC_UTF8	Japanese-90, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CI_AS	Japanese-90, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
Japanese_90_CI_AS_KS	Japanese-90, case-insensitive, accent-se nsitive, kanatype-sensitive, width-insensitive
Japanese_90_CI_AS_KS_SC	Japanese-90, case-insensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_90_CI_AS_KS_SC_UTF8	Japanese-90, case-insensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_90_CI_AS_KS_WS	Japanese-90, case-insensitive, accent-se nsitive, kanatype-sensitive, width-sensitive
Japanese_90_CI_AS_KS_WS_SC	Japanese-90, case-insensitive, accent-se nsitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_90_CI_AS_KS_WS_SC_UTF8	Japanese-90, case-insensitive, accent-se nsitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CI_AS_SC	Japanese-90, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive, supplementary characters

Kollation	Beschreibung
Japanese_90_CI_AS_SC_UTF8	Japanese-90, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Japanese_90_CI_AS_WS	Japanese-90, case-insensitive, accent-se nsitive, kanatype-insensitive, width-sensitive
Japanese_90_CI_AS_WS_SC	Japanese-90, case-insensitive, accent-se nsitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_90_CI_AS_WS_SC_UTF8	Japanese-90, case-insensitive, accent-se nsitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CS_AI	Japanese-90, case-sensitive, accent-insensitive , kanatype-insensitive, width-insensitive
Japanese_90_CS_AI_KS	Japanese-90, case-sensitive, accent-insensitive , kanatype-sensitive, width-insensitive
Japanese_90_CS_AI_KS_SC	Japanese-90, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_90_CS_AI_KS_SC_UTF8	Japanese-90, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_90_CS_AI_KS_WS	Japanese-90, case-sensitive, accent-insensitive , kanatype-sensitive, width-sensitive
Japanese_90_CS_AI_KS_WS_SC	Japanese-90, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters

Kollation	Beschreibung
Japanese_90_CS_AI_KS_WS_SC_UTF8	Japanese-90, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CS_AI_SC	Japanese-90, case-sensitive, accent-insensitive , kanatype-insensitive, width-insensitive, supplementary characters
Japanese_90_CS_AI_SC_UTF8	Japanese-90, case-sensitive, accent-insensitive , kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Japanese_90_CS_AI_WS	Japanese-90, case-sensitive, accent-insensitive , kanatype-insensitive, width-sensitive
Japanese_90_CS_AI_WS_SC	Japanese-90, case-sensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_90_CS_AI_WS_SC_UTF8	Japanese-90, case-sensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CS_AS	Japanese-90, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_90_CS_AS_KS	Japanese-90, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_90_CS_AS_KS_SC	Japanese-90, case-sensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_90_CS_AS_KS_SC_UTF8	Japanese-90, case-sensitive, accent-se nsitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8

Kollation	Beschreibung
Japanese_90_CS_AS_KS_WS	Japanese-90, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_90_CS_AS_KS_WS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_90_CS_AS_KS_WS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_90_CS_AS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters
Japanese_90_CS_AS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Japanese_90_CS_AS_WS	Japanese-90, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_90_CS_AS_WS_SC	Japanese-90, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_90_CS_AS_WS_SC_UTF8	Japanese-90, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_BIN	Japanese, binary sort
Japanese_BIN2	Japanese, binary code point comparison sort
Japanese_Bushu_Kakusu_100_BIN	Japanese-Bushu-Kakusu-100, binary sort
Japanese_Bushu_Kakusu_100_BIN2	Japanese-Bushu-Kakusu-100, binary code point comparison sort

Kollation	Beschreibung
Japanese_Bushu_Kakusu_100_CI_AI	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CI_AI_KS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CI_AI_KS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AI_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CI_AI_KS_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CI_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CI_AI_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_100_CI_AI_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CI_AI_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CI_AS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CI_AS_KS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CI_AS_KS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters

Kollation	Beschreibung
Japanese_Bushu_Kakusu_100_CI_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CI_AS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CI_AS_WS	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CI_AS_WS_SC	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CI_AS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CS_AI	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CS_AI_KS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CS_AI_KS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters

Kollation	Beschreibung
Japanese_Bushu_Kakusu_100_CS_AI_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CS_AI_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CS_AI_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CS_AI_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_100_CS_AS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CS_AS_KS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_Bushu_Kakusu_100_CS_AS_KS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_100_CS_AS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_100_CS_AS_WS	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_Bushu_Kakusu_100_CS_AS_WS_SC	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_Bushu_Kakusu_100_CS_AS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_Bushu_Kakusu_140_BIN	Japanese-Bushu-Kakusu-140, binary sort
Japanese_Bushu_Kakusu_140_BIN2	Japanese-Bushu-Kakusu-140, binary code point comparison sort
Japanese_Bushu_Kakusu_140_CI_AI	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CI_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplement ary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CI_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CI_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CI_AS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CI_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_WS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CI_AS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CI_AS_WS_VSS	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CI_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CS_AI	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kaku su-140, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AI_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8

Kollation	Beschreibung
Japanese_Bushu_Kakusu_140_CS_AS_WS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_Bushu_Kakusu_140_CS_AS_WS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_CI_AI	Japanese, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Japanese_CI_AI_KS	Japanese, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_CI_AI_KS_WS	Japanese, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_CI_AI_WS	Japanese, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Japanese_CI_AS	Japanese, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_CI_AS_KS	Japanese, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive

Kollation	Beschreibung
Japanese_CI_AS_KS_WS	Japanese, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_CI_AS_WS	Japanese, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_CS_AI	Japanese, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Japanese_CS_AI_KS	Japanese, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_CS_AI_KS_WS	Japanese, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_CS_AI_WS	Japanese, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Japanese_CS_AS	Japanese, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_CS_AS_KS	Japanese, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_CS_AS_KS_WS	Japanese, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_CS_AS_WS	Japanese, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_Unicode_BIN	Japanese-Unicode, binary sort
Japanese_Unicode_BIN2	Japanese-Unicode, binary code point comparison sort
Japanese_Unicode_CI_AI	Japanese-Unicode, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive

Kollation	Beschreibung
Japanese_Unicode_CI_AI_KS	Japanese-Unicode, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_Unicode_CI_AI_KS_WS	Japanese-Unicode, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_Unicode_CI_AI_WS	Japanese-Unicode, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Japanese_Unicode_CI_AS	Japanese-Unicode, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_Unicode_CI_AS_KS	Japanese-Unicode, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_Unicode_CI_AS_KS_WS	Japanese-Unicode, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_Unicode_CI_AS_WS	Japanese-Unicode, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_Unicode_CS_AI	Japanese-Unicode, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Japanese_Unicode_CS_AI_KS	Japanese-Unicode, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_Unicode_CS_AI_KS_WS	Japanese-Unicode, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_Unicode_CS_AI_WS	Japanese-Unicode, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
Japanese_Unicode_CS_AS	Japanese-Unicode, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive

Kollation	Beschreibung
Japanese_Unicode_CS_AS_KS	Japanese-Unicode, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_Unicode_CS_AS_KS_WS	Japanese-Unicode, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_Unicode_CS_AS_WS	Japanese-Unicode, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_XJIS_100_BIN	Japanese-XJIS-100, binary sort
Japanese_XJIS_100_BIN2	Japanese-XJIS-100, binary code point comparison sort
Japanese_XJIS_100_CI_AI	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Japanese_XJIS_100_CI_AI_KS	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
Japanese_XJIS_100_CI_AI_KS_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_XJIS_100_CI_AI_KS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_XJIS_100_CI_AI_KS_WS	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
Japanese_XJIS_100_CI_AI_KS_WS_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters

Kollation	Beschreibung
Japanese_XJIS_100_CI_AI_KS_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_XJIS_100_CI_AI_SC	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive, supplementary characters
Japanese_XJIS_100_CI_AI_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, UTF8
Japanese_XJIS_100_CI_AI_WS	Japanese-XJIS-100, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive
Japanese_XJIS_100_CI_AI_WS_SC	Japanese-XJIS-100, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_XJIS_100_CI_AI_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_XJIS_100_CI_AS	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-ins ensitive
Japanese_XJIS_100_CI_AS_KS	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_XJIS_100_CI_AS_KS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_XJIS_100_CI_AS_KS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8

Kollation	Beschreibung
Japanese_XJIS_100_CI_AS_KS_WS	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_XJIS_100_CI_AS_KS_WS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_XJIS_100_CI_AS_KS_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_XJIS_100_CI_AS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-ins ensitive, supplementary characters
Japanese_XJIS_100_CI_AS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, UTF8
Japanese_XJIS_100_CI_AS_WS	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_XJIS_100_CI_AS_WS_SC	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_XJIS_100_CI_AS_WS_SC_UTF8	Japanese-XJIS-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AI	Japanese-XJIS-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive
Japanese_XJIS_100_CS_AI_KS	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive

Kollation	Beschreibung
Japanese_XJIS_100_CS_AI_KS_SC	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_XJIS_100_CS_AI_KS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AI_KS_WS	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive
Japanese_XJIS_100_CS_AI_KS_WS_SC	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_XJIS_100_CS_AI_KS_WS_SC_UTF 8	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AI_SC	Japanese-XJIS-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive, supplementary characters
Japanese_XJIS_100_CS_AI_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AI_WS	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-insensitive, width-sensitive
Japanese_XJIS_100_CS_AI_WS_SC	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters
Japanese_XJIS_100_CS_AI_WS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, UTF8

Kollation	Beschreibung
Japanese_XJIS_100_CS_AS	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Japanese_XJIS_100_CS_AS_KS	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
Japanese_XJIS_100_CS_AS_KS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters
Japanese_XJIS_100_CS_AS_KS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AS_KS_WS	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive
Japanese_XJIS_100_CS_AS_KS_WS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters
Japanese_XJIS_100_CS_AS_KS_WS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AS_SC	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters
Japanese_XJIS_100_CS_AS_SC_UTF8	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Japanese_XJIS_100_CS_AS_WS	Japanese-XJIS-100, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
Japanese_XJIS_140_BIN	Japanese-XJIS-140, binary sort

Kollation	Beschreibung
Japanese_XJIS_140_BIN2	Japanese-XJIS-140, binary code point comparison sort
Japanese_XJIS_140_CI_AI	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CI_AI_KS	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CI_AI_KS_UTF8	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AI_KS_VSS	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CI_AI_KS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AI_KS_WS	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive

Kollation	Beschreibung
Japanese_XJIS_140_CI_AI_KS_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AI_KS_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CI_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AI_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AI_VSS	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CI_AI_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AI_WS	Japanese-XJIS-140, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive

Kollation	Beschreibung
Japanese_XJIS_140_CI_AI_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AI_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CI_AI_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-in sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CI_AS_KS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CI_AS_KS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AS_KS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive

Kollation	Beschreibung
Japanese_XJIS_140_CI_AS_KS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AS_KS_WS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CI_AS_KS_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AS_KS_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CI_AS_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive

Kollation	Beschreibung
Japanese_XJIS_140_CI_AS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CI_AS_WS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CI_AS_WS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CI_AS_WS_VSS	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CI_AS_WS_VSS_UTF8	Japanese-XJIS-140, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CS_AI	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AI_KS	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive

Kollation	Beschreibung
Japanese_XJIS_140_CS_AI_KS_UTF8	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CS_AI_KS_VSS	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AI_KS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CS_AI_KS_WS	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AI_KS_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CS_AI_KS_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-in sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8

Kollation	Beschreibung
Japanese_XJIS_140_CS_AI_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CS_AI_VSS	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AI_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CS_AI_WS	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AI_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CS_AI_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AI_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8

Kollation	Beschreibung
Japanese_XJIS_140_CS_AS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AS_KS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AS_KS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CS_AS_KS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AS_KS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CS_AS_KS_WS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AS_KS_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8

Kollation	Beschreibung
Japanese_XJIS_140_CS_AS_KS_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AS_KS_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CS_AS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector insensitive, UTF8
Japanese_XJIS_140_CS_AS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, variation selector sensitive, UTF8
Japanese_XJIS_140_CS_AS_WS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive
Japanese_XJIS_140_CS_AS_WS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector insensitive, UTF8

Kollation	Beschreibung
Japanese_XJIS_140_CS_AS_WS_VSS	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive
Japanese_XJIS_140_CS_AS_WS_VSS_UTF8	Japanese-XJIS-140, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive, supplementary characters, variation selector sensitive, UTF8
Korean_Wansung_CI_AS	Korean-Wansung, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_100_BIN	Latin1-General-100, binary sort
Latin1_General_100_BIN2	Latin1-General-100, binary code point comparison sort
Latin1_General_100_BIN2_UTF8	Latin1-General-100, binary code point comparison sort, UTF8
Latin1_General_100_CI_AS	Latin1-General-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive, supplementary characters, UTF8
Latin1_General_BIN	Latin1-General, binary sort
Latin1_General_BIN2	Latin1-General, binary code point comparison sort
Latin1_General_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive

Kollation	Beschreibung
Latin1_General_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
Latin1_General_CI_AS_KS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-sensitive, width-insensitive
Latin1_General_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
Modern_Spanish_CI_AS	Modern-Spanish, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
SQL_1xCompat_CP850_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 49 on Code Page 850 for non-Unicode Data
SQL_Latin1_General_CP1_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive for Unicode Data, SQL Server Sort Order 54 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_CP1_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 52 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_CP1_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 51 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_CP1250_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 82 on Code Page 1250 for non-Unicode Data

Kollation	Beschreibung
SQL_Latin1_General_CP1250_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 81 on Code Page 1250 for non-Unicode Data
SQL_Latin1_General_CP1251_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 106 on Code Page 1251 for non-Unicode Data
SQL_Latin1_General_CP1251_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 105 on Code Page 1251 for non-Unicode Data
SQL_Latin1_General_CP1253_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 124 on Code Page 1253 for non-Unicode Data
SQL_Latin1_General_CP1253_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 114 on Code Page 1253 for non-Unicode Data
SQL_Latin1_General_CP1253_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 113 on Code Page 1253 for non-Unicode Data
SQL_Latin1_General_CP1254_CI_AS	Turkish, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 130 on Code Page 1254 for non-Unicode Data

Kollation	Beschreibung
SQL_Latin1_General_CP1254_CS_AS	Turkish, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 129 on Code Page 1254 for non-Unicode Data
SQL_Latin1_General_CP1255_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 138 on Code Page 1255 for non-Unicode Data
SQL_Latin1_General_CP1255_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 137 on Code Page 1255 for non-Unicode Data
SQL_Latin1_General_CP1256_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 146 on Code Page 1256 for non-Unicode Data
SQL_Latin1_General_CP1256_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 145 on Code Page 1256 for non-Unicode Data
SQL_Latin1_General_CP1257_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 154 on Code Page 1257 for non-Unicode Data
SQL_Latin1_General_CP1257_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 153 on Code Page 1257 for non-Unicode Data

Kollation	Beschreibung
SQL_Latin1_General_CP437_BIN	Latin1-General, binary sort for Unicode Data, SQL Server Sort Order 30 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_CP437_BIN2	Latin1-General, binary code point comparison sort for Unicode Data, SQL Server Sort Order 30 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_CP437_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive for Unicode Data, SQL Server Sort Order 34 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_CP437_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 32 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_CP437_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 31 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_CP850_BIN	Latin1-General, binary sort for Unicode Data, SQL Server Sort Order 40 on Code Page 850 for non-Unicode Data
SQL_Latin1_General_CP850_BIN2	Latin1-General, binary code point comparison sort for Unicode Data, SQL Server Sort Order 40 on Code Page 850 for non-Unicode Data

Kollation	Beschreibung
SQL_Latin1_General_CP850_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 44 on Code Page 850 for non-Unicode Data
SQL_Latin1_General_CP850_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 42 on Code Page 850 for non-Unicode Data
SQL_Latin1_General_CP850_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 41 on Code Page 850 for non-Unicode Data
SQL_Latin1_General_Pref_CP1_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 53 on Code Page 1252 for non-Unicode Data
SQL_Latin1_General_Pref_CP437_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 33 on Code Page 437 for non-Unicode Data
SQL_Latin1_General_Pref_CP850_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 43 on Code Page 850 for non-Unicode Data
Thai_CI_AS	Thai, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive

Lokale Zeitzone für RDS benutzerdefinierte SQL Server-DB-Instances

Die Zeitzone einer RDS benutzerdefinierten SQL Server-DB-Instance ist standardmäßig festgelegt. Die aktuelle Standardeinstellung ist Coordinated Universal Time (UTC). Sie können die Zeitzone für Ihre DB-Instance stattdessen auf eine lokale Zeitzone einstellen, damit sie mit der Zeitzone Ihrer Anwendungen übereinstimmt.

Sie legen die Zeitzone bei der Erstellung Ihrer DB-Instance fest. Sie können Ihre DB-Instance erstellen AWS Management Console, indem Sie die Amazon RDS API reateDBInstanceC-Aktion oder den AWS CLI create-db-instanceBefehl verwenden.

Wenn Ihre DB-Instance Teil einer Multi-AZ-Bereitstellung ist, bleibt Ihre Zeitzone bei einem Failover auf die lokale Zeitzone eingestellt, die Sie festgelegt haben.

Wenn Sie eine point-in-time Wiederherstellung anfordern, geben Sie den Zeitpunkt für die Wiederherstellung an. Die Uhrzeit wird in Ihrer lokalen Zeitzone angezeigt. Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Folgende Beschränkungen gelten beim Festlegen der lokalen Zeitzone für Ihre DB-Instance:

- Sie können die Zeitzone für eine DB-Instance während der Instance-Erstellung konfigurieren, aber Sie können die Zeitzone einer vorhandenen RDS Custom for SQL Server-DB-Instance nicht ändern.
- Wenn die Zeitzone für eine bestehende RDS Custom for SQL Server-DB-Instance geändert wird, ändert RDS Custom den DB-Instance-Status auf `unsupported-configuration` und sendet Ereignisbenachrichtigungen.
- Sie können einen Snapshot aus einer DB-Instance in einer Zeitzone nicht in eine DB-Instance in einer anderen Zeitzone wiederherstellen.
- Es wird dringend davon abgeraten, eine Sicherungsdatei aus einer Zeitzone für eine andere Zeitzone wiederherzustellen. Wenn Sie eine Sicherungsdatei aus einer Zeitzone in einer anderen Zeitzone wiederherstellen, müssen Sie Ihre Abfragen und Anwendungen auf Auswirkungen durch die Zeitzonenänderung überprüfen. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Unterstützte Zeitzonen

Sie können Ihre lokale Zeitzone auf einen der in der folgenden Tabelle gelisteten Werte einstellen.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Afghanistan Standardzeit	(UTC+ 04:30)	Kabul	Diese Zeitzone berücksichtigt keine Sommerzeit.
Alaska Standardzeit	(UTC— 09:00)	Alaska	
Aleuten Normalzeit	(UTC— 10:00)	Aleuten-Inseln	
Altai Normalzeit	(UTC+ 07:00)	Barnaul, Gorno-Altaisk	
Arabische Normalzeit	(UTC+ 03:00)	Kuwait, Riad	Diese Zeitzone berücksichtigt keine Sommerzeit.
Arabische Standardzeit	(UTC+ 04:00)	Abu Dhabi, Muscat	
Arabische Normalzeit	(UTC+ 03:00)	Bagdad	Diese Zeitzone berücksichtigt keine Sommerzeit.
Argentinien Normalzeit	(UTC— 03:00)	Buenos Aires Stadt	Diese Zeitzone berücksichtigt keine Sommerzeit.
Astrachan Normalzeit	(UTC+ 04:00)	Astrachan, Uljanowsk	
Atlantik Standardzeit	(UTC— 04:00)	Atlantic Time (Kanada)	
AUSZentrale Standardzeit	(UTC+ 09:30)	Darwin	Diese Zeitzone berücksichtigt keine Sommerzeit.
Zentralaustralische Normalzeit	(UTC+ 08:45)	Eucla	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
AUSÖstliche Normalzeit	(UTC+ 10:00)	Canberra, Melbourne, Sydney	
Aserbaidschan Normalzeit	(UTC+ 04:00)	Baku	
Azoren Normalzeit	(UTC— 01:00)	Azoren	
Bahia Normalzeit	(UTC— 03:00)	Salvador	
Bangladesch Normalzeit	(UTC+ 06:00)	Dhaka	Diese Zeitzone berücksichtigt keine Sommerzeit.
Belarus Standardzeit	(UTC+ 03:00)	Minsk	Diese Zeitzone berücksichtigt keine Sommerzeit.
Bougainville Normalzeit	(UTC+ 11:00)	Bougainville-Insel	
Canada Central Standard Time	(UTC— 06:00)	Saskatchewan	Diese Zeitzone berücksichtigt keine Sommerzeit.
Kap Verde Standardzeit	(UTC— 01:00)	Kapverdische Inseln	Diese Zeitzone berücksichtigt keine Sommerzeit.
Kaukasus Normalzeit	(UTC+ 04:00)	Eriwan	
Cen. Australia Standard Time	(UTC+ 09:30)	Adelaide	
Mittelamerikanische Standardzeit	(UTC— 06:00)	Mittelamerika	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Zentralasiatische Standardzeit	(UTC+ 06:00)	Astana	Diese Zeitzone berücksichtigt keine Sommerzeit.
Central Brazilian Standard Time	(UTC— 04:00)	Cuiaba	
Mitteleuropäische Standardzeit	(UTC+ 01:00)	Belgrad, Bratislava, Budapest, Ljubljana, Prag	
Mitteleuropäische Standardzeit	(UTC+ 01:00)	Sarajevo, Skopje, Warschau, Zagreb	
Zentralpazifische Standardzeit	(UTC+ 11:00)	Salomon-Inseln, Neukaledonien	Diese Zeitzone berücksichtigt keine Sommerzeit.
Central Standard Time	(UTC— 06:00)	Central Time (USA und Kanada)	
Central Standard Time (Mexiko)	(UTC— 06:00)	Guadalajara, Mexiko-Stadt, Monterrey	
Chatham-Inseln Normalzeit	(UTC+ 12:45)	Chatham-Inseln	
China Standardzeit	(UTC+ 08:00)	Beijing, Chongqing, Hongkong, Urumqi	Diese Zeitzone berücksichtigt keine Sommerzeit.
Kuba Normalzeit	(UTC— 05:00)	Havanna	
Datumsgrenze, Normalzeit	(UTC— 12:00)	Internationale Datumsgrenze West	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
O. Afrikanische Standardzeit	(UTC+ 03:00)	Nairobi	Diese Zeitzone berücksichtigt keine Sommerzeit.
O. Australia Standard Time	(UTC+ 10:00)	Brisbane	Diese Zeitzone berücksichtigt keine Sommerzeit.
O. Europäische Standardzeit	(UTC+ 02:00)	Chisinau	
O. Südamerikanische Standardzeit	(UTC— 03:00)	Brasilia	
Osterinsel Normalzeit	(UTC— 06:00)	Osterinsel	
Ost Standardzeit	(UTC— 05:00)	Ostküstenzeit (USA und Kanada)	
Östliche Normalzeit (Mexiko)	(UTC— 05:00)	Chetumal	
Ägypten Normalzeit	(UTC+ 02:00)	Kairo	
Jekaterinburg Normalzeit	(UTC+ 05:00)	Jekaterinburg	
Fidschi Normalzeit	(UTC+ 12:00)	Fidschi	
FLEStandardzeit	(UTC+ 02:00)	Helsinki, Kiew, Riga, Sofia, Tallinn, Wilna	
Georgien Standardzeit	(UTC+ 04:00)	Tiflis	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
GMTStandardzeit	(UTC)	Dublin, Edinburgh, Lissabon, London	Diese Zeitzone ist nicht dieselbe wie Greenwich Mean Time. Diese Zeitzone berücksichtigt die Sommerzeit.
Grönland Standardzeit	(UTC— 03:00)	Grönland	
Greenwich Standardzeit	(UTC)	Monrovia, Reykjavik	Diese Zeitzone berücksichtigt keine Sommerzeit.
GTBStandardzeit	(UTC+ 02:00)	Athen, Bukarest	
Haiti Normalzeit	(UTC— 05:00)	Haiti	
Hawaii Standardzeit	(UTC— 10:00)	Hawaii	
Indien Standardzeit	(UTC+ 05:30)	Chennai, Kolkata, Mumbai, Neu-Delhi	Diese Zeitzone berücksichtigt keine Sommerzeit.
Iran Normalzeit	(UTC+ 03:30)	Teheran	
Israel Normalzeit	(UTC+ 02:00)	Jerusalem	
Jordanien Standardzeit	(UTC+ 02:00)	Amman	
Kaliningrad Normalzeit	(UTC+ 02:00)	Kaliningrad	
Kamtschatka Normalzeit	(UTC+ 12:00)	Petropawlowsk-Kamtschatski – Alt	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Korea Standardzeit	(UTC+ 09:00)	Seoul	Diese Zeitzone berücksichtigt keine Sommerzeit.
Libyen Normalzeit	(UTC+ 02:00)	Tripolis	
Linieninseln Normalzeit	(UTC+ 14:00)	Kiritimati-Insel	
Lord Howe Normalzeit	(UTC+ 10:30)	Lord-Howe-Insel	
Magadan Normalzeit	(UTC+ 11:00)	Magadan	Diese Zeitzone berücksichtigt keine Sommerzeit.
Magallan Normalzeit	(UTC— 03:00)	Punta Arenas	
Marquesas Normalzeit	(UTC— 09:30)	Marquesas-Inseln	
Mauritius Normalzeit	(UTC+ 04:00)	Port Louis	Diese Zeitzone berücksichtigt keine Sommerzeit.
Mittlerer Osten Standardzeit	(UTC+ 02:00)	Beirut	
Montevideo Normalzeit	(UTC— 03:00)	Montevideo	
Marokko Normalzeit	(UTC+ 01:00)	Casablanca	
Mountain Standard Time	(UTC— 07:00)	Mountain Time (USA und Kanada)	
Mountain Standard Time (Mexiko)	(UTC— 07:00)	Chihuahua, La Paz, Mazatlan	
Myanmar Normalzeit	(UTC+ 06:30)	Yangon (Rangun)	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
N. Zentralasiatische Standardzeit	(UTC+ 07:00)	Nowosibirsk	
Namibia Normalzeit	(UTC+ 02:00)	Windhuk	
Nepal Normalzeit	(UTC+ 05:45)	Kathmandu	Diese Zeitzone berücksichtigt keine Sommerzeit.
Neuseeland Standardzeit	(UTC+ 12:00)	Auckland, Wellington	
Neufundland Standardzeit	(UTC— 03:30)	Neufundland	
Norfolk Normalzeit	(UTC+ 11:00)	Norfolkinsel	
Ost-Nordasiatische Normalzeit	(UTC+ 08:00)	Irkutsk	
Nordasien Normalzeit	(UTC+ 07:00)	Krasnojarsk	
Nordkorea Normalzeit	(UTC+ 09:00)	Pjöngjang	
Omsk Normalzeit	(UTC+ 06:00)	Omsk	
Pacific SA Standard Time	(UTC— 03:00)	Santiago	
Pacific Standard Time	(UTC— 08:00)	Pacific Time (USA und Kanada)	
Pacific Standard Time (Mexiko)	(UTC— 08:00)	Baja California	
Pakistan Normalzeit	(UTC+ 05:00)	Islamabad, Karatschi	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Paraguay Normalzeit	(UTC— 04:00)	Asunción	
Romanische Normalzeit	(UTC+ 01:00)	Brüssel, Kopenhagen, Madrid, Paris	
Russland Zeitzone 10	(UTC+ 11:00)	Tschokurdach	
Russland Zeitzone 11	(UTC+ 12:00)	Anadyr, Petropawlowsk-Kamtschatski	
Russland Zeitzone 3	(UTC+ 04:00)	Ischewsk, Samara	
Russische Standardzeit	(UTC+ 03:00)	Moskau, St. Petersburg, Wolgograd	Diese Zeitzone berücksichtigt keine Sommerzeit.
Östl. Südamerika Normalzeit	(UTC— 03:00)	Cayenne, Fortaleza	Diese Zeitzone berücksichtigt keine Sommerzeit.
SA Pacific Standard Time	(UTC— 05:00)	Bogota, Lima, Quito, Rio Branco	Diese Zeitzone berücksichtigt keine Sommerzeit.
Mittl. Südamerika Normalzeit	(UTC— 04:00)	Georgetown, La Paz, Manaus, San Juan	Diese Zeitzone berücksichtigt keine Sommerzeit.
Saint Pierre Normalzeit	(UTC— 03:00)	St. Pierre und Miquelon	
Sachalin Normalzeit	(UTC+ 11:00)	Sachalin	
Samoa Normalzeit	(UTC+ 13:00)	Samoa	
São Tomé Normalzeit	(UTC+ 01:00)	São Tomé	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Saratow Normalzeit	(UTC+ 04:00)	Saratow	
Südostasiatische Standardzeit	(UTC+ 07:00)	Bangkok, Hanoi, Jakarta	Diese Zeitzone berücksichtigt keine Sommerzeit.
Singapur Standardzeit	(UTC+ 08:00)	Kuala Lumpur, Singapur	Diese Zeitzone berücksichtigt keine Sommerzeit.
Südafrika Normalzeit	(UTC+ 02:00)	Harare, Pretoria	Diese Zeitzone berücksichtigt keine Sommerzeit.
Sri Lanka Normalzeit	(UTC+ 05:30)	Sri Jayawarde nepura	Diese Zeitzone berücksichtigt keine Sommerzeit.
Sudan Normalzeit	(UTC+ 02:00)	Khartum	
Syrien Normalzeit	(UTC+ 02:00)	Damaskus	
Taipei Normalzeit	(UTC+ 08:00)	Taipeh	Diese Zeitzone berücksichtigt keine Sommerzeit.
Tasmanien Normalzeit	(UTC+ 10:00)	Hobart	
Tocantins Normalzeit	(UTC— 03:00)	Araguaina	
Japanische Standardzeit	(UTC+ 09:00)	Osaka, Sapporo, Tokio	Diese Zeitzone berücksichtigt keine Sommerzeit.
Tomsk Normalzeit	(UTC+ 07:00)	Tomsk	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Tonga Normalzeit	(UTC+ 13:00)	Nuku'alofa	Diese Zeitzone berücksichtigt keine Sommerzeit.
Transbaikal Normalzeit	(UTC+ 09:00)	Tschita	
Türkei Normalzeit	(UTC+ 03:00)	Istanbul	
Turks- und Caicosinseln Normalzeit	(UTC— 05:00)	Turks- und Caicosinseln	
Ulan-Bator Normalzeit	(UTC+ 08:00)	Ulan-Bator	Diese Zeitzone berücksichtigt keine Sommerzeit.
US Eastern Standard Time	(UTC— 05:00)	Indiana (Osten)	
US Mountain Standard Time	(UTC— 07:00)	Arizona	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC	UTC	Coordinated Universal Time	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC—02	(UTC— 02:00)	Coordinated Universal Time–02	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC—08	(UTC— 08:00 Uhr)	Coordinated Universal Time–08	
UTC—09	(UTC— 09:00 Uhr)	Coordinated Universal Time–09	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
UTC—11	(UTC— 11:00 Uhr)	Coordinated Universal Time–11	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC+12	(UTC+ 12:00)	Coordinated Universal Time+12	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC+13	(UTC+ 13:00)	Koordinierte Weltzeit+13	
Venezuela Normalzeit	(UTC— 04:00)	Caracas	Diese Zeitzone berücksichtigt keine Sommerzeit.
Wladiwostok Normalzeit	(UTC+ 10:00)	Wladiwostok	
Wolgograd Normalzeit	(UTC+ 04:00)	Wolgograd	
W. Australia Standard Time	(UTC+ 08:00)	Perth	Diese Zeitzone berücksichtigt keine Sommerzeit.
W. Zentralafrikanische Standardzeit	(UTC+ 01:00)	West-Zentralafrika	Diese Zeitzone berücksichtigt keine Sommerzeit.
W. Europäische Standardzeit	(UTC+ 01:00)	Amsterdam, Berlin, Bern, Rom, Stockholm, Wien	
W. Mongolei Normalzeit	(UTC+ 07:00)	Hovd	
Westasien Normalzeit	(UTC+ 05:00)	Aschgabat, Taschkent	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Westjordanland Normalzeit	(UTC+ 02:00)	Gaza, Hebron	
Westpazifische Normalzeit	(UTC+ 10:00)	Guam, Port Moresby	Diese Zeitzone berücksichtigt keine Sommerzeit.
Jakutsk Normalzeit	(UTC+ 09:00)	Jakutsk	

Verwenden eines Service-Hauptschlüssels mit RDS Custom for SQL Server

RDSCustom for SQL Server unterstützt die Verwendung eines Service Master Keys (SMK). RDSBenutzerdefiniert behält diesen Wert SMK während der gesamten Lebensdauer Ihrer RDS Custom for SQL Server-DB-Instance bei. Wenn Sie dasselbe beibehaltenSMK, kann Ihre DB-Instance Objekte verwenden, die mit dem verschlüsselt sindSMK, wie z. B. Kennwörter und Anmeldeinformationen für verknüpfte Server. Wenn Sie eine Multi-AZ-Bereitstellung verwenden, synchronisiert und verwaltet RDS Custom auch die Daten SMK zwischen primären und sekundären DB-Instances.

Themen

- Verfügbarkeit von Regionen und Versionen
- Unterstützte Features
- Verwenden TDE
- Funktionen konfigurieren
- Anforderungen und Einschränkungen

Verfügbarkeit von Regionen und Versionen

Die Verwendung von SMK wird in allen Regionen unterstützt, in denen RDS Custom for SQL Server verfügbar ist, und zwar für alle SQL Serverversionen, die auf RDS Custom verfügbar sind. Weitere Informationen zur Version und regionalen Verfügbarkeit von Amazon RDS with RDS Custom for SQL Server finden Sie unterUnterstützte Regionen und DB-Engines für RDS Custom for SQL Server.

Unterstützte Features

Bei Verwendung von RDS Custom for SQL Server werden die folgenden Funktionen unterstützt: SMK

- Transparente Datenverschlüsselung (TDE)
- Verschlüsselung auf Spaltenebene
- Datenbank-E-Mail
- Verknüpfte Server
- SQLServerintegrationsdienste (SSIS)

Verwenden TDE

An SMK ermöglicht die Konfiguration der transparenten Datenverschlüsselung (TDE), die Daten verschlüsselt, bevor sie in den Speicher geschrieben werden, und Daten automatisch entschlüsselt, wenn die Daten aus dem Speicher gelesen werden. Im Gegensatz zu RDS SQL Server erfordert die Konfiguration TDE auf einer RDS Custom for SQL Server-DB-Instance keine Verwendung von Optionsgruppen. Sobald Sie ein Zertifikat und einen Datenbankverschlüsselungsschlüssel erstellt haben, können Sie stattdessen den folgenden Befehl ausführen, um die Aktivierung auf TDE Datenbankebene vorzunehmen:

```
ALTER DATABASE [myDatabase] SET ENCRYPTION ON;
```

Weitere Informationen zur Verwendung von TDE with RDS for SQL Server finden Sie unterSupport für transparente Datenverschlüsselung im SQL Server.

Ausführliche Informationen zu Microsoft SQL Server finden Sie TDE in der Microsoft-Dokumentation unter Transparente Datenverschlüsselung.

Funktionen konfigurieren

Ausführliche Schritte zur Konfiguration von Funktionen, die a SMK mit RDS Custom for SQL Server verwenden, finden Sie in den folgenden Beiträgen im RDS Amazon-Datenbank-Blog:

- Verknüpfte Server: Konfiguration von Verbindungsservern auf RDS Custom for SQL Server.
- SSIS: Migrieren Sie SSIS Pakete zu RDS Custom for SQL Server.
- TDE: Schützen Sie Ihre Daten mit TDE RDS Custom for SQL Server.

Anforderungen und Einschränkungen

Beachten Sie bei der Verwendung einer DB-Instance SMK mit einer RDS Custom for SQL Server-DB-Instance die folgenden Anforderungen und Einschränkungen:

- Wenn Sie die SMK auf Ihrer DB-Instance neu generieren, sollten Sie sofort einen manuellen DB-Snapshot durchführen. Wir empfehlen, die Neugenerierung nach Möglichkeit zu vermeidenSMK.
- Sie müssen Backups der Serverzertifikate und der Datenbank-Hauptschlüsselkennwörter aufbewahren. Wenn Sie diese nicht sichern, kann dies zu Datenverlust führen.
- Wenn Sie eine Konfiguration SSIS vornehmen, sollten Sie ein SSM Dokument verwenden, um die RDS Custom for SQL Server-DB-Instance mit der Domäne zu verbinden, falls eine Rechenleistung oder ein Host-Ersatz erforderlich ist.
- Wenn TDE die Spaltenverschlüsselung aktiviert ist, werden Datenbank-Backups automatisch verschlüsselt. Wenn Sie eine Snapshot-Wiederherstellung oder eine Point-in-Time-Wiederherstellung durchführen, wird die DB-Instance SMK aus der Quelle wiederhergestellt, um die Daten für die Wiederherstellung zu entschlüsseln, und es SMK wird eine neue generiert, um Daten auf der wiederhergestellten Instance erneut zu verschlüsseln.

Weitere Informationen zu Service Master Keys in Microsoft SQL Server finden Sie in der Microsoft-Dokumentation unter SQLServer- und Datenbankverschlüsselungsschlüssel.

Unterstützung für Change Data Capture (CDC) mit RDS Custom for SQL Server

Übersicht

RDS Custom for SQL Server bietet native Unterstützung für Change Data Capture (CDC), sodass Sie Datenänderungen in Ihren SQL Server-Tabellen verfolgen und erfassen können. CDC speichert detaillierte Metadaten zu diesen Änderungen für den späteren Abruf und die Analyse. Ausführliche Informationen zur CDC-Funktionalität finden Sie in der Microsoft-Dokumentation unter Change Data Capture.

Der CDC-Vorgang in SQL Server erfordert übereinstimmende Werte zwischen dem lokalen Server (der den Wert `server_id` = 0 hat) `sys.servers` und `SERVERPROPERTY('ServerName')` den Bezeichnern. RDS Custom for SQL Server behält diese Synchronisation während des gesamten Lebenszyklus der Instanz automatisch bei, um sicherzustellen, dass das CDC auch dann kontinuierlich funktioniert, wenn Hosts während Wartungs- oder Wiederherstellungsvorgängen ausgetauscht werden.

Important

Nach einem Multi-AZ-Instance-Failover spiegelt die `SERVERPROPERTY('Servername')` Funktion automatisch Änderungen im Namen wider. network/computer Die `@@SERVERNAME` Funktion behält jedoch den alten Servernamen bei, bis der `MSSQLSERVER` Dienst neu gestartet wird. Die Abfrage von @ @SERVERNAME gibt nach einem Failover den vorherigen Servernamen zurück. Verwenden Sie die folgende SQL-Abfrage, um den genauen Servernamen nach einem Failover zu ermitteln:

```
SELECT name FROM sys.servers WHERE server_id=0
```

Diese Abfrage liefert die meisten up-to-date Servernameninformationen, ohne dass ein Neustart des Dienstes erforderlich ist.

Verfügbarkeit von Regionen und Versionen

Die CDC-Funktionalität wird überall unterstützt, AWS-Regionen wo RDS Custom for SQL Server verfügbar ist, und zwar für alle von RDS Custom unterstützten SQL Server-Versionen. Weitere Informationen zu den unterstützten Versionen und der regionalen Verfügbarkeit von RDS Custom for SQL Server finden Sie unterUnterstützte Regionen und DB-Engines für RDS Custom for SQL Server.

Anforderungen und Einschränkungen

Beachten Sie bei der Implementierung von CDC auf RDS Custom for SQL Server die folgenden wichtigen Überlegungen:

- Wenn Sie den `@@SERVERNAME` and/or lokalen Server manuell für die Verwendung von Funktionen wie MS Replication einrichten und der Wert des lokalen Servers (der`server_id = 0`) in auf ein Format gesetzt `sys.servers` ist, das mit `*.rds.amazonaws.com` oder übereinstimmt`*.awsrds.*.com`, versucht RDS Custom for SQL Server nicht, ihn entsprechend `SERVERPROPERTY('ServerName')` zu ändern. `sys.servers`
- RDS kann den lokalen Server (der das getan hat`server_id = 0`) nicht in `sys.servers` einen neuen Hostnamen ändern, solange Remoteanmeldungen oder Verbindungsserver aktiv den alten Hostnamen verwenden. Diese Einschränkung gilt in zwei Szenarien:
  - Wenn ein Verbindungsserver mithilfe einer Remote-Anmeldung, die dem alten Hostnamen zugeordnet ist, eine Verbindung zum lokalen Server herstellt

- Wenn eine RDS Custom for SQL Server-Instanz als Herausgeber oder Verteiler fungiert und Anmeldenamen, die dem alten Hostnamen zugeordnet sind, mit ihren Abonnenteninstanzen verknüpft hat.

Fehlerbehebung

Verwenden Sie die folgenden Abfragen, um Remote-Logins oder verknüpfte Logins zu identifizieren, die mit dem alten Servernamen verknüpft sind. Überprüfen Sie die Ergebnisse und entfernen Sie diese Anmeldungen, um sicherzustellen, dass das CDC ordnungsgemäß funktioniert.

```
SELECT * FROM sys.remote_logins WHERE server_id=0
```

or

```
select sss.srvname,ssp.name,srl.remote_name  from sys.server_principals ssp
inner join sys.remote_logins srl on srl.local_principal_id=ssp.principal_id
inner join sys.sysservers sss  on srl.server_id = sss.srvid
where sss.srvname = @@SERVERNAME
```

Einrichten Ihrer Umgebung für Amazon RDS Custom for SQL Server

Bevor Sie eine DB-Instance für Amazon RDS Custom for SQL Server DB-Instance erstellen und verwalten, müssen Sie die folgenden Aufgaben ausführen.

Inhalt

- Voraussetzungen für das Einrichten von RDS Custom für SQL Server
  - Automatisierte Erstellung von Instanzprofilen mit dem AWS Management Console
- Schritt 1: Erteilen Sie Ihrem IAM-Principal die erforderlichen Berechtigungen
- Schritt 2: Konfiguration von Netzwerk, Instanzprofil und Verschlüsselung
  - Konfiguration mit AWS CloudFormation
    - Parameter, die benötigt werden von CloudFormation
    - Laden Sie die AWS CloudFormation Vorlagendatei herunter
    - Konfiguration von Ressourcen mit CloudFormation
  - Manuelles Konfigurieren
    - Stellen Sie sicher, dass Sie über einen symmetrischen AWS KMS Verschlüsselungsschlüssel verfügen
    - Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils
      - Erstellen Sie die AWSRDSCustom SQLServer InstanceRole IAM-Rolle
      - Fügen Sie eine Zugriffsrichtlinie hinzu zu AWSRDSCustom SQLServer InstanceRole
      - Erstellen Sie Ihr RDS Custom for SQL Server-Instanzprofil
      - Fügen Sie AWSRDSCustom SQLServer InstanceRole es Ihrem Instanzprofil RDS Custom for SQL Server hinzu
    - Konfigurieren Ihrer VPC manuell
      - Konfigurieren Sie Ihre VPC-Sicherheitsgruppen wie folgt:
      - Konfigurieren Sie Endpunkte für abhängige AWS-Services
      - Konfigurieren des Instance-Metadaten-Service
- Instanzübergreifende Einschränkung

> **Note**
>
> Ein step-by-step Tutorial zum Einrichten der Voraussetzungen und zum Starten von Amazon RDS Custom for SQL Server finden Sie unter Erste Schritte mit Amazon RDS Custom for

SQL Server mithilfe einer CloudFormation Vorlage (Netzwerkkonfiguration) und Erkunden Sie die Voraussetzungen, die für die Erstellung einer Amazon RDS Custom for SQL Server-Instance erforderlich sind.

Voraussetzungen für das Einrichten von RDS Custom für SQL Server

Stellen Sie vor dem Erstellen einer DB-Instance von RDS Custom for SQL Server sicher, dass Ihre Umgebung die in diesem Thema beschriebenen Anforderungen erfüllt. Sie können die CloudFormation Vorlage auch verwenden, um die Voraussetzungen in Ihrem einzurichten AWS-Konto. Weitere Informationen finden Sie unter Konfiguration mit AWS CloudFormation.

Für RDS Custom for SQL Server müssen Sie die folgenden Voraussetzungen konfigurieren:

- Konfigurieren Sie die AWS Identity and Access Management (IAM-) Berechtigungen, die für die Instanzerstellung erforderlich sind. Dies ist der AWS Identity and Access Management (IAM-) Benutzer oder die Rolle, die benötigt wird, um eine `create-db-instance` Anfrage an RDS zu stellen.
- Konfigurieren Sie die erforderlichen Ressourcen, die für die DB-Instance RDS Custom for SQL Server erforderlich sind:
  - Konfigurieren Sie den AWS KMS Schlüssel, der für die Verschlüsselung der RDS-Custom-Instanz erforderlich ist. RDS Custom benötigt zum Zeitpunkt der Instanzerstellung einen vom Kunden verwalteten Schlüssel für die Verschlüsselung. Der KMS-Schlüssel ARN, ID, Alias-ARN oder Aliasname wird als `kms-key-id` Parameter in der Anforderung zur Erstellung der benutzerdefinierten RDS-DB-Instance übergeben.
  - Konfigurieren Sie die erforderlichen Berechtigungen innerhalb der DB-Instance RDS Custom for SQL Server. RDS Custom hängt der DB-Instance bei der Erstellung ein Instance-Profil an und verwendet es für die Automatisierung innerhalb der DB-Instance. Der Name des Instance-Profils ist `custom-iam-instance-profile` in der RDS-Anfrage zur benutzerdefinierten Erstellung auf festgelegt. Sie können aus dem ein Instanzprofil erstellen AWS Management Console oder Ihr Instanzprofil manuell erstellen. Weitere Informationen erhalten Sie unter Automatisierte Erstellung von Instanzprofilen mit dem AWS Management Console und Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils.
  - Konfigurieren Sie das Netzwerk-Setup gemäß den Anforderungen von RDS Custom for SQL Server. Benutzerdefinierte RDS-Instances befinden sich in den Subnetzen (konfiguriert mit der DB-Subnetzgruppe), die Sie bei der Instanzerstellung angeben. Diese Subnetze müssen

es RDS-Custom-Instances ermöglichen, mit Diensten zu kommunizieren, die für die RDS-Automatisierung erforderlich sind.

> **Note**
>
> Stellen Sie für die oben genannten Anforderungen sicher, dass es keine Dienststeuerungsrichtlinien (SCPs) gibt, die die Berechtigungen auf Kontoebene einschränken.
> Wenn das Konto, das Sie verwenden, Teil einer AWS Organisation ist, verfügt es möglicherweise über Dienststeuerungsrichtlinien (SCPs), die die Berechtigungen auf Kontoebene einschränken. Stellen Sie sicher, dass die Berechtigungen für Benutzer und Rollen, die Sie mithilfe der folgenden Verfahren erstellen, SCPs nicht eingeschränkt werden. Weitere Informationen zu SCPs finden Sie unter Richtlinien zur Dienststeuerung (SCPs) im AWS Organizations Benutzerhandbuch. Verwenden Sie den AWS CLI Befehl describe-organization, um zu überprüfen, ob Ihr Konto Teil einer AWS Organisation ist.
> Weitere Informationen zu AWS Organizations finden Sie unter Was ist AWS Organizations im AWS Organizations Benutzerhandbuch.

Allgemeine Anforderungen, die für RDS Custom for SQL Server gelten, finden Sie unter Allgemeine Anforderungen von RDS Custom for SQL Server.

Automatisierte Erstellung von Instanzprofilen mit dem AWS Management Console

Bei RDS Custom müssen Sie ein Instanzprofil erstellen und konfigurieren, um eine beliebige DB-Instance von RDS Custom for SQL Server zu starten. Verwenden Sie das AWS Management Console , um in einem einzigen Schritt ein neues Instanzprofil zu erstellen und anzuhängen. Diese Option ist im Abschnitt Benutzerdefinierte RDS-Sicherheit auf den Konsolenseiten Datenbank erstellen, Snapshot wiederherstellen und Zu einem bestimmten Zeitpunkt wiederherstellen verfügbar. Wählen Sie Neues Instanzprofil erstellen aus, um ein Namenssuffix für das Instanzprofil anzugeben. Das AWS Management Console erstellt ein neues Instanzprofil mit den Berechtigungen, die für benutzerdefinierte RDS-Automatisierungsaufgaben erforderlich sind. Um automatisch neue Instanzprofile zu erstellen, muss Ihr angemeldeter AWS Management Console Benutzer über die Berechtigungen`iam:CreateInstanceProfile`, `iam:AddRoleToInstanceProfile`iam:CreateRole, und verfügen. `iam:AttachRolePolicy`

> **Note**
>
> Diese Option ist nur in der verfügbar. AWS Management Console Wenn Sie die CLI oder das SDK verwenden, verwenden Sie die von RDS Custom bereitgestellte CloudFormation Vorlage oder erstellen Sie manuell ein Instanzprofil. Weitere Informationen finden Sie unter Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils.

Schritt 1: Erteilen Sie Ihrem IAM-Principal die erforderlichen Berechtigungen

Stellen Sie sicher, dass Sie über ausreichende Zugriffsrechte verfügen, um eine benutzerdefinierte RDS-Instanz zu erstellen. Die IAM-Rolle oder der IAM-Benutzer (als IAM-Prinzipal bezeichnet) für die Erstellung einer RDS Custom for SQL Server-DB-Instance mithilfe der Konsole oder CLI muss über eine der folgenden Richtlinien verfügen, damit die DB-Instance erfolgreich erstellt werden kann:

- Die Richtlinie `AdministratorAccess`
- Die folgende `AmazonRDSFullAccess`-Richtlinie zeigt die zusätzlichen Berechtigungen.

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
s3:CreateBucket
s3:PutBucketPolicy
s3:PutBucketObjectLockConfiguration
s3:PutBucketVersioning
kms:CreateGrant
kms:DescribeKey
kms:Decrypt
kms:ReEncryptFrom
kms:ReEncryptTo
kms:GenerateDataKeyWithoutPlaintext
kms:GenerateDataKey
ec2:DescribeImages
ec2:RunInstances
ec2:CreateTags
```

RDS Custom verwendet diese Berechtigungen bei der Instanzerstellung. Mit diesen Berechtigungen werden Ressourcen in Ihrem Konto konfiguriert, die für benutzerdefinierte RDS-Operationen erforderlich sind.

Weitere Informationen zu `kms:CreateGrant`-Berechtigung finden Sie unter AWS KMS key Verwaltung.

Die folgende Beispiel-JSON-Richtlinie gewährt die erforderlichen Berechtigungen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
```

```
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Der IAM-Prinzipal benötigt die folgenden zusätzlichen Berechtigungen, um mit benutzerdefinierten Engine-Versionen (CEVs) zu arbeiten:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConfigureKmsKeyEncryptionPermission",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key_id"
        },
        {
            "Sid": "CreateEc2Instance",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:RunInstances",
                "ec2:CreateTags"
            ],
            "Resource": "*"
        }
    ]
}
```

*account_id*Ersetzen Sie es durch die ID des Kontos, mit dem Sie Ihre Instanz erstellen. Ersetzen *region* AWS-Region Sie es durch die, mit der Sie Ihre Instanz erstellen. Ersetzen Sie *key_id* durch Ihre vom Kunden verwaltete Schlüssel-ID. Sie können nach Bedarf mehrere Schlüssel hinzufügen.

Weitere Informationen zu den Berechtigungen auf Ressourcenebene, die zum Starten einer EC2 Instance erforderlich sind, finden Sie unter Instanzen starten () RunInstances.

Außerdem erfordert der IAM-Prinzipal die `iam:PassRole`-Berechtigung für die IAM-Rolle. Diese muss an das Instance-Profil angehängt werden, das im `custom-iam-instance-profile`-Parameter in der Anforderung übergeben wird, um die RDS-Custom-DB-Instance zu erstellen. Das Instance-Profil und seine angehängte Rolle werden später in Schritt 2: Konfiguration von Netzwerk, Instanzprofil und Verschlüsselung erstellt.

> **Note**
>
> Stellen Sie sicher, dass die zuvor aufgeführten Berechtigungen nicht durch Dienststeuerungsrichtlinien (SCPs), Berechtigungsgrenzen oder Sitzungsrichtlinien eingeschränkt werden, die dem IAM-Prinzipal zugeordnet sind.

Schritt 2: Konfiguration von Netzwerk, Instanzprofil und Verschlüsselung

Sie können Ihre IAM-Instanzprofilrolle, Ihre Virtual Private Cloud (VPC) und Ihren AWS KMS symmetrischen Verschlüsselungsschlüssel mithilfe eines der folgenden Prozesse konfigurieren:

- Konfiguration mit AWS CloudFormation (empfohlen)
- Manuelles Konfigurieren

> **Note**
>
> Wenn Ihr Konto Teil eines Kontos ist AWS Organizations, stellen Sie sicher, dass die für die Instanzprofilrolle erforderlichen Berechtigungen nicht durch Dienststeuerungsrichtlinien () eingeschränkt werden. SCPs
> Die Netzwerkkonfigurationen in diesem Thema funktionieren am besten mit DB-Instances, auf die nicht öffentlich zugegriffen werden kann. Sie können von außerhalb der VPC keine direkte Verbindung zu solchen DB-Instances herstellen.

Konfiguration mit AWS CloudFormation

Um die Einrichtung zu vereinfachen, können Sie eine AWS CloudFormation Vorlagendatei verwenden, um einen CloudFormation Stapel zu erstellen. Eine CloudFormation Vorlage erstellt alle Netzwerk-, Instanzprofile und Verschlüsselungsressourcen gemäß den Anforderungen von RDS Custom.

Informationen zum Erstellen von Stacks finden Sie im AWS CloudFormation Benutzerhandbuch unter Erstellen eines Stacks auf der AWS CloudFormation Konsole.

Ein Tutorial zum Starten von Amazon RDS Custom for SQL Server mithilfe einer AWS CloudFormation Vorlage finden Sie unter Erste Schritte mit Amazon RDS Custom for SQL Server unter Verwendung einer AWS CloudFormation Vorlage im AWS Datenbank-Blog.

Themen

- Parameter, die benötigt werden von CloudFormation
- Laden Sie die AWS CloudFormation Vorlagendatei herunter
- Konfiguration von Ressourcen mit CloudFormation

Parameter, die benötigt werden von CloudFormation

Die folgenden Parameter sind erforderlich, um die erforderlichen Ressourcen für RDS Custom zu konfigurieren CloudFormation:

Parametergruppe	Parametername	Standardwert	Beschreibung
Konfiguration der Verfügbarkeit	Wählen Sie eine Verfügbarkeitskonfiguration für die Einrichtung der Voraussetzungen aus	Multi-AZ	Geben Sie an, ob die Voraussetzungen in einer Single-AZ- oder Multi-AZ-Konfiguration für benutzerdefinierte RDS-Instances eingerichtet werden sollen. Sie sollten die Multi-AZ-Konfiguration verwenden, wenn Sie in dieser Konfigura

Parametergruppe	Parametername	Standardwert	Beschreibung
			tion mindestens eine Multi-AZ-DB-Instance benötigen
Netzwerkkonfiguration	IPv4 CIDR-Block für VPC	10.0.0.0/16	Geben Sie einen IPv4 CIDR-Block (oder IP-Adressbereich) für Ihre VPC an. Diese VPC ist so konfiguriert, dass sie eine benutzerdefinierte RDS-DB-Instance erstellt und mit ihr arbeitet.
	IPv4 CIDR-Block für 1 von 2 privaten Subnetzen	10.0.128.0/20	Geben Sie einen IPv4 CIDR-Block (oder IP-Adressbereich) für Ihr erstes privates Subnetz an. Dies ist eines der beiden Subnetze, in denen die RDS Custom DB-Instance erstellt werden kann. Dies ist ein privates Subnetz ohne Internetzugang.

Parametergruppe	Parametername	Standardwert	Beschreibung
	IPv4 CIDR-Block für 2 von 2 privaten Subnetzen	10.0.144.0/20	Geben Sie einen IPv4 CIDR-Block (oder IP-Adressbereich) für Ihr zweites privates Subnetz an. Dies ist eines der beiden Subnetze, in denen die RDS Custom DB-Instance erstellt werden kann. Dies ist ein privates Subnetz ohne Internetzugang.
	IPv4 CIDR-Block für das öffentliche Subnetz	10.0.0.0/20	Geben Sie einen IPv4 CIDR-Block (oder IP-Adressbereich) für Ihr öffentliches Subnetz an. Dies ist eines der Subnetze, in denen die EC2 Instance eine Verbindung mit der RDS Custom DB-Instance herstelle n kann. Dies ist ein öffentliches Subnetz mit Internetzugang.

Parametergruppe	Parametername	Standardwert	Beschreibung
Konfiguration des RDP-Zugriffs	IPv4 CIDR-Block Ihrer Quelle	-	Geben Sie einen IPv4 CIDR-Block (oder IP-Adressbereich) Ihrer Quelle an. Dies ist der IP-Bereich, von dem aus Sie eine RDP-Verbindung zur EC2 Instanz im öffentlichen Subnetz herstellen. Wenn nicht festgelegt, ist die RDP-Verbindung zur EC2 Instanz nicht konfiguriert.
	Richten Sie den RDP-Zugriff auf RDS Custom für die SQL Server-Instanz ein	Nein	Geben Sie an, ob die RDP-Verbindung von der EC2 Instanz zur RDS Custom for SQL Server-Instanz aktiviert werden soll. Standardmäßig ist die RDP-Verbindung von der EC2 Instance zur DB-Instance nicht konfiguriert.

Ressourcen erstellt von CloudFormation

Wenn Sie den CloudFormation Stack erfolgreich mit den Standardeinstellungen erstellen, werden die folgenden Ressourcen in Ihrem erstellt AWS-Konto:

- Symmetrischer KMS-Verschlüsselungsschlüssel zur Verschlüsselung von Daten, die von RDS Custom verwaltet werden.

- Das Instanzprofil ist einer IAM-Rolle zugeordnet, `AmazonRDSCustomInstanceProfileRolePolicy` um die für RDS Custom erforderlichen Berechtigungen bereitzustellen. Weitere Informationen finden Sie unter Amazon RDSCustom ServiceRolePolicy im AWS Managed Policy Reference Guide.
- VPC mit dem als Parameter angegebenen CIDR-Bereich. CloudFormation Der Standardwert ist `10.0.0.0/16`.
- Zwei private Subnetze mit dem in den Parametern angegebenen CIDR-Bereich und zwei verschiedene Availability Zones in der AWS-Region. Die Standardwerte für das Subnetz CIDRs sind und. `10.0.128.0/20` `10.0.144.0/20`
- Ein öffentliches Subnetz mit dem in den Parametern angegebenen CIDR-Bereich. Der Standardwert für das Subnetz CIDR ist 10.0.0.0/20. Die EC2 Instance befindet sich in diesem Subnetz und kann verwendet werden, um eine Verbindung zur RDS-Custom-Instanz herzustellen.
- Die DHCP-Option wurde für die VPC mit Domänennamenauflösung für einen Amazon-DNS-Server (Domain Name System) festgelegt.
- Routing-Tabelle zur Verknüpfung mit zwei privaten Subnetzen und ohne Zugang zum Internet.
- Routing-Tabelle, die dem öffentlichen Subnetz zugeordnet werden soll und Zugriff auf das Internet hat.
- Mit der VPC verbundenes Internet-Gateway, um den Internetzugang zum öffentlichen Subnetz zu ermöglichen.
- Netzwerkzugriffskontrollliste (ACL) zur Verknüpfung mit zwei privaten Subnetzen und beschränkter Zugriff auf HTTPS und DB-Port innerhalb der VPC.
- Die VPC-Sicherheitsgruppe, die der RDS-Custom-Instance zugeordnet werden soll. Der Zugriff ist für ausgehendes HTTPS auf AWS-Service Endpunkte beschränkt, die für RDS Custom und für eingehende DB-Ports aus der Instance-Sicherheitsgruppe erforderlich sind. EC2
- VPC-Sicherheitsgruppe, die der EC2 Instance im öffentlichen Subnetz zugeordnet werden soll. Der Zugriff ist für den ausgehenden DB-Port zur Sicherheitsgruppe RDS Custom Instance beschränkt.
- VPC-Sicherheitsgruppe, die VPC-Endpunkten zugeordnet werden soll, die für Endpoints erstellt wurden, die für AWS-Service RDS Custom erforderlich sind.
- DB-Subnetzgruppe, in der RDS-Custom-Instances erstellt werden. Zwei private Subnetze, die mit dieser Vorlage erstellt wurden, werden der DB-Subnetzgruppe hinzugefügt.
- VPC-Endpunkte für jeden der AWS-Service Endpunkte, die für RDS Custom erforderlich sind.

Wenn Sie die Verfügbarkeitskonfiguration auf Multi-Az setzen, werden zusätzlich zur obigen Liste folgende Ressourcen erstellt:

- Netzwerk-ACL-Regeln, die die Kommunikation zwischen privaten Subnetzen ermöglichen.
- Eingehender und ausgehender Zugriff auf den Multi-AZ-Port innerhalb der VPC-Sicherheitsgruppe, die der benutzerdefinierten RDS-Instance zugeordnet ist.
- VPC-Endpunkte zu AWS Service-Endpunkten, die für die Multi-AZ-Kommunikation erforderlich sind.

Darüber hinaus werden durch die Einstellung der RDP-Zugriffskonfiguration die folgenden Ressourcen erstellt:

- Konfiguration des RDP-Zugriffs auf das öffentliche Subnetz von Ihrer Quell-IP-Adresse aus:
  - Netzwerk-ACL-Regeln, die eine RDP-Verbindung von Ihrer Quell-IP zum öffentlichen Subnetz ermöglichen.
  - Eingangszugriff auf den RDP-Port von Ihrer Quell-IP zur VPC-Sicherheitsgruppe, die der Instance zugeordnet ist. EC2
- Konfiguration des RDP-Zugriffs von der EC2 Instanz im öffentlichen Subnetz zur benutzerdefinierten RDS-Instanz in privaten Subnetzen:
  - Netzwerk-ACL-Regeln, die eine RDP-Verbindung vom öffentlichen Subnetz zu privaten Subnetzen ermöglichen.
  - Eingehender Zugriff auf den RDP-Port von der VPC-Sicherheitsgruppe, die der EC2 Instance zugeordnet ist, zur VPC-Sicherheitsgruppe, die der benutzerdefinierten RDS-Instance zugeordnet ist.

Verwenden Sie die folgenden Verfahren, um den CloudFormation Stack für RDS Custom for SQL Server zu erstellen.

Laden Sie die AWS CloudFormation Vorlagendatei herunter

So laden Sie die Vorlagendatei herunter

1. Öffnen Sie das Kontextmenü (Rechtsklick) für den Link (custom-sqlserver-onboard.zip) und wählen Sie Link speichern unter.
2. Speichern und extrahieren Sie die Datei auf Ihrem Computer.

Konfiguration von Ressourcen mit CloudFormation

Um Ressourcen zu konfigurieren mit CloudFormation

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Starten Sie den Assistenten zum Erstellen von Stacks und wählen Sie Create Stack (Stack erstellen) aus.

   Die Seite Create stack (Stack erstellen) wird angezeigt.
3. Wählen Sie für Voraussetzung - Vorlage vorbereiten die Option Vorlage ist bereit aus.
4. Gehen Sie für Specify template (Vorlage angeben) wie folgt vor:

   a. Wählen Sie unter Templete source (Vorlagenquelle) den Wert Upload a template file (Vorlagendatei hochladen) aus.

   b. Navigieren Sie unter Datei auswählen zu der entsprechenden Datei und wählen Sie sie aus.
5. Wählen Sie Weiter aus.

   Die Seite Specify DB Details (DB-Details angeben) wird angezeigt.
6. Geben Sie unter Stack name (Stack-Name) **`rds-custom-sqlserver`** ein.
7. Führen Sie für Parameters (Parameter) die folgenden Schritte aus:

   a. Wenn Sie die Standardoptionen beibehalten möchten, wählen Sie Next (Weiter) aus.

   b. Um Optionen zu ändern, wählen Sie die entsprechende Verfügbarkeitskonfiguration, Netzwerkkonfiguration und RDP-Zugriffskonfiguration aus, und klicken Sie dann auf Weiter.

      Lesen Sie die Beschreibung jedes Parameters sorgfältig durch, bevor Sie die Parameter ändern.

> **Note**
>
> Wenn Sie mindestens eine Multi-AZ-Instance in diesem CloudFormation Stack erstellen möchten, stellen Sie sicher, dass der CloudFormation Stack-Parameter Wählen Sie eine Verfügbarkeitskonfiguration für die Einrichtung der Voraussetzungen auf eingestellt ist. `Multi-AZ` Wenn Sie den CloudFormation Stack als Single-AZ-Konfiguration erstellen, aktualisieren Sie den CloudFormation Stack auf eine Multi-AZ-Konfiguration, bevor Sie die erste Multi-AZ-Instance erstellen.

8. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.
9. Führen Sie auf der Seite Review (Überprüfen) rds-custom-sqlserver die folgenden Schritte aus:

    a. Aktivieren Sie unter Capabilities (Funktionen) das Kontrollkästchen I acknowledge that AWS CloudFormation , das bestätigt, dass IAM-Ressourcen mit benutzerdefinierten Namen erstellen kann.

    b. Wählen Sie Stack erstellen aus.

> **Note**
>
> Aktualisieren Sie die aus diesem AWS CloudFormation Stack erstellten Ressourcen nicht direkt von den Ressourcenseiten aus. Dadurch wird verhindert, dass Sie future Updates mithilfe einer AWS CloudFormation Vorlage auf diese Ressourcen anwenden.

CloudFormation erstellt die Ressourcen, die RDS Custom for SQL Server benötigt. Wenn die Stack-Erstellung fehlschlägt, rufen Sie die Seite Events (Ereignisse) auf, um zu sehen, welche Ressourcenerstellung mit welchem Statusgrund fehlgeschlagen.

Die Registerkarte Ausgaben für diesen CloudFormation Stack in der Konsole sollte Informationen über alle Ressourcen enthalten, die als Parameter für die Erstellung einer RDS Custom for SQL Server-DB-Instance übergeben werden müssen. Stellen Sie sicher, dass Sie die VPC-Sicherheitsgruppe und die DB-Subnetzgruppe verwenden, die von CloudFormation for RDS Custom DB-Instances erstellt wurden. Standardmäßig versucht RDS, die VPC-Standardsicherheitsgruppe anzuhängen, die möglicherweise nicht den benötigten Zugriff hat.

Wenn Sie früher CloudFormation Ressourcen erstellt haben, können Sie das überspringen. Manuelles Konfigurieren

Den CloudFormation Stack aktualisieren

Sie können auch einen Teil der Konfiguration auf dem CloudFormation Stack nach der Erstellung aktualisieren. Die Konfigurationen, die aktualisiert werden können, sind:

- Verfügbarkeitskonfiguration für RDS Custom für SQL Server
    - Wählen Sie eine Verfügbarkeitskonfiguration für die Einrichtung der Voraussetzungen aus: Aktualisieren Sie diesen Parameter, um zwischen einer Single-AZ- und einer Multi-AZ-

Konfiguration zu wechseln. Wenn Sie diesen CloudFormation Stack für mindestens eine Multi-AZ-Instance verwenden, müssen Sie den Stack aktualisieren, um die Multi-AZ-Konfiguration zu wählen.

- RDP-Zugriffskonfiguration für RDS Benutzerdefiniert für SQL Server
  - IPv4 CIDR-Block Ihrer Quelle: Sie können den IPv4 CIDR-Block (oder den IP-Adressbereich) Ihrer Quelle aktualisieren, indem Sie diesen Parameter aktualisieren. Wenn Sie diesen Parameter auf leer setzen, wird die RDP-Zugriffskonfiguration aus Ihrem CIDR-Quellblock zum öffentlichen Subnetz entfernt.
  - RDP-Zugriff auf RDS Custom for SQL Server einrichten: Aktivieren oder deaktivieren Sie die RDP-Verbindung von der EC2 Instanz zur RDS Custom for SQL Server-Instanz.

Den Stapel löschen CloudFormation

Sie können den CloudFormation Stack löschen, nachdem Sie alle benutzerdefinierten RDS-Instances gelöscht haben, die Ressourcen aus dem Stack verwenden. RDS Custom verfolgt den CloudFormation Stack nicht und blockiert daher nicht das Löschen des Stacks, wenn es DB-Instances gibt, die Stack-Ressourcen verwenden. Stellen Sie sicher, dass es beim Löschen des Stacks keine benutzerdefinierten RDS-DB-Instances gibt, die die Stack-Ressourcen verwenden.

> **Note**
>
> Wenn Sie einen CloudFormation Stack löschen, werden alle vom Stack erstellten Ressourcen mit Ausnahme des KMS-Schlüssels gelöscht. Der KMS-Schlüssel wechselt in den Status „Ausstehende Löschung“ und wird nach 30 Tagen gelöscht. Um den KMS-Schlüssel zu behalten, führen Sie während der 30-tägigen Nachfrist einen CancelKeyDeletionVorgang durch.

Manuelles Konfigurieren

Wenn Sie Ressourcen manuell konfigurieren möchten, gehen Sie wie folgt vor.

> **Note**
>
> Um die Einrichtung zu vereinfachen, können Sie die AWS CloudFormation Vorlagendatei verwenden, um einen CloudFormation Stack zu erstellen, anstatt eine manuelle Konfiguration vorzunehmen. Weitere Informationen finden Sie unter Konfiguration mit AWS CloudFormation.

Sie können den auch verwenden AWS CLI , um diesen Abschnitt zu vervollständigen. Wenn ja, laden Sie die neueste CLI herunter und installieren Sie sie.

Themen

- Stellen Sie sicher, dass Sie über einen symmetrischen AWS KMS Verschlüsselungsschlüssel verfügen
- Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils
- Konfigurieren Ihrer VPC manuell

Stellen Sie sicher, dass Sie über einen symmetrischen AWS KMS Verschlüsselungsschlüssel verfügen

Für RDS Custom AWS KMS key ist eine symmetrische Verschlüsselung erforderlich. Wenn Sie eine RDS Custom for SQL Server-DB-Instance erstellen, stellen Sie sicher, dass Sie die KMS-Schlüssel-ID als Parameter angeben`kms-key-id`. Weitere Informationen finden Sie unter Eine DB-Instance für Amazon RDS Custom for SQL Server erstellen und eine Verbindung zu ihr herstellen.

Ihnen stehen folgende Optionen zur Verfügung:

- Wenn Sie bereits einen vom Kunden verwalteten KMS-Schlüssel in Ihrer haben AWS-Konto, können Sie ihn mit RDS Custom verwenden. Es sind keine weiteren Maßnahmen erforderlich.
- Wenn Sie bereits einen kundenverwalteten symmetrischen KMS-Verschlüsselungsschlüssel für eine andere RDS-Custom-Engine erstellt haben, können Sie denselben KMS-Schlüssel wiederverwenden. Es sind keine weiteren Maßnahmen erforderlich.
- Wenn Sie keinen vorhandenen kundenverwalteten symmetrischen KMS-Verschlüsselungsschlüssel in Ihrem Konto haben, erstellen Sie einen KMS-Schlüssel, indem Sie den Anweisungen unter Erstellen von Schlüsseln im AWS Key Management Service - Entwicklerhandbuch folgen.
- Wenn Sie eine benutzerdefinierte CEV- oder RDS-DB-Instance erstellen und sich Ihr KMS-Schlüssel in einer anderen befindet AWS-Konto, stellen Sie sicher, dass Sie den AWS CLI verwenden. Sie können die AWS Konsole nicht mit kontoübergreifenden KMS-Schlüsseln verwenden.

> **Important**
>
> RDS Custom unterstützt keine AWS verwalteten KMS-Schlüssel.

Stellen Sie sicher, dass Ihr symmetrischer Verschlüsselungsschlüssel Zugriff auf die `kms:Decrypt` und `kms:GenerateDataKey` -Operationen der AWS Identity and Access Management (IAM-) Rolle in Ihrem IAM-Instanzprofil gewährt. Wenn Sie einen neuen symmetrischen Verschlüsselungsschlüssel in Ihrem Konto haben, sind keine Änderungen erforderlich. Stellen Sie andernfalls sicher, dass die Richtlinie Ihres symmetrischen Verschlüsselungsschlüssels Zugriff auf diese Operationen erteilt.

Weitere Informationen finden Sie unter Schritt 4: Konfigurieren Sie IAM für RDS Custom für Oracle.

Erstellen Ihrer IAM-Rolle und Ihres Instance-Profils

Sie können manuell ein Instance-Profil erstellen und es verwenden, um benutzerdefinierte RDS-Instances zu starten. Wenn Sie planen, die Instanz in zu erstellen AWS Management Console, überspringen Sie diesen Abschnitt. Das AWS Management Console ermöglicht es Ihnen, ein Instance-Profil zu erstellen und an Ihre RDS Custom DB-Instances anzuhängen. Weitere Informationen finden Sie unter Automatisierte Erstellung von Instanzprofilen mit dem AWS Management Console.

Wenn Sie manuell ein Instanzprofil erstellen, übergeben Sie den Namen des Instanzprofils als `custom-iam-instance-profile` Parameter an Ihren `create-db-instance` CLI-Befehl. RDS Custom verwendet die diesem Instanzprofil zugeordnete Rolle, um die Automatisierung zur Verwaltung der Instanz auszuführen.

So erstellen Sie das IAM-Instance-Profil und IAM-Rollen für RDS Custom for SQL Server

1. Erstellen Sie die IAM-Rolle mit einem Namen `AWSRDSCustomSQLServerInstanceRole` mit einer Vertrauensrichtlinie, die es Amazon ermöglicht, diese Rolle zu EC2 übernehmen.
2. Fügen Sie die AWS verwaltete Richtlinie `AmazonRDSCustomInstanceProfileRolePolicy` zu `AWSRDSCustomSQLServerInstanceRole` hinzu.
3. Erstellen Sie ein IAM-Instance-Profil für RDS Custom for SQL Server namens `AWSRDSCustomSQLServerInstanceProfile`.
4. Fügen Sie dem Instance-Profil `AWSRDSCustomSQLServerInstanceRole` hinzu.

Erstellen Sie die AWSRDSCustom SQLServer InstanceRole IAM-Rolle

Im folgenden Beispiel wird eine `AWSRDSCustomSQLServerInstanceRole`-Rolle erstellt. Die Vertrauensrichtlinie ermöglicht es Amazon, die Rolle zu EC2 übernehmen.

```
aws iam create-role \
    --role-name AWSRDSCustomSQLServerInstanceRole \
    --assume-role-policy-document '{
        "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                  "Service": "ec2.amazonaws.com"
              }
            }
          ]
        }'
```

Fügen Sie eine Zugriffsrichtlinie hinzu zu AWSRDSCustom SQLServer InstanceRole

Um die erforderlichen Berechtigungen bereitzustellen, hängen Sie die AWS verwaltete Richtlinie `AmazonRDSCustomInstanceProfileRolePolicy` an an`AWSRDSCustomSQLServerInstanceRole`. `AmazonRDSCustomInstanceProfileRolePolicy`ermöglicht benutzerdefinierten RDS-Instances das Senden und Empfangen von Nachrichten sowie das Ausführen verschiedener Automatisierungsaktionen.

> **Note**
>
> Stellen Sie sicher, dass die Berechtigungen in der Zugriffsrichtlinie nicht durch SCPs die mit der Instanzprofilrolle verbundenen Berechtigungsgrenzen eingeschränkt sind.

Im folgenden Beispiel wird der `AWSRDSCustomSQLServerInstanceRole` Rolle eine AWS verwaltete Richtlinie `AmazonRDSCustomInstanceProfileRolePolicy` zugewiesen.

```
aws iam attach-role-policy \
    --role-name AWSRDSCustomSQLServerInstanceRole \
```

```
    --policy-arn arn:aws:iam::aws:policy/AmazonRDSCustomInstanceProfileRolePolicy
```

Erstellen Sie Ihr RDS Custom for SQL Server-Instanzprofil

Ein Instance-Profil ist ein Container, der eine einzelne IAM-Rolle enthält. RDS Custom verwendet das Instance-Profil, um die Rolle der Instance zu übergeben.

Wenn Sie die verwenden, AWS Management Console um eine Rolle für Amazon zu erstellen EC2, erstellt die Konsole automatisch ein Instance-Profil und weist diesem bei der Erstellung der Rolle denselben Namen zu. Erstellen Sie Ihr Instanzprofil wie folgt und benennen Sie es `AWSRDSCustomSQLServerInstanceProfile`.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile
```

Fügen Sie AWSRDSCustom SQLServer InstanceRole es Ihrem Instanzprofil RDS Custom for SQL Server hinzu

Fügen Sie die `AWSRDSCustomInstanceRoleForRdsCustomInstance` Rolle dem zuvor erstellten `AWSRDSCustomSQLServerInstanceProfile` Profil hinzu.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile \
    --role-name AWSRDSCustomSQLServerInstanceRole
```

Konfigurieren Ihrer VPC manuell

Ihre benutzerdefinierte RDS-DB-Instance befindet sich in einer Virtual Private Cloud (VPC), die auf dem Amazon VPC-Service basiert, genau wie eine EC2 Amazon-Instance oder Amazon RDS-Instance. Sie stellen Ihre eigene VPC zur Verfügung und konfigurieren sie. Somit haben Sie die volle Kontrolle über Ihr Instanznetzwerk-Setup.

RDS Custom sendet Kommunikation von Ihrer DB-Instance an andere AWS-Services. Stellen Sie sicher, dass die folgenden Dienste von dem Subnetz aus zugänglich sind, in dem Sie Ihre RDS Custom DB-Instances erstellen:

- Amazon CloudWatch (`com.amazonaws.`*`region`*`.monitoring`)
- CloudWatch Amazon-Protokolle (`com.amazonaws.`*`region`*`.logs`)
- CloudWatch Amazon-Veranstaltungen (`com.amazonaws.`*`region`*`.events`)

- Amazon EC2 (`com.amazonaws.region.ec2undcom.amazonaws.region.ec2messages`)
- Amazon EventBridge (`com.amazonaws.region.events`)
- Amazon S3 (`com.amazonaws.region.s3`)
- AWS Secrets Manager (`com.amazonaws.region.secretsmanager`)
- AWS Systems Manager (`com.amazonaws.region.ssmundcom.amazonaws.region.ssmmessages`)

Wenn Sie Multi-AZ-Bereitstellungen erstellen

- Amazon Simple Queue Service (`com.amazonaws.region.sqs`)

Wenn RDS Custom nicht mit den erforderlichen Diensten kommunizieren kann, veröffentlicht es die folgenden Ereignisse:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS
 services. Make sure port 443 (HTTPS) allows outbound connections, and try again.
 "Failed to connect to the following services: s3 events"
```

Um `incompatible-network` Fehler zu vermeiden, stellen Sie sicher, dass die VPC-Komponenten, die an der Kommunikation zwischen Ihrer RDS Custom DB-Instance beteiligt sind, die folgenden Anforderungen AWS-Services erfüllen:

- Die DB-Instance kann ausgehende Verbindungen an Port 443 mit anderen AWS-Services herstellen.
- Die VPC lässt eingehende Antworten auf Anfragen zu, die von Ihrer DB-Instance von RDS Custom stammen.
- RDS Custom kann die Domain-Namen von Endpunkten für jeden AWS-Service korrekt auflösen.

Wenn Sie eine VPC bereits für eine andere DB-Engine von RDS Custom konfiguriert haben, können Sie diese VPC wiederverwenden und diesen Prozess überspringen.

Themen

- Konfigurieren Sie Ihre VPC-Sicherheitsgruppen wie folgt:
- Konfigurieren Sie Endpunkte für abhängige AWS-Services
- Konfigurieren des Instance-Metadaten-Service

Konfigurieren Sie Ihre VPC-Sicherheitsgruppen wie folgt:

Eine Sicherheitsgruppe dient als virtuelle Firewall für Ihre VPC-Instance zur Steuerung von ein- und ausgehendem Datenverkehr. Einer RDS Custom DB-Instance ist eine Sicherheitsgruppe an die Netzwerkschnittstelle angehängt, die die Instance schützt. Stellen Sie sicher, dass Ihre Sicherheitsgruppe Datenverkehr zwischen RDS Custom und anderen AWS-Services über HTTPS zulässt. Sie übergeben diese Sicherheitsgruppe als `vpc-security-group-ids` Parameter in der Anfrage zur Instanzerstellung.

So konfigurieren Sie Ihre Sicherheitsgruppe für RDS Custom

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc.
2. Erlauben Sie RDS Custom, die Standardsicherheitsgruppe zu verwenden, oder erstellen Sie Ihre eigene Sicherheitsgruppe.

   Detaillierte Anweisungen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.
3. Stellen Sie sicher, dass Ihre Sicherheitsgruppe eingehende Verbindungen an Port 443 zulässt. RDS Custom benötigt diesen Port, um mit abhängigen AWS-Services zu kommunizieren.
4. Wenn Sie eine private VPC mit VPC-Endpunkten verwenden, stellen Sie sicher, dass die mit der DB-Instance verbundene Sicherheitsgruppe ausgehende Verbindungen mit VPC-Endpunkten an Port 443 zulässt. Stellen Sie außerdem sicher, dass die mit dem VPC-Endpunkt verknüpfte Sicherheitsgruppe eingehende Verbindungen an Port 443 von der DB-Instance zulässt.

   Wenn eingehende Verbindungen nicht zulässig sind, kann die RDS-Custom-Instance keine Verbindung zu den Endpunkten AWS Systems Manager und EC2 Amazon-Endpunkten herstellen. Weitere Informationen finden Sie unter Erstellen eines Virtual-Private-Cloud-Endpunkts im AWS Systems Manager -Benutzerhandbuch.
5. Stellen Sie bei Multi-AZ-Instances von RDS Custom for SQL Server sicher, dass die der DB-Instance zugeordnete Sicherheitsgruppe eingehende und ausgehende Verbindungen über Port 1120 zu dieser Sicherheitsgruppe selbst zulässt. Dies ist für eine Peer-Host-Verbindung auf einer Multi-AZ-RDS-Custom for SQL Server-DB-Instance erforderlich.

Weitere Informationen zu VPC-Sicherheitsgruppen finden Sie unter Sicherheitsgruppen für Ihre VPC im Amazon-VPC-Benutzerhandbuch.

Konfigurieren Sie Endpunkte für abhängige AWS-Services

Wir empfehlen Ihnen, Ihrer VPC Endpunkte für jeden Dienst mit den folgenden Anweisungen hinzuzufügen. Sie können jedoch jede Lösung verwenden, mit der Ihre VPC mit AWS Service-Endpunkten kommunizieren kann. Zum Beispiel können Sie Network Address Translation (NAT) oder AWS Direct Connect nutzen.

Um Endpunkte zu konfigurieren, für die RDS AWS-Services Custom funktioniert

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie in der Navigationsleiste mithilfe der Regionsauswahl AWS-Region Ihre Region aus.
3. Wählen Sie im Navigationsbereich Endpunkte aus. Wählen Sie im Hauptnavigationsbereich Create Endpoint (Endpunkt erstellen).
4. Wählen Sie für Service category (Servicekategorie) die Option AWS-Services aus.
5. Für Service-Name wählen Sie den in der Tabelle angezeigten Endpunkt aus.
6. Wählen Sie unter VPC Ihre VPC aus.
7. Wählen Sie unter Subnetze ein Subnetz für jede Availability Zone aus, die eingeschlossen werden soll.

   Der VPC-Endpunkt kann sich über mehrere Availability Zones erstrecken. AWS erstellt eine elastic network interface für den VPC-Endpunkt in jedem Subnetz, das Sie auswählen. Jede Netzwerkschnittstelle weist einen Domain-Name-System(DNS)-Hostnamen und eine private IP-Adresse auf.
8. Wählen Sie unter Security groups eine Sicherheitsgruppe aus oder erstellen Sie eine.

   Sie können mit Sicherheitsgruppen den Zugriff auf Ihren Endpunkt steuern, ähnlich wie bei der Verwendung einer Firewall. Stellen Sie sicher, dass die Sicherheitsgruppe eingehende Verbindungen von den DB-Instances über Port 443 zulässt. Weitere Informationen zu VPC-Sicherheitsgruppen finden Sie unter Sicherheitsgruppen für Ihre VPC im Amazon VPC-Benutzerhandbuch.
9. Optional können Sie eine Richtlinie an den VPC-Endpunkt anhängen. Endpunktrichtlinien können den Zugriff auf die Daten steuern AWS-Service , zu denen Sie eine Verbindung herstellen. Die Standardrichtlinie erlaubt es, dass alle Anfragen den Endpunkt passieren. Wenn

Sie eine benutzerdefinierte Richtlinie verwenden, stellen Sie sicher, dass Anfragen von der DB-Instance in der Richtlinie zulässig sind.

10. Wählen Sie Endpunkt erstellen aus.

In der folgenden Tabelle wird erläutert, wie Sie die Liste der Endpunkte finden, die Ihre VPC für ausgehende Kommunikation benötigt.

Service	Endpunkt-Format	Hinweise und Links
AWS Systems Manager	Verwenden Sie die folgenden Endpunktformate: • `ssm.region.amazonaws.com` • `ssmmessages. region.amazonaws.com`	Eine Liste aller Endpunkte in den einzelnen Regionen finden Sie unter AWS Systems Manager -Endpunkte und Kontingente im Allgemeine Amazon Web Services-Referenz.
AWS Secrets Manager	Verwenden Sie dabei das Endpunktformat `secretsmanager. region.amazonaws.com .`	Eine Liste aller Endpunkte in den einzelnen Regionen finden Sie unter AWS Secrets Manager -Endpunkte und Kontingente im Allgemeine Amazon Web Services-Referenz.
Amazon CloudWatch	Verwenden Sie die folgenden Endpunktformate: • Verwenden Sie für CloudWatch Metriken `monitoring. region.amazonaws.com` • Verwenden Sie für CloudWatch Ereignisse `events.region.amazonaws.com` • Verwenden Sie für CloudWatch Protokolle `logs.region.amazonaws.com`	Eine Liste der Endpunkte in jeder Region finden Sie unter: • CloudWatch Amazon-Endpunkte und Kontingente in der Allgemeine Amazon Web Services-Referenz • Amazon CloudWatch protokolliert Endpunkte und Kontingente in der Allgemeine Amazon Web Services-Referenz • Amazon CloudWatch Events-Endpunkte und Kontingente in

Service	Endpunkt-Format	Hinweise und Links
		der Allgemeine Amazon Web Services-Referenz
Amazon EC2	Verwenden Sie die folgenden Endpunktformate: • `ec2.`*`region`*`.amazonaws.com` • `ec2messages.`*`region`*`.amazonaws.com`	Eine vollständige Liste der Endpunkte in jeder Region finden Sie unter Endpunkte und Kontingente von Amazon Elastic Compute Cloud im Allgemeine Amazon Web Services-Referenz.
Amazon S3	Verwenden Sie dabei das Endpunktformat `s3.`*`region`*`.amazonaws.com` .	Eine vollständige Liste der Endpunkte in jeder Region finden Sie unter Endpunkte und Kontingente von Amazon Simple Storage Service im Allgemeine Amazon Web Services-Referenz. Weitere Informationen zu Gateway-Endpunkten für Amazon S3 finden Sie unterEndpunkte für Amazon S3imEntwicklerhandbuch für Amazon VPCaus. Informationen zum Erstellen eines Zugriffspunkts finden Sie unter Erstellen von Zugriffspunkten im Entwicklerhandbuch für Amazon VPC. Weitere Informationen zum Erstellen eines Gateway-Endpunkts für Amazon S3 finden Sie unter Gateway-VPC-Endpunkte.

Service	Endpunkt-Format	Hinweise und Links
Amazon Simple Queue Service	Verwenden Sie das Endpunktformat `sqs.region.amazonaws.com`	Eine Liste der Endpunkte in den einzelnen Regionen finden Sie unter Amazon Simple Queue Service-Endpunkte und Kontingente.

Konfigurieren des Instance-Metadaten-Service

Stellen Sie folgendermaßen sicher, dass die EC2-Instance eine Verbindung zu herstellen kann:

- Greifen Sie mit Instance Metadata Service Version 2 () IMDSv2 auf den Instance-Metadaten-Service zu.
- Lassen Sie ausgehende Kommunikation über Port 80 (HTTP) zur IMDS-Link-IP-Adresse zu.
- Fordern Sie `http://169.254.169.254` Instanz-Metadaten über den IMDSv2 Link an.

Weitere Informationen finden Sie unter Verwendung IMDSv2 im EC2 Amazon-Benutzerhandbuch.

Instanzübergreifende Einschränkung

Wenn Sie mithilfe der oben genannten Schritte ein Instanzprofil erstellen, verwendet es die AWS verwaltete Richtlinie, `AmazonRDSCustomInstanceProfileRolePolicy` um RDS Custom die erforderlichen Berechtigungen bereitzustellen, wodurch die Instanzverwaltung und -überwachung automatisiert werden können. Die verwaltete Richtlinie stellt sicher, dass die Berechtigungen nur den Zugriff auf die Ressourcen ermöglichen, die RDS Custom für die Ausführung der Automatisierung benötigt. Wir empfehlen, die verwaltete Richtlinie zu verwenden, um neue Funktionen zu unterstützen und Sicherheitsanforderungen zu erfüllen, die automatisch und ohne manuelles Eingreifen auf bestehende Instanzprofile angewendet werden. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSCusto InstanceProfileRolePolicym.

Die `AmazonRDSCustomInstanceProfileRolePolicy` verwaltete Richtlinie schränkt den kontoübergreifenden Zugriff des Instance-Profils ein, ermöglicht aber möglicherweise den Zugriff auf einige von RDS Custom verwaltete Ressourcen für RDS Custom-Instances innerhalb desselben Kontos. Je nach Ihren Anforderungen können Sie Berechtigungsgrenzen verwenden, um den instanzübergreifenden Zugriff weiter einzuschränken. Berechtigungsgrenzen definieren die maximalen Berechtigungen, die die identitätsbasierten Richtlinien einer Entität gewähren können,

gewähren aber selbst keine Berechtigungen. Weitere Informationen finden Sie unter Bewertung effektiver Berechtigungen mit Grenzen.

Die folgende Grenzrichtlinie schränkt beispielsweise die Instanzprofilrolle auf den Zugriff auf einen bestimmten AWS KMS Schlüssel ein und beschränkt den Zugriff auf von RDS Custom verwaltete Ressourcen auf Instances, die unterschiedliche AWS KMS Schlüssel verwenden.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyOtherKmsKeyAccess",
            "Effect": "Deny",
            "Action": "kms:*",
            "NotResource": "arn:aws:kms:us-east-1:111122223333:key/KMS_key_ID"
        }
    ]
}
```

> **Note**
>
> Stellen Sie sicher, dass die Berechtigungsgrenze keine Berechtigungen blockiert, die RDS Custom `AmazonRDSCustomInstanceProfileRolePolicy` gewährt werden.

Bringen Sie Ihre eigenen Medien mit RDS Custom for SQL Server

RDSCustom for SQL Server unterstützt zwei Lizenzmodelle: License Included (LI) und Bring Your Own Media (BYOM).

Mit BYOM können Sie Folgendes tun:

1. Stellen Sie Ihre eigenen Microsoft SQL Server-Binärdateien mit unterstützten kumulativen Updates (CU) auf einem AWS EC2 Windows bereit und installieren Sie sie. AMI
2. Speichern Sie das AMI als goldenes Abbild. Dabei handelt es sich um eine Vorlage, mit der Sie eine benutzerdefinierte Engine-Version () CEV erstellen können.
3. Erstellen Sie CEV aus Ihrem goldenen Bild ein.
4. Erstellen Sie neue RDS benutzerdefinierte für SQL Server-DB-Instances, indem Sie Ihre verwendenCEV.

Amazon verwaltet RDS dann Ihre DB-Instances für Sie.

> **Note**
>
> Wenn Sie auch über eine RDS Custom for SQL Server-DB-Instance (LI) (License Included) verfügen, können Sie die SQL Serversoftware dieser DB-Instance nicht mit verwendenBYOM. Sie müssen Ihre eigenen SQL Server-Binärdateien zu BYOM mitbringen.

Anforderungen für RDS Custom BYOM for Server SQL

Dieselben allgemeinen Anforderungen für benutzerdefinierte Engine-Versionen mit RDS Custom for SQL Server gelten auch fürBYOM. Weitere Informationen finden Sie unter Anforderungen für RDS Custom für SQL Server CEVs.

Stellen Sie bei der Verwendung BYOM sicher, dass Sie die folgenden zusätzlichen Anforderungen erfüllen:

- Verwenden Sie eine der folgenden unterstützten Editionen: SQL Server 2022 oder 2019 Enterprise, Standard oder Developer Edition.
- Erteilen Sie die SQL Serverrolle Serversysadmin (SA) für. `NT AUTHORITY\SYSTEM`
- Behalten Sie die Konfiguration des EC2 Windows-Serverbetriebssystems mit `UTC` der Zeit bei.

Die zugrunde liegenden Amazon EC2 Windows-Instances sind standardmäßig auf die UTC Zeitzone eingestellt. Weitere Informationen zum Anzeigen und Ändern der Uhrzeit für eine Windows-Instance finden Sie unter Zeit für eine Windows-Instance festlegen im EC2Amazon-Benutzerhandbuch.

- Öffnen Sie TCP Port 1433 und UDP Port 1434, um Verbindungen zuzulassenSSM.

Einschränkungen von BYOM für RDS Custom for Server SQL

Dieselben allgemeinen Einschränkungen für RDS Custom for SQL Server gelten auch fürBYOM. Weitere Informationen finden Sie unter Anforderungen und Einschränkungen für Amazon RDS Custom for SQL Server.

Bei BYOM gelten die folgenden zusätzlichen Einschränkungen:

- Nur die SQL Standard-Serverinstanz (MSSQLSERVER) wird unterstützt. Benannte SQL Server-Instanzen werden nicht unterstützt. RDSCustom for SQL Server erkennt und überwacht nur die SQL Standard-Serverinstanz.
- Auf jeder SQL Serverinstallation wird nur eine einzige Installation unterstütztAMI. Mehrere Installationen verschiedener SQL Serverversionen werden nicht unterstützt.
- SQLServer Web Edition wird mit nicht unterstütztBYOM.
- Testversionen von SQL Server-Editionen werden mit nicht unterstütztBYOM. Wenn Sie SQL Server installieren, aktivieren Sie nicht das Kontrollkästchen für die Verwendung einer Testversion.
- Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen erhalten Sie unter Regionale Verfügbarkeit für RDS Custom for SQL Server CEVs und Versionsunterstützung für RDS Custom für SQL Server CEVs.

Erstellen einer RDS benutzerdefinierten SQL Server-DB-Instance mit BYOM

Informationen zum Vorbereiten und Erstellen einer RDS benutzerdefinierten SQL Server-DB-Instance mit BYOM finden Sie unterVorbereitung einer CEV mit Bring Your Own Media (BYOM).

Arbeiten mit benutzerdefinierten Engine-Versionen für RDS Custom für SQL Server

Eine benutzerdefinierte Engine-Version (CEV) für RDS Custom für SQL Server ist ein Amazon Machine Image (AMI) mit Microsoft SQL Server.

Die grundlegenden Schritte des CEV-Workflows lauten wie folgt:

1. Wählen Sie ein AWS-EC2-Windows-AMI aus, das als Basis-Image für eine CEV verwendet werden soll. Sie haben die Möglichkeit, den vorinstallierten Microsoft SQL Server oder Ihre eigenen Medien zu verwenden, um SQL Server selbst zu installieren.
2. Installieren Sie andere Software auf dem Betriebssystem und passen Sie die Konfiguration des Betriebssystems und des SQL-Servers an die Anforderungen Ihres Unternehmens an.
3. Speichern Sie das AMI als goldenes Image.
4. Erstellen Sie aus Ihrem goldenen Image eine benutzerdefinierte Engine-Version (CEV).
5. Erstellen Sie neue DB-Instances von RDS Custom für SQL Server mithilfe der CEV.

Amazon RDS verwaltet diese DB-Instances dann für Sie.

Eine CEV ermöglicht es Ihnen, Ihre bevorzugte Basiskonfiguration des Betriebssystems und der Datenbank beizubehalten. Durch die Verwendung einer CEV wird sichergestellt, dass die Hostkonfiguration, z. B. die Installation von Drittanbieter-Agenten oder andere Betriebssystemanpassungen, auf DB-Instances von RDS Custom für SQL Server beibehalten wird. Darüber hinaus können Sie mit einer CEV schnell Flotten von DB-Instances von RDS Custom für SQL Server mit derselben Konfiguration bereitstellen.

Themen

- Vorbereitung der Erstellung einer CEV für RDS Custom für SQL Server
- Ein CEV für RDS Custom for SQL Server erstellen
- Ändern eines CEV für RDS Benutzerdefiniert für SQL Server
- Anzeigen von CEV-Details zu Amazon RDS Custom für SQL Server
- Löschen einer CEV für RDS Custom für SQL Server

Vorbereitung der Erstellung einer CEV für RDS Custom für SQL Server

Sie können eine CEV mit einem Amazon Machine Image (AMI) erstellen, das den vorinstallierten Microsoft SQL Server mit Lizenz inklusive (LI) enthält, oder mit einem AMI, auf dem Sie Ihre eigenen SQL-Server-Installationsmedien (BYOM) installieren.

Vorbereitung eines CEV

Gehen Sie wie folgt vor, um ein CEV mit Bring Your Own Media (BYOM) oder vorinstalliertem Microsoft SQL Server (LI) zu erstellen.

Vorbereitung einer CEV mit Bring Your Own Media (BYOM)

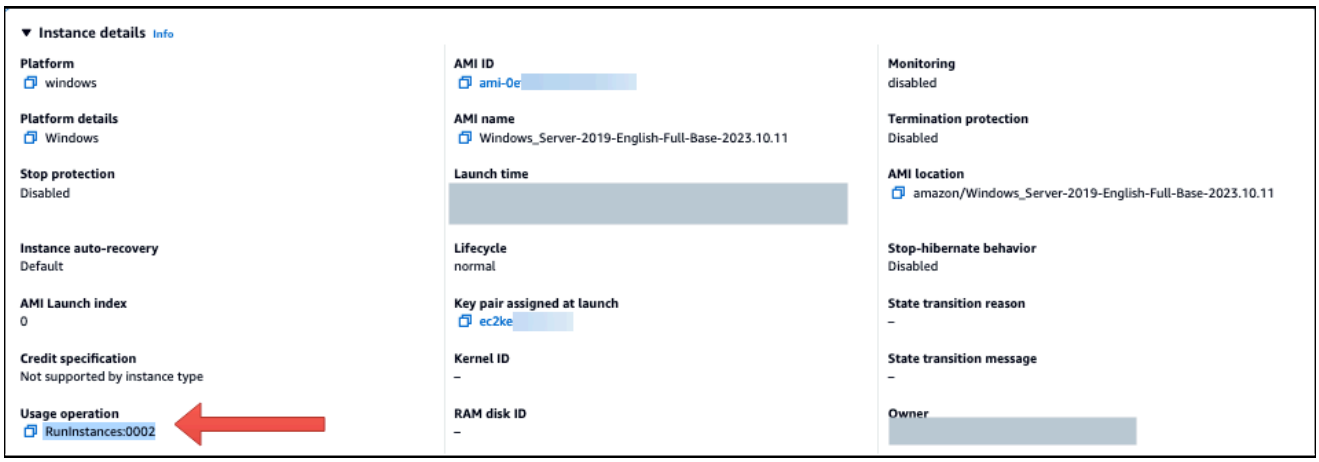
In den folgenden Schritten wird ein AMI mit Windows Server 2019 Base als Beispiel verwendet.

So erstellen Sie eine CEV mit BYOM

1. Wählen Sie auf der EC2 Amazon-Konsole Launch Instance aus.
2. Geben Sie unter Name den Namen der Instance ein.
3. Wählen Sie unter Schnellstart die Option Windows aus.
4. Wählen Sie Microsoft Windows Server 2019 Base.
5. Wählen Sie einen geeigneten Instance-Typ, ein key pair sowie Netzwerk- und Speichereinstellungen aus und starten Sie die Instance.
6. Stellen Sie nach dem Starten oder Erstellen der EC2 Instance sicher, dass das richtige Windows-AMI aus Schritt 4 ausgewählt wurde:

    a. Wählen Sie die EC2 Instance in der EC2 Amazon-Konsole aus.

    b. Überprüfen Sie im Abschnitt „Details“ den Vorgang „Nutzung“ und stellen Sie sicher, dass er RunInstances auf:0002 gesetzt ist.



7. Melden Sie sich bei der EC2 Instanz an und kopieren Sie Ihr SQL Server-Installationsmedium darauf.

   > **Note**
   >
   > Wenn Sie ein CEV mit der SQL Server Developer Edition erstellen, müssen Sie das Installationsmedium möglicherweise mit Ihrem Microsoft Visual Studio-Abonnement beziehen.

8. Installieren Sie SQL Server. Stellen Sie Folgendes sicher:

   a. Überprüfen Anforderungen für RDS Custom BYOM for Server SQL und. Versionsunterstützung für RDS Custom für SQL Server CEVs

   b. Legen Sie das Instance-Stammverzeichnis auf das Standardverzeichnis `C:\Program Files\Microsoft SQL Server\` fest. Ändern Sie dieses Verzeichnis nicht.

   c. Legen Sie den Kontonamen der SQL-Server-Datenbank-Engine entweder auf `NT Service \MSSQLSERVER` oder auf `NT AUTHORITY\NETWORK SERVICE` fest.

   d. Stellen Sie den SQL-Server-Startmodus auf Manuell ein.

   e. Wählen Sie für den SQL-Server-Authentifizierungsmodus Gemischt aus.

   f. Behalten Sie die aktuellen Einstellungen für die Standard-Datenverzeichnisse und TempDB-Speicherorte bei.

9. Gewähren Sie `NT AUTHORITY\SYSTEM` die Serverrollenberechtigung SQL Server sysadmin (SA):

```
USE [master]
GO
EXEC master..sp_addsrvrolemember @loginame = N'NT AUTHORITY\SYSTEM' , @rolename =
 N'sysadmin'
GO
```

10. Installieren Sie zusätzliche Software oder passen Sie das Betriebssystem und die Datenbankkonfiguration an Ihre Anforderungen an.

11. Führen Sie Sysprep auf der Instanz aus EC2 . Weitere Informationen finden Sie unter Erstellen eines EC2 Amazon-AMI mit Windows Sysprep.

12. Speichern Sie das AMI, das Ihre installierte SQL-Server-Version, andere Software und Anpassungen enthält. Dies ist Ihr goldenes Image.

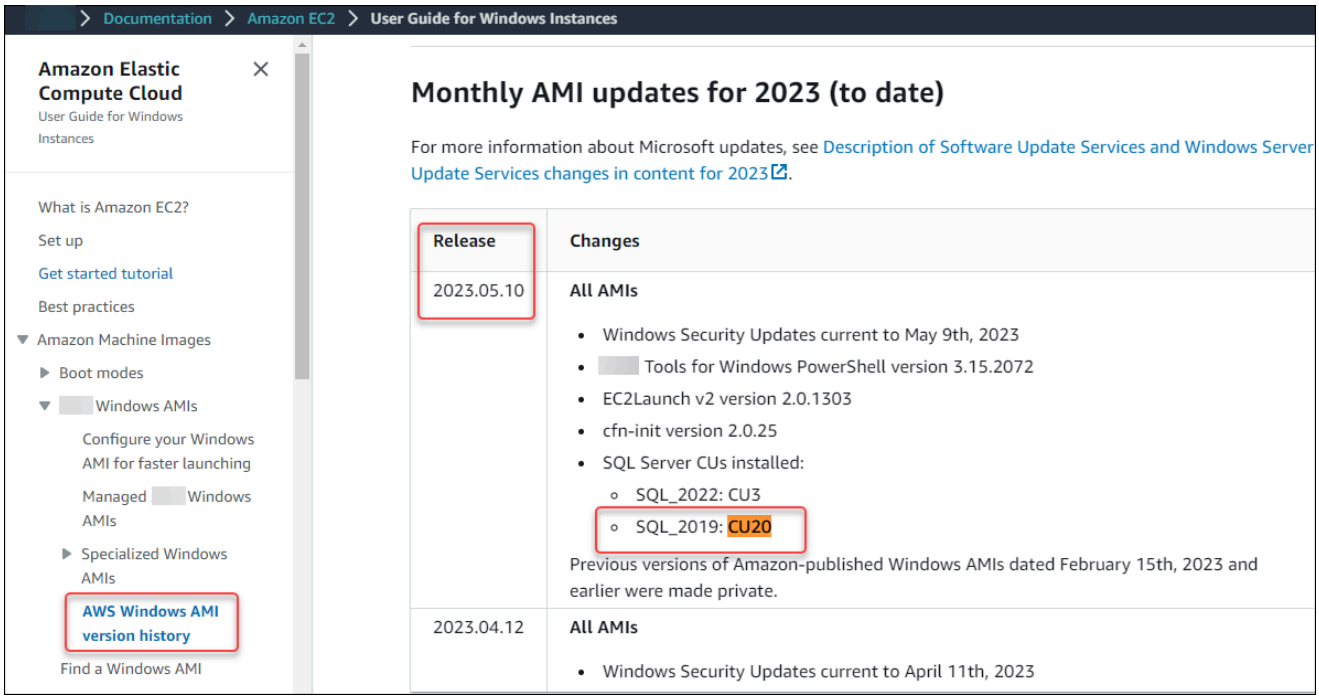
13. Erstellen Sie eine neue CEV, indem Sie die AMI-ID des von Ihnen erstellten Images angeben. Die detaillierten Schritte finden Sie unter Ein CEV für RDS Custom for SQL Server erstellen.
14. Erstellen Sie eine DB-Instance von RDS Custom für SQL Server mithilfe der CEV. Die detaillierten Schritte finden Sie unter Erstellen Sie eine RDS benutzerdefinierte SQL Server-DB-Instance aus einem CEV.

Vorbereitung einer CEV mit vorinstalliertem SQL Server (LI)

Die folgenden Schritte zum Erstellen eines CEV mit vorinstalliertem Microsoft SQL Server (LI) verwenden `2023.05.10` als Beispiel ein AMI mit der SQL Server CU2 0-Versionsnummer. Wenn Sie eine CEV erstellen, wählen Sie ein AMI mit der neuesten Versionsnummer aus. Dadurch wird sichergestellt, dass Sie eine unterstützte Version von Windows Server und SQL Server mit dem neuesten kumulativen Update (CU) verwenden.

So erstellen Sie eine CEV mit dem vorinstallierten Microsoft SQL Server (LI)

1. Wählen Sie das neueste verfügbare AWS EC2 Windows Amazon Machine Image (AMI) mit Lizenz (LI) für Microsoft Windows Server und SQL Server.
   a. Suchen Sie im Windows AMI-Versionsverlauf nach CU20.
   b. Notieren Sie sich die Versionsnummer. Für SQL Server 2019 CU2 0 lautet die Versionsnummer`2023.05.10`.



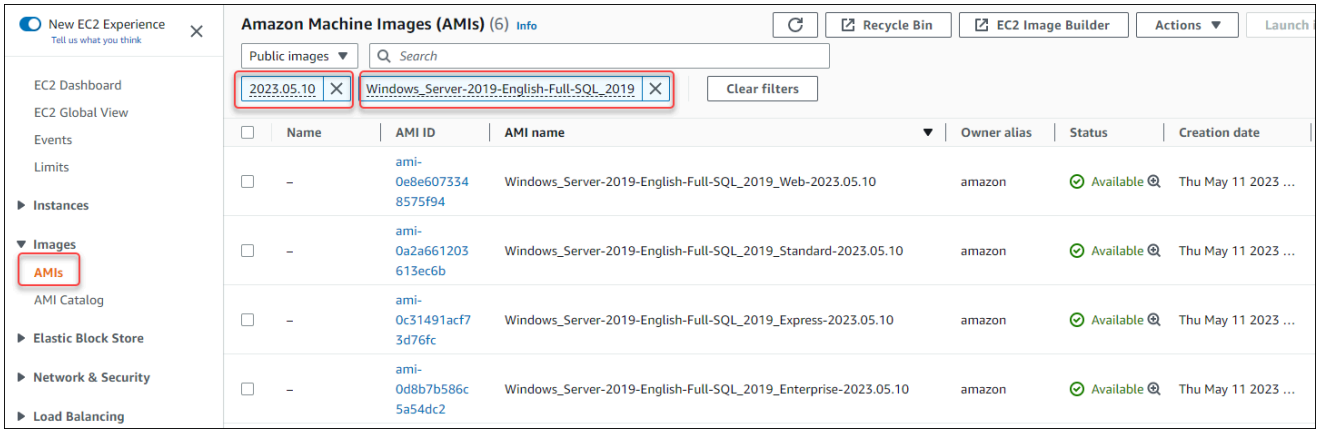
c. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.

d. Wählen Sie dann im linken Navigationsbereich der EC2 Amazon-Konsole Images aus AMIs.

e. Wählen Sie Öffentliche Abbilder aus.

f. Geben Sie `2023.05.10` in das Suchfeld ein. Eine Liste von AMIs wird angezeigt.

g. Geben Sie `Windows_Server-2019-English-Full-SQL_2019` in das Suchfeld ein, um die Ergebnisse zu filtern. Die folgenden Ergebnisse sollten angezeigt werden.



h. Wählen Sie das AMI mit der SQL-Server-Edition aus, die Sie verwenden möchten.

2. Erstellen oder starten Sie eine EC2 Instance von Ihrem ausgewählten AMI aus.

3. Melden Sie sich bei der EC2 Instance an und installieren Sie zusätzliche Software oder passen Sie das Betriebssystem und die Datenbankkonfiguration an Ihre Anforderungen an.

4. Führen Sie Sysprep auf der Instanz aus EC2 . Weitere Informationen zur Vorbereitung eines AMI mit Sysprep finden Sie unter Erstellen eines standardisierten Amazon Machine Image (AMI) mit Sysprep.

5. Speichern Sie das AMI, das Ihre installierte SQL-Server-Version, andere Software und Anpassungen enthält. Dies ist Ihr goldenes Image.

6. Erstellen Sie eine neue CEV, indem Sie die AMI-ID des von Ihnen erstellten Images angeben. Eine detaillierte Anleitung zum Erstellen einer CEV finden Sie unter Ein CEV für RDS Custom for SQL Server erstellen.

7. Erstellen Sie eine DB-Instance von RDS Custom für SQL Server mithilfe der CEV. Die detaillierten Schritte finden Sie unter Erstellen Sie eine RDS benutzerdefinierte SQL Server-DB-Instance aus einem CEV.

Regionale Verfügbarkeit für RDS Custom for SQL Server CEVs

Die CEV-Unterstützung (Custom Engine Version) für RDS Custom for SQL Server ist in den folgenden AWS-Regionen Ländern verfügbar:

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Oregon)
- USA West (Nordkalifornien)
- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Osaka)
- Asien-Pazifik (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Europe (Frankfurt)
- Europa (Irland)
- Europa (London)
- Europe (Paris)
- Europa (Stockholm)
- Südamerika (São Paulo)

Versionsunterstützung für RDS Custom für SQL Server CEVs

Die CEV-Erstellung für RDS Custom for SQL Server wird für die folgenden AWS EC2 AMIs Windows-Betriebssysteme unterstützt:

- Für die CEVs Verwendung vorinstallierter Medien, AWS EC2 Windows AMIs mit Lizenz enthalten (LI), Microsoft Windows Server 2019 (OS) und SQL Server 2022 oder 2019
- Für die CEVs Verwendung von Bring Your Own Media (BYOM), AWS EC2 Windows AMIs mit Microsoft Windows Server 2019 (OS)

Die CEV-Erstellung für RDS Custom für SQL Server wird für die folgenden Betriebssystem- und Datenbankeditionen unterstützt:

- Für die CEVs Verwendung vorinstallierter Medien:
  - SQL Server 2022 Enterprise, Standard oder Web, mit, CU9, -GDR CU13, CU14 -GDR, CU15, CU16 CU17, CU18 und. CU19
  - SQL Server 2019 Enterprise, Standard oder Web, mit, CU8,, CU2 0 CU17 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
- Für die CEVs Verwendung von Bring Your Own Media (BYOM):
  - SQL Server 2022 Enterprise, Standard oder Developer, mit, CU9, -GDR CU13, CU14 -GDR,, CU15 CU16, CU17 und. CU18 CU19
  - SQL Server 2019 Enterprise, Standard oder Developer, mit, CU8, CU17, CU2 0 CU18,, CU24, -GDR CU26, CU28 -GDR, CU29 0 und CU3. CU32
- Für die CEVs Verwendung vorinstallierter Medien oder Bring Your Own Media (BYOM) ist Windows Server 2019 das einzige unterstützte Betriebssystem.

Weitere Informationen finden Sie unter AWS Windows AMI-Versionsverlauf.

Anforderungen für RDS Custom für SQL Server CEVs

Die folgenden Anforderungen gelten für die Erstellung einer CEV für RDS Custom für SQL Server:

- Das AMI, das zur Erstellung einer CEV verwendet wird, basiert auf einer Betriebssystem- und Datenbankkonfiguration, die von RDS Custom für SQL Server unterstützt wird. Weitere Informationen zu unterstützten Konfigurationen finden Sie unter Anforderungen und Einschränkungen für Amazon RDS Custom for SQL Server.
- Die CEV muss einen eindeutigen Namen haben. Sie können keine CEV mit dem gleichen Namen wie eine bereits vorhandene CEV erstellen.
- Sie müssen die CEV anhand eines Benennungsmusters aus SQL Server Hauptversion + Nebenversion + benutzerdefinierter Zeichenfolge benennen. Die Hauptversion + Nebenversion müssen mit der SQL-Server-Version übereinstimmen, die mit dem AMI bereitgestellt wird. Sie können beispielsweise ein AMI mit SQL Server 2019 CU17 als 15.00.4249.2.my_cevtest benennen.
- Sie müssen ein AMI mit Sysprep vorbereiten. Weitere Informationen zur Vorbereitung eines AMI mit Sysprep finden Sie unter Erstellen eines standardisierten Amazon Machine Image (AMI) mit Sysprep.

- Sie sind für die Aufrechterhaltung des Lebenszyklus des AMI verantwortlich. Eine DB-Instance von RDS Custom für SQL Server, die anhand einer CEV erstellt wurde, speichert keine Kopie des AMI. Sie enthält einen Zeiger auf das AMI, das Sie zur Erstellung der CEV verwendet haben. Das AMI muss existieren, damit eine DB-Instance von RDS Custom für SQL Server funktionsfähig bleibt.

Einschränkungen für RDS Custom for SQL Server CEVs

Die folgenden Einschränkungen gelten für die Verwendung benutzerdefinierter Engine-Versionen mit RDS Custom für SQL Server:

- Sie können eine CEV nicht löschen, wenn ihr Ressourcen wie DB-Instances oder DB-Snapshots zugeordnet sind.
- Damit Sie eine DB-Instance von RDS Custom für SQL Server erstellen können, muss eine CEV den Status `pending-validation`, `available`, `failed` oder `validating` haben. Sie können keine DB-Instance von RDS Custom für SQL Server mit einer CEV erstellen, wenn der CEV-Status `incompatible-image-configuration` lautet.
- Wenn Sie eine DB-Instance von RDS Custom für SQL Server so ändern möchten, dass sie eine neue CEV verwendet, muss die CEV den Status `available` haben.
- Sie können kein AMI oder eine CEV von einer bestehenden DB-Instance von RDS Custom für SQL Server aus erstellen.
- Sie können eine bestehende CEV nicht ändern, um ein anderes AMI zu verwenden. Sie können jedoch eine DB-Instance von RDS Custom für SQL Server ändern, um eine andere CEV zu verwenden. Weitere Informationen finden Sie unter Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance.
- Die Verschlüsselung eines AMI oder CEV mit einem vom Kunden verwalteten KMS-Schlüssel, der sich von dem bei der Erstellung der DB-Instance bereitgestellten KMS-Schlüssel unterscheidet, wird nicht unterstützt.
- Eine regionsübergreifende Kopie von CEVs wird nicht unterstützt.
- Das kontoübergreifende Kopieren von wird CEVs nicht unterstützt.
- Eine einmal gelöschte CEV kann nicht mehr wiederhergestellt werden. Sie können jedoch eine neue CEV aus demselben AMI erstellen.
- Eine DB-Instance von RDS Custom für SQL Server speichert Ihre SQL-Server-Datenbankdateien im Laufwerk D:\. Das einer CEV zugeordnete AMI sollte die Systemdatenbankdateien von Microsoft SQL Server im Laufwerk C:\ speichern.

- Eine DB-Instance von RDS Custom für SQL Server behält Ihre an SQL Server vorgenommenen Konfigurationsänderungen bei. Konfigurationsänderungen am Betriebssystem einer laufenden DB-Instance von RDS Custom für SQL Server, die aus einer CEV erstellt wurde, werden nicht beibehalten. Wenn Sie eine permanente Konfigurationsänderung am Betriebssystem vornehmen müssen und diese als neue Basiskonfiguration beibehalten möchten, erstellen Sie eine neue CEV und ändern Sie die DB-Instance, um die neue CEV zu verwenden.

  > **Important**
  >
  > Das Ändern einer DB-Instance von RDS Custom für SQL Server zur Verwendung einer neuen CEV ist ein Offline-Vorgang. Sie können die Änderung sofort durchführen oder sie so planen, dass sie während eines wöchentlichen Wartungsfensters erfolgt.

- Wenn Sie eine CEV ändern, übergibt Amazon RDS diese Änderungen nicht an die zugehörigen DB-Instances von RDS Custom für SQL Server. Sie müssen jede DB-Instance von RDS Custom für SQL Server ändern, um die neue oder aktualisierte CEV zu verwenden. Weitere Informationen finden Sie unter Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance.

- > **Important**
  >
  > Wenn ein von einem CEV verwendetes AMI gelöscht wird, schlagen alle Änderungen fehl, die möglicherweise einen Host-Ersatz erfordern, z. B. Rechenleistung skalieren. Die DB-Instance von RDS Custom für SQL Server wird dann außerhalb des RDS-Support-Umfangs platziert. Wir empfehlen, das Löschen von AMIs zu vermeiden, die mit einer CEV verknüpft sind.

Ein CEV für RDS Custom for SQL Server erstellen

Sie können eine benutzerdefinierte Engine-Version (CEV) mit dem AWS Management Console oder dem erstellen AWS CLI. Anschließend können Sie die verwendenCEV, um eine RDS benutzerdefinierte SQL Server-DB-Instance zu erstellen.

Stellen Sie sicher, dass sich das Amazon Machine Image (AMI) in demselben AWS Konto und derselben Region wie Ihr befindetCEV. Andernfalls schlägt der Vorgang zum Erstellen eines CEV fehl.

Weitere Informationen finden Sie unter Eine DB-Instance für Amazon RDS Custom for SQL Server erstellen und eine Verbindung zu ihr herstellen.

> **Important**
>
> Die Schritte zum Erstellen einer Datei entsprechen CEV denen, die mit einem vorinstallierten SQL Server AMIs erstellt wurden, und denen, die mit Bring Your Own Media (BYOM) erstellt wurden.

Konsole

Um ein zu erstellen CEV

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus.Benutzerdefinierte Engine-Versionenaus.

   Auf der Seite Benutzerdefinierte Engine-Versionen werden alle derzeit CEVs vorhandenen Engine-Versionen angezeigt. Wenn Sie keine erstellt habenCEVs, ist die Tabelle leer.
3. Klicken Sie auf Erstellen einer benutzerdefinierten Engine-Version.
4. Wählen Sie als Engine-Typ Microsoft SQL Server aus.
5. Wählen Sie unter Edition die DB-Engine-Edition aus, die Sie verwenden möchten.
6. Wählen Sie unter Hauptversion die Hauptversion der Engine aus, die auf Ihrem installiert istAMI.
7. In :Details zur VersionGeben Sie einen gültigen Namen unterName der benutzerdefinierten Engineaus.

   Das Namenformat lautet*major-engine-version*.*minor-engine-version*.*customized_string*aus. Er darf nur 1-50 alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (-. _) enthalten. So könnten Sie beispielsweise **15.00.4249.2.my_cevtest** eingeben.

   Geben Sie optional eine Beschreibung für Ihre einCEV.
8. Suchen Sie für das Installationsmedium nach der AMI ID, mit der Sie das erstellen möchten, oder geben Sie sie ein. CEV
9. Fügen Sie im Abschnitt Tags alle Tags hinzu, um die zu identifizierenCEV.
10. Klicken Sie auf Erstellen einer benutzerdefinierten Engine-Version.

DieBenutzerdefinierte Engine-Versionen-Seite wird angezeigt. Ihr CEV wird mit dem Status ausstehende Validierung angezeigt

AWS CLI

Um eine mit CEV dem zu erstellen AWS CLI, führen Sie den Befehl -version aus. create-custom-db-engine

Die folgenden Optionen sind erforderlich:

- `--engine`
- `--engine-version`
- `--image-id`

Sie können auch die folgenden Optionen angeben:

- `--description`
- `--region`
- `--tags`

Im folgenden Beispiel wird ein CEV `15.00.4249.2.my_cevtest` benanntes erstellt. Stellen Sie sicher, dass Ihr Name mit der Versionsnummer der Hauptengine CEV beginnt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --image-id ami-0r93cx31t5r596482 \
    --description "Custom SQL Server EE 15.00.4249.2 cev test"
```

Die folgende Teilausgabe zeigt die Engine, die Parametergruppen und andere Informationen.

```
"DBEngineVersions": [
    {
```

```
    "Engine": "custom-sqlserver-ee",
    "MajorEngineVersion": "15.00",
    "EngineVersion": "15.00.4249.2.my_cevtest",
    "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for RDS Custom for
 SQL Server",
    "DBEngineVersionArn": "arn:aws:rds:us-east-1:<my-account-id>:cev:custom-sqlserver-
ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
    "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",

    "Image": [
        "ImageId": "ami-0r93cx31t5r596482",
        "Status": "pending-validation"
     ],
    "CreateTime": "2022-11-20T19:30:01.831000+00:00",
    "SupportsLogExportsToCloudwatchLogs": false,
    "SupportsReadReplica": false,
    "Status": "pending-validation",
    "SupportsParallelQuery": false,
    "SupportsGlobalDatabases": false,
    "TagList": []
    }
]
```

Wenn der Prozess zur Erstellung einer CEV fehlschlägt, gibt RDS Custom for SQL Server ein Problem `RDS-EVENT-0198` mit der Nachricht aus`Creation failed for custom engine version` *`major-engine-version.cev_name`*. Die Nachricht enthält Details über den Fehler. Das Ereignis druckt z. B. fehlende Dateien. Anregungen zur Behebung von Problemen bei der CEV Erstellung finden Sie unterBeheben von CEV-Fehlern für RDS Custom für SQL Server.

Erstellen Sie eine RDS benutzerdefinierte SQL Server-DB-Instance aus einem CEV

Nachdem Sie eine erfolgreich erstellt habenCEV, wird der CEVStatus angezeigt`pending-validation`. Sie können jetzt mit dem eine neue RDS benutzerdefinierte SQL Server-DB-Instance erstellenCEV. Informationen zum Erstellen einer neuen RDS benutzerdefinierten SQL Server-DB-Instance aus einer CEV finden Sie unterEine RDS benutzerdefinierte SQL Server-DB-Instance erstellen.

Lebenszyklus eines CEV

Der CEV Lebenszyklus umfasst die folgenden Status.

CEVStatus	Beschreibung	Vorschläge für die Fehlerbehebung	
`pending-validation`	A CEV wurde erstellt und die Validierung des zugehörigen steht noch ausAMI. A CEV bleibt so lange aktiv, `pending-validation` bis daraus eine RDS Custom for SQL Server-DB-Instance erstellt wird.	Wenn keine Aufgaben vorhanden sind, erstellen Sie eine neue RDS benutzerdefinierte SQL Server-DB-Instance aus demCEV. Beim Erstellen der RDS Custom for SQL Server-DB-Instance versucht das System, die zugehörige AMI für eine zu validierenCEV.	
`validating`	Eine Erstellungsaufgabe für die RDS Custom for SQL Server-DB-Instance, die auf einer neuen basiert, CEV ist im Gange. Beim Erstellen der RDS Custom for SQL Server-DB-Instance versucht das System, die zugehörige AMI DB-Instance zu überprüfenCEV.	Warten Sie, bis die Erstellungsaufgabe der vorhandenen RDS Custom for SQL Server-DB-Instance abgeschlossen ist. Sie können die RDS EVENTS Konsole verwenden , um detaillierte Ereignismeldungen zur Fehlerbehebung zu überprüfen.	
`available`	Das CEV wurde erfolgreich	Für CEV ist keine zusätzliche Validierung erforderlich. Es kann verwendet werden,	

CEVStatus	Beschreibung	Vorschläge für die Fehlerbehebung	
	validiert. A CEV wird in den `available` Status wechseln, sobald daraus erfolgreich eine RDS Custom for SQL Server-DB-Instance erstellt wurde.	um zusätzliche RDS benutzerdefinierte DB-Instances für SQL Server zu erstellen oder bestehende zu ändern.	
`inactive`	Der CEV wurde in einen inaktiven Zustand geändert.	Damit können Sie keine RDS benutzerd efinierte DB-Instance erstellen oder aktualisi erenCEV. Außerdem können Sie damit keinen DB-Snapshot wiederherstellen, um eine neue RDS benutzerdefinierte DB-Instance zu erstellenCEV. Informati onen zum Ändern des Status in `ACTIVE` finden Sie unter Ändern eines CEV für RDS Benutzerdefiniert für SQL Server.	

CEVStatus	Beschreibung	Vorschläge für die Fehlerbehebung	
`failed`	Der Schritt „DB-Instance erstellen“ schlug hierfür fehl, CEV bevor er die validiere n konnteAMI . Alternativ befindet sich der von der AMI verwendete Basiswert CEV nicht in einem verfügbaren Status.	Beheben Sie die Ursache dafür, dass das System die DB-Instance nicht erstellen konnte. Sehen Sie sich die ausführliche Fehlermeldung an und versuchen Sie erneut, eine neue DB-Instance zu erstellen . Stellen Sie sicher, dass sich der von der AMI verwendete Basiswert in einem verfügbaren Status CEV befindet.	

CEVStatus	Beschreibung	Vorschläge für die Fehlerbehebung	
`incompatible-image-configuration`	Bei der Validierung von ist ein Fehler aufgetreten. AMI	Sehen Sie sich die technischen Details des Fehlers an. Sie können nicht CEV erneut versuchen, das AMI damit zu validieren. Sehen Sie sich die folgenden Empfehlungen an: • Vergewissern Sie CEV sich, dass Ihr Name nach dem erforderlichen Benennungsmuster SQL Server-Haupt version + Nebenversion und benutzerdefinierter Zeichenfolge benannt ist. • Stellen Sie sicher, dass die SQL Serverversion im CEV Namen mit der Version übereinstimmt, die im Lieferumfang von enthalten istAMI. • Stellen Sie sicher, dass die Build-Version des Betriebssystems der mindestens erforderlichen Build-Version entspricht. • Stellen Sie sicher, dass die Build-Version des Betriebssystems der mindestens erforderlichen Hauptversion entspricht. Erstellen Sie eine neue CEV mit den richtigen Informationen. Erstellen Sie bei Bedarf eine neue EC2 Instanz mit einer unterstützten Instanz AMI und führen Sie den Sysprep-Prozess darauf aus.	

Ändern eines CEV für RDS Benutzerdefiniert für SQL Server

Sie können a CEV mit dem AWS Management Console oder dem ändern AWS CLI. Sie können die CEV Beschreibung oder den Verfügbarkeitsstatus ändern. Ihr CEV hat einen der folgenden Statuswerte:

- `available`— Sie können dies verwendenCEV, um eine neue RDS benutzerdefinierte DB-Instance zu erstellen oder eine DB-Instance zu aktualisieren. Dies ist der Standardstatus für eine neu erstellteCEV.
- `inactive`— Damit können Sie keine RDS benutzerdefinierte DB-Instance erstellen oder aktualisierenCEV. Sie können damit keinen DB-Snapshot wiederherstellen, um eine neue RDS benutzerdefinierte DB-Instance zu erstellenCEV.

Sie können den CEV Status von `available` bis `inactive` oder von `inactive` bis ändern`available`. Sie können den Status ändern, `INACTIVE` um die versehentliche Verwendung eines zu verhindern CEV oder um ein Produkt, das nicht mehr verwendet wurde, wieder zur Verwendung CEV berechtigt zu machen.

Konsole

So ändern Sie ein CEV

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus.Benutzerdefinierte Engine-Versionenaus.
3. Wählen Sie eine ausCEV, deren Beschreibung oder Status Sie ändern möchten.
4. Wählen Sie für Actions (Aktionen) die Option Modify (Ändern) aus.
5. Nehmen Sie eine oder alle der folgenden Änderungen vor:
   - Wählen Sie für die CEVStatuseinstellungen einen neuen Verfügbarkeitsstatus aus.
   - Geben Sie auf der Seite Update description (Beschreibung aktualisieren) eine Beschreibung für die neue Version ein.
6. Wählen Sie ÄndernCEV aus.

   Wenn der verwendet CEV wird, zeigt die Konsole den Status Sie können den CEV Status nicht ändern an. Beheben Sie die Probleme und versuchen Sie es erneut.

DieBenutzerdefinierte Engine-Versionen-Seite wird angezeigt.

AWS CLI

Um a mit CEV dem zu ändern AWS CLI, führen Sie den Befehl modify-custom-db-engine-version aus. Sie finden CEVs die Änderung, indem Sie den describe-db-engine-versionsBefehl ausführen.

Die folgenden Optionen sind erforderlich:

- `--engine`
- `--engine-version` *`cev`*, wo *`cev`* ist der Name der benutzerdefinierten Engine-Version, die Sie ändern möchten
- `--status` *`status`*, wo *`Status`* ist der Verfügbarkeitsstatus, den Sie dem zuweisen möchten CEV

Im folgenden Beispiel wird der aktuelle Status eines CEV Namens `15.00.4249.2.my_cevtest` auf geändert`inactive`.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --status inactive
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest ^
    --status inactive
```

Informationen dazu, wie Sie eine RDS benutzerdefinierte SQL Server-DB-Instance so ändern, dass sie eine neue verwendetCEV, finden Sie unterÄndern einer RDS benutzerdefinierten SQL Server-DB-Instance zur Verwendung einer neuen CEV.

Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance zur Verwendung einer neuen CEV

Sie können eine bestehende RDS Custom for SQL Server-DB-Instance so ändern, dass sie eine andere verwendetCEV. Sie können u. a. folgende Änderungen vornehmen:

- Ändern der CEV
- Ändern der -Instance-Klasse
- Ändern des Aufbewahrungszeitraum für Backups und des Backup-Fensters
- Ändern des Wartungsfensters

Konsole

Um eine RDS benutzerdefinierte SQL Server-DB-Instance zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Nehmen Sie nach Bedarf die folgenden Änderungen vor:
    a. Wählen Sie für die DB-Engine-Version eine andereCEV.
    b. Ändern Sie den Wert für DB instance class. Informationen zu unterstützten Klassen finden Sie unter Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server.
    c. Ändern Sie den Wert für Aufbewahrungszeitraum für Backups aus.
    d. Für Backup window, setzen Sie Werte für Beginnzeit und Duration (Dauer) ein.
    e. Für Wartungsfenster der DB-Instance, setzen Sie Werte für Starttag, Beginnzeit und Duration (Dauer) ein.
6. Klicken Sie auf Weiter.
7. Wählen Sie During the next scheduled maintenance window (Während des nächsten geplanten Wartungsfensters) oder Sofort aus.
8. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.

> **Note**
>
> Wenn Sie eine DB-Instance von einer CEV auf eine andere ändernCEV, z. B. beim Upgrade einer Nebenversion, werden die SQL Serversystemdatenbanken, einschließlich ihrer Daten und Konfigurationen, aus der aktuellen RDS Custom for SQL Server-DB-Instance übernommen.

AWS CLI

Um eine DB-Instance mithilfe CEV von so zu ändern, dass sie eine andere verwendet AWS CLI, führen Sie den modify-db-instanceBefehl aus.

Die folgenden Optionen sind erforderlich:

- `--db-instance-identifier`
- `--engine-version` *`cev`*, wo *`cev`* ist der Name der benutzerdefinierten Engine-Version, zu der Sie die DB-Instance ändern möchten.

Im folgenden Beispiel wird eine DB-Instance mit dem Namen „named“ so geändert`my-cev-db-instance`, dass sie einen CEV Namen verwendet, `15.00.4249.2.my_cevtest_new` und die Änderung wird sofort angewendet.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cev-db-instance \
    --engine-version 15.00.4249.2.my_cevtest_new \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cev-db-instance ^
    --engine-version 15.00.4249.2.my_cevtest_new ^
    --apply-immediately
```

Anzeigen von CEV-Details zu Amazon RDS Custom für SQL Server

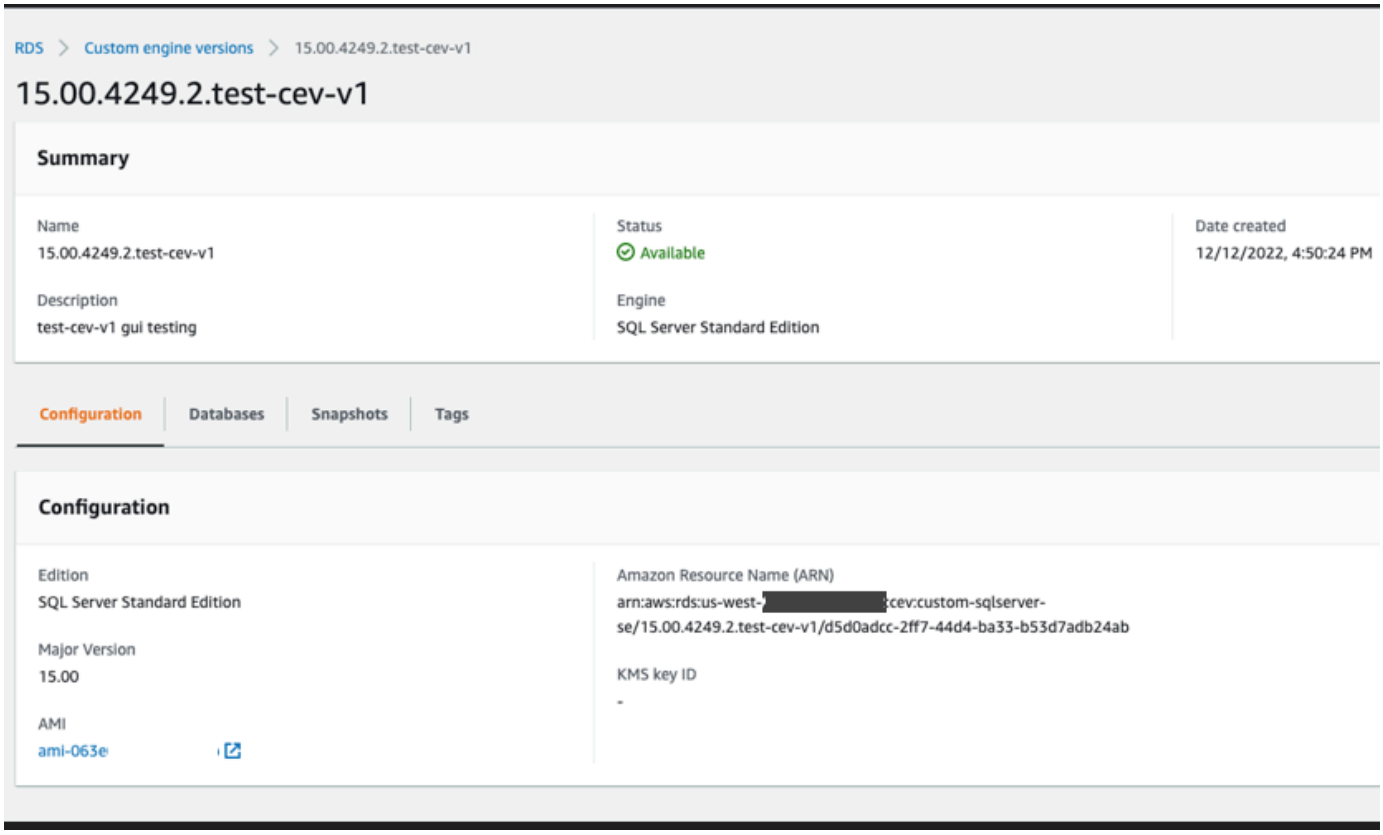
Sie können Details zu Ihrer CEV einsehen, indem Sie die AWS Management Console oder die AWS CLI verwenden.

Konsole

So zeigen Sie CEV-Details an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus.Benutzerdefinierte Engine-Versionenaus.

   DieBenutzerdefinierte Engine-VersionenSeite zeigt alle CEVs an, die derzeit existieren. Wenn Sie keine CEVs erstellt haben, ist die Seite leer.
3. Wählen Sie den Namen der CEV, die Sie anzeigen möchten.
4. Wählen Sie Configuration (Konfiguration) aus, um die Details anzeigen zu lassen.



AWS CLI

Führen Sie den Befehl describe-db-engine-versions aus, um Details zur CEV mit der AWS CLI anzuzeigen.

Sie können auch die folgenden Optionen angeben:

- `--include-all`, um alle CEVs mit einem beliebigen Lebenszyklusstatus anzuzeigen. Ohne die Option `--include-all` werden nur die CEVs zurückgegeben, die sich im Lebenszyklusstatus `available` befinden.

```
aws rds describe-db-engine-versions --engine custom-sqlserver-ee --engine-version
 15.00.4249.2.my_cevtest --include-all
{
    "DBEngineVersions": [
        {
            "Engine": "custom-sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "EngineVersion": "15.00.4249.2.my_cevtest",
            "DBParameterGroupFamily": "custom-sqlserver-ee-15.0",
            "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for custom
 RDS",
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:{my-account-id}:cev:custom-
sqlserver-ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
            "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",
            "Image": {
                "ImageId": "ami-0r93cx31t5r596482",
                "Status": "pending-validation"
            },
            "DBEngineMediaType": "AWS Provided",
            "CreateTime": "2022-11-20T19:30:01.831000+00:00",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": false,
            "SupportedFeatureNames": [],
            "Status": "pending-validation",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "TagList": [],
            "SupportsBabelfish": false
        }
    ]
```

```
}
```

Sie können Filter verwenden, um CEVs mit einem bestimmten Lebenszyklusstatus anzuzeigen. Zum Beispiel, um CEVs anzuzeigen, die den Lebenszyklusstatus `pending-validation`, `available` oder `failed` haben:

```
aws rds describe-db-engine-versions engine custom-sqlserver-ee
                region us-west-2 include-all query 'DBEngineVersions[?Status ==
 pending-validation ||
                Status == available || Status == failed]'
```

Löschen einer CEV für RDS Custom für SQL Server

Sie können eine CEV mithilfe der AWS Management Console oder der AWS CLI löschen. Dies dauert in der Regel einige Minuten.

Bevor Sie eine CEV löschen, stellen Sie sicher, dass sie von keiner der folgenden Optionen verwendet wird:

- Stopp einer RDS-DB-Instance.
- Ein Snapshot einer RDS Custom DB-Instance
- Eine automatisierte Sicherung Ihrer RDS Custom DB-Instance

Konsole

So löschen Sie eine VPC

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich und dann aus. Benutzerdefinierte Engine-Versionenaus.
3. Wählen Sie eine CEV aus, deren Beschreibung oder Status Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.

   Die Dialogbox Löschen***cev_name***? wird angezeigt.
5. Geben Sie **`delete me`** ein und klicken Sie auf Delete (Löschen).

   In derBenutzerdefinierte Engine-Versionen-Seite zeigt das Banner, dass Ihre CEV gelöscht wird.

AWS CLI

Um eine CEV mithilfe der zu löschenAWS CLI, führen Sie den Befehl delete-custom-db-engine-version aus.

Die folgenden Optionen sind erforderlich:

- `--engine custom-sqlserver-ee`
- `--engine-version` *cev*, wobei*cev*ist der Name der zu löschenden benutzerdefinierten Engine-Version

Im folgenden Beispiel wird ein Tresor namens `15.00.4249.2.my_cevtest` gelöscht.

Example

Für Linux, macOSoder Unix:

```
aws rds delete-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest
```

Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest
```

Eine DB-Instance für Amazon RDS Custom for SQL Server erstellen und eine Verbindung zu ihr herstellen

Sie können eine RDS benutzerdefinierte DB-Instance erstellen und dann mithilfe AWS Systems Manager des Remote Desktop Protocol (RDP) eine Verbindung zu ihr herstellen.

> **Important**
>
> Bevor Sie eine RDS Custom for SQL Server-DB-Instance erstellen oder eine Verbindung zu einer solchen herstellen können, stellen Sie sicher, dass Sie die unter aufgeführten Aufgaben abgeschlossen habenEinrichten Ihrer Umgebung für Amazon RDS Custom for SQL Server.
> Sie können RDS benutzerdefinierte DB-Instances kennzeichnen, wenn Sie sie erstellen, erstellen oder ändern Sie jedoch nicht das `AWSRDSCustom` Tag, das für die RDS benutzerdefinierte Automatisierung erforderlich ist. Weitere Informationen finden Sie unter Markieren von Ressourcen für RDS Custom for SQL Server.
> Wenn Sie zum ersten Mal eine RDS benutzerdefinierte SQL Server-DB-Instance erstellen, wird möglicherweise die folgende Fehlermeldung angezeigt: Die dienstbezogene Rolle wird gerade erstellt. Bitte versuchen Sie es später erneut. Wenn Sie dies der Fall sind, warten Sie einige Minuten und versuchen Sie dann erneut, die DB-Instance zu erstellen.

Themen

- Eine RDS benutzerdefinierte SQL Server-DB-Instance erstellen
- RDSBenutzerdefinierte serviceverknüpfte Rolle
- Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit AWS Systems Manager
- Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit RDP

Eine RDS benutzerdefinierte SQL Server-DB-Instance erstellen

Erstellen Sie eine Amazon RDS Custom for SQL Server-DB-Instance mit entweder dem AWS Management Console oder dem AWS CLI. Das Verfahren ähnelt dem Verfahren zum Erstellen einer Amazon RDS DB-Instance.

Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Konsole

Um eine RDS benutzerdefinierte SQL Server-DB-Instance zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie Create database (Datenbank erstellen) aus.
4. Wählen Sie Standard-Erstellung als Datenbankerstellungsmethode.
5. Wählen Sie für Engine-Optionen Microsoft SQL Server als Engine-Typ aus.
6. Wählen Sie als Datenbankverwaltungstyp Amazon RDS Custom aus.
7. Wählen Sie unter Edition die Edition der SQL Server-DB-Engine aus, die Sie verwenden möchten.
8. (Optional) Wenn Sie beabsichtigen, die DB-Instance aus einer zu erstellenCEV, aktivieren Sie das Kontrollkästchen Benutzerdefinierte Engine-Version verwenden (CEV). Wählen Sie CEV in der Drop-down-Liste Ihre aus.
9. Behalten Sie für Datenbankversion den Standardwert Version bei.
10. Wählen Sie für VorlagenProduktion.
11. Geben Sie unter Einstellungen einen neuen Namen für die DB-Instance-Kennung ein.
12. Gehen Sie wie folgt vor, um Ihr Masterpasswort einzugeben:

    a. Öffnen Sie im Abschnitt Settings (Einstellungen) die Option Credential Settings (Einstellungen zu Anmeldeinformationen).
    b. Deaktivieren Sie das Kontrollkästchen Auto generate a password (Automatisch ein Passwort generieren).
    c. (Optional) Ändern Sie den Wert für Master username (Masterbenutzername) und geben Sie in Master password (Masterpasswort) und Confirm password (Passwort bestätigen) dasselbe Passwort ein.

    Standardmäßig verwendet die neue RDS benutzerdefinierte DB-Instance ein automatisch generiertes Passwort für den Masterbenutzer.
13. In der Größe der DB-Instance wählen Sie einen Wert für DB instance class aus.

    Informationen zu unterstützten Klassen finden Sie unter Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server.

14. Klicken Sie aufSpeicher-Einstellungen.
15. Gehen Sie für RDSbenutzerdefinierte Sicherheit wie folgt vor:

    a. IAMBeispielsweise haben Sie zwei Möglichkeiten, das Instance-Profil für Ihre RDS Custom for SQL Server-DB-Instance auszuwählen.

       1. Wählen Sie Neues Instance-Profil erstellen und geben Sie ein Namenssuffix für das Instance-Profil ein. Weitere Informationen finden Sie unter Automatisierte Erstellung von Instanzprofilen mit dem AWS Management Console.
       2. Wählen Sie ein vorhandenes Instanzprofil aus. Wählen Sie aus der Dropdownliste ein Instanzprofil aus, das mit beginnt. `AWSRDSCustom`

    b. Wählen Sie für Verschlüsselung die Option Schlüssel eingeben aus, ARN um die verfügbaren AWS KMS Schlüssel aufzulisten. Wählen Sie dann Ihren Schlüssel aus der Liste aus.

       Für RDS Benutzerdefiniert ist ein AWS KMS Schlüssel erforderlich. Weitere Informationen finden Sie unter Stellen Sie sicher, dass Sie über einen symmetrischen AWS KMS Verschlüsselungsschlüssel verfügen.

16. Geben Sie für die verbleibenden Abschnitte Ihre bevorzugten RDS benutzerdefinierten DB-Instance-Einstellungen an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances. Die folgenden Einstellungen werden nicht in der Konsole angezeigt und werden nicht unterstützt:

    - Prozessorfunktionen
    - Automatische Speicherskalierung
    - Verfügbarkeit und Beständigkeit
    - Passwort und Kerberos-AuthentifizierungOption inDatenbankauthentifizierung(nurPasswortauthentifizierungwird unterstützt)
    - DatenbankoptionenGruppe inZusätzliche Konfiguration
    - Performance Insights
    - Protokollexporte
    - Auto minor version upgrade (Upgrade einer Unterversion automatisch durchführen)
    - Löschschutz

    Aufbewahrungszeitraum für Backups wird unterstützt, aber Sie können nicht 0 Tage auswählen.

17. Wählen Sie Datenbank erstellen aus.

    DieAnzeigen von Anmeldeinformationenerscheint auf der SchaltflächeDatenbankenangezeigten.

    Um den Master-Benutzernamen und das Passwort für die RDS benutzerdefinierte DB-Instance anzuzeigen, wählen Sie Credential-Details anzeigen.

    Verwenden Sie den angezeigten Benutzernamen und das angezeigte Passwort, um eine Verbindung zu DB-Instance als Hauptbenutzer herzustellen.

    > **Important**
    >
    > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern. Um das Masterbenutzer-Passwort zu ändern, nachdem die RDS benutzerdefinierte DB-Instance verfügbar ist, ändern Sie die DB-Instance. Weitere Informationen zum Ändern einer DB-Instance finden Sie unter Verwaltung einer Amazon RDS Custom for SQL Server-DB-Instance.

18. Wählen Sie Datenbanken, um die Liste der RDS benutzerdefinierten DB-Instances anzuzeigen.
19. Wählen Sie die RDS benutzerdefinierte DB-Instance aus, die Sie gerade erstellt haben.

    Auf der RDS Konsole werden die Details für die neue RDS benutzerdefinierte DB-Instance angezeigt:

    - Die DB-Instance hat den Status Wird erstellt, bis die RDS benutzerdefinierte DB-Instance erstellt und einsatzbereit ist. Wenn sich der Status in available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und dem dieser zugeteilten Speicher kann es einige Minuten dauern, bis die neue DB-Instance verfügbar ist.
    - Die Rolle hat den Wert Instance (RDSCustom).
    - RDSDer benutzerdefinierte Automatisierungsmodus hat den Wert Vollständige Automatisierung. Diese Einstellung bedeutet, dass die DB-Instance eine automatische Überwachung und Instanzwiederherstellung bietet.

AWS CLI

Sie erstellen eine RDS benutzerdefinierte DB-Instance mithilfe des create-db-instance AWS CLI Befehls.

Die folgenden Optionen sind erforderlich:

- `--db-instance-identifier`
- `--db-instance-class` (eine Liste der unterstützten Klassen, finden Sie unter Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server.
- `--engine` (`custom-sqlserver-ee`, `custom-sqlserver-se` oder `custom-sqlserver-web`)
- `--kms-key-id`
- `--custom-iam-instance-profile`

Im folgenden Beispiel wird eine RDS benutzerdefinierte SQL Server-DB-Instance mit dem Namen erstelltmy`-custom-instance`. Der Aufbewahrungszeitraum für Backups beträgt 3 Tage.

> **Note**
>
> Um eine DB-Instance aus einer benutzerdefinierten Engine-Version (CEV) zu erstellen, geben Sie dem `--engine-version` Parameter einen vorhandenen CEV Namen an. Beispiel: `--engine-version 15.00.4249.2.my_cevtest`

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4073.23.v1 \
    --db-instance-identifier my-custom-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 20 \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --no-multi-az \
    --port 8200 \
    --kms-key-id mykmskey \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4073.23.v1 ^
    --db-instance-identifier my-custom-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 20 ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --no-multi-az ^
    --port 8200 ^
    --kms-key-id mykmskey ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Rufen Sie Details zu Ihrer Instance ab, indem Sie die`describe-db-instances`befehl.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Die folgende Teilausgabe zeigt die Engine, die Parametergruppen und andere Informationen.

```
{
    "DBInstances": [
        {
            "PendingModifiedValues": {},
            "Engine": "custom-sqlserver-ee",
            "MultiAZ": false,
            "DBSecurityGroups": [],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-sqlserver-ee-15",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            "AutomationMode": "full",
            "DBInstanceIdentifier": "my-custom-instance",
            "TagList": []
```

```
        }
    ]
}
```

RDSBenutzerdefinierte serviceverknüpfte Rolle

Eine serviceverknüpfte Rolle gewährt Amazon RDS Custom Zugriff auf Ressourcen in Ihrem AWS-Konto. Dies erleichtert die Verwendung von RDS Custom, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. RDSBenutzerdefiniert die Berechtigungen seiner dienstbezogenen Rollen, und sofern nicht anders definiert, kann nur RDS Custom seine Rollen übernehmen. Zu den definierten Berechtigungen gehören die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen IAM Entität zugeordnet werden.

Wenn Sie eine RDS benutzerdefinierte DB-Instance erstellen, werden sowohl die mit Amazon RDS als auch die mit dem Service verknüpfte Rolle RDS Custom erstellt (sofern sie noch nicht vorhanden sind) und verwendet. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon RDS.

Wenn Sie zum ersten Mal eine RDS Custom for SQL Server-DB-Instance erstellen, wird möglicherweise die folgende Fehlermeldung angezeigt: Die serviceverknüpfte Rolle wird gerade erstellt. Bitte versuchen Sie es später erneut. Wenn Sie dies der Fall sind, warten Sie einige Minuten und versuchen Sie dann erneut, die DB-Instance zu erstellen.

Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit AWS Systems Manager

Nachdem Sie Ihre RDS benutzerdefinierte DB-Instance erstellt haben, können Sie mit AWS Systems Manager Session Manager eine Verbindung zu ihr herstellen. Session Manager ist eine Systems Manager Manager-Funktion, mit der Sie EC2 Amazon-Instances über eine browserbasierte Shell oder über die AWS CLI verwalten können. Weitere Informationen erhalten Sie unter AWS Systems Manager Session Manager.

Konsole

Herstellen einer Verbindung mit Ihrer Instance mithilfe von Session Manager

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte DB-Instance aus, zu der Sie eine Verbindung herstellen möchten.
3. Wählen Sie Konfiguration.

4. Beachten Sie dieRessourcen-IDWert für Ihre DB-Instance. Die Ressourcen-ID kann beispielsweise `db-ABCDEFGHIJKLMNOPQRS0123456` sein.
5. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
6. Wählen Sie im Navigationsbereich Instances aus.
7. Suchen Sie nach dem Namen Ihrer EC2 Instance und wählen Sie dann die damit verknüpfte Instance-ID aus. Die Instance-ID kann beispielsweise `i-abcdefghijklm01234` sein.
8. Wählen Sie Connect aus.
9. Klicken Sie auf Session Manager.
10. Wählen Sie Connect aus.

    Es öffnet sich ein Fenster für Ihre Sitzung.

AWS CLI

Sie können mit dem eine Verbindung zu Ihrer RDS benutzerdefinierten DB-Instance herstellen AWS CLI. Für diese Technik ist das Session Manager-Plugin für die AWS CLI. Informationen zum Installieren des Plugins finden Sie unter Installieren Sie das Session Manager-Plugin für das AWS CLI.

Um die DB-Ressourcen-ID Ihrer RDS benutzerdefinierten DB-Instance zu finden, verwenden Sie`describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Die folgende Beispielausgabe zeigt die Ressourcen-ID für Ihre RDS benutzerdefinierte Instance. Das Präfix lautet `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Um die EC2 Instance-ID Ihrer DB-Instance zu ermitteln, verwenden Sie`aws ec2 describe-instances`. Im folgenden Beispiel wird verwendet`db-ABCDEFGHIJKLMNOPQRS0123456`für die Ressourcen-ID.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
```

```
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

Die folgende Beispielausgabe zeigt die EC2 Instance-ID.

```
i-abcdefghijklm01234
```

Verwenden Sie den `aws ssm start-session` Befehl und geben Sie die EC2 Instanz-ID im `--target` Parameter an.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Ein erfolgreiches Ergebnis sieht wie folgt aus.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit RDP

Nachdem Sie Ihre RDS benutzerdefinierte DB-Instance erstellt haben, können Sie über einen RDP Client eine Verbindung zu dieser Instance herstellen. Das Verfahren ist das gleiche wie beim Herstellen einer Verbindung zu einer EC2 Amazon-Instance. Weitere Informationen finden Sie unter Verbinden mit Ihrer Windows-Instance.

Um eine Verbindung zur DB-Instance herzustellen, benötigen Sie das key pair, das der Instance zugeordnet ist. RDSBenutzerdefiniert erstellt das key pair für Sie. Der Paarname verwendet das Präfix `do-not-delete-rds-custom-`*`DBInstanceIdentifier`*. AWS Secrets Manager speichert Ihren privaten Schlüssel als Geheimnis.

Führen Sie die Aufgabe in dem folgenden Thema aus:

1. Konfigurieren Sie Ihre DB-Instance so, dass RDP Verbindungen zugelassen werden.
2. Rufen Sie Ihren geheimen Schlüssel ab.
3. Stellen Sie mithilfe des RDP Dienstprogramms eine Connect zu Ihrer EC2 Instance her.

Konfigurieren Sie Ihre DB-Instance so, dass RDP Verbindungen zugelassen werden

Um RDP Verbindungen zuzulassen, konfigurieren Sie Ihre VPC Sicherheitsgruppe und legen Sie eine Firewallregel auf dem Host fest.

Konfigurieren Sie Ihre VPC Sicherheitsgruppe

Stellen Sie sicher, dass die Ihrer DB-Instance zugeordnete VPC Sicherheitsgruppe eingehende Verbindungen auf Port 3389 für Transmission Control Protocol () TCP zulässt. Informationen zur Konfiguration Ihrer VPC Sicherheitsgruppe finden Sie unter. Konfigurieren Sie Ihre VPC-Sicherheitsgruppen wie folgt:

Legen Sie die Firewall-Regel auf dem Host fest

Um eingehende Verbindungen auf Port 3389 für zuzulassenTCP, richten Sie eine Firewallregel auf dem Host ein. In den folgenden Beispielen wird dies veranschaulicht.

Wir empfehlen, den spezifischen `-Profile`-Wert zu verwenden: `Public`, `Private` oder `Domain`. Die Verwendung von Any bezieht sich auf alle drei Werte. Sie können auch eine Kombination von Werten angeben, durch Kommas getrennt. Weitere Informationen zum Festlegen von Firewallregeln finden Sie unter Set- NetFirewallRule in der Microsoft-Dokumentation.

Mit Session Manager von Systems Manager eine Firewallregel einrichten

1. Connect Sie sich mit dem Sitzungsmanager wie in Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit AWS Systems Manager gezeigt.
2. Führen Sie den folgenden Befehl aus.

```
Set-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)" -Direction
 Inbound -LocalAddress Any -Profile Any
```

So verwenden Sie Systems Manager CLI Manager-Befehle, um eine Firewallregel festzulegen

1. Verwenden Sie den folgenden Befehl, um RDP auf dem Host zu öffnen.

```
OPEN_RDP_COMMAND_ID=$(aws ssm send-command --region $AWS_REGION \
    --instance-ids $RDS_CUSTOM_INSTANCE_EC2_ID \
    --document-name "AWS-RunPowerShellScript" \
    --parameters '{"commands":["Set-NetFirewallRule -DisplayName \"Remote Desktop -
 User Mode (TCP-In)\" -Direction Inbound -LocalAddress Any -Profile Any"]}' \
    --comment "Open RDP port" | jq -r ".Command.CommandId")
```

2. Verwenden Sie die in der Ausgabe zurückgegebene Befehls-ID, um den Status des vorherigen Befehls abzurufen. Um die folgende Abfrage für die Benutzer-ID zurückzugeben, stellen Sie sicher, dass Sie das jq-Plug-In installiert haben.

```
aws ssm list-commands \
    --region $AWS_REGION \
    --command-id $OPEN_RDP_COMMAND_ID
```

Rufen Sie Ihren geheimen Schlüssel ab

Rufen Sie Ihren geheimen Schlüssel mit einem AWS Management Console oder dem ab AWS CLI.

Konsole

So rufen Sie den geheimen Schlüssel ab

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte DB-Instance aus, zu der Sie eine Verbindung herstellen möchten.
3. Wählen Sie die Registerkarte Konfiguration aus.
4. Beachten Sie die ID der DB-Instance zum Beispiel für Ihre DB-Instance *my-custom-instance*.
5. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
6. Wählen Sie im Navigationsbereich Instances aus.
7. Suchen Sie nach dem Namen Ihrer EC2 Instance und wählen Sie dann die zugehörige Instance-ID aus.

   In diesem Beispiel ist das Instance-ID `i-abcdefghijklm01234`.
8. In -Details, finden Sie Schlüsselpaarname. Der Paarname enthält die DB-Kennung. In diesem Beispiel lautet der Domänenname `do-not-delete-rds-custom-`*`my-custom-instance-0d726c`*.
9. Suchen Sie in der Instanzübersicht nach Public IPv4 DNS. Das DNS könnte zum Beispiel die Öffentlichkeit sein`ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Öffnen Sie die AWS Secrets Manager Konsole unter https://console.aws.amazon.com/secretsmanager/.
11. Wählen Sie das Geheimnis aus, das den gleichen Namen wie Ihr key pair hat.
12. Wählen Sie Retrieve secret value (Secret-Wert abrufen) aus.

AWS CLI

Rufen Sie den privaten Schlüssel ab

1. Rufen Sie die Liste Ihrer RDS benutzerdefinierten DB-Instances ab, indem Sie den `aws rds describe-db-instances` Befehl aufrufen.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

2. Wählen Sie z. B. den DB-Instance-Bezeichner aus der Beispielausgabe `do-not-delete-rds-custom-`*`my-custom-instance`* aus.
3. Finden Sie die EC2 Instance-ID Ihrer DB-Instance, indem Sie den `aws ec2 describe-instances` Befehl aufrufen. Im folgenden Beispiel wird der EC2 Instance-Name verwendet, um die DB-Instance zu beschreiben.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=do-not-delete-rds-custom-my-custom-instance" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

   Die folgende Beispielausgabe zeigt die EC2 Instance-ID.

```
i-abcdefghijklm01234
```

4. Suchen Sie den Schlüsselnamen, indem Sie die EC2 Instanz-ID angeben, wie im folgenden Beispiel gezeigt.

```
aws ec2 describe-instances \
    --instance-ids i-abcdefghijklm01234 \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

   Die folgende Beispielausgabe zeigt den Schlüsselnamen, der das Präfix `do-not-delete-rds-custom-`*`DBInstanceIdentifier`* verwendet.

```
do-not-delete-rds-custom-my-custom-instance-0d726c
```

Stellen Sie mithilfe des RDP Dienstprogramms eine Connect zu Ihrer EC2 Instance her

Folgen Sie dem Verfahren unter Connect mit Ihrer Windows-Instance herstellen RDP im EC2Amazon-Benutzerhandbuch. Bei diesem Verfahren wird davon ausgegangen, dass Sie eine PEM-Datei erstellt haben, die Ihren privaten Schlüssel enthält.

Verwaltung einer Amazon RDS Custom for SQL Server-DB-Instance

Amazon RDS Custom for SQL Server unterstützt einen Teil der üblichen Verwaltungsaufgaben für Amazon RDS DB-Instances. Im Folgenden finden Sie Anweisungen für die unterstützten Verwaltungsaufgaben von RDS Custom for SQL Server mithilfe von AWS Management Console und. AWS CLI

Themen

- Anhalten und Wiederaufnehmen RDS der benutzerdefinierten Automatisierung
- Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance
- Ändern des Speichers für eine DB-Instance von RDS Custom für Oracle
- Markieren von Ressourcen für RDS Custom for SQL Server
- Löschen einer RDS Custom for SQL Server-DB-Instance
- Starten und Stoppen einer RDS Custom for SQL Server-DB-Instance

Anhalten und Wiederaufnehmen RDS der benutzerdefinierten Automatisierung

RDSCustom bietet automatisch die Überwachung und Instanzwiederherstellung für eine RDS Custom for SQL Server-DB-Instance. Wenn Sie die Instance anpassen müssen, gehen Sie wie folgt vor:

1. Unterbrechen Sie die RDS benutzerdefinierte Automatisierung für einen bestimmten Zeitraum. Durch die Pause wird sichergestellt, dass Ihre Anpassungen die RDS benutzerdefinierte Automatisierung nicht beeinträchtigen.
2. Passen Sie die DB-Instance „RDSBenutzerdefiniert für SQL Server“ nach Bedarf an.
3. Führen Sie eine der folgenden Aufgaben aus:
   - Nehmen Sie die Automatisierung manuell fort.
   - Warten Sie, bis der Pausezeitraum endet. In diesem Fall nimmt RDS Custom die Überwachung und Wiederherstellung der Instance automatisch wieder auf.

> **Important**
>
> Das Anhalten und Wiederaufnehmen der Automatisierung sind die einzigen unterstützten Automatisierungsaufgaben, wenn eine RDS Custom for SQL Server-DB-Instance geändert wird.

Konsole

Um die benutzerdefinierte Automatisierung anzuhalten oder fortzusetzen RDS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte DB-Instance aus, die Sie ändern möchten.
3. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.
4. Wählen Sie für den RDSbenutzerdefinierten Automatisierungsmodus eine der folgenden Optionen aus:

   - Pausiert unterbricht die Überwachung und Instance-Wiederherstellung für die RDS benutzerdefinierte DB-Instance. Geben Sie die Pausedauer ein, für die Sie (in Minuten) die Dauer des Automatisierungsmodus möchten. Der Standardwert beträgt 60 Minuten. Der Maximalwert beträgt 1440 Minuten.
   - Vollständige Automatisierung nimmt die Automatisierung wieder auf.
5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.

   Eine Meldung weist darauf hin, dass RDS Custom die Änderungen sofort anwendet.
6. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern. Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

   Auf der RDS Konsole werden die Details für die Änderung angezeigt. Wenn Sie die Automatisierung angehalten haben, gibt der Status Ihrer RDS benutzerdefinierten DB-Instance an, dass die Automatisierung angehalten wurde.
7. (Optional) Wählen Sie im Navigationsbereich Datenbanken und dann Ihre RDS benutzerdefinierte DB-Instance aus.

   Im Übersichtsbereich gibt der RDSbenutzerdefinierte Automatisierungsmodus den Automatisierungsstatus an. Wenn die Automatisierung angehalten ist, lautet der Wert Angehalten. Die Automatisierung wird fortgesetzt in ***num*** Minuten.

AWS CLI

Verwenden Sie den `modify-db-instance` AWS CLI Befehl, um die RDS benutzerdefinierte Automatisierung anzuhalten oder fortzusetzen. Identifizieren Sie die DB-Instance mit dem erforderlichen Parameter `--db-instance-identifier`. Steuern Sie den Automatisierungsmodus mit den folgenden Parametern:

- `--automation-mode` gibt den Pausestatus der DB-Instance an. Gültige Werte sind `all-paused`, was die Automatisierung anhält, und `full`, was es wieder aufnimmt.
- `--resume-full-automation-mode-minutes` gibt die Dauer der Pause an. Der Standardwert beträgt 60 Minuten.

> **Note**
>
> Unabhängig davon, ob Sie `--no-apply-immediately` oder angeben`--apply-immediately`, wendet RDS Benutzerdefiniert Änderungen so schnell wie möglich asynchron an.

`ResumeFullAutomationModeTime`Gibt in der Befehlsantwort die Wiederaufnahmezeit als UTC Zeitstempel an. Wenn der Automatisierungsmodus `all-paused` ist, können Sie `modify-db-instance` verwenden, um den Automatisierungsmodus fortzusetzen oder den Pausezeitraum zu verlängern. Es werden keine anderen `modify-db-instance`-Optionen unterstützt.

Das folgende Beispiel unterbricht die Automatisierung für `my-custom-instance` für 90 Minuten.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
```

```
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

Im folgenden Beispiel wird die Pausedauer um weitere 30 Minuten verlängert. Die 30 Minuten werden zur Originalzeit hinzugefügt, die in `ResumeFullAutomationModeTime` angezeigt wird.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

Im folgenden Beispiel wird die vollständige Automatisierung für `my-custom-instance` wieder aufgenommen.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

In der folgenden partiellen Beispielausgabe ist der ausstehende `AutomationMode`-Wert `full`.

```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
```

```
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```

Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance

Das Ändern einer RDS Custom for SQL Server-DB-Instance ähnelt dem Ändern für AmazonRDS, aber die Änderungen, die Sie vornehmen können, sind auf Folgendes beschränkt:

- Ändern der -Instance-Klasse
- Ändern des Aufbewahrungszeitraum für Backups und des Backup-Fensters
- Ändern des Wartungsfensters
- Upgrade der DB-Engine-Version, wenn eine neue Version verfügbar wird
- Ändern des zugewiesenen Speichers, des bereitgestellten IOPS Speichers und des Speichertyps
- Zulassen und Entfernen von Multi-AZ-Bereitstellungen

Für das Ändern einer RDS Custom for SQL Server-DB-Instance gelten die folgenden Einschränkungen:

- Benutzerdefinierte DB-Optionen und Parametergruppen werden nicht unterstützt.

- Alle Speichervolumes, die Sie manuell an Ihre RDS benutzerdefinierte DB-Instance anhängen, befinden sich außerhalb des Support-Perimeters.

  Weitere Informationen finden Sie unter Support-Perimeter in RDS Custom.

Konsole

Um eine RDS benutzerdefinierte SQL Server-DB-Instance zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Nehmen Sie nach Bedarf die folgenden Änderungen vor:
   a. Wählen Sie unter DB-Engine-Version die neue Version aus.
   b. Ändern Sie den Wert für DB instance class. Informationen zu unterstützten Klassen finden Sie unter Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server.
   c. Ändern Sie den Wert für Aufbewahrungszeitraum für Backups aus.
   d. Für Backup window, setzen Sie Werte für Beginnzeit und Duration (Dauer) ein.
   e. Für Wartungsfenster der DB-Instance, setzen Sie Werte für Starttag, Beginnzeit und Duration (Dauer) ein.
6. Klicken Sie auf Weiter.
7. Wählen Sie During the next scheduled maintenance window (Während des nächsten geplanten Wartungsfensters) oder Sofort aus.
8. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Verwenden Sie den modify-db-instance AWS CLI Befehl, um eine RDS Custom for SQL Server-DB-Instance zu ändern. Stellen Sie die folgenden Parameter nach Bedarf ein:

- `--db-instance-class` – Informationen zu unterstützten Klassen finden Sie unter Unterstützung für DB-Instance-Klassen für RDS Custom for SQL Server.

- `--engine-version` – Die Versionsnummer der Datenbank-Engine, auf die ein Upgrade durchgeführt werden soll.
- `--backup-retention-period`— Wie lange werden automatisierte Backups von 0 bis 35 Tagen beibehalten.
- `--preferred-backup-window` – Der tägliche Zeitraum, in dem automatische Backups erstellt werden.
- `--preferred-maintenance-window`— Der wöchentliche Zeitraum (inUTC), in dem die Systemwartung durchgeführt werden kann.
- `--apply-immediately` – Verwenden Sie `--apply-immediately`, um die Speicheränderungen sofort anzuwenden.

  Oder verwenden Sie `--no-apply-immediately` (Standardeinstellung), um die Änderungen während des nächsten Wartungsfensters anzuwenden.

Ändern des Speichers für eine DB-Instance von RDS Custom für Oracle

Das Ändern des Speichers für eine DB-Instance von RDS Custom für Oracle ähnelt dem Ändern des Speichers einer DB-Instance von Amazon RDS, Sie jedoch können nur die folgenden Vorgänge ausführen:

- Den zugewiesenen Speicher erhöhen.
- Den Speichertyp ändern. Sie können verfügbare Speichertypen wie Allzweck-Speicher oder bereitgestellte IOPS verwenden. Bereitgestellte IOPS werden für die Block-Express-Speichertypen gp3, io1 und io2 unterstützt.
- Ändern Sie die bereitgestellten IOPS, wenn Sie die Volumetypen verwenden, die bereitgestellte IOPS unterstützen.

Die folgenden Einschränkungen gelten für die Änderung des Speichers einer DB-Instance von RDS Custom für SQL Server:

- Die zugewiesene Mindestspeichergröße für RDS Custom for SQL Server beträgt 20 GiB. Das maximale Speicherlimit für io1, gp2 und gp3 beträgt 16 TiB, während io2 64 TiB unterstützt.
- Wie bei Amazon RDS können Sie den zugewiesenen Speicher nicht verringern. Dies ist eine Einschränkung von Amazon Elastic Block Store (Amazon EBS)-Volumes. Weitere Informationen finden Sie unter Arbeiten mit Speicher für Amazon RDS DB-Instances.

- Die automatische Skalierung von Speicher wird für DB-Instances von RDS Custom für SQL Server nicht unterstützt.
- Speicher-Volumes, die Sie Ihrer DB-Instance von RDS Custom manuell anfügen, werden bei der Speicherskalierung nicht berücksichtigt. Nur die von RDS bereitgestellten Standarddaten-Volumes, d. h. das D-Laufwerk, werden bei der Speicherskalierung berücksichtigt.

  Weitere Informationen finden Sie unter Support-Perimeter in RDS Custom.
- Die Skalierung des Speichers verursacht normalerweise keine Ausfälle oder Leistungseinbußen der DB-Instance. Nachdem Sie die Speichergröße für eine DB-Instance geändert haben, lautet der Status der DB-Instance Speicheroptimierung.
- Die Speicheroptimierung kann mehrere Stunden dauern. Sie können keine weiteren Speicheränderungen für sechs (6) Stunden vornehmen oder bis die Speicheroptimierung auf der Instance abgeschlossen ist, je nachdem, welcher Zeitraum länger ist. Weitere Informationen finden Sie unter Arbeiten mit Speicher für Amazon RDS DB-Instances.

Weitere Informationen über Speicher finden Sie unter Amazon RDS-DB-Instance-Speicher.

Allgemeine Informationen zur Änderung des Speichers finden Sie unter Arbeiten mit Speicher für Amazon RDS DB-Instances.

> **Important**
>
> Ändern Sie den Speicher für Ihre RDS Custom for SQL Server-DB-Instance nicht mithilfe von Amazon EC2 - oder Amazon EBS-Konsolen oder APIs. Direkte Speicheränderungen außerhalb der Amazon RDS-Konsole oder APIs führen zu einem `unsupported-configuration` Status für Ihre Datenbank.
> Wenn Sie direkte Speicheränderungen mit Amazon EC2 oder Amazon EBS vornehmen, kann Amazon RDS den Status Ihrer Datenbank-Instance nicht verfolgen oder verwalten. Dies könnte Folgendes zur Folge haben:
>
> - Hochverfügbarkeits-Failover-Mechanismen funktionieren nicht richtig
> - Zu unterbrechende Konfigurationen für die Datenbankreplikation
> - Redundanzfunktionen werden ausfallen

Ändern Sie den Speicher nur über die Amazon RDS-Konsole oder APIs um Ihre Datenbank in einem unterstützten Zustand zu halten. Schritte Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server zur Wiederherstellung finden Sie unter.

Konsole

So ändern Sie den Speicher für eine DB-Instance von RDS Custom für SQL Server

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Nehmen Sie nach Bedarf die folgenden Änderungen vor:
   a. Geben Sie einen neuen Wert für Allocated storage (Zugewiesener Speicherplatz) ein. Er muss größer als der aktuelle Wert und zwischen 20 GiB und 16 TiB sein.
   b. Ändern Sie den Wert für Speichertyp. Sie können zwischen den verfügbaren Speichertypen für allgemeine Zwecke und bereitgestellte IOPS wählen. Provisioned IOPS wird für die Block Express-Speichertypen gp3, io1 und io2 unterstützt.
   c. Wenn Sie einen Speichertyp angeben, der bereitgestellte IOPS unterstützt, können Sie den Wert Provisioned IOPS definieren.
6. Klicken Sie auf Weiter.
7. Wählen Sie During the next scheduled maintenance window (Während des nächsten geplanten Wartungsfensters) oder Sofort aus.
8. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Verwenden Sie den Befehl, um den Speicher für eine RDS Custom for SQL Server-DB-Instance zu ändern. modify-db-instance AWS CLI Stellen Sie die folgenden Parameter nach Bedarf ein:

- `--allocated-storage`: Größe des zuzuteilenden Speichers für die DB-Instance in Gibibytes. Er muss größer als der aktuelle Wert und zwischen 20 und 16 384 GiB sein.

- `--storage-type`— Der Speichertyp, zum Beispiel gp2, gp3, io1 oder io2.
- `--iops` – Bereitgestellte IOPS für die DB-Instance. Sie können dies nur für Speichertypen angeben, die bereitgestellte IOPS unterstützen (gp3, io1 und io2).
- `--apply-immediately` – Verwenden Sie `--apply-immediately`, um die Speicheränderungen sofort anzuwenden.

  Oder verwenden Sie `--no-apply-immediately` (Standardeinstellung), um die Änderungen während des nächsten Wartungsfensters anzuwenden.

Im folgenden Beispiel wird die Speichergröße my-custom-instance auf 200 GiB, der Speichertyp auf io1 und die bereitgestellten IOPS auf 3000 geändert.

Example

FürLinux, oder: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 200 \
    --apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 200 ^
    --apply-immediately
```

Markieren von Ressourcen für RDS Custom for SQL Server

Sie können RDS Custom Ressourcen wie bei Amazon RDS-Ressourcen kennzeichnen, jedoch mit einigen wichtigen Unterschieden:

- Erstellen oder modifizieren Sie die`AWSRDSCustom`-Tag, das für die RDS Custom Automatisierung erforderlich ist. Wenn Sie dies tun, könnten Sie die Automatisierung unterbrechen.

- Das `Name` Tag wird den benutzerdefinierten RDS-Ressourcen mit dem Präfix hinzugefügt`do-not-delete-rds-custom`. Vom Kunden übergebene Werte für den Schlüssel werden überschrieben.
- Tags, die während der Erstellung zu RDS Custom DB-Instanzen hinzugefügt wurden, werden an alle anderen verwandten RDS Custom Ressourcen weitergegeben.
- Tags werden nicht propagiert, wenn Sie sie nach der Erstellung der DB-Instance zu RDS Custom Ressourcen hinzufügen.

Allgemeine Informationen zum Markieren von Ressourcen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.

Löschen einer RDS Custom for SQL Server-DB-Instance

Gehen Sie wie folgt vor, um eine RDS Custom for SQL Server-DB-Instance zu löschen:

- Geben Sie den Namen der Instance an.
- Aktivieren oder deaktivieren Sie die Option zum Erstellen eines endgültigen DB-Snapshots der DB-Instance.
- Wählen oder deaktivieren Sie die Option zum Speichern automatisierter Sicherungen.

Sie können eine RDS benutzerdefinierte SQL Server-DB-Instance mit der Konsole oder dem löschenCLI. Die zum Löschen der DB-Instance erforderliche Zeit kann je nach Aufbewahrungszeitraum für Backups (d. h. wie viele Backups gelöscht werden sollen), wie viele Daten gelöscht werden und ob ein endgültiger Snapshot erstellt wird, variieren.

> **Warning**
>
> Durch das Löschen einer RDS Custom for SQL Server-DB-Instance werden die EC2 Instance und die zugehörigen EBS Amazon-Volumes dauerhaft gelöscht. Sie sollten diese Ressourcen niemals beenden oder löschen, da andernfalls das Löschen und die Erstellung des finalen Snapshots fehlschlagen könnten.

> **Note**
>
> Sie können keinen endgültigen DB-Snapshot Ihrer DB-Instance erstellen, wenn sie sich im Status `creating`, `failed`, `incompatible-create`, `incompatible-restore` oder

`incompatible-network` befindet. Weitere Informationen finden Sie unter Anzeigen von Amazon RDS DB-Instance-Status.

> **Important**
>
> Wenn Sie einen endgültigen Snapshot erstellen möchten, empfehlen wir, möglichst keine Daten in Ihre DB-Instance zu schreiben, während der Löschvorgang der DB-Instance läuft. Sobald das Löschen der DB-Instance eingeleitet wurde, kann nicht garantiert werden, dass Datenänderungen vom endgültigen Snapshot erfasst werden.

Konsole

Um eine RDS Custom DB-Instance zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die RDS benutzerdefinierte SQL Server-DB-Instance aus, die Sie löschen möchten. RDSBenutzerdefiniert für SQL Server-DB-Instances zeigen die Rolleninstanz (RDSBenutzerdefiniert für SQL Server) an.
3. Klicken Sie bei Actions auf Delete.
4. Wenn Sie einen endgültigen Snapshot erstellen möchten, wählen Sie Create final snapshot (Abschließenden Snapshot erstellen) aus und geben Sie einen Namen im Feld Final snapshot name (Name des finalen Snapshots) ein.
5. Wählen Sie Retain automated backups (Automatisierte Sicherungen aufbewahren), um automatisierte Sicherungen aufzubewahren.
6. Geben Sie **`delete me`** in das Feld ein.
7. Wählen Sie Löschen.

AWS CLI

Sie löschen eine RDS benutzerdefinierte DB-Instance für SQL Server mithilfe des delete-db-instance AWS CLI Befehls. Identifizieren Sie die DB-Instance mit dem erforderlichen Parameter `--db-instance-identifier`. Die übrigen Parameter sind dieselben wie für eine RDS Amazon-DB-Instance.

Im folgenden Beispiel wird die angegebene DB-Instance „RDSCustom for SQL Server“ gelöschtmy-custom-instance, ein letzter Snapshot erstellt und automatische Backups beibehalten.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier my-custom-instance \
    --no-skip-final-snapshot \
    --final-db-snapshot-identifier my-custom-instance-final-snapshot \
    --no-delete-automated-backups
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --no-skip-final-snapshot ^
    --final-db-snapshot-identifier my-custom-instance-final-snapshot ^
    --no-delete-automated-backups
```

Um einen endgültigen Snapshot zu erstellen, ist die Option `--final-db-snapshot-identifier` erforderlich und muss angegeben werden.

Wenn Sie den endgültigen Snapshot überspringen möchten, geben Sie im Befehl die Option `--skip-final-snapshot` anstelle der Optionen `--no-skip-final-snapshot` und `--final-db-snapshot-identifier` an.

Damit automatische Backups gelöscht werden, geben Sie im Befehl die Option `--delete-automated-backups` anstelle der Option `--no-delete-automated-backups` an.

Starten und Stoppen einer RDS Custom for SQL Server-DB-Instance

Sie können Ihre RDS Custom for SQL Server-DB-Instance starten und beenden. Für das Stoppen und Starten Ihrer RDS Custom RDS for SQL Server-DB-Instances gelten dieselben allgemeinen Anforderungen und Einschränkungen wie für SQL Server-DB-Instances. Weitere Informationen finden Sie unter Eine Amazon RDS-DB-Instance temporär stoppen.

Die folgenden Überlegungen gelten auch für das Starten und Stoppen Ihrer RDS Custom for SQL Server-DB-Instance:

- Das Ändern eines EC2 Instance-Attributs einer RDS Custom for SQL Server-DB-Instance, während die DB-Instance aktiv ist, wird `STOPPED` nicht unterstützt.
- Sie können eine RDS Custom for SQL Server-DB-Instance nur beenden und starten, wenn sie für eine einzelne Availability Zone konfiguriert ist. Sie können eine RDS Custom for SQL Server-DB-Instance in einer Multi-AZ-Konfiguration nicht stoppen.
- Ein `SYSTEM` Snapshot wird erstellt, wenn Sie eine RDS Custom for SQL Server-DB-Instance beenden. Der Snapshot wird automatisch gelöscht, wenn Sie die RDS Custom for SQL Server-DB-Instance erneut starten.
- Wenn Sie Ihre EC2 Instance löschen, während Ihre RDS Custom for SQL Server-DB-Instance gestoppt ist, wird das `C:` Laufwerk ersetzt, wenn Sie die RDS Custom for SQL Server-DB-Instance erneut starten.
- Das `C:\` Laufwerk, der Hostname und Ihre benutzerdefinierten Konfigurationen werden beibehalten, wenn Sie eine RDS Custom for SQL Server-DB-Instance beenden, solange Sie den Instance-Typ nicht ändern.
- Die folgenden Aktionen führen dazu, dass RDS Custom die DB-Instance außerhalb des Support-Bereichs platziert und Ihnen trotzdem die DB-Instance-Stunden in Rechnung gestellt werden:
  - Die zugrunde liegende EC2 Instance wird gestartet, während Amazon gestoppt RDS ist. Um das Problem zu lösen, können Sie `start-db-instance` Amazon anrufen oder das System beenden RDSAPI, EC2 sodass die RDS benutzerdefinierte Instance zurückkehrt`STOPPED`.
  - Die zugrunde liegende EC2 Instance wird gestoppt, wenn die RDS Custom for SQL Server-DB-Instance aktiviert ist`ACTIVE`.

Weitere Informationen zum Anhalten und Starten von DB-Instances finden Sie unter Eine Amazon RDS-DB-Instance temporär stoppen und Starten einer Amazon RDS DB-Instance, die zuvor gestoppt wurde.

Arbeiten mit Microsoft Active Directory mit RDS Custom for SQL Server

RDSMit Custom for SQL Server können Sie Ihre Instanzen mit einem selbstverwalteten Active Directory (AD) verbinden oder AWS Managed Microsoft AD. Dies gilt unabhängig davon, wo Ihr AD gehostet wird, z. B. in einem lokalen Rechenzentrum, bei Amazon EC2 oder bei einem anderen Cloud-Dienstanbieter.

Für die Authentifizierung von Benutzern und Diensten können Sie die Kerberos-Authentifizierung in Ihrer RDS Custom for SQL Server-DB-Instance verwendenNTLM, ohne dazwischengeschaltete Domänen und Forest Trusts zu verwenden. Wenn ein Benutzer versucht, sich auf Ihrer RDS Custom for SQL Server-DB-Instance mit einem selbst verbundenen Active Directory zu authentifizieren, werden Authentifizierungsanforderungen an ein selbstverwaltetes oder von Ihnen angegebenes AD weitergeleitet. AWS Managed Microsoft AD

In den folgenden Abschnitten finden Sie Informationen zur Arbeit mit Self Managed Active Directory und AWS Managed Active Directory für RDS Custom for SQL Server.

Themen

- Verfügbarkeit von Regionen und Versionen
- Konfiguration von selbstverwaltetem oder lokalem AD
- Konfigurieren Sie Microsoft Active Directory mit AWS Directory Service
- Portregeln für die Netzwerkkonfiguration
- Netzwerkvalidierung
- Windows-Authentifizierung für RDS Custom für SQL Server-Instanzen einrichten
- Verwalten einer DB-Instance in einer Domäne
- Grundlegendes zur Domänenmitgliedschaft
- Problembehebung bei Active Directory

Verfügbarkeit von Regionen und Versionen

RDSBenutzerdefiniert für SQL Server unterstützt sowohl Self Managed AD als auch die AWS Managed Microsoft AD Verwendung von NTLM oder Kerberos in allen Regionen, in denen RDS Benutzerdefiniert für SQL Server unterstützt wird. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom.

Konfiguration von selbstverwaltetem oder lokalem AD

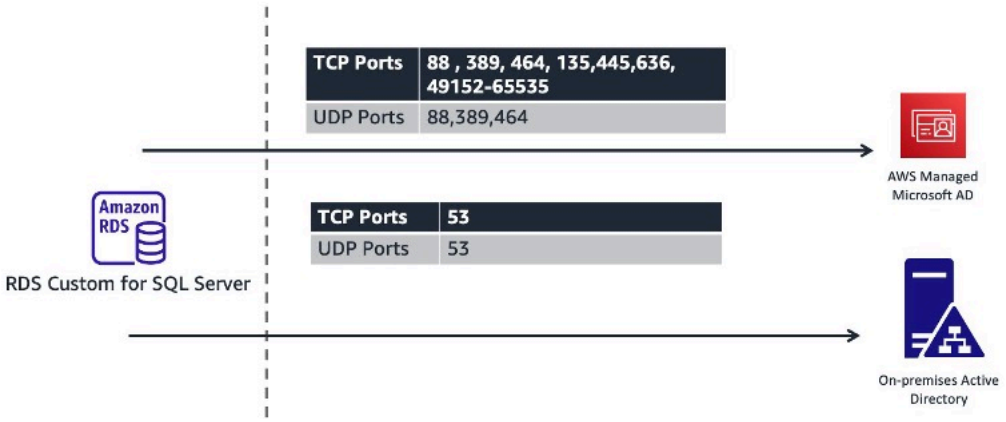
Um Ihr lokales oder selbstverwaltetes Microsoft AD mit Ihrer RDS Custom for SQL Server-DB-Instance zu verbinden, muss Ihre Active Domain wie folgt konfiguriert sein:

- Definieren Sie die Subnetze in der mit Ihrer RDS Custom for SQL Server-DB-Instance VPC verknüpften Subnetze in Ihrem selbstverwalteten oder lokalen AD. Vergewissern Sie sich, dass keine Konflikte zwischen den Subnetzen in Ihren VPC und den Subnetzen in Ihren AD-Sites bestehen.
- Ihr AD-Domain-Controller weist die Domain-Funktionsebene Windows Server 2008 R2 oder höher auf.
- Ihr AD-Domainname darf nicht im Format Single Lable Domain () SLD vorliegen. RDS Custom for SQL Server unterstützt keine SLD Domänen.
- Der vollqualifizierte Domainname (FQDN) für Ihr AD darf 47 Zeichen nicht überschreiten.

Konfigurieren Ihrer Netzwerkkonnektivität

Konfigurieren Sie Ihre selbst verwaltete oder lokale AD-Netzwerkkonnektivität wie folgt:

- Richten Sie Konnektivität zwischen AmazonVPC, auf dem Ihre RDS Custom for SQL Server-Instance ausgeführt wird, und Ihrem AD ein. Verwenden Sie AWS Direct Connect AWS VPN, AWS Transit Gateway, und VPC Peering.
- Erlauben Sie Datenverkehr über die Ports Ihrer RDS Custom for SQL Server-Sicherheitsgruppen und Ihres Netzwerks ACLs zu Ihrem selbstverwalteten oder lokalen AD. Weitere Informationen finden Sie unter Portregeln für die Netzwerkkonfiguration.



DNSAuflösung konfigurieren

Richten Sie die folgenden Anforderungen ein, um die DNS Auflösung mit selbstverwalteten oder lokalen ADs zu konfigurieren:

- Konfigurieren Sie die DNS Auflösung innerhalb IhresVPC, um den vollqualifizierten Domänennamen () Ihres selbst gehosteten Active Directory-Kontos aufzulösen. FQDN Ein Beispiel für ein FQDN ist`corp.example.local`. Um die DNS Auflösung zu konfigurieren, konfigurieren Sie den VPC DNS Resolver so, dass er Anfragen für bestimmte Domänen mit einem ausgehenden Amazon Route 53-Endpunkt und einer Resolver-Regel weiterleitet. Weitere Informationen finden Sie unter Konfiguration eines ausgehenden Route 53 53-Resolver-Endpunkts zur Auflösung von Datensätzen. DNS
- Bei Workloads, die VPCs sowohl Ressourcen als auch lokale Ressourcen nutzen, müssen Sie lokal gehostete Datensätze auflösenDNS. Lokale Ressourcen müssen möglicherweise Namen auflösen, die auf gehostet werden. AWS

  Um ein Hybrid-Cloud-Setup zu erstellen, verwenden Sie Resolver-Endpunkte und Regeln für bedingte Weiterleitungen, um DNS Anfragen zwischen Ihren lokalen und benutzerdefinierten Ressourcen zu lösen. VPC Weitere Informationen finden Sie unter Auflösen von DNS Abfragen zwischen VPCs und Ihrem Netzwerk im Amazon Route 53-Entwicklerhandbuch.

> **Important**
>
> Wenn Sie die DNS Resolver-Einstellungen der Netzwerkschnittstelle auf dem RDS Custom for SQL Server ändern, funktionieren VPC Endgeräte, die DNS -fähig sind, nicht mehr korrekt. DNS-aktivierte VPC Endpunkte sind für Instances in privaten Subnetzen ohne Internetzugang erforderlich.

Konfigurieren Sie Microsoft Active Directory mit AWS Directory Service

AWS Managed Microsoft AD erstellt ein vollständig verwaltetes Microsoft Active Directory AWS , das auf Windows Server 2019 basiert und auf den Funktionsebenen 2012 R2 Forest und Domain betrieben wird. AWS Directory Service erstellt die Domain-Controller in verschiedenen Subnetzen in einem AmazonVPC, sodass Ihr Verzeichnis auch bei einem Ausfall hochverfügbar ist.

Informationen zum Erstellen eines Verzeichnisses mit AWS Managed Microsoft AD finden Sie unter Erste Schritte mit AWS Managed Microsoft AD im AWS Directory Service Administratorhandbuch.

Konfigurieren Ihrer Netzwerkkonnektivität

Aktivieren Sie den VPC Querverkehr zwischen dem Verzeichnis und der DB-Instance

Um das Verzeichnis und die DB-Instance im selben Verzeichnis zu findenVPC, überspringen Sie diesen Schritt und fahren Sie mit dem nächsten Schritt fortPortregeln für die Netzwerkkonfiguration.

Um das Verzeichnis und die DB-Instance unterschiedlich zu lokalisierenVPCs, konfigurieren Sie VPC Cross-Traffic mithilfe von VPC Peering oder AWS Transit Gateway. Weitere Informationen zur Verwendung von VPC Peering finden Sie unter Was ist VPC Peering? im Amazon VPC Peering Guide und was ist AWS Transit Gateway? in den Amazon VPC Transit Gateways.

Aktivieren Sie VPC Cross-Traffic mithilfe von Peering VPC

1. Richten Sie geeignete VPC Routing-Regeln ein, um sicherzustellen, dass der Netzwerkverkehr in beide Richtungen fließen kann.
2. Erlauben Sie der Sicherheitsgruppe der DB-Instance, eingehenden Datenverkehr von der Sicherheitsgruppe des Verzeichnisses zu empfangen. Weitere Informationen finden Sie unter Portregeln für die Netzwerkkonfiguration.
3. Die Netzwerkzugriffskontrollliste (ACL) darf den Datenverkehr nicht blockieren.

Wenn das Verzeichnis einer anderen Person AWS-Konto gehört, müssen Sie das Verzeichnis gemeinsam nutzen. Um das Verzeichnis, AWS-Konto in dem sich die RDS Custom for SQL Server-Instanz befindet, gemeinsam zu nutzen, folgen Sie dem Tutorial: Sharing your AWS Managed Microsoft AD for seamless EC2 domain-join im AWS Directory Service Administrationshandbuch.

Gemeinsames Verwenden eines Verzeichnisses zwischen AWS-Konten

1. Melden Sie sich mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole an und überprüfen Sie, ob die Domain den `SHARED` Status hat, bevor Sie fortfahren.
2. Nachdem Sie sich mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole angemeldet haben, notieren Sie sich den Wert für die Verzeichnis-ID. Sie verwenden diese ID, um die DB-Instance mit der Domain zu verbinden.

DNSAuflösung konfigurieren

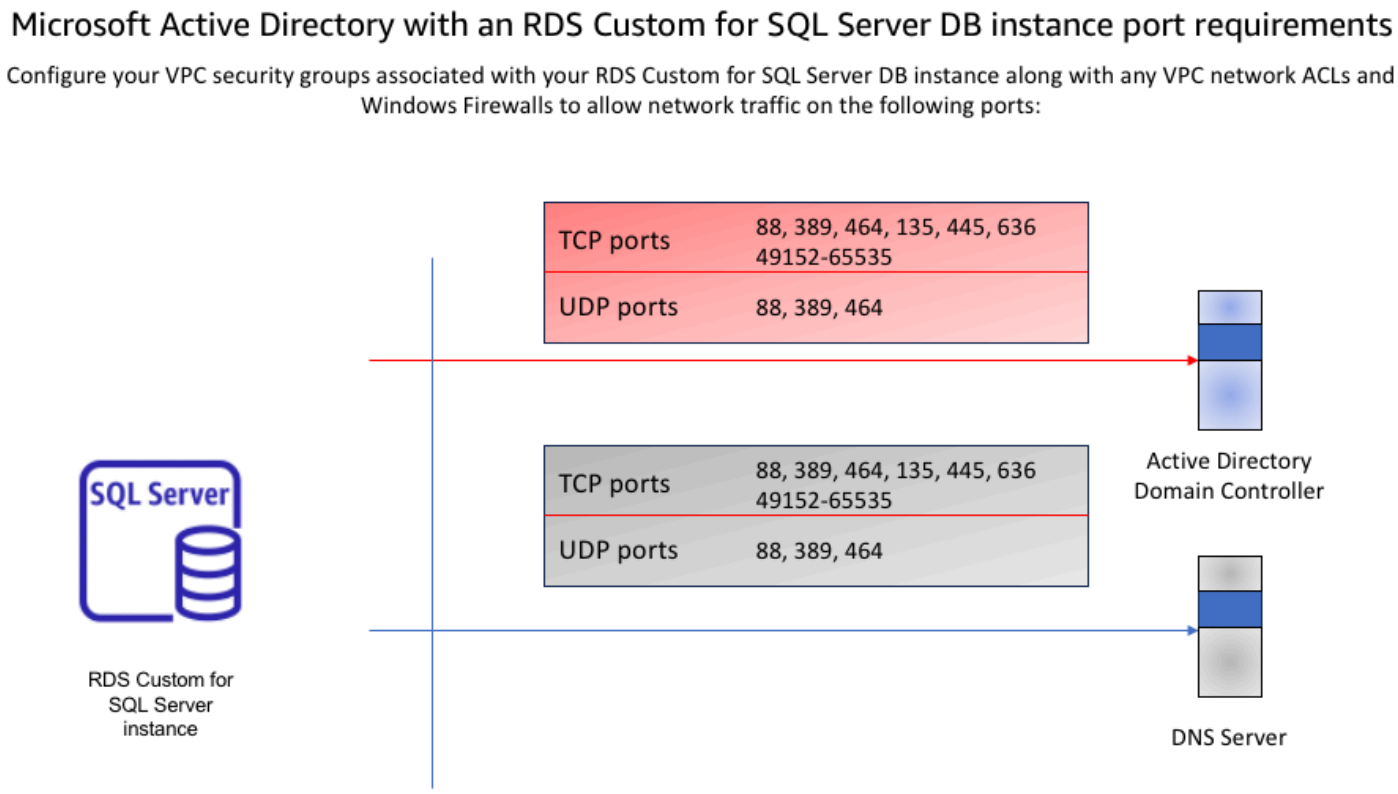
Wenn Sie ein Verzeichnis mit erstellen AWS Managed Microsoft AD, AWS Directory Service erstellt zwei Domänencontroller und fügt den DNS Dienst in Ihrem Namen hinzu.

Wenn Sie über eine bestehende AWS Managed Microsoft AD oder beabsichtigen, eine solche Instance in einer VPC anderen als Ihrer RDS Custom for SQL Server-DB-Instance zu starten, konfigurieren Sie den VPC DNS Resolver so, dass er Abfragen für bestimmte Domänen mit einer Route 53 53-ausgehenden und Resolver-Regel weiterleitet. Weitere Informationen finden Sie unter Konfiguration eines ausgehenden Route 53-Resolver-Endpunkts zur Auflösung von Datensätzen. DNS

Portregeln für die Netzwerkkonfiguration

Stellen Sie sicher, dass die folgenden Netzwerkkonfigurationsanforderungen erfüllt sind:

- Konfigurierte Konnektivität zwischen dem AmazonVPC, auf dem Sie die RDS Custom for SQL Server-DB-Instance erstellen möchten, und Ihrem selbstverwalteten Active Directory oder AWS Managed Microsoft AD. Richten Sie für selbstverwaltetes Active Directory die Konnektivität mithilfe von AWS Direct Connect AWS VPN, VPC Peering oder AWS Transit Gateway ein. Für: AWS Managed Microsoft AD Richten Sie Konnektivität mithilfe VPC von Peering ein.
- Stellen Sie sicher, dass die Sicherheitsgruppe und das VPC Netzwerk ACLs für die Subnetze, in denen Sie Ihre RDS Custom for SQL Server-DB-Instance erstellen, Datenverkehr auf den Ports und in den Richtungen zulassen, die in der folgenden Abbildung dargestellt sind.



In der folgenden Tabelle ist die Rolle der einzelnen Ports aufgeführt.

Protokoll	Ports	Rolle
TCP/UDP	53	Domänennamensystem (DNS)
TCP/UDP	88	Kerberos-Authentifizierung
TCP/UDP	464	Passwort ändern/festlegen
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
TCP	135	Verteilte Computerumgebung/ End Point Mapper (DCE/ EPMAP)
TCP	445	SMBDateifreigabe durch Verzeichnisdienste
TCP	636	Lightweight Directory Access Protocol überTLS/S SL(LDAPS)
TCP	49152–65535	Kurzlebige Ports für RPC

- Im Allgemeinen befinden sich die DNS Domänenserver in den AD-Domänencontrollern. Sie müssen den VPC DHCP Optionssatz nicht konfigurieren, um diese Funktion verwenden zu können. Weitere Informationen finden Sie unter DHCPOptionssätze im VPCAmazon-Benutzerhandbuch.

> **Important**
>
> Wenn Sie ein VPC Netzwerk verwendenACLs, müssen Sie auch ausgehenden Datenverkehr über dynamische Ports (49152-65535) von Ihrer RDS Custom for Server-DB-Instance aus zulassen. SQL Stellen Sie sicher, dass diese Verkehrsregeln auch auf den Firewalls widergespiegelt werden, die für alle AD-Domänencontroller, Server und Custom for Server-DB-Instances gelten. DNS RDS SQL

Während VPC Sicherheitsgruppen verlangen, dass Ports nur in der Richtung geöffnet werden, in der der Netzwerkverkehr initiiert wird, ACLs erfordern die meisten Windows-Firewalls und VPC Netzwerke, dass Ports in beide Richtungen geöffnet sind.

Netzwerkvalidierung

Bevor Sie Ihre RDS-Custom-Instanz entweder zu selbstverwaltetem oder hinzufügen AWS Managed Microsoft AD, überprüfen Sie Folgendes von einer EC2 Instance in derselben VPC, in der Sie die RDS Custom for SQL Server-Instanz starten möchten.

- Prüfen Sie, ob Sie in der Lage sind, den vollqualifizierten Domänennamen (FQDN) auf einen Domänencontroller aufzulösen. IPs

  ```
  nslookup corp.example.com
  ```

  Der Befehl muss eine ähnliche Ausgabe zurückgeben:

  ```
  Server:  ip-10-0-0-2.us-west-2.compute.internal
  Address:  25.0.0.2

  Non-authoritative answer:
  Name:    corp.example.com
  Addresses:  40.0.9.25 (DC1 IP)
              40.0.50.123 (DC2 IP)
  ```

- Lösen Sie AWS Dienste von einer EC2 Instance in der VPC auf, in der Sie Ihre benutzerdefinierte RDS-Instance starten:

  ```
  $region='input-your-aws-region'
  $domainFQDN='input-your-domainFQDN'

  function Test-DomainPorts {
      param (
          [string]$Domain,
          [array]$Ports
      )

      foreach ($portInfo in $Ports) {
          try {
              $conn = New-Object System.Net.Sockets.TcpClient
  ```

```
            $connectionResult = $conn.BeginConnect($Domain, $portInfo.Port, $null,
 $null)
            $success = $connectionResult.AsyncWaitHandle.WaitOne(1000) # 1 second
 timeout
            if ($success) {
                $conn.EndConnect($connectionResult)
                $result = $true
            } else {
                $result = $false
            }
        }
        catch {
            $result = $false
        }
        finally {
            if ($null -ne $conn) {
                $conn.Close()
            }
        }
        Write-Host "$($portInfo.Description) port open: $result"
    }
}

# Check if ports can be reached
$ports = @(
    @{Port = 53;   Description = "DNS"},
    @{Port = 88;   Description = "Kerberos"},
    @{Port = 389;  Description = "LDAP"},
    @{Port = 445;  Description = "SMB"},
    @{Port = 5985; Description = "WinRM"},
    @{Port = 636;  Description = "LDAPS"},
    @{Port = 3268; Description = "Global Catalog"},
    @{Port = 3269; Description = "Global Catalog over SSL"},
    @{Port = 9389; Description = "AD DS"}
)

function Test-DomainReachability {
    param (
        [string]$DomainName
    )

    try {
        $dnsResults = Resolve-DnsName -Name $DomainName -ErrorAction Stop
```

```
        Write-Host "Domain $DomainName is successfully resolving to following IP
 addresses: $($dnsResults.IpAddress)"
        Write-Host ""
        return $true
    }
    catch {
        Write-Host ""
        Write-Host "Error Message: $($_.Exception.Message)"
        Write-Host "Domain $DomainName reachability check failed, please Configure
 DNS resolution"
        return $false
    }
}

$domain = (Get-WmiObject Win32_ComputerSystem).Domain
if ($domain -eq 'WORKGROUP') {
    Write-Host ""
    Write-Host "Host $env:computername is still part of WORKGROUP and not part of any
 domain"
    }
else {
    Write-Host ""
    Write-Host "Host $env:computername is joined to $domain domain"
    Write-Host ""
    }


$isReachable = Test-DomainReachability -DomainName $domainFQDN
if ($isReachable) {
    write-Host "Checking if domain $domainFQDN is reachable on required ports  "
    Test-DomainPorts -Domain $domainFQDN -Ports $ports
}
else {
    Write-Host "Port check skipped. Domain not reachable"
}



# Get network adapter configuration
$networkConfig = Get-WmiObject Win32_NetworkAdapterConfiguration |
                 Where-Object { $_.IPEnabled -eq $true } |
                 Select-Object -First 1

# Check DNS server settings
```

```
$dnsServers = $networkConfig.DNSServerSearchOrder

if ($dnsServers) {
    Write-Host "`nDNS Server settings:"
    foreach ($server in $dnsServers) {
        Write-Host "  - $server"
    }
} else {
    Write-Host "`nNo DNS servers configured or unable to retrieve DNS server
 information."
}

write-host ""

# Checks reachability to dependent services
$services = "s3", "ec2", "secretsmanager", "logs", "events", "monitoring", "ssm",
 "ec2messages", "ssmmessages"

function Get-TcpConnectionAsync {
    param (
        $ServicePrefix,
        $region
    )
    $endpoint = "${ServicePrefix}.${region}.amazonaws.com"
    $tcp = New-Object Net.Sockets.TcpClient
    $result = $false

    try {
        $connectTask = $tcp.ConnectAsync($endpoint, 443)
        $timedOut = $connectTask.Wait(3000)
        $result = $tcp.Connected
    }
    catch {
        $result = $false
    }
    return $result
}

foreach ($service in $services) {
    $validationResult = Get-TcpConnectionAsync -ServicePrefix $service -Region
 $region
    Write-Host "Reachability to $service is $validationResult"
}
```

Der `TcpTestSucceeded` Wert muss `True` fürs`3`,,`ec2`,`secretsmanager`,`logs`, `events monitoring` `ssmec2messages`, und `ssmmessages` zurückgegeben werden.

Windows-Authentifizierung für RDS Custom für SQL Server-Instanzen einrichten

Wir empfehlen, für alle Benutzer, die eine RDS Custom for SQL Server-DB-Instance besitzen, die mit Ihrer AWS-Konto AD-Domäne verknüpft ist, eine eigene Organisationseinheit und Dienstanmeldedaten für diese Organisationseinheit zu erstellen. Indem Sie eine Organisationseinheit und Dienstanmeldedaten zuweisen, vermeiden Sie widersprüchliche Berechtigungen und folgen dem Prinzip der geringsten Rechte.

Gruppenrichtlinien auf Active Directory-Ebene können zu Konflikten mit AWS Automatisierungen und Berechtigungen führen. Wir empfehlen, GPOs auszuwählen, die nur für die Organisationseinheit gelten, die Sie für RDS Custom for SQL Server erstellen.

- Um OU- und AD-Domänenbenutzer in Ihrem selbstverwalteten oder lokalen AD zu erstellen, können Sie den Domänencontroller als Domänenadministrator verbinden.
- Um Benutzer und Gruppen in einem AWS Directory Service Verzeichnis zu erstellen, müssen Sie mit einer Verwaltungsinstanz verbunden sein und außerdem als Benutzer mit Rechten zum Erstellen von Benutzern und Gruppen angemeldet sein. Weitere Informationen finden Sie unter Benutzer- und AWS Directory Service Gruppenverwaltung AWS Managed Microsoft AD im Administratorhandbuch.
- Um Ihr Active Directory von der Amazon EC2 Windows Server-Instance aus zu verwalten, müssen Sie die Active Directory-Domänendienste und die Active Directory Lightweight Directory-Dienste auf der EC2 Instance installieren. Weitere Informationen finden Sie unter Installation der Active Directory-Verwaltungstools für AWS Managed Microsoft AD im AWS Directory Service Administratorhandbuch.
- Wir empfehlen, dass Sie diese Tools für die Verwaltung auf einer separaten EC2 Instanz und nicht auf Ihrer RDS Custom for SQL Server-DB-Instance installieren, um die Verwaltung zu vereinfachen.

Im Folgenden sind die Anforderungen für ein AD-Domänendienstkonto aufgeführt:

- Sie benötigen ein Dienstkonto in Ihrer AD-Domäne mit delegierten Berechtigungen, um Computer mit der Domäne verbinden zu können. Ein Domänendienstkonto ist ein Benutzerkonto in Ihrem AD, dem die Berechtigung zur Ausführung bestimmter Aufgaben delegiert wurde.

- Delegieren Sie die folgenden Berechtigungen an Ihr Domänendienstkonto in der Organisationseinheit, der Sie Ihrer RDS Custom for SQL Server-Instanz beitreten möchten:
  - Überprüfte Fähigkeit zum Schreiben in den DNS-Hostnamen
  - Überprüfte Fähigkeit zum Schreiben in den Prinzipalnamen des Service
  - Erstellen und Löschen von Computerobjekten
- Für selbstverwaltetes und lokales AD muss das Domänendienstkonto Mitglied der Gruppe „AWS Delegated Domain Name System Administrators“ sein.
- Denn AWS Managed Microsoft AD das Domänendienstkonto sollte Mitglied der Gruppe "" DnsAdmins sein.

Dies sind die Mindestberechtigungen, die erforderlich sind, um Computerobjekte mit Ihrem selbstverwalteten AD und AWS Managed Microsoft AD zu verknüpfen. Weitere Informationen finden Sie in der Microsoft Windows Server-Dokumentation unter Fehler: Zugriff verweigert, wenn Benutzer ohne Administratorrechte, denen die Steuerung delegiert wurde, versuchen, Computer mit einem Domänencontroller zu verbinden.

> **Important**
>
> Verschieben Sie keine Computerobjekte, die RDS Custom for SQL Server erstellt, in der Organisationseinheit (OU), nachdem Ihre DB-Instance erstellt wurde. Das Verschieben zugeordneter Objekte kann dazu führen, dass Ihre RDS Custom for SQL Server-DB-Instance falsch konfiguriert wird. Wenn Sie die von Amazon RDS erstellten Computerobjekte verschieben müssen, verwenden Sie die DBInstance Aktion Ändern, um die Domänenparameter mit dem gewünschten Speicherort der Computerobjekte zu ändern.

Themen

- Schritt 1: Erstellen Sie eine Organisationseinheit (OU) in Ihrem AD
- Schritt 2: Erstellen Sie einen AD-Domänenbenutzer
- Schritt 3: Delegieren Sie die Kontrolle an den AD-Benutzer im selbstverwalteten Modus oder AWS Managed Microsoft AD
- Schritt 4: Erstellen Sie ein Geheimnis
- Schritt 5: Erstellen oder ändern Sie eine RDS Custom for SQL Server-DB-Instance
- Schritt 6: Erstellen Sie die SQL Server-Anmeldung für die Windows-Authentifizierung

- Schritt 7: Verwenden der Kerberos- oder NTLM-Authentifizierung

Schritt 1: Erstellen Sie eine Organisationseinheit (OU) in Ihrem AD

Gehen Sie wie folgt vor, um eine Organisationseinheit in Ihrem AD zu erstellen:

Erstellen Sie eine Organisationseinheit in Ihrem AD

1. Stellen Sie als Domainadministrator eine Connect zu Ihrem Domain-AD her.
2. Öffnen Sie Active Directory-Benutzer und -Computer und wählen Sie die Domain aus, in der Sie Ihre Organisationseinheit erstellen möchten.
3. Klicken Sie mit der rechten Maustaste auf die Domain und wählen Sie Neu und dann Organisationseinheit aus.
4. Geben Sie einen Namen für die Organisationseinheit ein.

   Aktivieren Sie die Option Container vor versehentlichem Löschen schützen.
5. Wählen Sie OK aus. Ihre neue Organisationseinheit wird unter Ihrer Domain angezeigt.

Denn AWS Managed Microsoft AD der Name dieser Organisationseinheit basiert auf dem NetBIOS-Namen, den Sie bei der Erstellung Ihres Verzeichnisses eingegeben haben. Diese Organisationseinheit gehört all Ihren AWS zugehörigen Verzeichnisobjekten AWS und enthält diese, über die Sie volle Kontrolle haben. Standardmäßig OUs gibt es in dieser Organisationseinheit zwei untergeordnete Elemente, nämlich Computer und Benutzer. Neu OUs , die RDS Custom erstellt, sind untergeordnete Elemente der Organisationseinheit, die auf dem NetBIOS basiert.

Schritt 2: Erstellen Sie einen AD-Domänenbenutzer

Die Anmeldeinformationen des Domänenbenutzers werden für das Geheimnis in Secrets Manager verwendet.

Erstellen Sie einen AD-Domänenbenutzer in Ihrem AD

1. Öffnen Sie Active Directory-Benutzer und -Computer und wählen Sie die Domäne und Organisationseinheit aus, in der Sie den Benutzer erstellen möchten.
2. Klicken Sie mit der rechten Maustaste auf das Objekt Benutzer und wählen Sie Neu und dann Benutzer aus.
3. Geben Sie einen Vornamen, Nachnamen und Anmeldenamen für den Benutzer ein. Klicken Sie auf Weiter.

4. Geben Sie ein Passwort für den Benutzer ein. Wählen Sie nicht „Benutzer muss das Passwort bei der nächsten Anmeldung ändern“ oder „Konto ist deaktiviert“. . Klicken Sie auf Weiter.
5. Klicken Sie auf OK. Ihr neuer Benutzer erscheint unter Ihrer Domain.

Schritt 3: Delegieren Sie die Kontrolle an den AD-Benutzer im selbstverwalteten Modus oder AWS Managed Microsoft AD

So delegieren Sie die Kontrolle an den AD-Domain-Benutzer in Ihrer Domain

1. Öffnen Sie das MMC-Snap-In „Active Directory-Benutzer und -Computer“ und wählen Sie Ihre Domäne aus.
2. Klicken Sie mit der rechten Maustaste auf die zuvor erstellte Organisationseinheit und wählen Sie Delegate Control aus.
3. Klicken Sie im Delegierungssteuerungsassistenten auf Weiter.
4. Klicken Sie im Abschnitt Benutzer oder Gruppen auf Hinzufügen.
5. Geben Sie im Feld Benutzer, Computer oder Gruppen auswählen den AD-Benutzer ein, den Sie erstellt haben, und klicken Sie auf Namen überprüfen. Wenn Ihre AD-Benutzerprüfung erfolgreich ist, klicken Sie auf OK.
6. Vergewissern Sie sich im Abschnitt Benutzer oder Gruppen, dass Ihr AD-Benutzer hinzugefügt wurde, und klicken Sie auf Weiter.
7. Wählen Sie im Abschnitt Zu delegierende Aufgaben die Option Benutzerdefinierte Aufgabe zum Delegieren erstellen aus und klicken Sie auf Weiter.
8. Gehen Sie im Abschnitt Active Directory-Objekttyp wie folgt vor:

   Wählen Sie ONly die folgenden Objekte im Ordner aus.

   Wählen Sie Computerobjekte

   Wählen Sie Ausgewählte Objekte in diesem Ordner erstellen

   Wählen Sie Ausgewählte Objekte in diesem Ordner löschen aus und klicken Sie auf Weiter.
9. Gehen Sie im Abschnitt „Berechtigungen“ wie folgt vor:

   Behalten Sie die Auswahl von Allgemein bei.

   Wählen Sie Überprüfter Schreibvorgang in den DNS-Hostnamen aus.

Wählen Sie Überprüfter Schreibvorgang in den Service-Prinzipalnamen aus und klicken Sie auf Weiter.

10. Bestätigen Sie unter Abschließen des Assistenten zur Delegierung von Kontrollbefugnissen Ihre Einstellungen und klicken Sie auf Fertig stellen.

Schritt 4: Erstellen Sie ein Geheimnis

Erstellen Sie den geheimen Schlüssel in derselben AWS-Konto Region, die die DB-Instance RDS Custom for SQL Server enthält, die Sie in Ihr Active Directory aufnehmen möchten. Speichern Sie die Anmeldeinformationen des AD-Domänenbenutzers, der in erstellt wurdeSchritt 2: Erstellen Sie einen AD-Domänenbenutzer.

Console

- Wählen AWS Secrets Manager Sie unter Neues Geheimnis speichern aus.
- Als Secret-Typ wählen Sie Anderer Secret-Typ aus.
- Fügen Sie für Schlüssel/Wert-Paare zwei Schlüssel hinzu:
  - Geben Sie den ersten Schlüssel ein `SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME` und geben Sie den Namen Ihres AD-Benutzers (ohne das Domainpräfix) als Wert ein.
  - Geben Sie für den zweiten Schlüssel das Passwort für Ihren AD-Benutzer in Ihrer Domain ein `SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD` und geben Sie es ein.
- Geben Sie als Verschlüsselungsschlüssel denselben AWS KMS Schlüssel ein, mit dem Sie die Instanz RDS Custom for SQL Server erstellt haben.
- Wählen Sie unter Geheimer Name den geheimen Namen aus, der mit beginntdo`-not-delete-rds-custom-`, damit Ihr Instanzprofil auf dieses Geheimnis zugreifen kann. WENN Sie einen anderen Namen für das Geheimnis wählen möchten, aktualisieren Sie, `RDSCustomInstanceProfile` um auf Geheimer Name zuzugreifen.
- (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für den Secret-Namen ein.
- Fügen Sie die Tags hinzu `Key="AWSRDSCustom",Value="custom-sqlserver"`
- Klicken Sie auf Speichern und dann auf Weiter.
- Behalten Sie für Rotationseinstellungen konfigurieren die Standardwerte bei und wählen Sie Weiter aus.
- Überprüfen Sie die Einstellungen für das Secret und klicken Sie auf Speichern.

- Wählen Sie das neue Geheimnis und kopieren Sie den Wert für Secret ARN. Wir verwenden dies im nächsten Schritt, um Ihr Active Directory einzurichten.

CLI

Führen Sie den folgenden Befehl in Ihrer CLI aus, um ein Geheimnis zu erstellen:

```
# Linux based
aws secretsmanager create-secret \
--name do-not-delete-rds-custom-DomainUserCredentails \
--description "Active directory user credentials for managing RDS Custom" \
--secret-string "{\"SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME\":\"tester\",
\"SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD\":\"xxxxxxxx\"}" \
--kms-key-id <RDSCustomKMSKey> \
--tags Key="AWSRDSCustom",Value="custom-sqlserver"

# Windows based
aws secretsmanager create-secret ^
--name do-not-delete-rds-custom-DomainUserCredentails ^
--description "Active directory user credentials for managing RDS Custom" ^
--secret-string "{\"SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME\":\"tester\",
\"SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD\":\"xxxxxxxx\"}" ^
--kms-key-id <RDSCustomKMSKey> ^
--tags Key="AWSRDSCustom",Value="custom-sqlserver"
```

Schritt 5: Erstellen oder ändern Sie eine RDS Custom for SQL Server-DB-Instance

Erstellen oder ändern Sie eine RDS Custom for SQL Server-DB-Instance zur Verwendung mit Ihrem Verzeichnis. Sie können die Konsole, CLI oder RDS-API verwenden, um eine DB-Instance einem Verzeichnis zuzuordnen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Erstellen Sie mit der Konsole, dem create-db-instanceCLI-Befehl oder dem Vorgang Create DBInstance RDS API eine neue SQL Server-DB-Instance.

  Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine vorhandene SQL Server-DB-Instance mithilfe der Konsole, des modify-db-instanceCLI-Befehls oder der Operation Modify DBInstance RDS API.

  Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.

- Stellen Sie mithilfe der Konsole, des CLI-Befehls -db-snapshot oder des API-Vorgangs Restore From DBSnapshot RDS eine SQL restore-db-instance-fromServer-DB-Instance DBInstance aus einem DB-Snapshot wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.

- Stellen Sie eine SQL Server-DB-Instance point-in-time mithilfe der Konsole, des Befehls restore-db-instance-to- point-in-time CLI oder der Operation Restore DBInstance ToPointInTime RDS API in einer wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

> **Note**
>
> Wenn Ihre RDS Custom for SQL Server-Instanz bereits manuell mit einem AD verbunden ist, überprüfen Sie die Einstellungen für Portregeln für die NetzwerkkonfigurationNetzwerkvalidierung, und führen Sie die Schritte 1 bis Schritt 4 aus. Aktualisieren Sie `--domain-fqdn`-`-domain-ou`, und `--domain-auth-secret-arn` für Ihr AD, sodass Anmeldeinformationen und Konfigurationen für den Domänenbeitritt bei RDS Custom registriert sind, um CNAME zu überwachen, zu registrieren und Wiederherstellungsmaßnahmen zu ergreifen.

Wenn Sie den verwenden AWS CLI, sind die folgenden Parameter erforderlich, damit die DB-Instance das von Ihnen erstellte Verzeichnis verwenden kann:

- Verwenden Sie für den `--domain-fqdn` Parameter den vollqualifizierten Domainnamen Ihres selbstverwalteten AD.
- Verwenden Sie für den Parameter `--domain-ou` die Organisationseinheit, die Sie in Ihrem selbstverwalteten AD erstellt haben.
- Verwenden Sie für den `--domain-auth-secret-arn` Parameter den Wert des Secret ARN, den Sie erstellt haben.

> **Important**
>
> Wenn Sie eine DB-Instance so ändern, dass sie einer selbstverwalteten AD-Domäne beitritt oder aus ihr entfernt wird AWS Managed Microsoft AD, ist ein Neustart der DB-Instance erforderlich, damit die Änderung wirksam wird. Sie können wählen, ob Sie die Änderungen sofort übernehmen oder bis zum nächsten Wartungsfenster warten möchten. Wenn Sie die Option Sofort anwenden wählen, führt dies zu Ausfallzeiten für eine Single-AZ-DB-Instance. Ein Multi-AZ-DB-Cluster führt vor Abschluss eines Neustarts einen Failover durch. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Der folgende CLI-Befehl erstellt eine neue RDS Custom for SQL Server-DB-Instance und verbindet sie mit einer selbstverwalteten oder einer AWS Managed Microsoft AD Domäne.

Für LinuxmacOS, oderUnix:

```
aws rds create-db-instance  \
--engine custom-sqlserver-se \
--engine-version 15.00.4312.2.v1 \
--db-instance-identifier my-custom-instance \
--db-instance-class db.m5.large \
--allocated-storage 100 --storage-type io1 --iops 1000 \
--master-username my-master-username \
--master-user-password my-master-password \
--kms-key-id  my-RDSCustom-key-id \
--custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance  \
--domain-fqdn "corp.example.com" \
--domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" \
--domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-not-
delete-rds-custom-my-AD-test-secret-123456" \
--db-subnet-group-name my-DB-subnet-grp \
--vpc-security-group-ids  my-securitygroup-id \
--no-publicly-accessible \
--backup-retention-period 3 \
--port 8200 \
--region us-west-2 \
--no-multi-az
```

Windows:

```
aws rds create-db-instance  ^
--engine custom-sqlserver-se ^
--engine-version 15.00.4312.2.v1 ^
--db-instance-identifier my-custom-instance ^
--db-instance-class db.m5.large ^
--allocated-storage 100 --storage-type io1 --iops 1000 ^
--master-usernamemy-master-username ^
--master-user-password my-master-password ^
--kms-key-id  my-RDSCustom-key-id ^
--custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance  ^
--domain-fqdn "corp.example.com" ^
--domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" ^
--domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-not-
delete-rds-custom-my-AD-test-secret-123456" ^
--db-subnet-group-name my-DB-subnet-grp ^
--vpc-security-group-ids  my-securitygroup-id ^
--no-publicly-accessible ^
--backup-retention-period 3 ^
--port 8200 ^
--region us-west-2 ^
--no-multi-az
```

> **Important**
>
> Wenn Ihr NetBIOS für corpexample AWS Managed Microsoft AD ist, wird es selbst als OU angezeigt. Jede zuvor erstellte neue Organisationseinheit wird als verschachtelte Organisationseinheit angezeigt. Für AWS Managed Microsoft AD, auf eingestellt`--domain-ou. "OU=RDSCustomOU,OU=corpexample,DC=corp,DC=example,DC=com"`

Mit dem folgenden Befehl wird eine bestehende RDS-Custom for SQL Server-DB-Instance so geändert, dass sie eine Active Directory-Domäne verwendet.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --domain-fqdn "corp.example.com" \
    --domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-
not-delete-rds-custom-my-AD-test-secret-123456" \
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --domain-fqdn "corp.example.com" ^
    --domain-ou "OU=RDSCustomOU,DC=corp,DC=example,DC=com" ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:do-not-delete-rds-custom-my-AD-test-secret-123456" ^
```

Mit dem folgenden CLI-Befehl wird eine DB-Instance vom Typ RDS Custom for SQL Server aus einer Active Directory-Domäne entfernt.

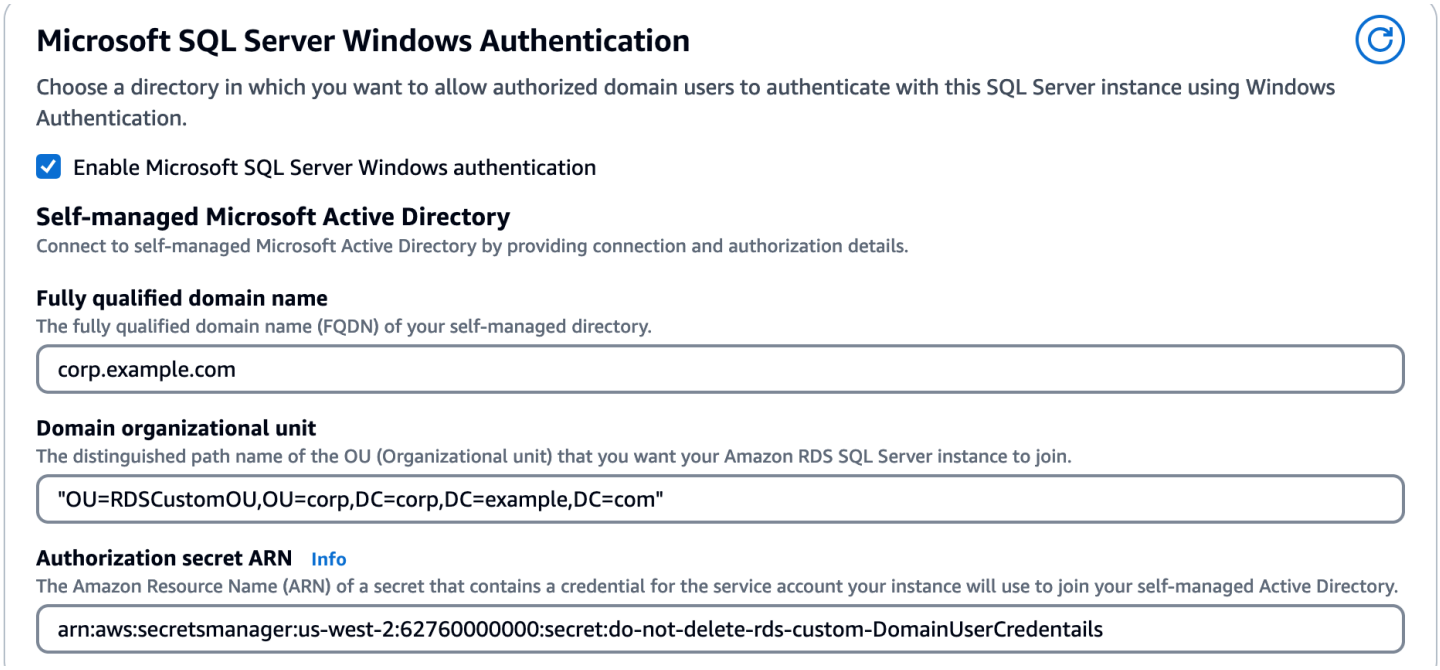
Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --disable-domain
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --disable-domain
```

Wenn Sie die Konsole verwenden, um Ihre Instanz zu erstellen oder zu ändern, klicken Sie auf Microsoft SQL Server Windows Authentication aktivieren, um die folgenden Optionen zu sehen.



Sie sind dafür verantwortlich, sicherzustellen, dass Ihr Domänen-FQDN in die IP-Adressen des Domänencontrollers aufgelöst wird. Wenn der Domänencontroller nicht IPs aufgelöst wird, schlagen die Domänenbeitrittsvorgänge fehl, aber die Erstellung der RDS-Instanz Custom for SQL Server ist erfolgreich. Informationen zur Problembehebung finden Sie unter Problembehebung bei Active Directory.

Schritt 6: Erstellen Sie die SQL Server-Anmeldung für die Windows-Authentifizierung

Verwenden Sie die Anmeldeinformationen für den Amazon-RDS-Masterbenutzer, um eine Verbindung zur SQL-Server-DB-Instance herzustellen, wie Sie es bei jeder anderen DB-Instance tun würden. Da die DB-Instance mit der AD-Domäne verbunden ist, können Sie SQL Server-Logins und -Benutzer bereitstellen. Sie tun dies über das AD-Hilfsprogramm für Benutzer und Gruppen in Ihrer AD-Domäne. Datenbankberechtigungen werden über die Standard-SQL-Server-Berechtigungen verwaltet, die für die Windows-Anmeldungen gewährt und widerrufen werden.

Damit sich ein AD-Benutzer bei SQL Server authentifizieren kann, muss ein SQL Server-Windows-Anmeldename für den AD-Benutzer oder eine Active Directory-Gruppe vorhanden sein, der der Benutzer angehört. Eine differenzierte Zugriffskontrolle wird durch das Gewähren und Widerrufen von Berechtigungen für diese SQL Server-Anmeldungen gewährleistet. Ein AD-Benutzer, der keinen SQL Server-Anmeldenamen hat oder zu einer AD-Gruppe mit einem solchen Anmeldenamen gehört, kann nicht auf die SQL Server-DB-Instance zugreifen.

Die `ALTER ANY LOGIN` Berechtigung ist erforderlich, um eine AD SQL Server-Anmeldung zu erstellen. Wenn Sie keine Logins mit dieser Berechtigung erstellt haben, stellen Sie mithilfe der SQL

Server-Authentifizierung eine Verbindung als Masterbenutzer der DB-Instance her und erstellen Sie Ihre AD SQL Server-Logins im Kontext des Masterbenutzers.

Sie können einen DDL-Befehl (Data Definition Language) wie den folgenden ausführen, um einen SQL Server-Anmeldenamen für einen AD-Benutzer oder eine AD-Gruppe zu erstellen.

```
USE [master]
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
 DEFAULT_LANGUAGE = [us_english];
GO
```

Benutzer (sowohl Menschen als auch Anwendungen) aus Ihrer Domäne können jetzt mithilfe der Windows-Authentifizierung von einem Client-Computer aus, der in eine Domäne eingebunden ist, eine Verbindung zur RDS Custom for SQL Server-Instanz herstellen.

Schritt 7: Verwenden der Kerberos- oder NTLM-Authentifizierung

NTLM-Authentifizierung mithilfe des RDS-Endpunkts

Jede Amazon RDS-DB-Instance hat einen Endpunkt und jeder Endpunkt hat einen DNS-Namen und eine Portnummer für die DB-Instance. Um eine Verbindung mir Ihrer DB-Instance mit einer SQL-Client-Anwendung herzustellen, benötigen Sie den DNS-Namen und die Portnummer für Ihre DB-Instance. Um sich mithilfe der NTLM-Authentifizierung zu authentifizieren, müssen Sie eine Verbindung zum RDS-Endpunkt herstellen.

Während einer geplanten Datenbankwartung oder einer ungeplanten Serviceunterbrechung führt Amazon RDS automatisch einen Failover zur up-to-date sekundären Datenbank durch, sodass der Betrieb schnell und ohne manuelles Eingreifen wieder aufgenommen werden kann. Die primären und sekundären Instances verwenden denselben Endpunkt, dessen physische Netzwerkadresse im Rahmen des Failover-Prozesses auf die sekundäre übertragen wird. Sie müssen Ihre Anwendung nicht neu konfigurieren, wenn ein Failover auftritt.

Kerberos-Authentifizierung

Für die Kerberos-basierte Authentifizierung für RDS Custom for SQL Server müssen Verbindungen zu einem bestimmten Service Principal Name (SPN) hergestellt werden. Nach einem Failover-Ereignis ist der Anwendung der neue SPN jedoch möglicherweise nicht bekannt. Um dieses Problem zu lösen, bietet RDS Custom for SQL Server einen Kerberos-basierten Endpunkt.

Der Kerberos-basierte Endpunkt folgt einem bestimmten Format. Wenn Ihr RDS-Endpunkt dies ist*rds-instance-name*.*account-region-hash*.*aws-region*.rds.amazonaws.com, wäre dies der entsprechende Kerberos-basierte Endpunkt. *rds-instance-name*.*account-region-hash*.*aws-region*.awsrds.*fully qualified domain name (FQDN)*

Wenn der RDS-Endpunkt beispielsweise lautet `ad-test.cocv6zwtircu.us-east-1.rds.amazonaws.com` und der Domänenname lautet`corp-ad.company.com`, wäre dies der Kerberos-basierte Endpunkt. `ad-test.cocv6zwtircu.us-east-1.awsrds.corp-ad.company.com`

Dieser Kerberos-basierte Endpunkt kann zur Authentifizierung bei der SQL Server-Instanz mithilfe von Kerberos verwendet werden, auch nach einem Failover-Ereignis, da der Endpunkt automatisch aktualisiert wird, sodass er auf den neuen SPN der primären SQL Server-Instanz verweist.

Finden Sie Ihren CNAME

Um Ihren CNAME zu finden, stellen Sie eine Verbindung zu Ihrem Domain-Controller her und öffnen Sie den DNS-Manager. Navigieren Sie zu Forward Lookup Zones und Ihrem FQDN.

Navigieren Sie durch Auszeichnungen, AWS-Region und konto - und regionsspezifischen Hash.

Wenn Sie die benutzerdefinierte EC2 RDS-Instance verbinden und versuchen, mithilfe von CNAME eine lokale Verbindung zur Datenbank herzustellen, verwendet Ihre Verbindung die NTLM-Authentifizierung anstelle von Kerberos.

Wenn nach der Verbindung mit CNAME vom Remote-Client aus eine NTLM-Verbindung zurückgegeben wird, überprüfen Sie, ob die erforderlichen Ports auf der Zulassungsliste stehen.

Führen Sie die folgende Abfrage aus, um zu überprüfen, ob Ihre Verbindung Kerberos verwendet:

```
SELECT net_transport, auth_scheme
    FROM sys.dm_exec_connections
    WHERE session_id = @@SSPID;
```

Verwalten einer DB-Instance in einer Domäne

Sie können die Konsole oder Amazon verwenden AWS CLI, RDS API um Ihre DB-Instance und ihre Beziehung zu Ihrer Domain zu verwalten. Beispielsweise können Sie die DB-Instance in Domänen, aus Domänen oder zwischen Domänen verschieben.

Mit Amazon können Sie RDS API beispielsweise Folgendes tun:

- Um einen erneuten Domainbeitritt für eine fehlgeschlagene Mitgliedschaft zu versuchen, verwenden Sie die odifyDBInstanceAPIM-Operation und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft an.
- Um den IAM Rollennamen für die Mitgliedschaft zu aktualisieren, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft und die neue IAM Rolle an.
- Um eine DB-Instance aus einer Domäne zu entfernen, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie `none` als Domänenparameter an.
- Um eine DB-Instance von einer Domain in eine andere zu verschieben, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie den Domain-Identifier der neuen Domain als Domänenparameter an.
- Verwenden Sie die escribeDBInstances API Operation D, um die Mitgliedschaft für jede DB-Instance aufzulisten.

Wiederherstellen einer RDS benutzerdefinierten SQL Server-DB-Instance und Hinzufügen zu einer Active Directory-Domäne

Sie können einen DB-Snapshot wiederherstellen oder point-in-time recovery (PITR) für eine SQL Server-DB-Instance ausführen und sie dann zu einer Active Directory-Domäne hinzufügen. Sobald die DB-Instance wiederhergestellt ist, ändern Sie die Instance mithilfe des unter Hinzufügen der DB-Instance Schritt 5: Erstellen oder ändern Sie eine RDS Custom for SQL Server-DB-Instance zu einer AD-Domäne erläuterten Verfahrens.

Grundlegendes zur Domänenmitgliedschaft

Nachdem Sie Ihre DB-Instance erstellt oder modifiziert haben, wird die Instance ein Mitglied der Domäne. Die AWS Konsole zeigt den Status der Domänenmitgliedschaft für die DB-Instance an. Der Status der DB-Instance kann einer der folgenden sein:

- joined – Die Instance ist Mitglied der Domäne.
- joining – Die Instance ist gerade im Prozess, Mitglied einer Domäne zu werden.
- pending-join – Die Mitgliedschaft der Instance steht noch aus.
- pending-maintenance-join— versucht, AWS die Instance während des nächsten geplanten Wartungsfensters zu einem Mitglied der Domain zu machen.
- pending-removal – Das Entfernen der Instance von der Domäne steht noch aus.

- pending-maintenance-removal— versucht, AWS die Instanz während des nächsten geplanten Wartungsfensters aus der Domain zu entfernen.
- failed – Ein Konfigurationsproblem hat verhindert, dass die Instance mit der Domäne verbunden werden konnte. Überprüfen und korrigieren Sie Ihre Konfiguration, bevor Sie den Befehl zu Änderung der Instance erneut ausführen.
- removing – Die Instance wird gerade von der Domäne entfernt.

Eine Anfrage, Mitglied einer Domäne zu werden, kann aufgrund eines Problems mit der Netzwerkverbindung oder einer falschen IAM Rolle fehlschlagen. Beispielsweise können Sie eine DB-Instance erstellen oder eine vorhandene Instance ändern und der Versuch, dass die DB-Instance Mitglied einer Domäne wird, wird fehlschlagen. Geben Sie in diesem Fall entweder den Befehl, um die DB-Instance zu erstellen oder zu ändern, neu aus oder ändern Sie die neu erstellte Instance, um der Domäne beizutreten.

Problembehebung bei Active Directory

Die folgenden Probleme können auftreten, wenn Sie ein AD einrichten oder ändern.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 2 / 0x2	Die angegebene Datei wurde nicht gefunden.	Das Format oder der Speicherort für die Organisationseinheit, das/der mit dem Parameter `–domain-ou` angegeben wurde, ist ungültig. Das über AWS Secrets Manager angegebene Domänendienstkonto verfügt nicht über die erforderlichen Berechtigungen, um der Organisationseinheit beizutreten.	Überprüfen Sie den Parameter `–domain-ou` . Stellen Sie sicher, dass das Domain-Servicekonto über die richtigen Berechtigungen für die Organisationseinheit verfügt.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 5 / 0x5	Zugriff verweigert.	Falsch konfigurierte Berechtigungen für das Domain-Servicekonto oder das Computerkonto ist bereits in der Domain vorhanden.	Überprüfen Sie die Berechtigungen für das Domänendienstkonto in der Domäne und stellen Sie sicher, dass das RDS Computerkonto in der Domäne nicht doppelt vorhanden ist. Sie können den Namen des RDS Computerkontos überprüfen, indem Sie es `SELECT @@SERVERNAME` auf Ihrer RDS Custom for SQL Server-DB-Instance ausführen. Wenn Sie Multi-AZ verwenden, versuchen Sie einen Neustart mit Failover und überprüfen Sie dann erneut, ob das RDS Computerkonto vorhanden ist. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 87 / 0x57	Der Parameter ist falsch.	Das über AWS Secrets Manager angegebene Domänendienstkonto verfügt nicht über die richtigen Berechtigungen. Auch das Benutzerprofil ist möglicherweise beschädigt.	Prüfen Sie die Anforderungen für das Domain-Servicekonto.
Fehler 234 / 0xEA	Die angegebene Organisationseinheit ist nicht vorhanden.	Die mit dem `–domain-ou` Parameter angegebene Organisationseinheit ist in Ihrem AD nicht vorhanden.	Überprüfen Sie den `–domain-ou` Parameter und stellen Sie sicher, dass die angegebene Organisationseinheit in Ihrem AD vorhanden ist.
Fehler 1326 / 0x52E	Der Benutzername oder das Passwort ist falsch.	Die in AWS Secrets Manager angegebenen Anmeldeinformationen für das Domänendienstkonto enthalten einen unbekannten Benutzernamen oder ein falsches Passwort. Das Domänenkonto kann auch in Ihrem AD deaktiviert sein.	Stellen Sie sicher, dass die in AWS Secrets Manager angegebenen Anmeldeinformationen korrekt sind und das Domänenkonto in Ihrem Active Directory aktiviert ist.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 1355 / 0x54B	Die angegebene Domain ist nicht vorhanden oder konnte nicht kontaktiert werden.	Die Domäne ist ausgefallen, die angegebenen Gruppen DNS IPs sind nicht erreichbar, oder die angegebene Domäne FQDN ist nicht erreichbar.	Stellen Sie sicher, dass die Parameter `–domain-dns-ips` und `–domain-fqdn` korrekt sind. Überprüfen Sie die Netzwerkkonfiguration Ihrer RDS Custom for SQL Server-DB-Instance und stellen Sie sicher, dass Ihr AD erreichbar ist.
Fehler 1722/0x6BA	Der RPC Server ist nicht verfügbar.	Beim Zugriff auf den RPC Dienst Ihrer AD-Domain ist ein Problem aufgetreten. Dies könnte auf einen Service- oder Netzwerkfehler zurückzuführen sein.	Stellen Sie sicher, dass der RPC Dienst auf Ihren Domänencontrollern ausgeführt wird und dass die TCP Ports `135` und in Ihrer Domain von Ihrer RDS Custom for SQL Server-DB-Instance aus erreichbar `49152-65535` sind.
Fehler 2224 / 0x8B0	Das Benutzerkonto besteht bereits.	Das Computerkonto, das zu Ihrem AD hinzugefügt werden soll, ist bereits vorhanden.	Identifizieren Sie das Computerkonto, indem Sie es `SELECT @@SERVERNAME` auf Ihrer RDS Custom for SQL Server-DB-Instance ausführen, und entfernen Sie es dann vorsichtig aus Ihrem AD.

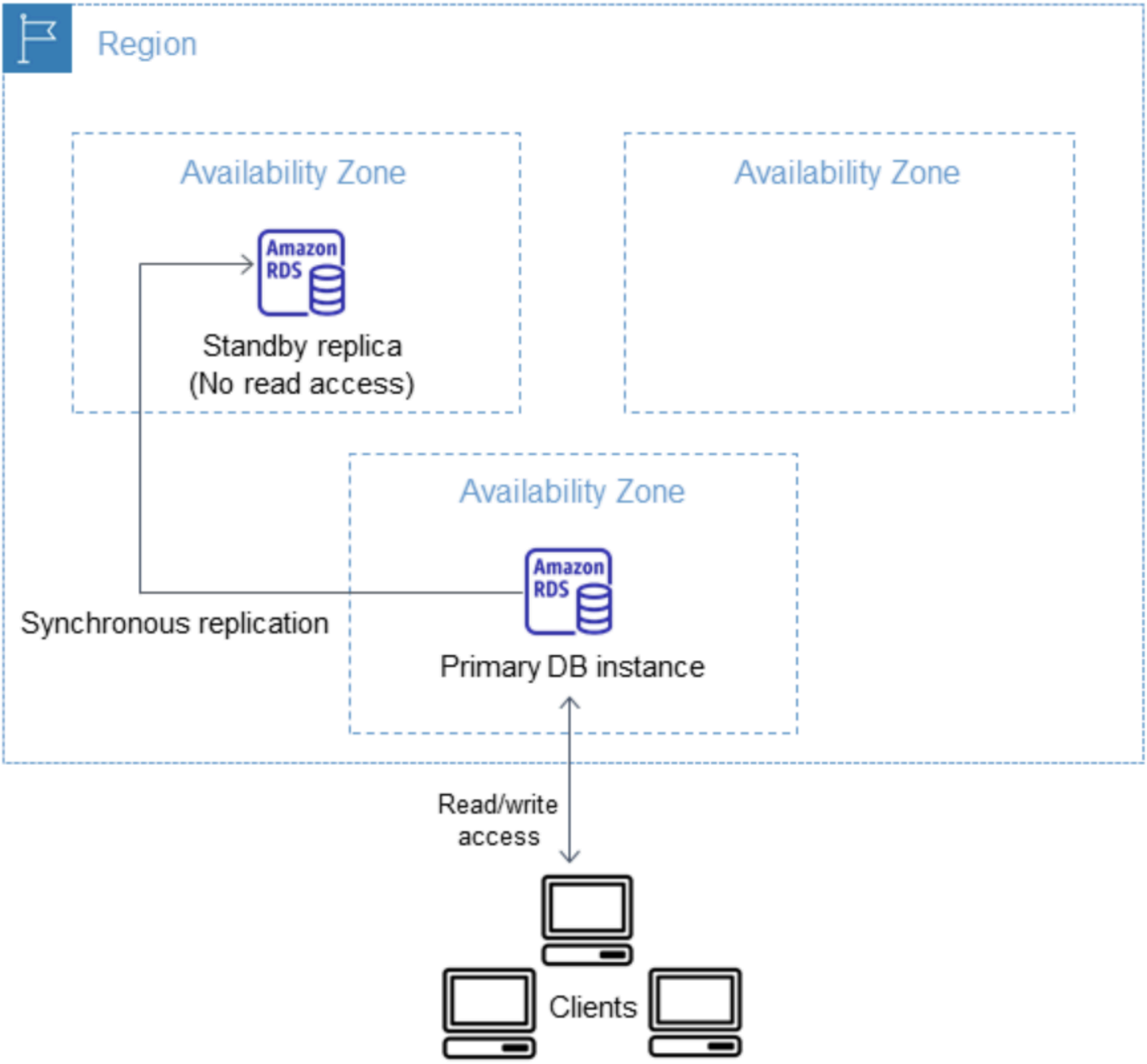
Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 2242 / 0x8c2	Das Passwort dieses Benutzers ist abgelaufen.	Das Passwort für das über AWS Secrets Manager angegebene Domänendienstkonto ist abgelaufen.	Aktualisieren Sie das Passwort für das Domänendienstkonto, das verwendet wurde, um Ihre RDS Custom for SQL Server-DB-Instance mit Ihrem AD zu verbinden.

Verwaltung einer Multi-AZ-Bereitstellung für RDS Custom for Server SQL

In einer Multi-AZ-DB-Instance-Bereitstellung für RDS Custom for SQL Server stellt Amazon RDS automatisch ein synchrones Standby-Replikat in einer anderen Availability Zone (AZ) bereit und verwaltet es. Die primäre DB-Instance wird über die Availability Zone synchron auf ein Standby-Replikat repliziert, um Datenredundanz zu erzielen.

> **Important**
>
> Eine Multi-AZ-Bereitstellung für RDS Custom for SQL Server unterscheidet sich von Multi-AZ für Server. RDS SQL Im Gegensatz zu Multi-AZ RDS for SQL Server müssen Sie die Voraussetzungen für RDS Custom for SQL Server einrichten, bevor Sie Ihre Multi-AZ-DB-Instance erstellen, da RDS Custom innerhalb Ihres eigenen Kontos ausgeführt wird, wofür Berechtigungen erforderlich sind.
> Wenn Sie die Voraussetzungen nicht erfüllen, kann Ihre Multi-AZ-DB-Instance möglicherweise nicht ausgeführt werden oder wird automatisch in eine Single-AZ-DB-Instance geändert. Weitere Informationen über Voraussetzungen finden Sie unter Voraussetzungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Server SQL.

Wenn Sie eine DB-Instance mit hoher Verfügbarkeit ausführen, kann dies die Verfügbarkeit bei geplanten Systemwartungen verbessern. Im Falle einer geplanten Datenbankwartung oder einer ungeplanten Serviceunterbrechung wechselt Amazon RDS automatisch zur up-to-date sekundären DB-Instance. Mit dieser Funktion können Datenbankoperationen schnell ohne manuellen Eingriff fortgesetzt werden. Die Primär- und Standby-Instances verwenden denselben Endpunkt, dessen physische Netzwerkadresse als Teil des Failoverprozesses am sekundären Replica gespiegelt wird. Sie müssen Ihre Anwendung nicht neu konfigurieren, wenn ein Failover auftritt.



Sie können eine Multi-AZ-Bereitstellung „RDSBenutzerdefiniert für SQL Server“ erstellen, indem Sie bei der Erstellung einer RDS benutzerdefinierten DB-Instance Multi-AZ angeben. Sie können die Konsole verwenden, um bestehende RDS Custom for SQL Server-DB-Instances in Multi-AZ-Bereitstellungen zu konvertieren, indem Sie die DB-Instance ändern und die Multi-AZ-Option angeben. Sie können auch eine Multi-AZ-DB-Instance-Bereitstellung mit dem AWS CLI oder Amazon RDS API angeben.

In der RDS Konsole wird die Availability Zone des Standby-Replikats (der sekundären AZ) angezeigt. Sie können auch den `describe-db-instances` CLI Befehl oder die `DescribeDBInstances` API Operation verwenden, um die sekundäre AZ zu finden.

RDSBenutzerdefiniert für SQL Server-DB-Instances mit Multi-AZ-Bereitstellung können im Vergleich zu einer Single-AZ-Bereitstellung eine höhere Schreib- und Commit-Latenz aufweisen. Dieser Anstieg kann aufgrund der synchronen Datenreplikation zwischen DB-Instances entstehen. Es kann zu einer Änderung der Latenz kommen, wenn bei Ihrer Bereitstellung ein Failover auf das Standby-Replikat erfolgt, obwohl es auf Netzwerkkonnektivität mit niedriger Latenz zwischen Availability Zones ausgelegt AWS ist.

> **Note**
>
> Für Produktionsworkloads empfehlen wir, eine DB-Instance-Klasse mit Provisioned IOPS (Eingabe-/Ausgabeoperationen pro Sekunde) zu verwenden, um eine schnelle, konsistente Leistung zu erzielen. Weitere Informationen zu DB-Instance-Klassen finden Sie unter Anforderungen und Einschränkungen für Amazon RDS Custom for SQL Server.

Themen

- Verfügbarkeit von Regionen und Versionen
- Einschränkungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Server SQL
- Voraussetzungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Server SQL
- Eine RDS benutzerdefinierte SQL Multi-AZ-Bereitstellung für Server erstellen
- Ändern einer Single-AZ-Bereitstellung von RDS Custom für SQL Server in eine Multi-AZ-Bereitstellung
- Ändern einer Multi-AZ-Bereitstellung „RDSBenutzerdefiniert für SQL Server“ in eine Single-AZ-Bereitstellung
- Failover-Prozess für eine Multi-AZ-Bereitstellung „RDSCustom for SQL Server“

Verfügbarkeit von Regionen und Versionen

Multi-AZ-Bereitstellungen für RDS Custom for SQL Server werden für die folgenden Server-Editionen unterstützt: SQL

- SQLServer 2022 und 2019: Enterprise, Standard, Web und Developer Edition

> **Note**
>
> Multi-AZ-Bereitstellungen für RDS Custom for SQL Server werden auf SQL Server 2019 CU8 (15.00.4073.23) oder niedrigeren Versionen nicht unterstützt.

Multi-AZ-Bereitstellungen für RDS Custom for SQL Server sind in allen Regionen verfügbar, in denen Custom for Server verfügbar ist. RDS SQL Weitere Informationen zur regionalen Verfügbarkeit von Multi-AZ-Bereitstellungen für RDS Custom for SQL Server finden Sie unter. Unterstützte Regionen und DB-Engines für RDS Custom for SQL Server

Einschränkungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Server SQL

Multi-AZ-Bereitstellungen mit RDS Custom for SQL Server haben die folgenden Einschränkungen:

- Regionsübergreifende Multi-AZ-Bereitstellungen werden nicht unterstützt.
- Sie können die sekundäre DB-Instance nicht so konfigurieren, dass sie die Datenbankleseaktivität akzeptiert.
- Wenn Sie eine Custom Engine-Version (CEV) mit einer Multi-AZ-Bereitstellung verwenden, verwendet auch Ihre sekundäre DB-Instance dieselbe. CEV Die sekundäre DB-Instance kann keine andere CEV verwenden.

Voraussetzungen für eine Multi-AZ-Bereitstellung mit RDS Custom for Server SQL

Wenn Sie bereits über eine Single-AZ-Bereitstellung „RDSBenutzerdefiniert für SQL Server“ verfügen, sind die folgenden zusätzlichen Voraussetzungen erforderlich, bevor Sie sie in eine Multi-AZ-Bereitstellung ändern können. Sie können wählen, ob Sie die Voraussetzungen manuell oder mit der bereitgestellten CloudFormation Vorlage erfüllen möchten. Die neueste CloudFormation Vorlage enthält die Voraussetzungen für Single-AZ- und Multi-AZ-Bereitstellungen.

> **Important**
>
> Um die Einrichtung zu vereinfachen, empfehlen wir, die neueste AWS CloudFormation Vorlagendatei zu verwenden, die in den Anweisungen zur Netzwerkeinrichtung enthalten ist, um die Voraussetzungen zu erstellen. Weitere Informationen finden Sie unter Konfiguration mit AWS CloudFormation.

> **Note**
>
> Wenn Sie eine bestehende Single-AZ-Bereitstellung RDS benutzerdefiniert für SQL Server in eine Multi-AZ-Bereitstellung ändern, müssen Sie die folgenden Voraussetzungen erfüllen. Wenn Sie die Voraussetzungen nicht erfüllen, schlägt die Multi-AZ-Einrichtung fehl. Führen Sie zum Erfüllen dieser Voraussetzungen die Schritte unter Ändern einer Single-AZ-Bereitstellung von RDS Custom für SQL Server in eine Multi-AZ-Bereitstellung aus.

- Aktualisieren Sie die Regeln für eingehende und ausgehende Nachrichten der RDS Sicherheitsgruppe, sodass Port 1120 zulässig ist.
- Fügen Sie Ihrer privaten Netzwerk-Zugriffskontrollliste (ACL) eine Regel hinzu, die TCP Ports `0-65535` für die DB-Instance zulässt. VPC
- Erstellen Sie neue SQS VPC Amazon-Endpunkte, mit denen die RDS Custom for SQL Server-DB-Instance kommunizieren SQS kann.
- Aktualisieren Sie die SQS Berechtigungen in der Instance-Profilrolle.

Eine RDS benutzerdefinierte SQL Multi-AZ-Bereitstellung für Server erstellen

Gehen Sie wie unter beschrieben vor, um eine RDS benutzerdefinierte Multi-AZ-Bereitstellung für SQL Server zu erstellen. Eine DB-Instance für Amazon RDS Custom for SQL Server erstellen und eine Verbindung zu ihr herstellen

> **Important**
>
> Um die Einrichtung zu vereinfachen, empfehlen wir, die neueste AWS CloudFormation Vorlagendatei zu verwenden, die in den Anweisungen zur Netzwerkeinrichtung enthalten ist. Weitere Informationen finden Sie unter Konfiguration mit AWS CloudFormation.

Das Erstellen einer Multi-AZ-Bereitstellung kann einige Minuten in Anspruch nehmen.

Ändern einer Single-AZ-Bereitstellung von RDS Custom für SQL Server in eine Multi-AZ-Bereitstellung

Sie können eine vorhandene DB-Instance von RDS Custom für SQL Server von einer Single-AZ-Bereitstellung in eine Multi-AZ-Bereitstellung ändern. Wenn Sie die DB-Instance ändern, führt Amazon RDS mehrere Aktionen aus:

- Es wird ein Snapshot der primären DB-Instance aufgenommen.
- Erstellt neue Volumes für das Standby-Replikat aus dem Snapshot. Diese Volumes werden im Hintergrund initialisiert, und die maximale Volume-Leistung wird erreicht, nachdem die Daten vollständig initialisiert wurden.
- Die synchrone Replikation auf Blockebene zwischen der primären und sekundären DB-Instance wird aktiviert.

> **Important**
>
> Wir empfehlen, Ihre DB-Instance von RDS Custom für SQL Server möglichst nicht in Zeiten hoher Aktivität von einer Single-AZ- in eine Multi-AZ-Bereitstellung auf einer Produktions-DB-Instance zu ändern.

AWS verwendet einen Snapshot, um die Standby-Instance zu erstellen, um Ausfallzeiten bei der Konvertierung von Single-AZ zu Multi-AZ zu vermeiden, aber die Leistung kann während und nach der Konvertierung zu Multi-AZ beeinträchtigt werden. Diese Auswirkung kann bei Workloads erheblich sein, die empfindlich auf Schreiblatenz reagieren. Diese Funktion ermöglicht zwar die schnelle Wiederherstellung großer Volumes aus Snapshots, kann jedoch aufgrund der synchronen Replikation zu einer erhöhten Latenz von I/O Vorgängen führen. Diese Latenz kann sich auf die Leistung Ihrer Datenbank auswirken.

> **Note**
>
> Wenn Sie Ihre DB-Instance RDS Custom for SQL Server vor dem 29. August 2024 erstellt haben, aktualisieren Sie vor der Änderung auf die neueste Nebenversion.
>
> - Führen Sie für SQL Server 2019-Instances ein Upgrade der DB-Engine-Version auf `15.00.4410.1.v1` oder höher durch.

- Aktualisieren Sie für SQL Server 2022-Instanzen die DB-Engine-Version auf `16.00.4150.1.v1` oder höher.

Themen

- Konfiguration der Voraussetzungen für die Änderung einer Single-AZ- zu einer Multi-AZ-Bereitstellung mit CloudFormation
- Konfigurieren der Voraussetzungen für die manuelle Änderung einer Single-AZ- in eine Multi-AZ-Bereitstellung
- Ändern Sie mithilfe der RDS-Konsole, AWS CLI oder RDS-API.

Konfiguration der Voraussetzungen für die Änderung einer Single-AZ- zu einer Multi-AZ-Bereitstellung mit CloudFormation

Um eine Multi-AZ-Bereitstellung zu verwenden, müssen Sie sicherstellen, dass Sie die neueste CloudFormation Vorlage mit den Voraussetzungen angewendet haben, oder die neuesten Voraussetzungen manuell konfigurieren. Wenn Sie die neueste Vorlage CloudFormation für die Voraussetzungen bereits angewendet haben, können Sie diese Schritte überspringen.

Um die Voraussetzungen für die Multi-AZ-Bereitstellung von RDS Custom for SQL Server zu konfigurieren, verwenden Sie CloudFormation

1. Öffnen Sie die CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
2. Um den Assistenten zum Erstellen von Stacks zu starten, wählen Sie den vorhandenen Stack aus, mit dem Sie eine Single-AZ-Bereitstellung erstellt haben, und klicken Sie auf Aktualisieren.

   Die Seite Stack aktualisieren wird angezeigt.
3. Wählen Sie für Voraussetzung – Vorlage vorbereiten die Option Aktuelle Vorlage ersetzen aus.
4. Gehen Sie für Specify template (Vorlage angeben) wie folgt vor:

   a. Laden Sie die neueste AWS CloudFormation Vorlagendatei herunter. Öffnen Sie das Kontextmenü (Rechtsklick) für den Link (custom-sqlserver-onboard.zip) und wählen Sie Link speichern unter.

   b. Speichern und extrahieren Sie die Datei `custom-sqlserver-onboard.json` auf Ihrem Computer.

c. Wählen Sie unter Templete source (Vorlagenquelle) den Wert Upload a template file (Vorlagendatei hochladen) aus.

d. Navigieren Sie unter Choose file (Datei auswählen) zur Datei `custom-sqlserver-onboard.json` und wählen Sie sie aus.

5. Wählen Sie Weiter aus.

   Die Seite Specify DB Details (DB-Details angeben) wird angezeigt.

6. Wenn Sie die Standardoptionen beibehalten möchten, wählen Sie Next (Weiter) aus.

   Die Seite Erweiterte Optionen wird angezeigt.

7. Wenn Sie die Standardoptionen beibehalten möchten, wählen Sie Next (Weiter) aus.

8. Wenn Sie die Standardoptionen beibehalten möchten, wählen Sie Next (Weiter) aus.

9. Führen Sie auf der Seite Änderungen überprüfen die folgenden Schritte aus:

   a. Aktivieren Sie unter Capabilities (Funktionen) das Kontrollkästchen I acknowledge that AWS CloudFormation , das bestätigt, dass IAM-Ressourcen mit benutzerdefinierten Namen erstellen kann.

   b. Wählen Sie Absenden aus.

10. Vergewissern Sie sich, dass das Update erfolgreich ist. Für den Status eines erfolgreichen Vorgangs wird `UPDATE_COMPLETE` angezeigt.

Wenn das Update fehlschlägt, wird jede neue Konfiguration, die im Aktualisierungsprozess angegeben wurde, rückgängig gemacht. Die vorhandene Ressource ist weiterhin verwendbar. Wenn Sie beispielsweise Netzwerk-ACL-Regeln mit den Nummern 18 und 19 hinzufügen, es aber bereits Regeln mit diesen Nummern gibt, würde das Update den folgenden Fehler zurückgeben: `Resource handler returned message: "The network acl entry identified by 18 already exists.` In diesem Szenario können Sie die vorhandenen ACL-Regeln so ändern, dass sie eine Nummer unter 18 verwenden, und dann die Aktualisierung erneut versuchen.

Konfigurieren der Voraussetzungen für die manuelle Änderung einer Single-AZ- in eine Multi-AZ-Bereitstellung

> **Important**
>
> Um die Einrichtung zu vereinfachen, empfehlen wir, die neueste AWS CloudFormation Vorlagendatei zu verwenden, die in den Anweisungen zur Netzwerkeinrichtung enthalten ist.

Weitere Informationen finden Sie unter Konfiguration der Voraussetzungen für die Änderung einer Single-AZ- zu einer Multi-AZ-Bereitstellung mit CloudFormation.

Wenn Sie Voraussetzungen manuell konfigurieren möchten, führen Sie die folgenden Schritte aus.

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie Endpunkt aus. DieErstellen eines Endpunktswird angezeigt.
3. Wählen Sie unter Servicekategorie die Option AWS -Services aus.
4. Suchen Sie unter Dienste nach *SQS*
5. Wählen Sie unter VPC die VPC aus, in der Ihre DB-Instance von RDS Custom für SQL Server bereitgestellt wird.
6. Wählen Sie unter Subnetze die Subnetze aus, in denen Ihre DB-Instance von RDS Custom für SQL Server bereitgestellt wird.
7. Wählen Sie unter Sicherheitsgruppen die *-vpc-endpoint-sg* Gruppe aus.
8. Wählen Sie für Richtlinie die Option Benutzerdefiniert aus.
9. Ersetzen Sie in Ihrer benutzerdefinierten Richtlinie die Werte *AWS partitionRegion*,*accountId*, und *IAM-Instance-role* durch Ihre eigenen Werte.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                }
            },
            "Action": [
                "SQS:SendMessage",
                "SQS:ReceiveMessage",
                "SQS:DeleteMessage",
                "SQS:GetQueueUrl"
            ],
            "Resource": "arn:aws:sqs:us-east-1:111122223333:do-not-delete-
rds-custom-*",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/{IAM-Instance-role}"
            }
        }
    ]
}
```

10. Aktualisieren Sie das Instance-Profil mit der Berechtigung, auf Amazon SQS zuzugreifen. Ersetzen Sie die *AWS partitionRegion*, und *accountId* durch Ihre eigenen Werte.

```
                    {
    "Sid": "SendMessageToSQSQueue",
    "Effect": "Allow",
    "Action": [
      "SQS:SendMessage",
      "SQS:ReceiveMessage",
      "SQS:DeleteMessage",
      "SQS:GetQueueUrl"

    ],
    "Resource": [
      {
        "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
      }
    ],
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
      }
    }
  }
                    >
```

11. Aktualisieren Sie die Regeln für eingehenden und ausgehenden Datenverkehr der Amazon-RDS-Sicherheitsgruppe, um Port 1120 zuzulassen.

    a. Wählen Sie unter Sicherheitsgruppen die *-rds-custom-instance-sg* Gruppe aus.

    b. Erstellen Sie für eingehende Regeln eine benutzerdefinierte TCP-Regel, um Ports *1120* aus der *-rds-custom-instance-sg* Quellgruppe zuzulassen.

c. Erstellen Sie für ausgehende Regeln eine benutzerdefinierte TCP-Regel, um den Port zur *1120* *-rds-custom-instance-sg* Zielgruppe zuzulassen.

12. Fügen Sie Ihrer privaten Netzwerk-Zugriffssteuerungsliste (ACL) eine Regel hinzu, die TCP-Ports `0-65535` für das Quellsubnetz der DB-Instance zulässt.

> **Note**
>
> Notieren Sie sich beim Erstellen einer Regel für eingehenden Datenverkehr und einer Regel für ausgehenden Datenverkehr die höchste vorhandene Regelnummer. Die neuen Regeln, die Sie erstellen, müssen eine Regelnummer unter 100 haben und dürfen mit keiner vorhandenen Regelnummer übereinstimmen.

a. Wählen Sie unter Netzwerk ACLs die *-private-network-acl* Gruppe aus.

b. Erstellen Sie für eingehende Regeln eine All-TCP-Regel, um TCP-Ports `0-65535` mit einer Quelle von *privatesubnet1* und *privatesubnet2* zuzulassen.

c. Erstellen Sie für ausgehende Regeln eine All-TCP-Regel, um TCP-Ports den Zugang zu Ziel *privatesubnet1* - und `0-65535` zu ermöglichen. *privatesubnet2*

Ändern Sie mithilfe der RDS-Konsole, AWS CLI oder RDS-API.

Nachdem Sie die Voraussetzungen erfüllt haben, können Sie eine RDS Custom for SQL Server-DB-Instance mithilfe der RDS-Konsole, AWS CLI oder RDS-API von einer Single-AZ- zu einer Multi-AZ-Bereitstellung ändern.

Konsole

So ändern Sie eine Single-AZ-Bereitstellung von RDS Custom für SQL Server in eine Multi-AZ-Bereitstellung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der Amazon RDS-Konsole Databases (Datenbanken) aus.

   Der Bereich Databases (Datenbanken) wird angezeigt.
3. Wählen Sie die DB-Instance von RDS Custom für SQL Server aus, die Sie ändern möchten.
4. Wählen Sie unter Aktionen die Option In Multi-AZ-Bereitstellung konvertieren aus.

5. Damit Änderungen sofort übernommen werden, wählen Sie die Option Sofort anwenden auf der Seite Bestätigung aus. Die Auswahl dieser Option verursacht keine Ausfallzeiten, kann jedoch zur Beeinträchtigung der Leistung führen. Sie können die Aktualisierung auch im nächsten Wartungsfenster übernehmen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.
6. Wählen Sie auf der Seite Bestätigung die Option Konvertieren in Multi-AZ aus.

AWS CLI

Um mithilfe von zu einer Multi-AZ-DB-Instance-Bereitstellung zu konvertieren AWS CLI, rufen Sie den modify-db-instanceBefehl auf und legen Sie die `--multi-az` Option fest. Geben Sie die DB-Instance-Kennung und die Werte für andere Optionen an, die geändert werden sollen. Informationen zu den jeweiligen Optionen finden Sie unter Einstellungen für DB-Instances.

Example

Mit dem folgenden Code wird `mycustomdbinstance` geändert, indem die Option `--multi-az` hinzugefügt wird. Die Änderungen werden während des nächsten Wartungsfensters (mit) übernommen `--no-apply-immediately`. Verwenden Sie `--apply-immediately`, damit Änderungen sofort angewendet werden. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --multi-az \
    --no-apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --multi-az  \ ^
    --no-apply-immediately
```

RDS-API

Um mit der RDS-API in eine Multi-AZ-DB-Instance-Bereitstellung zu konvertieren, rufen Sie den DBInstance Vorgang Modify auf und setzen Sie den `MultiAZ` Parameter auf true.

Ändern einer Multi-AZ-Bereitstellung „RDSBenutzerdefiniert für SQL Server“ in eine Single-AZ-Bereitstellung

Sie können eine bestehende RDS Custom for SQL Server-DB-Instance von einer Multi-AZ- zu einer Single-AZ-Bereitstellung ändern.

Konsole

Um eine RDS Custom for SQL Server-DB-Instance von einer Multi-AZ- in eine Single-AZ-Bereitstellung zu ändern.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.

   Der Bereich Databases (Datenbanken) wird angezeigt.
3. Wählen Sie die DB-Instance „RDSBenutzerdefiniert für SQL Server“, die Sie ändern möchten.
4. Wählen Sie für die Multi-AZ-Bereitstellung die Option Nein aus.
5. Damit Änderungen sofort übernommen werden, wählen Sie die Option Sofort anwenden auf der Seite Bestätigung aus. Die Auswahl dieser Option verursacht keine Ausfallzeiten, kann jedoch zur Beeinträchtigung der Leistung führen. Sie können die Aktualisierung auch im nächsten Wartungsfenster übernehmen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.
6. Klicken Sie auf der Seite Bestätigung auf DB-Instance ändern.

AWS CLI

Um eine Multi-AZ-Bereitstellung mithilfe von in eine Single-AZ-Bereitstellung umzuwandeln AWS CLI, rufen Sie den modify-db-instanceBefehl auf und fügen Sie die Option hinzu. `--no-multi-az` Geben Sie die DB-Instance-Kennung und die Werte für andere Optionen an, die geändert werden sollen. Informationen zu den jeweiligen Optionen finden Sie unter Einstellungen für DB-Instances.

Example

Mit dem folgenden Code wird `mycustomdbinstance` geändert, indem die Option `--no-multi-az` hinzugefügt wird. Die Änderungen werden während des nächsten Wartungsfensters (mit) übernommen `--no-apply-immediately`. Verwenden Sie `--apply-immediately`, damit

Änderungen sofort angewendet werden. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --no-multi-az  \
    --no-apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --no-multi-az \ ^
    --no-apply-immediately
```

RDS API

Um eine Multi-AZ-Bereitstellung mithilfe von in eine Single-AZ-Bereitstellung umzuwandeln RDSAPI, rufen Sie die odifyDBInstanceM-Operation auf und setzen Sie den `MultiAZ` Parameter auf. `false`

Failover-Prozess für eine Multi-AZ-Bereitstellung „RDSCustom for SQL Server“

Wenn ein geplanter oder ungeplanter Ausfall Ihrer DB-Instance auf einen Infrastrukturdefekt zurückzuführen ist, wechselt Amazon RDS automatisch zu einem Standby-Replikat in einer anderen Availability Zone, wenn Sie Multi-AZ aktiviert haben. Die Dauer, bis der Failover-Prozess abgeschlossen ist, hängt von der Datenbankaktivität sowie von anderen Bedingungen zu dem Zeitpunkt ab, an dem die primäre DB-Instance ausgefallen ist. Der Failover-Prozess dauert normalerweise 60-120 Sekunden. Diese Failover-Dauer kann sich verlängern, wenn umfangreiche Transaktionen oder zeitintensive Wiederherstellungsprozesse durchgeführt werden. Wenn der Failover abgeschlossen ist, kann es länger dauern, bis die RDS Konsole die neue Availability Zone anzeigt.

> **Note**
>
> Sie können ein Failover manuell erzwingen, wenn Sie eine DB-Instance mit Failover neu starten. Weitere Informationen über das Neustarten einer DB-Instance finden Sie unter Neustarten einer DB-Instance.

Amazon RDS wickelt Failovers automatisch ab, sodass Sie den Datenbankbetrieb so schnell wie möglich ohne administrativen Eingriff wieder aufnehmen können. Die primäre DB-Instance schaltet automatisch auf das Standby-Replikat um, wenn eine der in der folgenden Tabelle beschriebenen Bedingungen eintritt: Sie können diese Gründe für das Failover im RDS Ereignisprotokoll einsehen.

Failover-Grund	Beschreibung
`The operating system for the RDS Custom for SQL Server Multi-AZ DB instance is being patched in an offline operation`	Ein Failover wurde während des Wartungsfensters für einen Betriebssystem-Patch oder ein Sicherheitsupdate ausgelöst. Weitere Informationen finden Sie unter Warten einer DB-Instance.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unhealthy.`	Die Multi-AZ-DB-Instance-Bereitstellung hat eine beeinträchtigte primäre DB-Instance erkannt und ein Failover eingeleitet.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unreachable due to loss of network connectivity.`	RDSBei der Überwachung wurde ein Fehler bei der Netzwerke rreichbarkeit der primären DB-Instance festgestellt und ein Failover ausgelöst.
`The RDS Custom for SQL Server Multi-AZ DB instance was modified by the customer.`	Eine Änderung der DB-Instance hat ein Failover ausgelöst. Weitere Informationen finden Sie unter Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance.
`The storage volume of the primary host of the RDS Custom`	Die Multi-AZ-DB-Instance-Bereitstellung hat ein Speicherproblem der primären DB-Instance erkannt und ein Failover eingeleitet.

Failover-Grund	Beschreibung
`for SQL Server Multi-AZ DB instance experienced a failure.`	
`The user requested a failover of the RDS Custom for SQL Server Multi-AZ DB instance.`	Die Multi-AZ-DB-Instance „RDSCustom for SQL Server“ wurde mit einem Failover neu gestartet. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.
`The RDS Custom for SQL Server Multi-AZ primary DB instance is busy or unresponsive.`	Die primäre DB-Instance reagiert nicht. Wir empfehlen, die folgenden Schritte auszuprobieren: • Untersuchen Sie die Ereignisprotokolle und CloudWatch Protokolle auf übermäßige Speicher- oder CPU Auslagerungsspeicherbelegung. Weitere Informationen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten. • Erstellen Sie eine Regel, die bei einem RDS Amazon-Ereignis ausgelöst wird. Weitere Informationen finden Sie unter Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird. • Prüfen Sie Ihren Workload, um festzustellen, ob Sie die angemessene DB-Instance-Klasse verwenden. Weitere Informationen finden Sie unter .

Um festzustellen, ob Ihre Multi-AZ-DB-Instance erfolgreich ausgeführt wurde, können Sie Folgendes tun:

- Richten Sie DB-Event-Abonnements ein, um Sie per E-Mail oder darüber zu informierenSMS, dass ein Failover eingeleitet wurde. Weitere Informationen über -Ereignisse finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.
- Zeigen Sie Ihre DB-Ereignisse mithilfe der RDS Konsole oder mithilfe von API Operations an.
- Zeigen Sie den aktuellen Status Ihrer Multi-AZ-DB-Instance-Bereitstellung auf RDS Custom for SQL Server anCLI, indem Sie die RDS Konsole oder die API Operationen verwenden.

Time-to-Live-Einstellungen (TTL) bei Anwendungen, die eine Multi-AZ-Bereitstellung „RDSCustom for SQL Server“ verwenden

Der Failover-Mechanismus ändert den Domain Name System (DNS) -Datensatz der DB-Instance automatisch so, dass er auf die Standby-DB-Instance verweist. Als Ergebnis müssen alle bestehenden Verbindungen zu Ihrer DB-Instance neu hergestellt werden. Stellen Sie sicher, dass der Wert für die Konfiguration von DNS cache time-to-live (TTL) niedrig ist, und stellen Sie sicher, dass Ihre Anwendung nicht über einen DNS längeren Zeitraum zwischengespeichert wird. Ein hoher TTL Wert kann verhindern, dass Ihre Anwendung nach einem Failover schnell wieder eine Verbindung zur DB-Instance herstellt.

Sicherung und Wiederherstellung einer Amazon RDS Custom for SQL Server-DB-Instance

Wie Amazon RDS erstellt und speichert RDS Custom automatische Backups Ihrer RDS Custom for SQL Server-DB-Instance, wenn die Aufbewahrung von Backups aktiviert ist. Sie können Ihre DB-Instance auch sichern, indem Sie manuell einen DB-Snapshot erstellen. Die automatisierten Backups bestehen aus Snapshot-Backups und Transaktionsprotokoll-Backups. Snapshot-Backups werden für das gesamte Speichervolumen der DB-Instance während des von Ihnen angegebenen Backup-Fensters erstellt. In regelmäßigen Abständen werden Transaktionsprotokoll-Backups für die Datenbanken erstellt, für die diese Option in PITR Frage kommt. RDSBenutzerdefiniert speichert die automatisierten Backups Ihrer DB-Instance gemäß dem von Ihnen angegebenen Aufbewahrungszeitraum für Backups. Sie können automatisierte Backups verwenden, um Ihre DB-Instance bis zu einem bestimmten Zeitpunkt innerhalb der Aufbewahrungsfrist für Backups wiederherzustellen.

Sie können Snapshot-Backups auch manuell erstellen. Sie können aus diesen Snapshot-Backups jederzeit eine neue DB-Instance erstellen. Weitere Informationen zum manuellen Erstellen eines DB-Snapshots finden Sie unter Einen RDS benutzerdefinierten Snapshot für SQL Server erstellen.

Snapshot-Backups dienen zwar betrieblich als vollständige Backups, Ihnen wird jedoch nur die inkrementelle Speichernutzung in Rechnung gestellt. Der erste Snapshot einer RDS benutzerdefinierten DB-Instance enthält die Daten für die vollständige DB-Instance. Bei den nachfolgenden Snapshots derselben Datenbank handelt es sich um inkrementelle Snapshots, d. h. es werden nur die Daten gespeichert, die sich seit der letzten Snapshot-Speicherung geändert haben.

Themen

- Einen RDS benutzerdefinierten Snapshot für SQL Server erstellen
- Wiederherstellung aus einem RDS Custom for SQL Server DB-Snapshot
- Wiederherstellung einer RDS Custom for SQL Server-Instanz auf einen bestimmten Zeitpunkt
- Löschen eines Snapshots vom Typ „RDSBenutzerdefiniert für SQL Server“
- Automatisierte Backups vom Typ „RDSBenutzerdefiniert für SQL Server“ löschen

Einen RDS benutzerdefinierten Snapshot für SQL Server erstellen

RDSCustom for SQL Server erstellt einen Speicher-Volume-Snapshot Ihrer DB-Instance und sichert dabei die gesamte DB-Instance und nicht nur einzelne Datenbanken. Wenn Sie einen Snapshot

erstellen, geben Sie an, welche RDS Custom for SQL Server-DB-Instance gesichert werden soll. Geben Sie dann dem DB-Snapshot einen Namen, sodass über diesen eine Wiederherstellung zu einem späteren Zeitpunkt möglich ist.

Wenn Sie einen Snapshot erstellen, erstellt RDS Custom for SQL Server einen EBS Amazon-Snapshot für das Volume. Dabei handelt es sich um das Datenbank-Volume(`D:`), das der DB-Instance zugeordnet ist. Damit Snapshots einfach mit einer bestimmten DB-Instance verknüpft werden können, werden sie mit `DBSnapshotIdentifier`, `DbiResourceId` und `VolumeType` getaggt.

Das Erstellen eines DB-Snapshots führt zu einer kurzen I/O-Suspendierung. Diese Suspendierung kann je nach Größe und Klasse Ihrer DB-Instance einige Sekunden bis einige Minuten dauern. Die Dauer der Snapshot-Erstellung hängt von der Gesamtzahl und Größe Ihrer Datenbanken ab. Weitere Informationen zur Anzahl der Datenbanken, die für einen Point-in-Time-Restore-Vorgang (PITR) in Frage kommen, finden Sie unterAnzahl der Datenbanken, für die PITR pro Instance-Klassentyp infrage kommen.

Da der Snapshot das gesamte Speichervolume umfasst, wirkt sich die Größe von Dateien, wie z. B. temporäre Dateien, auch auf die Zeit aus, die zum Erstellen des Snapshots benötigt wird. Weitere Informationen zum Erstellen von Snapshots finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Erstellen Sie mit der Konsole oder dem einen RDS benutzerdefinierten SQL Server-Snapshot AWS CLI.

Konsole

Um einen RDS benutzerdefinierten Snapshot zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie in der Liste der RDS benutzerdefinierten DB-Instances die Instance aus, für die Sie einen Snapshot erstellen möchten.
4. Wählen Sie für Aktionen die Option Take snapshot (Snapshot aufnehmen).

   Das Fenster Take DB Snapshot (DB-Snapshot erstellen) wird angezeigt.
5. Geben Sie den Namen des Snapshots in das Feld Snapshot name (Snapshot-Name) ein.
6. Wählen Sie Take Snapshot (Snapshot erstellen) aus.

AWS CLI

Sie erstellen einen Snapshot einer RDS benutzerdefinierten DB-Instance mithilfe des create-db-snapshot AWS CLI Befehls.

Folgende Optionen stehen Ihnen zur Verfügung:

- `--db-instance-identifier`— Identifiziert, welche RDS benutzerdefinierte DB-Instance Sie sichern werden
- `--db-snapshot-identifier`— Benennt Ihren RDS benutzerdefinierten Snapshot, damit Sie ihn später wiederherstellen können

In diesem Beispiel erstellen Sie einen DB-Snapshot mit dem Namen *`my-custom-snapshot`* für eine RDS benutzerdefinierte DB-Instance namens*`my-custom-instance`*.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Wiederherstellung aus einem RDS Custom for SQL Server DB-Snapshot

Wenn Sie eine RDS Custom for SQL Server-DB-Instance wiederherstellen, geben Sie den Namen des DB-Snapshots und einen Namen für die neue Instance an. Sie können keine Wiederherstellung von einem Snapshot auf eine bestehende RDS benutzerdefinierte DB-Instance durchführen. Bei der Wiederherstellung wird eine neue RDS Custom for SQL Server-DB-Instance erstellt.

Bei der Wiederherstellung aus einem Snapshot wird das Speichervolume auf den Zeitpunkt zurückgesetzt, zu dem der Snapshot erstellt wurde. Dies schließt alle Datenbanken und alle anderen Dateien ein, die auf dem `(D:)` Volume vorhanden waren.

Konsole

Um eine RDS benutzerdefinierte DB-Instance aus einem DB-Snapshot wiederherzustellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot für die Wiederherstellung aus.
4. Wählen Sie in Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen) aus.
5. Geben Sie auf der Seite DB-Instance wiederherstellen als DB-Instance-ID den Namen für Ihre wiederhergestellte RDS benutzerdefinierte DB-Instance ein.
6. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

AWS CLI

Sie stellen einen RDS benutzerdefinierten DB-Snapshot mithilfe des Befehls restore-db-instance-from AWS CLI -db-snapshot wieder her.

Wenn der Snapshot, von dem Sie wiederherstellen, für eine private DB-Instance bestimmt ist, geben Sie beide die richtigen `db-subnet-group-name` und `no-publicly-accessible` an. Andernfalls ist die DB-Instance standardmäßig öffentlich zugänglich. Die folgenden Optionen sind erforderlich:

- `db-snapshot-identifier` — Identifiziert den Snapshot, aus dem wiederhergestellt werden soll
- `db-instance-identifier`— Gibt den Namen der RDS benutzerdefinierten DB-Instance an, die aus dem DB-Snapshot erstellt werden soll
- `custom-iam-instance-profile`— Gibt das Instance-Profil an, das der zugrunde liegenden EC2 Amazon-Instance einer RDS Custom DB-Instance zugeordnet ist.

Der folgende Code stellt den Snapshot mit dem Namen `my-custom-snapshot` zu `my-custom-instance` her.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
```

```
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Wiederherstellung einer RDS Custom for SQL Server-Instanz auf einen bestimmten Zeitpunkt

Sie können eine DB-Instance zu einem bestimmten Zeitpunkt (PITR) wiederherstellen und so eine neue DB-Instance erstellen. Zur Unterstützung PITR muss für Ihre DB-Instances die Aufbewahrung von Backups aktiviert sein.

Die letzte wiederherstellbare Zeit für eine RDS Custom for SQL Server-DB-Instance hängt von mehreren Faktoren ab, liegt aber in der Regel innerhalb von 5 Minuten nach der aktuellen Uhrzeit. Um die letzte wiederherstellbare Zeit für eine DB-Instance zu ermitteln, verwenden Sie den AWS CLI describe-db-instancesBefehl und sehen Sie sich den Wert an, der im `LatestRestorableTime` Feld für die DB-Instance zurückgegeben wird. Um den letzten wiederherstellbaren Zeitpunkt für jede DB-Instance in der RDS Amazon-Konsole zu sehen, wählen Sie Automatisierte Backups.

Sie können die Backup auf jeden beliebigen Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups vornehmen. Um den frühesten wiederherstellbaren Zeitpunkt für jede DB-Instance zu sehen, wählen Sie in der RDS Amazon-Konsole Automatisierte Backups aus.

Allgemeine Informationen zu PITR finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Themen

- PITRÜberlegungen zu RDS Custom for Server SQL
- Anzahl der Datenbanken, für die PITR pro Instance-Klassentyp infrage kommen
- Datenbanken ungeeignet machen für PITR
- Transaktionsprotokolle in Amazon S3

- PITRWiederherstellung mit dem AWS Management ConsoleAWS CLI, dem oder dem. RDS API

PITRÜberlegungen zu RDS Custom for Server SQL

In RDS Custom for SQL Server PITR unterscheidet sich in folgenden wichtigen Punkten von PITR AmazonRDS:

- PITRstellt nur die Datenbanken in der DB-Instance wieder her. Es stellt das Betriebssystem oder die Dateien auf dem Laufwerk C: nicht wieder her.
- Bei einer RDS Custom for SQL Server-DB-Instance wird eine Datenbank automatisch gesichert und kommt PITR nur unter den folgenden Bedingungen in Frage:
  - Die Datenbank ist online.
  - Sein Wiederherstellungsmodell ist auf FULL gesetzt.
  - Es ist beschreibbar.
  - Es hat seine physischen Dateien auf dem Laufwerk D:.
  - Sie ist nicht in der `rds_pitr_blocked_databases`-Tabelle aufgeführt. Weitere Informationen finden Sie unter Datenbanken ungeeignet machen für PITR.
- Welche Datenbanken in Frage kommen, hängt von der Reihenfolge ihrer Datenbank-ID ab. PITR RDSCustom for SQL Server ermöglicht bis zu 5.000 Datenbanken pro DB-Instance. Die maximale Anzahl von Datenbanken, die durch einen PITR Vorgang für eine RDS Custom for SQL Server-DB-Instance wiederhergestellt werden, hängt jedoch vom Instance-Klassentyp ab. Weitere Informationen finden Sie unter Anzahl der Datenbanken, für die PITR pro Instance-Klassentyp infrage kommen.

  Andere Datenbanken, die nicht Teil von sind, PITR können aus DB-Snapshots wiederhergestellt werden, einschließlich der automatisierten Snapshot-Backups, für PITR die sie verwendet werden.
- Durch das Hinzufügen einer neuen Datenbank, das Umbenennen einer Datenbank oder das Wiederherstellen einer Datenbank, die dafür in Frage kommt, wird ein Snapshot der DB-Instance PITR initiiert.
- Die maximale Anzahl von Datenbanken, die PITR geändert werden können, wenn die Datenbank-Instance einen Rechenvorgang mit Skalierung durchläuft, hängt vom Klassentyp der Zielinstanz ab. Wenn die Instanz hochskaliert wird, sodass mehr Datenbanken auf der Instance in Frage kommenPITR, wird ein neuer Snapshot erstellt.
- Wiederhergestellte Datenbanken haben denselben Namen wie in der Quell-DB-Instance. Wenn Sie möchten, können Sie einen anderen Namen eingeben.

- `AWSRDSCustomSQLServerIamRolePolicy`erfordert Zugriff auf andere AWS Dienste. Weitere Informationen finden Sie unter Fügen Sie eine Zugriffsrichtlinie hinzu zu AWSRDSCustom SQLServer InstanceRole.
- Zeitzonenänderungen werden für RDS Custom for SQL Server nicht unterstützt. Wenn Sie die Zeitzone des Betriebssystems oder der DB-Instance ändern, funktioniert das PITR (und andere Automatisierungen) nicht.

Anzahl der Datenbanken, für die PITR pro Instance-Klassentyp infrage kommen

Die folgende Tabelle zeigt die maximale Anzahl von Datenbanken, für die PITR je nach Instance-Klassentyp in Frage kommt.

Typ der Instance-Klasse	Maximale Anzahl PITR geeigneter Datenbanken
db.*.large	100
db.*.xlarge zu db.*.2xlarge	150
db.*.4xlarge zu db.*.8xlarge	300
db.*.12xlarge nach db.*.16xlarge	600
db.*.24xlarge, db.*32xlarge	1000

*Stellt verschiedene Instanzklassentypen dar.

Die maximale Anzahl von Datenbanken, für die PITR eine DB-Instance in Frage kommt, hängt vom Instance-Klassentyp ab. Die Anzahl reicht von 100 bei den kleinsten Instance-Klassentypen bis zu 1000 bei den größten Instance-Klassentypen, die von RDS Custom for SQL Server unterstützt

werden. SQLServersystemdatenbanken (`master, model, msdb, tempdb`) sind in diesem Limit nicht enthalten. Wenn eine DB-Instance je nach Typ der Zielinstanzklasse nach oben oder unten skaliert wird, aktualisiert RDS Custom automatisch die Anzahl der Datenbanken, für PITR die sie in Frage kommen. RDSCustom for SQL Server sendet, `RDS-EVENT-0352` wenn sich die maximale Anzahl der Datenbanken, die für eine DB-Instance in Frage kommen, PITR ändert. Weitere Informationen finden Sie unter Benutzerdefinierte Engine-Versionsereignisse.

> **Note**
>
> PITRUnterstützung für mehr als 100 Datenbanken ist nur für DB-Instances verfügbar, die nach dem 26. August 2023 erstellt wurden. Für Instances, die vor dem 26. August 2023 erstellt wurden, PITR liegt die maximale Anzahl von Datenbanken, für die infrage kommt, unabhängig von der Instance-Klasse bei 100. Um die PITR Unterstützung für mehr als 100 Datenbanken auf DB-Instances zu aktivieren, die vor dem 26. August 2023 erstellt wurden, können Sie die folgende Aktion ausführen:
>
> - Aktualisieren Sie die DB-Engine-Version auf 15.00.4322.2.v1 oder höher

Während eines PITR Vorgangs stellt RDS Custom alle Datenbanken wieder her, die zum Zeitpunkt der Wiederherstellung Teil einer Quell-DB-Instance PITR waren. Sobald die Ziel-DB-Instance die Wiederherstellungsvorgänge abgeschlossen hat und die Aufbewahrung von Backups aktiviert ist, beginnt die DB-Instance mit der Sicherung auf der Grundlage der maximalen Anzahl von Datenbanken, für PITR die die Ziel-DB-Instance in Frage kommt.

Wenn Ihre DB-Instance beispielsweise auf einer mit `db.*.xlarge` 200 Datenbanken läuft:

1. RDSCustom for SQL Server wählt die ersten 150 Datenbanken, sortiert nach ihrer Datenbank-ID, für die PITR Sicherung aus.
2. Sie ändern die Instanz so, dass sie auf db.*.4xlarge skaliert wird.
3. Sobald der Rechenvorgang für die Skalierung abgeschlossen ist, wählt RDS Custom for SQL Server die ersten 300 Datenbanken, sortiert nach ihrer Datenbank-ID, für die Sicherung aus. PITR Jede der 200 Datenbanken, die die PITR Anforderungsbedingungen erfüllen, kommt nun in FragePITR.
4. Sie ändern jetzt die Instanz, sodass sie wieder auf db.*.xlarge herunterskaliert wird.
5. Sobald der Rechenvorgang zur Skalierung abgeschlossen ist, wählt RDS Custom for SQL Server erneut die ersten 150 Datenbanken, sortiert nach ihrer Datenbank-ID, für die Sicherung aus. PITR

Datenbanken ungeeignet machen für PITR

Sie können wählen, ob einzelne Datenbanken von PITR ausgeschlossen werden sollen. Um dies zu tun, legen Sie ihre `database_id`-Werte in eine `rds_pitr_blocked_databases`-Tabelle. Verwenden Sie das folgende SQL Skript, um die Tabelle zu erstellen.

So erstellen Sie die Tabelle „rds_pitr_blocked_databases“

- Führen Sie das folgende SQL Skript aus.

```
create table msdb..rds_pitr_blocked_databases
(
database_id INT NOT NULL,
database_name SYSNAME NOT NULL,
db_entry_updated_date datetime NOT NULL DEFAULT GETDATE(),
db_entry_updated_by SYSNAME NOT NULL DEFAULT CURRENT_USER,
PRIMARY KEY (database_id)
);
```

Eine Liste der berechtigten und nicht berechtigten Datenbanken finden Sie in der `RI.End`-Datei im `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`*`/TransactionLogMetadata`-Verzeichnis im Amazon S3-Bucket `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. Weitere Informationen zur Datei `RI.End` finden Sie unter Transaktionsprotokolle in Amazon S3.

Sie können auch die Liste der Datenbanken ermitteln, die für die PITR Verwendung des folgenden SQL Skripts in Frage kommen. Setzen Sie die `@limit` Variable auf die maximale Anzahl von Datenbanken, die PITR für die Instance-Klasse in Frage kommen. Weitere Informationen finden Sie unter Anzahl der Datenbanken, für die PITR pro Instance-Klassentyp infrage kommen.

Um die Liste der geeigneten Datenbanken für PITR eine DB-Instance-Klasse zu ermitteln

- Führen Sie das folgende SQL Skript aus.

```
DECLARE @Limit INT;
SET @Limit = (insert-database-instance-limit-here);

USE msdb;
IF (EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND
 TABLE_NAME = 'rds_pitr_blocked_databases'))
```

```
   WITH TABLE0 AS (
       SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
       FROM sys.dm_hadr_database_replica_states hdrs
       INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
       WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
       OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
(sdb.name, 'Updateability') = 'READ_ONLY')
   ),
   TABLE1 as (
           SELECT dbs.database_id as DatabaseId, sysdbs.name as DatabaseName,
'OPTOUT' as Reason,
           CASE WHEN dbs.database_name = sysdbs.name THEN NULL ELSE
dbs.database_name END AS DatabaseNameOnPitrTable
           FROM msdb.dbo.rds_pitr_blocked_databases dbs
           INNER JOIN sys.databases sysdbs ON dbs.database_id = sysdbs.database_id
           WHERE sysdbs.database_id > 4
           ),
   TABLE2 as (
           SELECT
           db.name AS DatabaseName,
           db.create_date AS CreateDate,
           db.state_desc AS DatabaseState,
           db.database_id AS DatabaseId,
           rs.database_guid AS DatabaseGuid,
           rs.last_log_backup_lsn AS LastLogBackupLSN,
           rs.recovery_fork_guid RecoveryForkGuid,
           rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
           db.recovery_model_desc AS RecoveryModel,
           db.is_auto_close_on AS IsAutoClose,
           db.is_read_only as IsReadOnly,
           NEWID() as FileName,
           CASE WHEN(db.state_desc = 'ONLINE'
                   AND db.recovery_model_desc != 'SIMPLE'
                   AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
OR db.is_auto_close_on = 1))
                   AND db.is_read_only != 1
                   AND db.user_access = 0
                   AND db.source_database_id IS NULL
                   AND db.is_in_standby != 1
                   THEN 1 ELSE 0 END AS IsPartOfSnapshot,
           CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
           FROM sys.databases db
```

```
            INNER JOIN sys.database_recovery_status rs
            ON db.database_id = rs.database_id
            WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE1) AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
        ),
        TABLE3 as(
            Select @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE2
 where TABLE2.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        SELECT TOP(SELECT TotalNumberOfDatabases from TABLE3)
 DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE2 where
 TABLE2.IsPartOfSnapshot=1
        ORDER BY TABLE2.DatabaseID ASC
ELSE
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
 'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
        OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
 (sdb.name, 'Updateability') = 'READ_ONLY')
    ),
    TABLE1 as (
            SELECT
            db.name AS DatabaseName,
            db.create_date AS CreateDate,
            db.state_desc AS DatabaseState,
            db.database_id AS DatabaseId,
            rs.database_guid AS DatabaseGuid,
            rs.last_log_backup_lsn AS LastLogBackupLSN,
            rs.recovery_fork_guid RecoveryForkGuid,
            rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
            db.recovery_model_desc AS RecoveryModel,
            db.is_auto_close_on AS IsAutoClose,
            db.is_read_only as IsReadOnly,
            NEWID() as FileName,
            CASE WHEN(db.state_desc = 'ONLINE'
                    AND db.recovery_model_desc != 'SIMPLE'
                    AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
 OR db.is_auto_close_on = 1))
                    AND db.is_read_only != 1
                    AND db.user_access = 0
```

```
                    AND db.source_database_id IS NULL
                    AND db.is_in_standby != 1
                    THEN 1 ELSE 0 END AS IsPartOfSnapshot,
            CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
            FROM sys.databases db
            INNER JOIN sys.database_recovery_status rs
            ON db.database_id = rs.database_id
            WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
        ),
        TABLE2 as(
            SELECT @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE1
where TABLE1.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        select top(select TotalNumberOfDatabases from TABLE2)
DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE1 where
TABLE1.IsPartOfSnapshot=1
        ORDER BY TABLE1.DatabaseID ASC
```

> **Note**
>
> Die Datenbanken, bei denen es sich nur um symbolische Links handelt, sind ebenfalls von Datenbanken ausgeschlossen, die für PITR Operationen in Frage kommen. Die obige Abfrage filtert nicht nach diesen Kriterien.

Transaktionsprotokolle in Amazon S3

Die Aufbewahrungsfrist für Backups bestimmt, ob Transaktionsprotokolle für RDS Custom for SQL Server-DB-Instances automatisch extrahiert und auf Amazon S3 hochgeladen werden. Ein Wert ungleich Null bedeutet, dass automatische Backups erstellt werden und dass der RDS Custom Agent die Transaktionsprotokolle alle 5 Minuten auf S3 hochlädt.

Transaktionsprotokolldateien auf S3 werden im Ruhezustand mit dem AWS KMS key die Sie angegeben haben, als Sie Ihre DB-Instance erstellt haben. Weitere Informationen finden Sie unter Schutz von Daten durch serverseitige Verschlüsselung im Amazon Simple Storage Service User Guide.

Die Transaktionsprotokolle für jede Datenbank werden in einen S3-Bucket namens `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`* hochgeladen. Das `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`*-Verzeichnis im S3-Bucket enthält zwei Unterverzeichnisse:

- `TransactionLogs` — Enthält die Transaktionsprotokolle für jede Datenbank und ihre jeweiligen Metadaten.

  Der Name der Transaktionslog-Datei folgt dem Muster *`yyyyMMddHHmm`*`.`*`database_id`*`.`*`timestamp`*, Beispiel:

  ```
  202110202230.11.1634769287
  ```

  Derselbe Dateiname mit dem Suffix `_metadata` enthält Informationen über das Transaktionslog wie Log-Sequenznummern, Datenbankname und `RdsChunkCount`. `RdsChunkCount`bestimmt, wie viele physische Dateien eine einzelne Transaktionslogdatei darstellen. Möglicherweise sehen Sie Dateien mit Suffixen `_0001`, `_0002` und so weiter, was die physischen Teile einer Transaktionslogdatei bedeutet. Wenn Sie eine Chunked Transaktionslogdatei verwenden möchten, müssen Sie die Chunks nach dem Herunterladen zusammenführen.

  Betrachten Sie ein Szenario, in dem Sie folgende Dateien haben:

  - `202110202230.11.1634769287`
  - `202110202230.11.1634769287_0001`
  - `202110202230.11.1634769287_0002`
  - `202110202230.11.1634769287_metadata`

  Das `RdsChunkCount` ist `3`. Die Reihenfolge zum Zusammenführen der Dateien ist die folgende: `202110202230.11.1634769287`, `202110202230.11.1634769287_0001`, `202110202230.11.1634769287_0002`.

- `TransactionLogMetadata` — Enthält Metadateninformationen über jede Iteration der Transaktionslog-Extraktion.

  Die `RI.End` enthält Informationen für alle Datenbanken, deren Transaktionsprotokolle extrahiert wurden, und für alle Datenbanken, die vorhanden sind, aber ihre Transaktionsprotokolle nicht extrahiert wurden. Der `RI.End`-Dateiname folgt dem Muster *`yyyyMMddHHmm`*`.RI.End.`*`timestamp`*, Beispiel:

```
202110202230.RI.End.1634769281
```

PITRWiederherstellung mit dem AWS Management ConsoleAWS CLI, dem oder dem. RDS API

Sie können eine RDS benutzerdefinierte SQL Server-DB-Instance mit dem AWS Management Console, dem oder dem auf einen AWS CLI bestimmten Zeitpunkt zurücksetzen RDSAPI.

Konsole

So stellen Sie eine RDS benutzerdefinierte DB-Instance auf einen bestimmten Zeitpunkt wieder her

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
3. Wählen Sie die RDS benutzerdefinierte DB-Instance aus, die Sie wiederherstellen möchten.
4. Wählen Sie unter Aktionen die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

   Anschließend wird das Fenster Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) angezeigt.
5. Wählen Sie Späteste Wiederherstellungszeit, um auf den spätesten möglichen Zeitpunkt wiederherzustellen oder wählen Sie Benutzerdefiniert, um eine Zeit auszuwählen.

   Geben Sie bei der Auswahl von Custom das Datum und die Uhrzeit ein, zu der Sie den Instance-Cluster wiederherstellen möchten.

   Die Zeiten werden in Ihrer lokalen Zeitzone angezeigt, was durch einen Offset von der koordinierten Weltzeit (UTC) gekennzeichnet ist. Zum Beispiel ist UTC -5 Eastern Standard Time/Central Daylight Time.
6. Geben Sie als DB-Instance-ID den Namen der wiederhergestellten RDS benutzerdefinierten DB-Ziel-Instance ein. Der Name muss eindeutig sein.
7. Wählen Sie bei Bedarf andere Optionen aus, z. B. DB-Instance-Class.
8. Wählen Sie Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

AWS CLI

Sie stellen eine DB-Instance zu einem bestimmten Zeitpunkt wieder her, indem Sie den point-in-time AWS CLI Befehl restore-db-instance-to- verwenden, um eine neue RDS benutzerdefinierte DB-Instance zu erstellen.

Verwenden Sie eine der folgenden Optionen, um die Sicherung anzugeben, von der wiederhergestellt werden soll:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*
- `--source-db-instance-automated-backups-arn` *`backupARN`*

Die Option `custom-iam-instance-profile` ist erforderlich.

Der folgende Befehl stellt `my-custom-db-instance` auf eine neue DB-Instance namens `my-restored-custom-db-instance` wieder her, und zwar zum angegebenen Zeitpunkt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Löschen eines Snapshots vom Typ „RDSBenutzerdefiniert für SQL Server“

Sie können von RDS Custom for SQL Server verwaltete DB-Snapshots löschen, wenn Sie sie nicht mehr benötigen. Das Löschverfahren ist für Amazon- RDS und RDS Custom-DB-Instances identisch.

Die EBS Amazon-Snapshots für die Binär- und Root-Volumes verbleiben länger in Ihrem Konto, da sie möglicherweise mit einigen Instances verknüpft sind, die in Ihrem Konto ausgeführt werden, oder mit anderen RDS Custom for SQL Server-Snapshots. Diese EBS Snapshots werden automatisch gelöscht, wenn sie keinen Bezug mehr zu vorhandenen RDS Custom for SQL Server-Ressourcen (DB-Instances oder Backups) haben.

Konsole

Um einen Snapshot einer RDS benutzerdefinierten DB-Instance zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot aus, den Sie löschen möchten.
4. Wählen Sie unter Actions (Aktionen) die Option Delete Snapshot (Snapshot löschen) aus.
5. Wählen Sie auf der Bestätigungsseite die Option Delete (Löschen) aus.

AWS CLI

Verwenden Sie den AWS CLI Befehl, um einen RDS benutzerdefinierten Snapshot zu löschen delete-db-snapshot.

Die folgenden Optionen sind erforderlich:

- `--db-snapshot-identifier` — Der zu löschende Snapshot

Das folgende Beispiel löscht den Cluster-Snapshot `my-custom-snapshot`.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-snapshot ^
```

```
  --db-snapshot-identifier my-custom-snapshot
```

Automatisierte Backups vom Typ „RDSBenutzerdefiniert für SQL Server“ löschen

Sie können gespeicherte automatische Backups für RDS Custom for SQL Server löschen, wenn sie nicht mehr benötigt werden. Das Verfahren entspricht dem Verfahren zum Löschen von RDS Amazon-Backups.

Konsole

So löschen Sie eine aufbewahrte automatisierte Backup:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.
3. Wählen Sie Retained (Aufbewahrt).
4. Wählen Sie die aufbewahrte automatisierte Sicherung, die Sie löschen möchten.
5. Klicken Sie bei Actions auf Delete.
6. Geben Sie auf der Bestätigungsseite **`delete me`** und wählen Sie Löschen aus.

AWS CLI

Sie können ein gespeicherter automatisierter Backup mit dem AWS CLI Befehl delete-db-instance-automated-backup löschen.

Zum Löschen einer aufbewahrten automatisierten Sicherung werden die folgenden Optionen verwendet.

- `--dbi-resource-id`— Die Ressourcen-ID für die RDS benutzerdefinierte Quell-DB-Instance.

  Sie können die Ressourcen-ID für die Quell-DB-Instance eines gespeicherten automatisierten Backups mithilfe des AWS CLI Befehls describe-db-instance-automated-backups ermitteln.

Im folgenden Beispiel wird das aufbewahrte automatisierte Backup mit der Quell-DB-Instance-Ressourcenkennung `custom-db-123ABCEXAMPLE` gelöscht.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Kopieren eines Amazon RDS Custom for SQL Server-DB-Snapshots

Mit RDS Custom for SQL Server können Sie automatische Backups und manuelle DB-Snapshots kopieren. Nach dem Kopieren eines Snapshots handelt es sich bei der von Ihnen erstellten Kopie um einen manuellen Snapshot. Sie können mehrere Kopien eines automatisierten Backups oder eines manuellen Snapshots erstellen, aber jede Kopie muss eine eindeutige Kennung haben.

Sie können einen Snapshot nur innerhalb desselben AWS Kontos auf verschiedene Standorte kopieren, AWS-Regionen an denen RDS Custom for SQL Server verfügbar ist. Die folgenden Operationen werden derzeit nicht unterstützt:

- Kopieren von DB-Snapshots innerhalb desselben AWS-Region.
- Kopieren von DB-Snapshots zwischen Konten. AWS

RDS Custom for SQL Server unterstützt das inkrementelle Kopieren von Snapshots. Weitere Informationen finden Sie unter Überlegungen zum inkrementellen Kopieren von Snapshots.

Themen

- Einschränkungen
- Umgang mit Verschlüsselungen
- Regionsübergreifendes Kopieren
- Snapshots von DB-Instances, die mit Custom Engine Versions (CEV) erstellt wurden
- Erteilen Sie die erforderlichen Berechtigungen für Ihren IAM-Prinzipal
- Kopieren eines DB-Snapshots

Einschränkungen

Die folgenden Einschränkungen gelten für das Kopieren eines DB-Snapshots für RDS Custom for SQL Server:

- Wenn Sie einen Quell-Snapshot löschen, bevor der Ziel-Snapshot verfügbar ist, kann das Kopieren des Snapshots fehlschlagen. Stellen Sie sicher, dass der Ziel-Snapshot den Status hat, `AVAILABLE` bevor Sie den Quell-Snapshot löschen.
- Sie können in Ihrer Anfrage zum Kopieren von DB-Snapshots keinen Namen für eine Optionsgruppe angeben oder eine Optionsgruppe kopieren.

- Wenn Sie vor oder während des Kopiervorgangs abhängige AWS Ressourcen des Quell-DB-Snapshots löschen, kann Ihre Anfrage zum Kopieren des Snapshots asynchron fehlschlagen.
  - Wenn Sie die Service Master Key (SMK) -Backup-Datei für Ihre Quell-DB-Instance löschen, die im RDS Custom Managed S3-Bucket in Ihrem Konto gespeichert ist, wird die DB-Snapshot-Kopie asynchron erfolgreich ausgeführt. Bei Funktionen von SQL Server, die von SMK abhängig sind, wie z. B. TDE-fähige Datenbanken, treten jedoch Probleme auf. Weitere Informationen finden Sie unter Problembehandlung beim Status PENDING_RECOVERY für TDE-fähige Datenbanken in RDS Custom for SQL Server.
- Das Kopieren von DB-Snapshots innerhalb derselben AWS-Region wird derzeit nicht unterstützt.
- Das Kopieren von DB-Snapshots zwischen AWS Konten wird derzeit nicht unterstützt.

Die Einschränkungen beim Kopieren eines DB-Snapshots für Amazon RDS gelten auch für RDS Custom for SQL Server. Weitere Informationen finden Sie unter Einschränkungen.

Umgang mit Verschlüsselungen

Alle DB-Instances und DB-Snapshots von RDS Custom for SQL Server sind mit KMS-Schlüsseln verschlüsselt. Sie können nur einen verschlüsselten Snapshot in einen verschlüsselten Snapshot kopieren. Daher müssen Sie einen KMS-Schlüssel angeben, der im Ziel AWS-Region für Ihre Anfrage zum Kopieren von DB-Snapshots gültig ist.

Der Quell-Snapshot bleibt den gesamten Kopiervorgang über verschlüsselt. Amazon RDS verwendet Umschlagverschlüsselung, um Daten während des Kopiervorgangs mit dem angegebenen AWS-Region Ziel-KMS-Schlüssel zu schützen. Weitere Informationen zur Envelope-Verschlüsselung finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Regionsübergreifendes Kopieren

Sie können DB-Snapshots über AWS-Regionen hinweg kopieren. Es gibt jedoch bestimmte Einschränkungen und Überlegungen für das Kopieren von regionsübergreifenden Snapshots.

Autorisierung von RDS zur Kommunikation beim Kopieren AWS-Regionen von Snapshots

Nachdem eine regionsübergreifende Anfrage zum Kopieren von DB-Snapshots erfolgreich verarbeitet wurde, startet RDS den Kopiervorgang. Eine Autorisierungsanfrage für RDS für den Zugriff auf den Quell-Snapshot wird erstellt. Diese Autorisierungsanfrage verknüpft den Quell-DB-Snapshot mit dem Ziel-DB-Snapshot. Dadurch kann RDS nur auf den angegebenen Ziel-Snapshot kopieren.

RDS verifiziert die Autorisierung mithilfe der `rds:CrossRegionCommunication` Berechtigung in der serviceverknüpften IAM-Rolle. Wenn die Kopie autorisiert ist, kann RDS mit der Quellregion kommunizieren und den Kopiervorgang abschließen.

RDS hat keinen Zugriff auf DB-Snapshots, die nicht zuvor durch eine DBSnapshot Kopieranforderung autorisiert wurden. Die Autorisierung wird nach Abschluss des Kopiervorgangs widerrufen.

RDS verwendet die serviceverknüpfte Rolle, um die Autorisierung in der Quellregion zu überprüfen. Der Kopiervorgang schlägt fehl, wenn Sie die mit dem Dienst verknüpfte Rolle während des Kopiervorgangs löschen.

Weitere Informationen finden Sie im AWS Identity and Access Management Benutzerhandbuch unter Verwenden von serviceverknüpften Rollen.

Verwenden von AWS Security Token Service -Anmeldeinformationen

Sitzungstoken vom Endpunkt global AWS Security Token Service (AWS STS) sind nur in Bereichen gültig AWS-Regionen , die standardmäßig aktiviert sind (kommerzielle Regionen). Wenn Sie Anmeldeinformationen aus dem `assumeRole` API-Vorgang in verwenden AWS STS, verwenden Sie den regionalen Endpunkt, wenn es sich bei der Quellregion um eine Opt-in-Region handelt. Andernfalls schlägt die Anforderung fehl. Ihre Anmeldeinformationen müssen in beiden Regionen gültig sein. Dies gilt nur für Opt-in-Regionen, wenn Sie den regionalen AWS STS Endpunkt verwenden.

Um den globalen Endpunkt zu verwenden, stellen Sie sicher, dass er für beide Regionen in den Vorgängen aktiviert ist. Stellen Sie den globalen Endpunkt `Valid` in den AWS STS Kontoeinstellungen AWS-Regionen auf „Alle“ ein.

Weitere Informationen finden Sie unter Verwaltung AWS STSAWS-Region in einem im AWS Identity and Access Management Benutzerhandbuch.

Snapshots von DB-Instances, die mit Custom Engine Versions (CEV) erstellt wurden

Bei einem DB-Snapshot einer DB-Instance, die eine Custom Engine Version (CEV) verwendet, ordnet RDS den CEV dem DB-Snapshot zu. Um einen Quell-DB-Snapshot, der einer CEV zugeordnet ist, hinüber zu kopieren AWS-Regionen, kopiert RDS den CEV zusammen mit dem Quell-DB-Snapshot in die Zielregion.

Wenn Sie mehrere DB-Snapshots, die demselben CEV zugeordnet sind, in dieselbe Zielregion kopieren, kopiert die erste Kopieranforderung das zugehörige CEV. Der Kopiervorgang der folgenden

Anfragen findet die ursprünglich kopierte CEV und ordnet sie den folgenden DB-Snapshot-Kopien zu. Die vorhandene CEV-Kopie muss den `AVAILABLE` Status haben, in dem sie den DB-Snapshot-Kopien zugeordnet werden kann.

Um einen mit einem CEV verknüpften DB-Snapshot zu kopieren, muss die IAM-Richtlinie des Anforderers über die erforderlichen Berechtigungen verfügen, um sowohl das Kopieren von DB-Snapshots als auch das zugehörige CEV-Kopieren zu autorisieren. Die folgenden Berechtigungen sind in der IAM-Richtlinie Ihres Anforderers erforderlich, um das zugehörige CEV-Kopieren zu ermöglichen:

- `rds:CopyCustomDBEngineVersion`- Ihr IAM-Principal, der die Anfrage gestellt hat, muss die Berechtigung haben, das Quell-CEV zusammen mit dem Quell-DB-Snapshot in die Zielregion zu kopieren. Die Anforderung zum Kopieren des Snapshots schlägt aufgrund von Autorisierungsfehlern fehl, wenn Ihr IAM-Principal, der den Antrag gestellt hat, nicht autorisiert ist, den Quell-CEV zu kopieren.
- `ec2:CreateTags`- Das zugrunde liegende EC2 AMI der Quell-CEV wird als Teil der CEV-Kopie in die Zielregion kopiert. RDS Custom versucht, das AMI mit dem `AWSRDSCustom` Tag zu kennzeichnen, bevor das AMI kopiert wird. Stellen Sie sicher, dass Ihr IAM-Principal, der die Anfrage gestellt hat, berechtigt ist, das Tag für das AMI zu erstellen, das dem Quell-CEV in der Quellregion zugrunde liegt.

Weitere Informationen zu CEV-Kopierberechtigungen finden Sie unter. Erteilen Sie die erforderlichen Berechtigungen für Ihren IAM-Prinzipal

Erteilen Sie die erforderlichen Berechtigungen für Ihren IAM-Prinzipal

Stellen Sie sicher, dass Sie über ausreichende Zugriffsrechte verfügen, um einen DB-Snapshot von RDS Custom for SQL Server zu kopieren. Die IAM-Rolle oder der IAM-Benutzer (als IAM-Prinzipal bezeichnet) zum Kopieren eines DB-Snapshots mithilfe der Konsole oder CLI muss über eine der folgenden Richtlinien verfügen, damit eine DB-Instance erfolgreich erstellt werden kann:

- Die `AdministratorAccess` Richtlinie oder
- Die folgende `AmazonRDSFullAccess`-Richtlinie zeigt die zusätzlichen Berechtigungen.

```
s3:CreateBucket
s3:GetBucketPolicy
s3:PutBucketPolicy
kms:CreateGrant
```

```
kms:DescribeKey
ec2:CreateTags
```

RDS Custom verwendet diese Berechtigungen beim Kopieren von Snapshots AWS-Regionen. Mit diesen Berechtigungen werden Ressourcen in Ihrem Konto konfiguriert, die für benutzerdefinierte RDS-Operationen erforderlich sind. Weitere Informationen zu `kms:CreateGrant`-Berechtigung finden Sie unter AWS KMS key Verwaltung.

Die folgende JSON-Beispielrichtlinie gewährt zusätzlich zur `AmazonRDSFullAccess` Richtlinie die erforderlichen Berechtigungen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateS3BucketAndReadWriteBucketPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateEc2Tags",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Stellen Sie sicher, dass die aufgelisteten Berechtigungen nicht durch Dienststeuerungsrichtlinien (SCPs), Berechtigungsgrenzen oder Sitzungsrichtlinien eingeschränkt werden, die dem IAM-Prinzipal zugeordnet sind.

Wenn Sie Bedingungen mit Kontextschlüsseln in der IAM-Richtlinie des Anforderers verwenden, können bestimmte Bedingungen dazu führen, dass die Anfrage fehlschlägt. Weitere Informationen zu häufigen Fallstricken aufgrund von IAM-Richtlinienbedingungen finden Sie unter. Anfordern einer Regionsübergreifende Kopie für DB-Snapshots

Kopieren eines DB-Snapshots

Verwenden Sie die folgenden Verfahren, um einen DB-Snapshot zu kopieren. Für jedes AWS Konto können Sie bis zu 20 DB-Snapshots gleichzeitig von einem AWS-Region zum anderen kopieren. Wenn Sie einen DB-Snapshot in einen anderen kopieren AWS-Region, erstellen Sie einen manuellen DB-Snapshot, der in diesem AWS-Region gespeichert wird. Beim Kopieren eines DB-Snapshots aus der Quelle AWS-Region fallen Amazon RDS-Datenübertragungsgebühren an. Weitere Informationen zu den Kosten von Datenübertragungen finden Sie unter Amazon RDS – Preise.

Nachdem die DB-Snapshot-Kopie in der neuen Version erstellt wurde AWS-Region, verhält sich die DB-Snapshot-Kopie genauso wie alle anderen DB-Snapshots in dieser Version. AWS-Region

Sie können einen DB-Snapshot mit der AWS Management Console AWS CLI, oder der Amazon RDS-API kopieren.

Console

Mit dem folgenden Verfahren wird ein DB-Snapshot von RDS Custom for SQL Server mithilfe von kopiert AWS Management Console.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den DB-Snapshot RDS Custom for SQL Server aus, den Sie kopieren möchten.
4. Wählen Sie in der Dropdownliste Aktionen die Option Snapshot kopieren aus.

RDS > Snapshots > Copy snapshot

Copy snapshot

Settings

Source DB snapshot
DB snapshot identifier for the snapshot being copied.
db1-snapshot

New DB snapshot identifier
DB snapshot identifier for the new snapshot.
copied-db1-snapshot
Must start with a letter and only contain letters, digits, or hyphens.

Destination Region Info
Choose a Region

Resource tags Info
☐ Copy tags from the source DB snapshot

Depending on the amount of data to be copied and the Region you choose, this operation can take several hours to complete. The display on the progress bar might be delayed until setup is complete.

Encryption

Encryption Info
☑ Enable Encryption
Choose to encrypt the copy of the source DB snapshot. Master key IDs and aliases appear in the list after they have been created using KMS. You cannot remove encryption from an encrypted DB snapshot.

AWS KMS key Info
Enter a key ARN

Amazon Resource Name (ARN)
[redacted]
Example: arn:aws:kms:<region>:<accountID>:key/<key-id>

Account
[redacted]

KMS key ID
[redacted]

5. Um den DB-Snapshot in einen anderen zu kopieren AWS-Region, setzen Sie Destination Region auf den erforderlichen Wert.

> **Note**
>
> Für das Ziel AWS-Region muss dieselbe Version der Datenbank-Engine wie für die Quelle verfügbar sein AWS-Region.

6. Geben Sie unter Neuer DB-Snapshot-ID einen eindeutigen Namen für den DB-Snapshot ein. Sie können mehrere Kopien eines automatisierten Backups oder eines manuellen Snapshots erstellen, aber jede Kopie muss eine eindeutige Kennung haben.
7. (Optional) Wählen Sie Copy Tags (Tags kopieren), um Tags und Werte aus dem Snapshot in die Kopie des Snapshots zu übernehmen.
8. Geben Sie für Verschlüsselung die KMS-Schlüssel-ID an, die zum Verschlüsseln der DB-Snapshot-Kopie verwendet werden soll.

> **Note**
>
> RDS Custom for SQL Server verschlüsselt alle DB-Snapshots. Sie können keinen unverschlüsselten DB-Snapshot erstellen.

9. Wählen Sie Copy Snapshot (Snapshot kopieren) aus.

RDS Custom for SQL Server erstellt eine DB-Snapshot-Kopie Ihrer DB-Instance in der AWS-Region von Ihnen ausgewählten.

AWS CLI

Sie können einen DB-Snapshot von RDS Custom for SQL Server kopieren, indem Sie den AWS CLI Befehl verwenden copy-db-snapshot. Wenn Sie den Snapshot in einen neuen kopieren AWS-Region, führen Sie den Befehl im neuen aus AWS-Region. Zum Kopieren eines DB-Snapshots werden die folgenden Optionen verwendet. Nicht alle Optionen werden in allen Szenarien benötigt.

- `--source-db-snapshot-identifier`- Der Bezeichner für den Quell-DB-Snapshot.
  - Wenn sich der Quell-Snapshot in einer anderen Version AWS-Region als der Kopie befindet, geben Sie einen gültigen DB-Snapshot-ARN an. Beispiel, `arn:aws:rds:us-west-2:123456789012:snapshot:instance1-snapshot-12345678`.

- `--target-db-snapshot-identifier`- Die Kennung für die neue Kopie des DB-Snapshots.
- `--kms-key-id`-Die KMS-Schlüssel-ID für einen verschlüsselten DB-Snapshot. Der KMS-Schlüsselbezeichner ist der Amazon Resource Name (ARN), der Schlüsselbezeichner oder der Schlüsselalias für den KMS-Schlüssel.
  - Wenn Sie einen verschlüsselten Snapshot in eine andere AWS-Region kopieren, müssen Sie einen KMS-Schlüssel für die Ziel- AWS-Region angeben. KMS-Schlüssel sind spezifisch für AWS-Region das, in dem sie erstellt wurden, und Sie können keine Verschlüsselungsschlüssel von einem AWS-Region zum anderen verwenden, AWS-Region es sei denn, es wird ein Schlüssel für mehrere Regionen verwendet. Weitere Informationen über multiregionale Schlüssel finden Sie unter Verwenden von multiregionalen Schlüsseln in AWS KMS.
- `--copy-tags`- Fügen Sie die Tags und Werte aus dem Quell-Snapshot in die Kopie des Snapshots ein.

Die folgenden Optionen werden für das Kopieren eines DB-Snapshots vom Typ RDS Custom for SQL Server nicht unterstützt:

- `--copy-option-group`
- `--option-group-name`
- `--pre-signed-url`
- `--target-custom-availability-zone`

Im folgenden Codebeispiel wird ein verschlüsselter DB-Snapshot aus der Region USA West (Oregon) in die Region USA Ost (Nord-Virginia) kopiert. Führen Sie den Befehl in der Zielregion (us-east-1) aus.

Für Linux, macOS oder Unix:

```
aws rds copy-db-snapshot \
     --region us-east-1 \
    --source-db-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:snapshot:instance1-snapshot-12345678 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --kms-key-id a1b2c3d4-1234-5678-wxyz-a1b2c3d4d5e6
```

Für Windows:

```
aws rds copy-db-snapshot ^
     --region us-east-1 ^
    --source-db-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:snapshot:instance1-snapshot-12345678 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --kms-key-id a1b2c3d4-1234-5678-wxyz-a1b2c3d4d5e6
```

RDS API

Sie können einen DB-Snapshot von RDS Custom for SQL Server mithilfe der Amazon RDS-API-Operation Copy kopieren DBSnapshot. Wenn Sie den Snapshot in eine neue AWS-Region kopieren, führen Sie die Aktion in der neuen AWS-Region aus. Zum Kopieren eines DB-Snapshots werden die folgenden Parameter verwendet. Nicht alle Parameter sind erforderlich:

- `SourceDBSnapshotIdentifier`- Der Bezeichner für den Quell-DB-Snapshot.
  - Wenn sich der Quell-Snapshot in einer anderen Version AWS-Region als der Kopie befindet, geben Sie einen gültigen DB-Snapshot-ARN an. Beispiel, `arn:aws:rds:us-west-2:123456789012:snapshot:instance1-snapshot-12345678`.
- `TargetDBSnapshotIdentifier`- Die Kennung für die neue Kopie des DB-Snapshots.
- `KmsKeyId`- Die KMS-Schlüssel-ID für einen verschlüsselten DB-Snapshot. Der KMS-Schlüsselbezeichner ist der Amazon Resource Name (ARN), der Schlüsselbezeichner oder der Schlüsselalias für den KMS-Schlüssel.
  - Wenn Sie einen verschlüsselten Snapshot in eine andere AWS-Region kopieren, müssen Sie einen KMS-Schlüssel für die Ziel- AWS-Region angeben. KMS-Schlüssel sind spezifisch für AWS-Region das, in dem sie erstellt wurden, und Sie können nur AWS-Region dann Verschlüsselungsschlüssel verwenden, AWS-Region wenn ein Schlüssel für mehrere Regionen verwendet wird. Weitere Informationen über multiregionale Schlüssel finden Sie unter Verwenden von multiregionalen Schlüsseln in AWS KMS.
- `CopyTags`- Stellen Sie diesen Parameter auf ein`true`, um Tags und Werte aus dem Quell-Snapshot in die Kopie des Snapshots zu kopieren. Der Standardwert ist `false`.

Die folgenden Optionen werden beim Kopieren eines DB-Snapshots vom Typ RDS Custom for SQL Server nicht unterstützt:

- `CopyOptionGroup`
- `OptionGroupName`

- `PreSignedUrl`
- `TargetCustomAvailabilityZone`

Der folgende Code erstellt eine Kopie eines Snapshots mit dem neuen Namen `mydbsnapshotcopy` in der US East (N. Virginia)-Region.

```
https://rds.us-east-1.amazonaws.com/
    ?Action=CopyDBSnapshot
    &KmsKeyId=a1b2c3d4-1234-5678-wxyz-a1b2c3d4d5e6
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-
west-2%3A123456789012%3Asnapshot%3Ainstance1-snapshot-12345678
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-
date
    &X-Amz-
Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Migration einer lokalen Datenbank zu Amazon RDS Custom for SQL Server

Sie können den folgenden Prozess verwenden, um eine lokale Microsoft SQL Server-Datenbank mithilfe einer nativen Sicherung und Backup auf Amazon RDS Custom for SQL Server zu migrieren:

1. Machen Sie eine vollständige Sicherung der Datenbank auf der lokalen DB-Instance.
2. Laden Sie die Sicherungsdatei in Amazon S3 hoch.
3. Laden Sie die Sicherungsdatei von S3 auf Ihre RDS Custom für SQL-DB-Instance herunter.
4. Stellen Sie eine Datenbank mit der heruntergeladenen Sicherungsdatei auf der RDS Custom for SQL Server DB-Instance wieder her.

Dieser Prozess erklärt die Migration einer Datenbank von lokal zu RDS Custom for SQL Server unter Verwendung einer nativen vollständigen Sicherung und Backup. Um die Schnittzeit während des Migrationsprozesses zu verkürzen, können Sie auch Differentialsicherungen oder Protokollsicherungen verwenden.

Allgemeine Informationen zur nativen Sicherung und Wiederherstellung für RDS for SQL Server finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Themen

- Voraussetzungen
- Sichern der lokalen Datenbank
- Hochladen der Sicherungsdatei auf Amazon S3
- Herunterladen der Sicherungsdatei aus Amazon S3
- Wiederherstellen der Sicherungsdatei in der RDS Custom für SQL-DB-Instance

Voraussetzungen

Führen Sie vor der Migration der Datenbank folgende Aufgaben aus:

1. Konfigurieren Sie Remotedesktopverbindung (RDP) für Ihre RDS Custom für SQL-DB-Instance. Weitere Informationen finden Sie unter Verbindung zu Ihrer RDS Custom DB-Instance herstellen mit RDP.
2. Konfigurieren Sie den Zugriff auf Amazon S3, damit Sie die Datenbanksicherungsdatei hochladen und herunterladen können.

Sichern der lokalen Datenbank

Sie verwenden das native SQL Server-Backup, um eine vollständige Sicherung der Datenbank auf der lokalen DB-Instance durchzuführen.

Das folgende Beispiel zeigt eine Sicherung einer Datenbank namens`mydatabase` mit der `COMPRESSION`-Option angegeben, um die Größe der Sicherungsdatei zu reduzieren.

So sichern Sie die lokale Datenbank

1. Stellen Sie mithilfe von SQL Server Management Studio (SSMS) eine Verbindung mit der lokalen SQL Server-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein zu markieren.

```
backup database mydatabase to
disk ='C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup\mydb-full-compressed.bak'
with compression;
```

Hochladen der Sicherungsdatei auf Amazon S3

Sie verwenden den AWS Management Console , um die Sicherungsdatei `mydb-full-compressed.bak` auf Amazon S3 hochzuladen.

Laden Sie die Sicherungsdatei nach S3 hoch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in den Ihre Dateien hochgeladen werden sollen.
3. Klicken Sie auf Upload.
4. Führen Sie im Fenster Upload einen der folgenden Schritte aus:
   - Drag-and-drop `mydb-full-compressed.bak` zum Hochladen-Fenster.
   - Klicken Sie auf Datei hinzufügen, wählen Sie `mydb-full-compressed.bak` und klicken Sie auf Öffnen.

Amazon S3 lädt Ihre Sicherungsdatei als S3-Objekt hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: status eine Erfolgsmeldung angezeigt.

Herunterladen der Sicherungsdatei aus Amazon S3

Sie verwenden die Konsole, um die Sicherungsdatei von S3 auf die RDS Custom for SQL Server DB-Instanz herunterzuladen.

Laden Sie die Sicherungsdatei aus S3 herunter

1. Stellen Sie mithilfe von RDP eine Verbindung mit Ihrer RDS Custom für SQL-DB-Instance her.
2. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in den Ihre Dateien hochgeladen werden sollen.
4. Wählen Sie die Sicherungsdatei `mydb-full-compressed.bak` aus.
5. Wählen Sie für Aktionen die Option Herunterladen aus.
6. Öffnen Sie das Kontextmenü (rechte Maustaste) für den bereitgestellten Link und wählen Sie Save Link As (Link speichern unter) aus.
7. Speichern Sie die `mydb-full-compressed.bak`-Datei im Verzeichnis `D:\rdsdbdata\BACKUP`.

Wiederherstellen der Sicherungsdatei in der RDS Custom für SQL-DB-Instance

Verwenden Sie die native SQL Server-Backup, um die Sicherungsdatei in Ihrer RDS Custom für SQL-DB-Instance wiederherzustellen.

In diesem Beispiel wird die Aktion `MOVE` angegeben, da sich die Daten- und Protokolldateiverzeichnisse von der lokalen DB-Instance unterscheiden.

So stellen Sie die Sicherungsdatei wieder her

1. Stellen Sie mithilfe von SSMS eine Verbindung mit Ihrer RDS Custom für SQL-DB-Instance her.
2. Führen Sie den folgenden Befehl aus, um ein zu markieren.

```
restore database mydatabase from disk='D:\rdsdbdata\BACKUP\mydb-full-
compressed.bak'
with move 'mydatabase' to 'D:\rdsdbdata\DATA\mydatabase.mdf',
move 'mydatabase_log' to 'D:\rdsdbdata\DATA\mydatabase_log.ldf';
```

Upgrade einer Amazon RDS Custom for SQL Server-DB-Instance

Sie können eine Amazon RDS Custom for SQL Server-DB-Instance aktualisieren, indem Sie sie so ändern, dass sie eine neue DB-Engine-Version verwendet. Allgemeine Informationen zum Upgrade von DB-Instances finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Themen

- Überblick über Upgrades in RDS Custom für SQL Server
- Aktualisierung der Haupt- und Nebenversion der Engine
- Datenbank-Kompatibilitätsstufe

Überblick über Upgrades in RDS Custom für SQL Server

Amazon RDS Custom for SQL Server unterstützt Haupt- und Nebenversions-Upgrades. Upgrades kleinerer Versionen können Sicherheitspatches, Bugfixes und Engine-Verbesserungen beinhalten. Microsoft veröffentlicht diese Updates als kumulative Updates (CUs). Größere Versionsupgrades führen neue Funktionen und Engine-Änderungen zwischen den Versionen ein, z. B. ein Upgrade von SQL Server 2019 auf 2022. Sie können beide Upgrades sofort oder während der geplanten Wartungsfenster anwenden. Um mögliche Probleme mit der Abwärtskompatibilität zu vermeiden, empfehlen wir, Ihre Anwendungen vor dem Upgrade in einer Umgebung außerhalb der Produktionsumgebung zu testen.

Mit RDS Custom for SQL Server können Sie eine RDS Provided Engine Version (RPEV) oder eine Custom Engine Version (CEV) aktualisieren.

- Von RDS bereitgestellte Engine-Versionen (RPEV) enthalten up-to-date Betriebssystem-Patches (OS) und kumulative Updates (CU) für SQL Server.
- Für eine benutzerdefinierte Engine-Version (CEV) müssen Sie einen zweistufigen Prozess befolgen. Erstellen Sie zunächst ein neues CEV mit Ihrer SQL Server-Zielversion, siehe. Vorbereitung der Erstellung einer CEV für RDS Custom für SQL Server Diese Zielversion muss Ihrer aktuellen Version entsprechen oder neuer sein. Sobald das neue CEV erstellt wurde, ändern Sie Ihre Datenbank-Instance so, dass sie diese neue Version verwendet. Weitere Informationen finden Sie unter Durchführen eines kleineren Versions-Upgrades für Amazon RDS Custom for SQL Server CEV mit Multi-AZ.

  Wenden Sie keine kumulativen Updates für SQL Server direkt auf Ihre laufende benutzerdefinierte RDS-Instance an. Sobald Sie ein CEV mit einer bestimmten SQL Server-Version (z. B. SQL Server

2022 CU16) erstellt haben, wird die Instanz durch direktes Anwenden eines neueren kumulativen Updates aus dem Support-Perimeter entfernt und es wird ein Fehler gemeldet. `SP-S3006` Um eine vorhandene SQL Server-Instanz mithilfe eines CEV zu patchen, erstellen Sie ein neues CEV, das das gewünschte kumulative Update enthält, und ändern Sie dann Ihre vorhandene Instanz, um zum neuen CEV zu wechseln.

Wenn Sie eine RDS Custom for SQL Server-DB-Instance in einer Multi-AZ-Bereitstellung aktualisieren, führt RDS Custom for SQL Server fortlaufende Upgrades für Ihre Instanz durch. Bei diesem Ansatz werden Ausfallzeiten minimiert, indem jeweils eine Instanz aktualisiert wird. RDS führt die folgenden Aktionen durch, um fortlaufende Upgrades durchzuführen:

1. Führen Sie ein Upgrade der Standby-DB-Instance durch.
2. Failover zur aktualisierten Standby-DB-Instance, wodurch sie zur neuen primären DB-Instance wird.
3. Führen Sie ein Upgrade der neuen Standby-DB-Instance durch.

Die Ausfallzeit der DB-Instance bei Multi-AZ-Bereitstellungen entspricht der Zeit, die für den Failover benötigt wird.

Beim Upgrade einer RDS Custom for SQL Server-DB-Instance gelten die folgenden Einschränkungen:

- Benutzerdefinierte DB-Optionen und Parametergruppen werden nicht unterstützt.
- Alle zusätzlichen Speichervolumes, die Sie an Ihre RDS Custom for SQL Server-DB-Instance anhängen, werden nach dem Upgrade nicht angehängt.
- Denn die CEVs direkte Anwendung von kumulativen Updates für SQL Server wird nicht unterstützt und führt dazu, dass die Instanz aus dem Support-Perimeter entfernt wird.

Aktualisierung der Haupt- und Nebenversion der Engine

Upgrades sowohl der Haupt- als auch der Nebenversionen der Engine sind irreversibel und müssen immer auf eine neuere Version aktualisiert werden. Um verfügbare Zielversionen zu identifizieren, verwenden Sie die AWS Management Console und wählen Sie aus den verfügbaren Versionen aus, wenn Sie Ihre DB-Instance ändern. Verwenden Sie alternativ den `describe-db-engine-versions`CLI-Befehl oder den RDS-API-Befehl Describe Versions. DBEngine

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-engine-versions \
    --engine custom-sqlserver-se \
    --engine-version 15.00.4322.2.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Windows:

```
aws rds describe-db-engine-versions ^
    --engine custom-sqlserver-se ^
    --engine-version 15.00.4322.2.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

Die Ausgabe zeigt die verfügbaren Versionen der Ziel-Engine:

```
-------------------------
|DescribeDBEngineVersions|
+------------------------+
|      EngineVersion     |
+------------------------+
|  15.00.4410.1.v1       |
|  15.00.4415.2.v1       |
|  15.00.4430.1.v1       |
|  16.00.4165.4.v1       |
|  16.00.4175.1.v1       |
|  16.00.4185.3.v1       |
+------------------------+
```

Nachdem Sie Ihre Zielversion identifiziert haben, verwenden Sie die AWS Management Console und folgen Sie den Anweisungen unterÄndern einer RDS benutzerdefinierten SQL Server-DB-Instance. Verwenden Sie alternativ den `modify-db-instance`CLI-Befehl oder den Befehl Modify DBInstance RDS API.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier DB_INSTANCE_IDENTIFIER \
    --engine-version ENGINE_VERSION \
    --allow-major-version-upgrade \
    --region Region \
```

```
    --no-apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier DB_INSTANCE_IDENTIFIER ^
    --engine-version ENGINE_VERSION ^
    --allow-major-version-upgrade ^
    --region Region ^
    --no-apply-immediately
```

> **Note**
>
> Sie müssen den `--allow-major-version-upgrade` Parameter angeben, um größere Versionsupgrades durchzuführen.

Datenbank-Kompatibilitätsstufe

Sie können Microsoft SQL Server-Kompatibilitätsgrade verwenden, um einige Verhaltensweisen von Datenbanken zu justieren und somit vorherige Versionen von SQL Server zu simulieren. Weitere Informationen finden Sie unter Compatibility Level in der Microsoft-Dokumentation.

Wenn Sie Ihre DB-Instance upgraden, behalten alle bestehenden Datenbanken ihren ursprünglichen Kompatibilitätsgrad. Wenn Sie beispielsweise ein Upgrade von SQL Server 2019 auf SQL Server 2022 durchführen, haben alle vorhandenen Datenbanken einen Kompatibilitätsgrad von 150. Jede neue Datenbank, die nach dem Upgrade erstellt wurde, hat den Kompatibilitätsgrad 160.

Sie können den Kompatibilitätsgrad einer Datenbank ändern, indem Sie den Befehl ALTER DATABASE verwenden. Um beispielsweise eine Datenbank mit dem Namen so `customeracct` zu ändern, dass sie mit SQL Server 2022 kompatibel ist, geben Sie den folgenden Befehl ein:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 160
```

Beheben von DB-Problemen für Amazon RDS Custom for SDL Server

Das Modell der gemeinsamen Verantwortung von RDS Custom bietet Zugriff auf Betriebssystem-Shell-Ebene und Zugriff auf Datenbankadministratoren. RDS Custom führt Ressourcen in Ihrem Konto aus, im Gegensatz zu Amazon RDS, das Ressourcen in einem Systemkonto ausführt. Mit einem größeren Zugang kommt eine größere Verantwortung mit sich. In den folgenden Abschnitten erfahren Sie, wie Sie Probleme mit Amazon RDS Custom for SQL Server-DB-Instances beheben können.

> **Note**
>
> In diesem Abschnitt wird die Problembehandlung mit RDS Custom for SQL Server behandelt. Zur Fehlerbehebung bei RDS Custom for Oracle vgl. Beheben von DB-Problemen für Amazon RDS Custom für Oracle.

Themen

- Anzeigen von benutzerdefinierten RDS-Ereignissen
- Abonnieren von benutzerdefinierten RDS-Ereignissen
- Beheben von CEV-Fehlern für RDS Custom für SQL Server
- Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server
- Problembehandlung Storage-Full in RDS Custom für SQL Server
- Problembehandlung beim Status PENDING_RECOVERY für TDE-fähige Datenbanken in RDS Custom for SQL Server

Anzeigen von benutzerdefinierten RDS-Ereignissen

Das Verfahren zum Anzeigen von Ereignissen ist für RDS Custom- und Amazon RDS DB-Instances gleich. Weitere Informationen finden Sie unter RDSAmazon-Ereignisse anzeigen.

Verwenden Sie den `describe-events` Befehl AWS CLI, um die benutzerdefinierte RDS-Ereignisbenachrichtigung mit dem anzuzeigen. RDS Custom führt mehrere neue Ereignisse ein. Die Ereigniskategorien sind die gleichen wie für Amazon RDS. Eine Liste der Ereignisse finden Sie unter Amazon RDS-Ereigniskategorien und Ereignisnachrichten .

Im folgenden Beispiel werden Details zu den Ereignissen abgerufen, die für die angegebene RDS Custom DB-Instance aufgetreten sind.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Abonnieren von benutzerdefinierten RDS-Ereignissen

Das Verfahren zum Abonnieren von Ereignissen ist für RDS Custom- und Amazon RDS DB-Instances gleich. Weitere Informationen finden Sie unter RDSAmazon-Veranstaltungsbenachrichtigung abonnieren.

Verwenden Sie den Befehl `create-event-subscription`, um Ereignisbenachrichtigungen von RDS Custom zu abonnieren. Nutzen Sie die folgenden erforderlichen Parameter:

- `--subscription-name`
- `--sns-topic-arn`

Im folgenden Beispiel wird ein Abonnement für Backup- und Wiederherstellungsereignisse für eine RDS Custom DB-Instance im aktuellen AWS -Konto. Sie können auch anfordern, dass Benachrichtigungen an ein bestimmtes Amazon-SNS-Thema (Amazon Simple Notification Service) gesendet werden, die `--sns-topic-arn` bestimmt.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Beheben von CEV-Fehlern für RDS Custom für SQL Server

Wenn Sie versuchen, eine CEV zu erstellen, kann dieser Vorgang fehlschlagen. In diesem Fall gibt RDS Custom die Ereignismeldung `RDS-EVENT-0198` aus. Weitere Informationen zum Anzeigen von RDS-Ereignissen finden Sie unter Amazon RDS-Ereigniskategorien und Ereignisnachrichten .

Verwenden Sie die folgenden Informationen, um mögliche Ursachen zu beheben.

Fehlermeldung	Vorschläge für die Fehlerbehebung		
`Custom Engine Version creation expected a`	Führen Sie Sysprep auf der EC2 Instance aus, die Sie über das AMI		

Fehlermeldung	Vorschläge für die Fehlerbehebung
`Sysprep'd AMI. Retry creation using a Sysprep'd AMI.`	erstellt haben. Weitere Informationen zur Vorbereitung eines AMI mit Sysprep finden Sie unter Erstellen eines standardisierten Amazon Machine Image (AMI) mit Sysprep.
`EC2 Image permissions for image (AMI_ID) weren't found for customer (Customer_ID). Verify customer (Customer_ID) has valid permissions on the EC2 Image.`	Stellen Sie sicher, dass Ihr Konto und Ihr Profil, die für die Erstellung verwendet wurden, über die erforderlichen Berechtigungen für `create EC2 Instance` und für `Describe Images` für das ausgewählte AMI verfügen.
`Failed to rebuild databases with server collation (collation name) due to missing setup.exe file for SQL Server.`	Stellen Sie sicher, dass die `setup`-Datei unter `C:\Program Files\Microsoft SQL Server\nnn\Setup Bootstrap\SQLnnnn\setup.exe` verfügbar ist.
`Image (AMI_ID) doesn't exist in your account (ACCOUNT_ID). Verify (ACCOUNT_ID) is the owner of the EC2 image.`	Stellen Sie sicher, dass das AMI im selben Kundenkonto vorhanden ist.
`Image id (AMI_ID) isn't valid. Specify a valid image id, and try again.`	Der Name des AMI ist falsch. Stellen Sie sicher, dass die richtige AMI-ID angegeben ist.

Fehlermeldung	Vorschläge für die Fehlerbehebung
`Image (AMI_ID) operating system platform isn't supported. Specify a valid image, and try again.`	Wählen Sie ein unterstütztes AMI, das Windows Server mit SQL Server Enterprise, Standard oder Web Edition enthält. Wählen Sie ein AMI mit einem der folgenden Nutzungsoperationscodes aus dem EC2 Marketplace aus: • RunInstancesWählen Sie ein AMI mit einem der folgenden Verwendungsoperationscodes aus dem Marketplace: ----sep----:0102 - Windows with SQL Server Enterprise • RunInstances:0102 - Windows mit SQL Server Enterprise ----sep----:0006 - Windows mit SQL Server Standard • RunInstances:0006 - Windows mit SQL Server Standard ----sep----:0202 - Windows mit SQL Server Web
`SQL Server Web Edition isn't supported for creating a Custom Engine Version using Bring Your Own Media. Specify a valid image, and try again.`	Verwenden Sie ein AMI, das eine unterstützte Edition von SQL Server enthält. Weitere Informationen finden Sie unter Versionsunterstützung für RDS Custom für SQL Server CEVs.

Fehlermeldung	Vorschläge für die Fehlerbehebung
`The custom engine version can't be the same as the OEV engine version. Specify a valid CEV, and try again.`	Classic Engine-Versionen von RDS Custom für SQL Server werden nicht unterstützt. Zum Beispiel Version 15.00.4073.23.v1. Verwenden Sie eine unterstützte Versionsnummer.
`The custom engine version isn't in an active state. Specify a valid CEV, and try again.`	Die CEV muss sich im Status `AVAILABLE` befinden, damit der Vorgang abgeschlossen werden kann. Ändern Sie die CEV von `INACTIVE` in `AVAILABLE` .
`The custom engine version isn't valid for an upgrade. Specify a valid CEV with an engine version greater or equal to (X), and try again.`	Die Ziel-CEV ist nicht gültig. Überprüfen Sie die Anforderungen für einen gültigen Upgrade-Pfad.
`The custom engine version isn't valid. Names can include only lowercase letters (a-z), dashes (-), underscores (_), and periods (.). Specify a valid CEV, and try again.`	Halten Sie sich an die erforderl iche CEV-Benennungskonvention. Weitere Informationen finden Sie unter Anforderungen für RDS Custom für SQL Server CEVs.
`The custom engine version isn't valid. Specify valid database engine version, and try again. Example: 15.00.4073.23-cev123.`	Eine nicht unterstützte DB-Engine -Version wurde bereitgestellt. Verwenden Sie eine unterstützte DB-Engine-Version.

Fehlermeldung	Vorschläge für die Fehlerbehebung		
`The expected architecture is (X) for image (AMI_ID), but architecture (Y) was found.`	Verwenden Sie ein AMI, das auf der x86_64-Architektur basiert.		
`The expected owner of image (AMI_ID) is customer account ID (ACCOUNT_ID), but owner (ACCOUNT_ID) was found.`	Erstellen Sie die EC2 Instance aus dem AMI, für das Sie berechtigt sind. Führen Sie Sysprep auf der EC2 Instanz aus, um ein Basis-Image zu erstellen und zu speichern.		
`The expected platform is (X) for image (AMI_ID), but platform (Y) was found.`	Verwenden Sie ein AMI, das auf der Windows-Plattform basiert.		
`The expected root device type is (X) for image %s, but root device type (Y) was found.`	Erstellen Sie das AMI mit dem EBS-Gerätetyp.		

Fehlermeldung	Vorschläge für die Fehlerbehebung
`The expected SQL Server edition is (X), but (Y) was found.`	Wählen Sie ein unterstütztes AMI, das Windows Server mit SQL Server Enterprise, Standard oder Web Edition enthält. Wählen Sie ein AMI mit einem der folgenden Nutzungsoperationscodes aus dem EC2 Marketplace aus: • RunInstancesWählen Sie ein AMI mit einem der folgenden Verwendungsoperationscodes aus dem Marketplace: ----sep----:0102 - Windows with SQL Server Enterprise • RunInstances:0102 - Windows mit SQL Server Enterprise ----sep----:0006 - Windows mit SQL Server Standard • RunInstances:0006 - Windows mit SQL Server Standard ----sep----:0202 - Windows mit SQL Server Web
`The expected state is (X) for image (AMI_ID), but the following state was found: (Y).`	Stellen Sie sicher, dass sich das AMI im Status `AVAILABLE` befindet.
`The provided Windows OS name (X) isn’t valid. Make sure the OS is one of the following: (Y).`	Verwenden Sie ein unterstütztes Windows-Betriebssystem.

Fehlermeldung	Vorschläge für die Fehlerbehebung
`Unable to find bootstrap log file in path.`	Stellen Sie sicher, dass die Protokolldatei unter `C:\Program Files\Microsoft SQL Server\nnn\Setup Bootstrap\Log\Summary.txt` verfügbar ist.
`RDS expected a Windows build version greater than or equal to (X), but found version (Y)..`	Verwenden Sie ein AMI mit einer Betriebssystem-Build-Version von mindestens 14393.
`RDS expected a Windows major version greater than or equal to (X), but found version (Y)..`	Verwenden Sie ein AMI mit einer Betriebssystem-Hauptversion von mindestens 10.0 oder höher.

Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server

Aufgrund des Modells der geteilten Verantwortung liegt es in Ihrer Verantwortung, Konfigurationsprobleme zu beheben, die Ihre RDS Custom for SQL Server-DB-Instance in den Zustand `unsupported-configuration` versetzen. Wenn das Problem in der AWS Infrastruktur liegt, können Sie es mit der Konsole oder AWS CLI dem beheben. Wenn das Problem mit dem Betriebssystem oder der Datenbankkonfiguration besteht, können Sie sich beim Host anmelden, um es zu beheben.

> **Note**
>
> In diesem Abschnitt wird erläutert, wie Sie nicht unterstützte Konfigurationen in RDS Custom for SQL Server beheben. Weitere Informationen zum Konfigurieren von RDS Custom for Oracle finden Sie unter Fehlerbehebung bei nicht unterstützten Konfigurationen in RDS Custom für Oracle.

In den folgenden Tabellen finden Sie Beschreibungen der Benachrichtigungen und Ereignisse, die vom Support-Perimeter gesendet werden, und Informationen zu deren Behebung. Diese

Benachrichtigungen und der Support-Umfang können sich ändern. Hintergrundinformationen zum Support-Perimeter finden Sie unter Support-Perimeter in RDS Custom. Beschreibungen der Ereignisse finden Sie unter Amazon RDS-Ereigniskategorien und Ereignisnachrichten .

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S0000	Manuelle, nicht unterstützte Konfiguration	Der Status der benutzerdefinierten RDS-DB-Instance ist aus folgenden Gründen auf [Nicht unterstützte Konfiguration] gesetzt:. X	Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.

AWS Ressource (Infrastruktur)

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1001	EC2 Status der Instanz	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die zugrunde liegende EC2 Instance %s wurde gestoppt, ohne die RDS-Instance anzuhalten. Sie können dieses Problem lösen,	Um den Status einer DB-Instance zu überprüfen, verwenden Sie die Konsole oder führen Sie den folgenden AWS CLI Befehl aus: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
		indem Sie die zugrunde liegende EC2 Instance starten und sicherstellen, dass die Binär- und Datenvolumes angehängt sind. Wenn Sie die RDS-Instance beenden möchten, stellen Sie zunächst sicher, dass sich die zugrunde liegende EC2 Instance im Status AVAILABLE befindet, und beenden Sie dann die RDS-Instance mithilfe der RDS-Konsole oder CLI.	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S1002	EC2 Status der Instanz	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Der Status der RDS-DB-Instance ist auf gesetzt, STOPPED aber die zugrunde liegende EC2 Instance %s wurde gestartet. Sie können dieses Problem lösen, indem Sie die zugrunde liegende EC2 Instance beenden. Wenn Sie die RDS-Instance starten möchten, verwenden Sie die Konsole oder CLI.	Verwenden Sie den folgenden AWS CLI Befehl, um den Status einer DB-Instance zu überprüfen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep DBInstanceStatus`
			Sie können den Status der EC2 Instance auch über die EC2 Konsole überprüfen.
			Um eine DB-Instance zu starten, verwenden Sie die Konsole oder führen Sie den folgenden AWS CLI Befehl aus:
			`aws rds start-db-instance \ --db-instance-identifier db-instance-name`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1003	EC2 Instance-Klasse	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Es besteht eine Diskrepanz zwischen der erwarteten und der konfigurierten DB-Instance-Klasse des EC2 Hosts. Sie können dieses Problem lösen, indem Sie die DB-Instance-Klasse auf ihren ursprünglichen Klassentyp ändern.	Verwenden Sie den folgenden CLI-Befehl, um die erwartete DB-Instance-Klasse zu überprüfen: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep DBInstanceClass`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1004	Auf das EBS-Speichervolume kann nicht zugegriffen werden	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Auf das ursprüngliche EBS-Speichervolume %s, das der EC2 Instance zugeordnet war, kann derzeit nicht zugegriffen werden.	
SP-S1005	Das EBS-Speichervolume wurde getrennt	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Das ursprüngliche EBS-Speichervolume „Volume-ID“ ist nicht angehängt. Sie können dieses Problem lösen, indem Sie das der Instance zugeordnete EBS-Volume anhängen. EC2	Verwenden Sie nach dem erneuten Anhängen des EBS-Volumes die folgenden CLI-Befehle, um zu überprüfen, ob das EBS-Volume 'volume-id' ordnungsgemäß an die RDS-Instance angehängt ist: `aws ec2 describe-volumes \ --volume-ids volume-id | grep InstanceId`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S1006	Größe des EBS-Speicher-Volumes	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Es besteht eine Diskrepanz zwischen den erwarteten und den konfigurierten Einstellungen des EBS-Speichervolumes „Volume-ID“. Die Volume-Größe wurde manuell auf der EC2 Ebene ihrer ursprünglichen Werte von [%s] geändert. Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	Verwenden Sie den folgenden CLI-Befehl, um die Volume-Größe der Volume-ID-Details des EBS-Volumes mit den Details der RDS-Instanz zu vergleichen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep AllocatedStorage`
			Verwenden Sie den folgenden CLI-Befehl, um die tatsächliche Größe des zugewiesenen Volumes anzuzeigen:
			`aws ec2 describe-volumes \ --volume-ids |grep Size`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S1007	Konfiguration des EBS-Speichervolumes	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Es besteht eine Diskrepanz zwischen den erwarteten und den konfigurierten Einstellungen des EBS-Speichervolumes „Volume-ID“. Sie können dieses Problem beheben, indem Sie die Konfiguration des EBS-Speicher-Volumes [IOPS, Durchsatz, Volume-Typ] auf die ursprünglichen Werte [IOPS: %s, Durchsatz: %s, Volume-Typ: %s] auf der Ebene ändern. EC2 Verwenden Sie für future Speicheränderungen die RDS-Konsole oder	Verwenden Sie den folgenden CLI-Befehl, um den Volumetyp der Volume-ID-Details des EBS-Volumes und die RDS-Instance-Details zu vergleichen. Stellen Sie sicher, dass die Werte auf EBS-Ebene mit den Werten auf RDS-Ebene übereinstimmen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`
			So ermitteln Sie den erwarteten Wert für den Speicherdurchsatz auf RDS-Ebene:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageThroughput`
			So ermitteln Sie den erwarteten Wert für Volume-IOPS auf RDS-Ebene:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep Iops`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

		CLI. Die Größe des Datenträgers wurde ebenfalls manuell gegenüber dem ursprünglichen Wert (en) von [%s] geändert. EC2 Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	Um den aktuellen Speichertyp auf der EC2 Ebene abzurufen:
			`aws ec2 describe-volumes \ --volume-ids |grep VolumeType`
			Um den aktuellen Wert für den Speicherdurchsatz auf der EC2 Ebene abzurufen:
			`aws ec2 describe-volumes \ --volume-ids |grep Throughput`
			Um den aktuellen Wert für Volume-IOPS auf der EC2 Ebene abzurufen:
			`aws ec2 describe-volumes \ --volume-ids |grep Iops`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S1008	Größe und Konfiguration des EBS-Speichervolumes	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Es besteht eine Diskrepanz zwischen den erwarteten und den konfigurierten Einstellungen des EBS-Speichervolumes „Volume-ID“. Sie können dieses Problem beheben, indem Sie die Konfiguration des EBS-Speicher-Volumes [IOPS, Durchsatz, Volume-Typ] auf die ursprünglichen Werte [IOPS: %s, Durchsatz: %s, Volume-Typ: %s] auf der Ebene ändern. EC2 Verwenden Sie für future Speicheränderungen die RDS-Konsole oder	Verwenden Sie den folgenden CLI-Befehl, um den Volumetyp der Volume-ID-Details des EBS-Volumes und die RDS-Instance-Details zu vergleichen. Stellen Sie sicher, dass die Werte auf der EBS-Ebene mit den Werten auf der RDS-Ebene übereinstimmen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`
			So ermitteln Sie den erwarteten Wert für den Speicherdurchsatz auf RDS-Ebene:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageThroughput`
			So ermitteln Sie den erwarteten Wert für Volume-IOPS auf RDS-Ebene:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep Iops`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

		CLI. Die Größe des Datenträgers wurde ebenfalls manuell gegenüber dem ursprünglichen Wert (en) von [%s] geändert. EC2 Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	Um den aktuellen Speichertyp auf der EC2 Ebene abzurufen:
			`aws ec2 describe-volumes \ --volume-ids |grep VolumeType`
			Um den aktuellen Wert für den Speicherdurchsatz auf der EC2 Ebene abzurufen:
			`aws ec2 describe-volumes \ --volume-ids |grep Throughput`
			Um den aktuellen Wert für Volume-IOPS auf der EC2 Ebene abzurufen:
			`aws ec2 describe-volumes \ --volume-ids |grep Iops`
			So ermitteln Sie die erwartete Größe des zugewiesenen Volumes:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep AllocatedStorage`
			Um die tatsächliche Größe des zugewiesenen Volumes zu ermitteln:
			`aws ec2 describe-volumes \ --volume-ids |grep Size`

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1009	SQS-Berechtigungen	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Amazon Simple Queue Service (SQS) -Berechtigungen für das IAM-Instance-Profil fehlen. Sie können dieses Problem lösen, indem Sie sicherstellen, dass das mit dem Host verknüpfte IAM-Profil über die folgenden Berechtigungen verfügt: ["SQS: „, "SQS: SendMessage „, "SQS: ReceiveMessage „, "SQS: „]. DeleteMessage GetQueueUrl	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1010	SQS VPC-Endpunkt	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Eine VPC-Endpunktrichtlinie blockiert die Amazon Simple Queue Service (SQS) -Operationen. Sie können dieses Problem lösen, indem Sie Ihre VPC-Endpunktrichtlinie ändern, um die erforderlichen SQS-Aktionen zuzulassen.	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1011	Richtlinie für den Event-Bus	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die ressourcenbasierte Richtlinie für Ihren Event-Bus arn:aws:events:region- 1:123456789012:event-bus/default verweigert Amazon events: actions. CloudWatch PutEvents Beheben Sie dieses Problem, indem PutEvents Sie Ihre ressourcenbasierte Richtlinie so ändern, dass Ereignisse zulässig sind: Aktionen für %s. EventBus	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1012	CloudWatch VPC-Berechtigungen	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Einer VPC-Endpunktrichtlinie fehlen Berechtigungen für den Zugriff auf CloudWatch Amazon-Ereignisse. Beheben Sie dieses Problem, indem Sie Ihre VPC-Endpunktrichtlinie so ändern, dass Ereignisse zulässig sind: PutEvents on EventBus arn:aws:events:region- 1:123456789012:event-bus/default.	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1013	Service-Kontrollrichtlinie	Der Status der RDS Custom DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: In Ihrer AWS Organizations Servicekontrollrichtlinie fehlen die Zugriffsberechtigungen für CloudWatch Amazon-Ereignisse. Beheben Sie dieses Problem, indem Sie Ihre Servicesteuerungsrichtlinie so ändern, dass Ereignisse zulässig sind: PutEvents on EventBus arn:aws:events:region- 1:123456789012:event-bus/default.	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1014	IAM-Instance-Profil	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die %s-Berechtigungen Ihres IAM-Instance-Profils verweigern CloudWatch Amazon-Ereignisse. Beheben Sie dieses Problem, indem Sie ["events: PutEvents „] auf 'Zulassen' setzen und Ereignisse zulassen: PutEvents on EventBus arn:aws:events:region- 1:123456789012:event-bus/default in Ihrem mit der Instance verknüpften IAM-Profil.	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1015	IAM-Instance-Profil	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: In Ihrem IAM-Instance-Profil %s fehlen CloudWatch Amazon-Ev entberechtigungen. Beheben Sie dieses Problem, indem Sie die Berechtigungen ["events: PutEvents „] und das Zulassen von Events: PutEvents on EventBus arn:aws:events:region- 1:123456789012:event-bus/default in Ihr mit der Instance verknüpftes IAM-Profil aufnehmen.	

Code des Ereignisses	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S1016	Grenze der IAM-Berechtigungen	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Ihr IAM-Instance-Profil %s hat eine Berechtigungsgrenze, die CloudWatch Amazon-Ereignisse ablehnt. Beheben Sie dieses Problem, indem Sie ["events: PutEvents „] für EventBus arn:aws:events:region- 1:123456789012:event-bus/default in der Berechtigungsgrenze Ihres IAM-Instance-Profils auf „Zulassen“ setzen.	

Betriebssystem

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S2001	Status des SQL-Dienstes	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Der SQL Server-Dienst wurde nicht gestartet. Sie können dieses Problem beheben, indem Sie den SQL Server-Dienst auf dem Host neu starten. Wenn es sich bei dieser DB-Instance um eine Multi-AZ-DB-Instance handelt und der Neustart fehlschlägt, beenden und starten Sie den Host, um einen Failover einzuleiten.	

SP-S2002	Status des benutzerdefinierten RDS-Agenten	Der Status der RDS Custom DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Der RDS Custom Agent-Dienst ist nicht installiert	Melden Sie sich beim Host an und stellen Sie sicher, dass der RDS Custom Agent ausgeführt wird.
			
			Sie können die folgenden Befehle verwenden, um den Agentenstatus anzuzeigen.
			
			`$name = "RDSCustomAgent"`
			`$service = Get-Service $name`

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

		oder konnte nicht gestartet werden. Sie können dieses Problem lösen, indem Sie das Windows-Ereignisprotokoll überprüfen, um festzustellen, warum der Dienst nicht gestartet werden kann, und die entsprechenden Maßnahmen ergreifen, um das Problem zu beheben. Wenn Sie weitere Unterstützung benötigen, erstellen Sie eine Support-Anfrage.	`Write-Host $service.Status`
			
			Wenn der Status nicht `Running` ist, können Sie den Service mit dem folgenden Befehl starten:
			
			`Start-Service $name`
			
			Wenn der Agent nicht gestartet werden kann, sehen Sie in den Windows-Ereignissen nach, warum er nicht gestartet werden kann. Der Agent benötigt einen Windows-Benutzer, um den Dienst zu starten. Stellen Sie sicher, dass ein Windows-Benutzer vorhanden ist und über die Rechte zum Ausführen des Dienstes verfügt.

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
`SP-S2003`	Status des SSM-Agenten	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Der Amazon SSM Agent-Service ist nicht erreichbar. Sie können dieses Problem beheben, indem Sie den Servicestatus mit dem `Get-Service AmazonSSMAgent` PowerShell Befehl überprüfen oder den Service mit starten. `Start-Service AmazonSSMAgent` Stellen Sie sicher, dass ausgehender HTTPS-Verkehr (Port 443) zu den regionalen Endpunkten ssm, ssmmessages und ec2messages zulässig ist.	Weitere Informationen finden Sie unter Fehlerbehebung bei SSM-Agent. Informationen zur Fehlerbehebung bei SSM-Endpunkten finden Sie unter Es konnte keine Verbindung zu SSM-Endpunkten hergestellt werden und Mithilfe von ssm-cli kann die Verfügbarkeit verwalteter Knoten behoben werden.

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S2004	Anmeldung für benutzerdefinierte RDS-Agenten	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Bei der SQL-Anmeldung "$HOSTNAME/RDSAgent” ist ein unerwartetes Problem aufgetreten. Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
`SP-S2005`	Zeitzone	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Die Zeitzone auf der EC2 Amazon-Instance [%s] wurde geändert. Sie können dieses Problem lösen, indem Sie die Zeitzone wieder auf die Einstellung ändern, die bei der Instance-Erstellung angegeben wurde. Wenn Sie eine Instanz mit einer bestimmten Zeitzone erstellen möchten, finden Sie weitere Informationen in der Dokumentation zu RDS Custom.	Führen Sie den `Get-Timezone` PowerShell Befehl aus, um die Zeitzone zu bestätigen. Weitere Informationen finden Sie unter Lokale Zeitzone für RDS benutzerdefinierte SQL Server-DB-Instances.

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S2006	Version der Softwarelösung mit hoher Verfügbarkeit	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die Hochverfügbarkeits-Softwarelösung der aktuellen Instance unterscheidet sich von der erwarteten Version. Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S2007	Konfiguration der Softwarelösung mit hoher Verfügbarkeit	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die Konfigurationseinstellungen der Softwarelösung für hohe Verfügbarkeit wurden auf unerwartete Werte für die Instanz %s geändert. Um dieses Problem zu beheben, starten Sie die EC2 Instance neu. Wenn Sie die EC2 Instance neu starten, werden die Einstellungen automatisch auf die erforderliche Konfiguration für die Hochverfügbarkeits-Softwarelösung aktualisiert.	

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S2008	SQL Server-Dienst	Die benutzerdefinierte RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] eingestellt: Der Dienst SQLServer (MSSQLServer) ist auf dem Host nicht vorhanden. Um dieses Problem zu lösen, erstellen Sie einen Support-Fall.	Sie können die folgenden Befehle verwenden, um den Agentenstatus anzuzeigen.
			`$name = "MSSQLServer"`
			`$service = Get-Service $name`
			`Write-Host $service.Status`

SP-2009	SSL-Zertifikat	Die benutzerdefinierte RDS-DB-Instance ist aus folgenden Gründen auf [Nicht unterstützte Konfiguration] gesetzt: Nicht selbstsignierte SSL-Zertifikate führen zu Störungen in RDS. Um dieses Problem zu beheben, entfernen Sie die nicht selbstsignierten Zertifikate aus dem vertrauenswürdigen Stammzertifikatsspeicher.	Führen Sie den folgenden PowerShell Befehl aus, um nicht selbstsignierte Zertifikate zu überprüfen. `Get-ChildItem cert:\LocalMachine\root -Recurse	Where-Object {$_.Issuer -ne $_.Subject -and $_.Issuer -notlike "*RDSCustomAgentCA*"}` Weitere Informationen finden Sie unter HTTP-Fehler 403.16, wenn Sie versuchen, auf eine Website zuzugreifen, die auf IIS gehostet wird.

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-2010	Speicherstatus des Root-Volumes	Die benutzerdefinierte RDS-DB-Instance ist aus folgendem Grund auf [Nicht unterstützte Konfiguration] gesetzt: Der Speicher auf dem Root-Volume ist voll. Um dieses Problem zu beheben, geben Sie mindestens 500 MiB Speicherplatz auf dem EBS-Stammvolume „volume-id“ frei oder erhöhen Sie die Volume-Größe und ändern Sie die Größe des Laufwerks C auf der EC2 Instance „instance-id“. Die Änderungen an der Größe des Root-Volumes bleiben nicht bestehen, wenn Sie die Instance austauschen. EC2	Verwenden Sie den folgenden Befehl, um den verfügbaren Speicherplatz auf dem Root-Volume (C:) anzuzeigen. `(Get-PSDrive -Name C).Free / 1MB` Weitere Informationen zum Ändern des EBS-Root-Volumes finden Sie unter How

Datenbank

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S3001	Protokoll für gemeinsam genutzten Speicher von SQL Server	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Das SQL Server-Protokoll für gemeinsam genutzten Speicher ist deaktiviert. Sie können dieses Problem lösen, indem Sie das Shared Memory-Protokoll im SQL Server Configuration Manager aktivieren.	Sie können dies überprüfen, indem Sie Folgendes überprüfen: SQL Server-Konfigurations-Manager > SQL Server-Netzwerkkonfiguration > Protokolle für MSSQLSERVER > Shared Memory as Enabled. Nachdem Sie das Protokoll aktiviert haben, starten Sie den SQL Server-Prozess neu.
SP-S3002	Hauptschlüssel für den Dienst	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgendem Grund: RDS Automation kann die Sicherung des Service Master Key (SMK) nicht als Teil der neuen	

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
		SMK-Generation übernehmen. Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	
SP-S3003	Hauptschlüssel für den Service	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, weil: Die Metadaten zum Service Master Key (SMK) fehlen oder sind unvollständig. Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S3004	Version und Edition der DB-Engine	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Es besteht eine Diskrepanz zwischen der erwarteten und der installierten Version und Edition von SQL Server: Das Ändern der SQL Server-Edition wird auf RDS Custom for SQL Server nicht unterstützt. Außerdem wird das manuelle Ändern der SQL Server-Version auf der EC2 RDS-Custom-Instanz nicht unterstützt. Um dieses Problem zu beheben, erstellen Sie eine Support-Anfrage.	Führen Sie die folgende Abfrage aus, um die SQL-Version abzurufen:
			`select @@version`
			Führen Sie den folgenden AWS CLI Befehl aus, um die Version und Edition der RDS-SQL-Engine abzurufen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion`
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Engine`
			Weitere Informationen erhalten Sie unter Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance und Upgrade der Engine-Version für eine DB-Instance.

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S3005	DB-Engine-Edition	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die aktuelle SQL Server-Edition entspricht nicht der erwarteten SQL Server-Edition [%s]: Das Ändern der SQL Server-Edition wird auf RDS Custom for SQL Server nicht unterstützt. Um dieses Problem zu lösen, erstellen Sie eine Support-Anfrage.	Führen Sie die folgende Abfrage aus, um die SQL-Edition abzurufen:
			Example
			`select @@version`
			Führen Sie den folgenden AWS CLI Befehl aus, um die RDS SQL Engine-Edition abzurufen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Engine`

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S3006	DB Engine Version	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die aktuelle SQL Server-Version entspricht nicht der erwarteten SQL Server-Version [%s]: Sie können die SQL Server-Version auf der benutzerdefinierten EC2 RDS-Instanz nicht manuell ändern. Um dieses Problem zu lösen, erstellen Sie einen Support-Fall. Für future Änderungen an der SQL Server-Version können Sie die Instanz über die AWS RDS-Konsole oder über den modify-db-instance CLI-Befehl ändern.	Führen Sie die folgende Abfrage aus, um die SQL-Version abzurufen:
			Example
			`select @@version`
			Führen Sie den folgenden AWS CLI Befehl aus, um die RDS-SQL-Engine-Version abzurufen:
			`aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion`
			Weitere Informationen erhalten Sie unter Ändern einer RDS benutzerdefinierten SQL Server-DB-Instance und Upgrade der Engine-Version für eine DB-Instance.

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess

SP-S3007	Speicherort der Datenbankdatei	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Datenbankdateien werden außerhalb des Laufwerks D:\ konfiguriert. Sie können dieses Problem lösen, indem Sie sicherstellen, dass alle Datenbankdateien, einschließlich ROW, LOG, FILESTREAM usw., auf dem Laufwerk D:\ gespeichert sind.	Führen Sie die folgende Abfrage aus, um den Speicherort von Datenbankdateien aufzulisten, die sich nicht im Standardpfad befinden:
			`USE master;`
			`SELECT physical_name as files_not_in_default_path`
			`FROM sys.master_files`
			`WHERE SUBSTRING(physical_name,1,3)!='D:\';`

Ereigniscode	Bereich „Konfiguration“	RDS Event-Meldung	Validierungsprozess
SP-S3008	Das Limit für die Datenbankanzahl wurde überschritten	Der Status der benutzerdefinierten RDS-DB-Instance ist auf [Nicht unterstützte Konfiguration] gesetzt, und zwar aus folgenden Gründen: Die Gesamtzahl der Datenbanken in der DB-Instance überschreitet die maximale Grenze von 5000. Um dieses Problem zu beheben, reduzieren Sie die Anzahl der Datenbanken unter das maximal unterstützte Limit.	Verwenden Sie den folgenden Befehl, um die Gesamtzahl der Datenbanken anzuzeigen: `SELECT COUNT(name) as databaseCount FROM sys.databases WHERE name not in ('tempdb','master','model','msdb','DWDiagnostics','DWConfiguration','DWQueue');`

Problembehandlung **`Storage-Full`** in RDS Custom für SQL Server

RDS Custom überwacht auch das Root-Volume (C:). Die DB-Instance RDS Custom for SQL Server wechselt in den `unsupported-configuration` Zustand, in dem auf dem Root-Volume weniger als 500 MiB Festplattenspeicher verfügbar sind. Weitere Informationen finden Sie unter `Event SP-S2010` in Korrigieren von nicht unterstützten Konfigurationen in RDS Custom for SQL Server.

Problembehandlung beim Status PENDING_RECOVERY für TDE-fähige Datenbanken in RDS Custom for SQL Server

SQL Server-Datenbanken mit aktivierter transparenter Datenverschlüsselung (TDE) bleiben möglicherweise im `PENDING_RECOVERY` Status, wenn bei der automatischen Entschlüsselung Probleme auftreten. Dies ist in der Regel nach einer Wiederherstellung der DB-Instance der Fall,

wenn die Service Master Key (SMK) -Backup-Datei der Quell-DB-Instance, die im RDS Custom Managed S3-Bucket in Ihrem Konto gespeichert ist, vor Abschluss der Wiederherstellung gelöscht wurde.

Um die automatische Entschlüsselung zu aktivieren und die TDE-fähigen Datenbanken online zu schalten, müssen Sie den Database Master Key (DMK) mit seinem Passwort öffnen und den DMK mithilfe der SMK verschlüsseln.

Verwenden Sie die folgenden SQL Server-Befehle als Referenz:

```
-- Identify PENDING_RECOVERY TDE databases
USE MASTER;
GO
SELECT name, is_encrypted, state_desc FROM sys.databases;
GO

-- Open DMK using password
OPEN MASTER KEY DECRYPTION BY PASSWORD = '<password>';
GO

-- Encrypt DMK using SMK
ALTER MASTER KEY ADD ENCRYPTION BY SERVICE MASTER KEY;
GO

-- Close SMK
CLOSE MASTER KEY;
GO

-- Bring the TDE databases online
ALTER DATABASE <database_name> SET ONLINE;
GO

-- Verify TDE databases are now in ONLINE state
SELECT name, is_encrypted, state_desc FROM sys.databases;
GO
```

Amazon RDS auf AWS Outposts

Amazon RDS on AWS Outposts erweitert RDS für SQL Server-, RDS für MySQL- und RDS für PostgreSQL-Datenbanken auf AWS Outposts Rack Gen1- und Gen2-Umgebungen. AWS Outposts verwendet dieselbe Hardware wie in der Öffentlichkeit, AWS-Regionen um AWS Dienste, Infrastruktur und Betriebsmodelle vor Ort bereitzustellen. Mit RDS unter Outposts können Sie verwaltete DB-Instances nahe Geschäftsanwendungen bereitstellen, die lokal ausgeführt werden müssen. Weitere Informationen AWS Outposts dazu finden Sie in der AWS Outposts Dokumentation und auf der AWS Outposts Produktseite.

Sie verwenden dieselbe AWS Management Console, AWS CLI, und RDS-API zur Bereitstellung und Verwaltung von lokalen RDS auf Outposts-DB-Instances wie für RDS-DB-Instances, die in der ausgeführt werden. AWS Cloud RDS unter Outposts automatisiert Aufgaben wie Datenbankbereitstellung, Betriebssystem- und Datenbank-Patching, Backup und Langzeitarchivierung in Amazon S3.

RDS unter Outposts unterstützt automatisierte Backups von DB-Instances. Für die Sicherung und Wiederherstellung von DB-Instances AWS-Region ist eine Netzwerkverbindung zwischen Ihrem Outpost und Ihrem erforderlich. Alle DB-Snapshots und Transaktionsprotokolle von einem Outpost werden in Ihrem gespeichert. AWS-Region Sie können aus Ihrer AWS -Region eine DB-Instance aus einem DB-Snapshot in einem anderen Outpost wiederherstellen. Weitere Informationen finden Sie unter Einführung in Backups.

RDS unter Outposts unterstützt automatisierte Wartungsvorgänge und automatisierte Upgrades von DB-Instances. Weitere Informationen finden Sie unter Warten einer DB-Instance.

RDS auf Outposts verwendet Verschlüsselung im Ruhezustand für DB-Instanzen und DB-Snapshots unter Verwendung Ihrer AWS KMS key. Weitere Informationen zur Verschlüsselung im Ruhezustand finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.

Standardmäßig können EC2 Instances in Outposts-Subnetzen den Amazon Route 53 DNS-Service verwenden, um Domainnamen in IP-Adressen aufzulösen. Abhängig von der Pfadlatenz zwischen Ihrem Outpost und der AWS-Region können längere DNS-Behebungszeiten mit Route 53 auftreten. In solchen Fällen können Sie die in Ihrer lokalen Umgebung installierten DNS-Server verwenden. Weitere Informationen finden Sie unter DNS im AWS Outposts -Benutzerhandbuch.

Wenn die Netzwerkverbindung zu dem nicht AWS-Region verfügbar ist, wird Ihre DB-Instance weiterhin lokal ausgeführt. Sie können weiterhin mit DNS-Namensauflösungen auf DB-Instances

zugreifen, indem Sie einen lokalen DNS-Server als sekundären Server konfigurieren. Sie können jedoch keine neuen DB-Instances erstellen oder bestehende DB-Instances ändern. Automatische Backups werden nicht durchgeführt, wenn keine Konnektivität vorhanden ist. Wenn ein DB-Instancefehler auftritt, wird die DB-Instance erst dann automatisch ersetzt, wenn die Konnektivität wiederhergestellt wurde. Wir empfehlen, die Netzwerkkonnektivität so schnell wie möglich wiederherzustellen.

Themen

- Voraussetzungen für Amazon RDS auf AWS Outposts
- Amazon RDS zur AWS Outposts Unterstützung von RDS Amazon-Funktionen
- Unterstützte DB-Instance-Klassen für Amazon RDS auf AWS Outposts
- Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts
- Arbeiten mit Multi-AZ-Bereitstellungen für Amazon RDS in AWS Outposts
- Erstellen von DB-Instances für Amazon RDS in AWS Outposts
- Read Replicas für Amazon RDS erstellen auf AWS Outposts
- Überlegungen zum Wiederherstellen von DB-Instances in Amazon RDS on AWS Outposts

Voraussetzungen für Amazon RDS auf AWS Outposts

Dies sind die Voraussetzungen für die Verwendung von Amazon RDS in AWS Outposts:

- Installieren Sie es AWS Outposts in Ihrem lokalen Rechenzentrum. Weitere Informationen finden Sie unter Installation eines AWS Outposts Servers im AWS Outposts Serverinstallationshandbuch.
- Stellen Sie sicher, dass mindestens ein Subnetz für RDS unter Outposts zur Verfügung steht. Sie können dasselbe Subnetz auch für andere Workloads verwenden.
- Es muss eine zuverlässige Netzwerkverbindung zwischen Ihrem Outpost und einer AWS -Region vorhanden sein.

Amazon RDS zur AWS Outposts Unterstützung von RDS Amazon-Funktionen

In der folgenden Tabelle werden die RDS Amazon-Funktionen beschrieben, die von Amazon RDS am unterstützt werden AWS Outposts.

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen

Bereitstellung von DB-Instances	Ja	Sie können DB-Instances nur RDS für SQL Server-SQL, RDS My- und RDS SQL Postgre-DB-Engines erstellen. Die folgenden Versionen werden unterstützt:
		• SQLMicrosoft-Server:
		  • 15.00.4043.16.v1 und höhere 2019 Versionen
		  • 14.00.3294.2.v1 und höhere 2017 Versionen
		  • 13.00.5820.21.v1 und höhere 2016 Versionen
		• Meine Versionen SQL 8.0 und 8.4
		• Alle Postgre SQL 16 & 15 & 14 & 13 Versionen sowie Postgre SQL Version 12.5 und höhere SQL Postgre 12-Versionen
			Erstellen von DB-Instances für Amazon RDS in AWS Outposts

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Stellen Sie mit Microsoft SQL Server Management Studio eine Connect zu einer Microsoft SQL Server-DB-Instance her	Ja	Einige TLS Versionen und Verschlüsselungsschiffren sind möglicherweise nicht sicher. Um sie auszuschalten, befolgen Sie die Anweisungen unter Konfiguration von SQL Serversicherheitsprotokollen und Chiffren.	Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen
Ändern des Master-Benutzerpassworts	Ja	None	Ändern einer Amazon RDS DB-Instance
Umbenennen einer DB-Instance	Ja	None	Ändern einer Amazon RDS DB-Instance
Neustarten einer DB-Instance	Ja	None	Neustarten einer DB-Instance
Anhalten einer DB-Instance	Ja	None	Eine Amazon RDS-DB-Instance temporär stoppen
Starten einer DB-Instance	Ja	None	Starten einer Amazon RDS DB-Instance, die zuvor gestoppt wurde

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Multi-AZ-Bereitstellungen	Ja	Multi-AZ-Bereitstellungen werden auf My- SQL und Postgre-DB-Instances unterstützt. SQL Multi-AZ-Bereitstellungen unterstützen Direct Routing () nicht. VPC DVR	Erstellen von DB-Instances für Amazon RDS in AWS Outposts Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS
DB-Parametergruppen	Ja	None	Parametergruppen für Amazon RDS
Read Replicas	Ja	Read Replicas werden für My SQL - und SQL Postgre-DB-Instances unterstützt. Read Replicas unterstützen Direct VPC Routing () nicht. DVR	Read Replicas für Amazon RDS erstellen auf AWS Outposts
Verschlüsselung im Ruhezustand	Ja	RDSon Outposts unterstützt keine unverschlüsselten DB-Instances.	Verschlüsseln von Amazon RDS-Ressourcen
AWS Identity and Access Management (IAM) Datenbankauthentifizierung	Nein	None	IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL
Zuordnen einer IAM Rolle zu einer DB-Instance	Nein	None	add-role-to-dbBefehl - instance AWS CLI AddRoleToDBInstanceRDSAPIBetrieb

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Kerberos-Authentifizierung	Nein	None	Kerberos-Authentifizierung
RDSAmazon-Ressourcen taggen	Ja	None	Verschlagwortung von Amazon-Ressourcen RDS
Optionsgruppen	Ja	None	Arbeiten mit Optionsgruppen
Ändern des Wartungsfensters	Ja	None	Warten einer DB-Instance
Automatischer Unterversion-Upgrade	Ja	None	Automatisches Upgraden der Engine-Unterversion
Ändern des Backup-Fensters	Ja	None	Einführung in Backups Ändern einer Amazon RDS DB-Instance
Ändern der DB-Instance-Klasse	Ja	None	Ändern einer Amazon RDS DB-Instance
Ändern des zugewiesenen Speichers	Ja	None	Ändern einer Amazon RDS DB-Instance
Automatische Speicherskalierung	Ja	None	Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Manuelle und automatische Snapshots der DB-Instance	Ja	Sie können automatisierte Backups und manuelle Snapshots in Ihrer AWS-Region speichern. Sie können sie auch lokal auf Ihrem Outpost speichern. Lokale Backups werden auf My SQL - und SQL Postgre-DB-Instances unterstützt. Um Backups in Ihrem Außenposten zu speichern, stellen Sie sicher, dass Sie Amazon S3 on Outposts konfiguriert haben. Lokale Backups werden für Multi-AZ-Instance-Bereitstellungen nicht unterstützt.	Erstellen von DB-Instances für Amazon RDS in AWS Outposts Amazon S3 in Outposts Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS
Wiederherstellen aus einem DB-Snapshot	Ja	Sie können automatisierte Backups und manuelle Snapshots für die wiederhergestellte DB-Instance in der übergeordneten AWS-Region oder lokal in Ihrem Outpost speichern.	Überlegungen zum Wiederherstellen von DB-Instances in Amazon RDS on AWS Outposts Wiederherstellung auf einer DB-Instance
Wiederherstellen einer DB-Instance aus Amazon S3	Nein	None	Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Exportieren von Snapshot-Daten nach Amazon S3	Nein	None	Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS
Point-in-time Wiederherstellung	Ja	Sie können mit einer Ausnahme automatisierte Backups und manuelle Snapshots für die wiederhergestellte DB-Instance in der übergeordneten AWS-Region oder lokal in Ihrem Outpost speichern.	Überlegungen zum Wiederherstellen von DB-Instances in Amazon RDS on AWS Outposts Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS
„Enhanced Monitoring“ (Verbesserte Überwachung)	Nein	None	Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“ (Erweiterte Überwachung)
CloudWatch Amazon-Überwachung	Ja	Sie können die gleichen Metriken anzeigen, die für Ihre Datenbanken in der AWS-Region verfügbar sind.	Überwachung von Amazon RDS mit Amazon CloudWatch
CloudWatch Datenbank-Engine-Protokolle in Logs veröffentlichen	Ja	None	Datenbankprotokolle in Amazon CloudWatch Logs veröffentlichen

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Ereignisbenachrichtigung	Ja	None	Mit RDS Amazon-Event-Benachrichtigungen arbeiten
RDSPerformance Insights von Amazon	Nein	None	Überwachung mit Performance Insights auf Amazon RDS
Anzeigen oder Herunterladen von Datenbankprotokollen	Nein	RDSon Outposts unterstützt nicht das Anzeigen von Datenbankprotokollen mit der Konsole oder das Beschreiben von Datenbankprotokollen mit dem AWS CLI oder RDSAPI. RDSon Outposts unterstützt nicht das Herunterladen von Datenbankprotokollen über die Konsole oder das Herunterladen von Datenbankprotokollen mit dem AWS CLI oder RDSAPI.	Überwachung von Amazon RDS Amazon
RDSAmazon-Proxy	Nein	None	Amazon RDS-Proxy
Gespeicherte Prozeduren für Amazon RDS for My SQL	Ja	None	RDSals Referenz für meine SQL gespeicherte Prozedur

Funktionsmerkmal	Unterstützt	Hinweise	Weitere Informationen
Replikation mit externen Datenbanken für RDS for My SQL	Nein	None	Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance
Systemeigene Sicherung und Wiederherstellung für Amazon RDS für Microsoft SQL Server	Ja	None	Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung

Unterstützte DB-Instance-Klassen für Amazon RDS auf AWS Outposts

Amazon RDS on AWS Outposts unterstützt die folgenden DB-Instance-Klassen:

- Universelle DB-Instance-Klassen
  - db.m7i.48xlarge
  - db.m7i.24x groß
  - db.m7i.16xlarge
  - db.m7i.12xlarge
  - db.m7i.8 x groß
  - db.m7i.4x groß
  - db.m7i.2xgroß
  - db.m7i.xlarge
  - db.m7i.groß
  - db.m5.24xlarge
  - db.m5.16xlarge

- db.m5.12xlarge
- db.m5.8xlarge
- db.m5.4xlarge
- db.m5.2xlarge
- db.m5.xlarge
- db.m5.large

- Arbeitsspeicheroptimierte DB-Instance-Klassen
  - db.r7i.48x groß
  - db.r7i.24x groß
  - db.r7i.16xlarge
  - db.r7i.12xlarge
  - db.r7i.8 x groß
  - db.r7i.4x groß
  - db.r7i.2xgroß
  - db.r7i.xlarge
  - db.r7i.groß
  - db.r5.24xlarge
  - db.r5.16xlarge
  - db.r5.12xlarge
  - db.r5.8xlarge
  - db.r5.4xlarge
  - db.r5.2xlarge
  - db.r5.xlarge
  - db.r5.large

Je nachdem, wie Sie Ihren Outpost konfiguriert haben, stehen Ihnen möglicherweise nicht alle diese Klassen zur Verfügung. Wenn Sie beispielsweise die db.r5-Klassen nicht für Ihren Outpost gekauft haben, können Sie sie nicht mit RDS on Outposts verwenden.

Für RDS unter Outposts-DB-Instance-Klassen wird nur universeller SSD-Speicher unterstützt. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Amazon RDS verwaltet Wartung und Recovery für Ihre DB-Instances und benötigt dafür aktive Kapazität auf dem Outpost. Wir empfehlen, dass Sie EC2 N+1-Instances für jede DB-Instance-Klasse in Ihren Produktionsumgebungen konfigurieren. RDS on Outposts kann die zusätzliche Kapazität dieser EC2 Instances für Wartungs- und Reparaturarbeiten nutzen. Wenn Ihre Produktionsumgebungen beispielsweise über 3 db.m5.large- und 5 db.r5.xlarge DB-Instance-Klassen verfügen, empfehlen wir, dass sie über mindestens 4 m5.large-Instances und 6 r5.xlarge-Instances verfügen. EC2 EC2 Weitere Informationen finden Sie unter Ausfallsicherheit in AWS Outposts im AWS Outposts -Benutzerhandbuch.

Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts

Amazon RDS in AWS Outposts verwendet Informationen, die Sie über Ihr lokales Netzwerk angeben, um einen Adresspool zu erstellen. Dieser Pool wird als kundeneigener IP-Adresspool (CoIP-Pool) bezeichnet. Kundeneigene IP-Adressen (CoIPs) bieten lokale oder externe Konnektivität zu Ressourcen in Ihren Outpost-Subnetzen über Ihr lokales Netzwerk. Weitere Informationen zu CoIPs finden Sie unter Kundeneigene IP-Adressen im AWS Outposts-Benutzerhandbuch.

Jede RDS unter Outposts-DB-Instance hat eine private IP-Adresse für den Datenverkehr innerhalb ihrer Virtual Private Cloud (VPC). Diese private IP-Adresse ist nicht öffentlich zugänglich. Mit der Option Public (Öffentlich) können Sie festlegen, ob die DB-Instance neben der privaten IP-Adresse auch eine öffentliche IP-Adresse hat. Die Verwendung der öffentlichen IP-Adresse für Verbindungen leitet sie über das Internet und kann in einigen Fällen zu hohen Latenzen führen.

Anstatt diese privaten und öffentlichen IP-Adressen zu verwenden, unterstützt RDS on Outposts die Verwendung von CoIPs für DB-Instances über ihre Subnetze. Wenn Sie eine CoIP für eine RDS-on-Outposts-DB-Instance verwenden, stellen Sie eine Verbindung mit der DB-Instance über den DB-Instance-Endpunkt her. RDS on Outposts verwendet die CoIP automatisch für alle Verbindungen innerhalb und außerhalb der VPC.

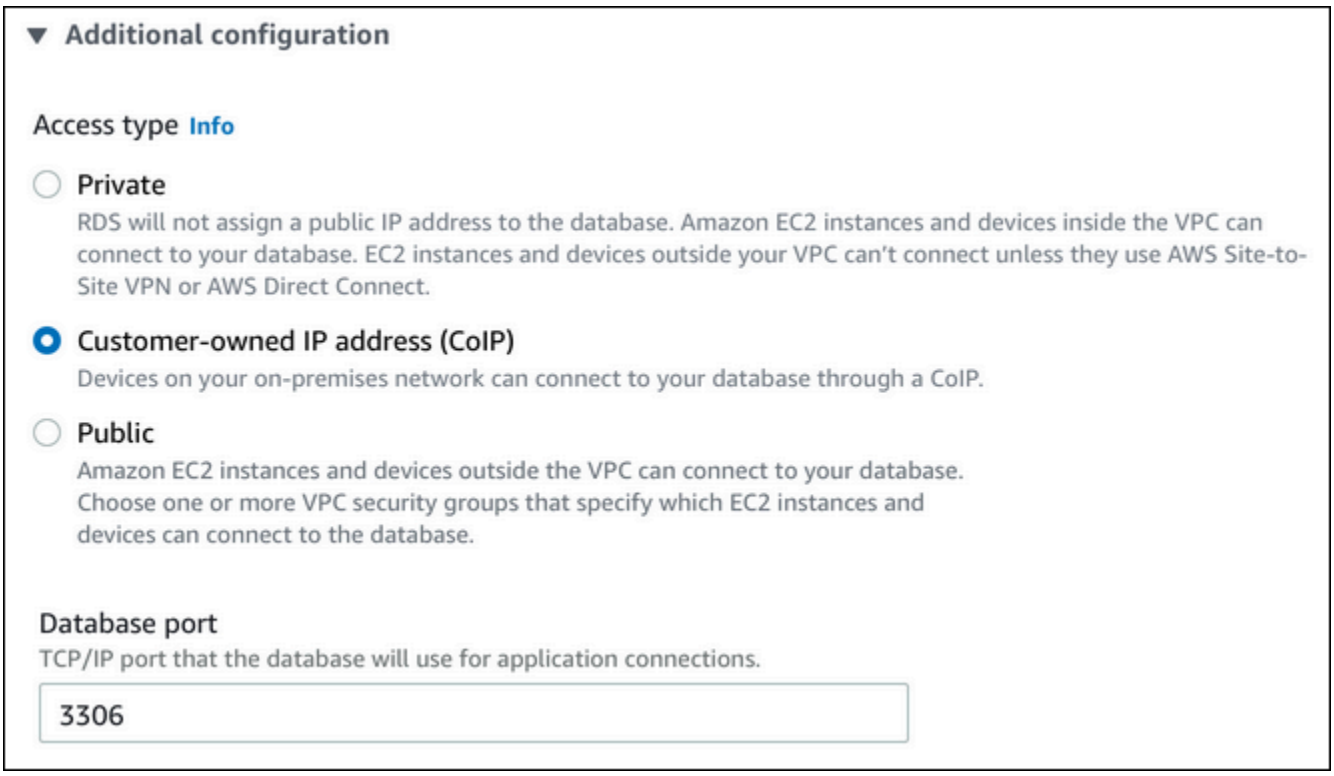
CoIPs können die folgenden Vorteile für RDS unter Outposts-DB-Instances bieten:

- Geringere Verbindungslatenz
- Erhöhte Sicherheit

Verwendung von CoIPs

Sie können eine CoIP für eine RDS-on-Outposts-DB-Instance mithilfe der AWS Management Console, der AWS CLI oder der RDS-API aktivieren oder deaktivieren:

- Verwenden Sie mit der AWS Management Console die Einstellung Customer-owned IP address (CoIP) (Kundeneigene IP-Adresse (CoIP)) unter Access type (Zugriffstyp), um CoIPs zu verwenden. Wählen Sie eine der anderen Einstellungen aus, um sie auszuschalten.



- Verwenden Sie mit der AWS CLI die Option `--enable-customer-owned-ip | --no-enable-customer-owned-ip`.
- Verwenden Sie mit der RDS-API den Parameter `EnableCustomerOwnedIp`.

Sie können CoIPs aktivieren oder deaktivieren, wenn Sie eine der folgenden Aktionen ausführen:

- Erstellen einer DB-Instance.

  Weitere Informationen finden Sie unter Erstellen von DB-Instances für Amazon RDS in AWS Outposts.
- Ändern einer DB-Instance

  Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Erstellen eines Lesereplikats

  Weitere Informationen finden Sie unter Read Replicas für Amazon RDS erstellen auf AWS Outposts.
- DB-Instance aus einem Snapshot wiederherstellen

  Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.

- Wiederherstellen einer DB-Instance zu einer bestimmten Zeit

  Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

> **Note**
>
> In einigen Fällen können Sie CoIPs für eine DB-Instance aktivieren, ohne dass Amazon RDS der DB-Instance eine CoIP zuweisen kann. In solchen Fällen wird der Status der DB-Instance in incompatible-network (Inkomaptibles Netzwerk) geändert. Weitere Informationen zum DB-Instance-Status finden Sie unter Anzeigen von Amazon RDS DB-Instance-Status.

Einschränkungen

Die folgenden Einschränkungen gelten für die CoIP-Unterstützung für RDS unter Outposts-DB-Instances:

- Wenn eine CoIP für eine DB-Instance verwendet wird, stellen Sie sicher, dass die öffentliche Zugänglichkeit für die DB-Instance deaktiviert ist.
- Stellen Sie sicher, dass die eingehenden Regeln für Ihre VPC-Sicherheitsgruppen den CoIP-Adressbereich (CIDR-Block) enthalten. Weitere Informationen zum Einrichten von Sicherheitsgruppen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.
- Sie können eine CoIP nicht aus einem CoIP-Pool einer DB-Instance zuweisen. Wenn Sie eine CoIP für eine DB-Instance verwenden, weist Amazon RDS der DB-Instance automatisch eine CoIP aus einem CoIP-Pool zu.
- Sie müssen das AWS-Konto verwenden, das die Outpost-Ressourcen (Eigentümer) besitzt, oder die folgenden Ressourcen mit anderen AWS-Konten (Verbrauchern) in derselben Organisation teilen:
  - Outpost
  - Die Routing-Tabelle des lokalen Gateways (LGW) für die VPC der DB-Instance
  - Der CoIP-Pool oder die Pools für die LGW-Routing-Tabelle

  Weitere Informationen finden Sie unter Arbeiten mit freigegebenen AWS Outposts-Ressourcen im AWS Outposts-Benutzerhandbuch.

Arbeiten mit Multi-AZ-Bereitstellungen für Amazon RDS in AWS Outposts

Für Multi-AZ-Bereitstellungen erstellt Amazon RDS eine primäre DB-Instance auf einem AWS Outpost. RDS repliziert die Daten synchron auf eine Standby-DB-Instance auf einem anderen Outpost.

Multi-AZ-Bereitstellungen in AWS Outposts funktionieren wie Multi-AZ-Bereitstellungen in AWS-Regionen, jedoch mit folgenden Unterschieden:

- Sie benötigen eine lokale Verbindung zwischen zwei oder mehr Outposts.
- Sie benötigen Customer-owned IP Pools (CoIP, kundeneigene IP-Pools). Weitere Informationen finden Sie unter Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts.
- Die Replikation läuft in Ihrem lokalen Netzwerk.

Multi-AZ in AWS Outposts ist für alle unterstützten Versionen von MySQL und PostgreSQL in RDS on Outposts verfügbar. Lokale Backups werden für Multi-AZ-Bereitstellungen nicht unterstützt. Weitere Informationen finden Sie unter Erstellen von DB-Instances für Amazon RDS in AWS Outposts.

Arbeiten mit dem Modell der geteilten Verantwortung

Obwohl AWS wirtschaftlich angemessene Anstrengungen unternimmt, um DB-Instances bereitzustellen, die für hohe Verfügbarkeit konfiguriert sind, wird für die Verfügbarkeit ein Modell der geteilten Verantwortung verwendet. Die Fähigkeit von RDS on Outposts zum Failover und Reparieren von DB-Instances erfordert, dass jeder Ihrer Outposts mit seiner AWS-Region verbunden ist.

RDS on Outposts erfordert auch Konnektivität zwischen dem Outpost, der die primäre DB-Instance hostet, und dem Outpost, der die Standby-DB-Instance für die synchrone Replikation hostet. Jede Auswirkung auf diese Verbindung kann verhindern, dass RDS on Outposts ein Failover durchführt.

Aufgrund der synchronen Datenreplikation kann es zu erhöhten Latenzen für eine standardmäßige Bereitstellung einer DB-Instance kommen. Die Bandbreite und Latenz der Verbindung zwischen dem Outpost, der die primäre DB-Instance hostet, und dem Outpost, der die Standby-DB-Instance hostet, wirken sich direkt auf Latenzen aus. Weitere Informationen finden Sie unter Voraussetzungen.

Verbessern der Verfügbarkeit

Wir empfehlen die folgenden Maßnahmen zur Verbesserung der Verfügbarkeit:

- Weisen Sie Ihren unternehmenskritischen Anwendungen genügend zusätzliche Kapazität zu, um Wiederherstellung und Failover zu ermöglichen, wenn ein zugrunde liegendes Hostproblem vorliegt. Dies gilt für alle Outposts, die Subnetze in Ihrer DB-Subnetzgruppe enthalten. Weitere Informationen finden Sie unter Ausfallsicherheit in AWS Outposts.
- Stellen Sie redundante Netzwerkkonnektivität für Ihre Outposts bereit.
- Verwenden Sie mehr als zwei Outposts. Mit mehr als zwei Outposts kann Amazon RDS eine DB-Instance wiederherstellen. RDS führt diese Wiederherstellung durch, indem es die DB-Instance auf einen anderen Outpost verschiebt, wenn der aktuelle Outpost ausfällt.
- Stellen Sie zwei Stromquellen und redundante Netzwerkkonnektivität für Ihren Outpost bereit.

Wir empfehlen Folgendes für Ihre lokalen Netzwerke:

- Die Latenz der Round Trip Time (RTT, Umlaufzeit) zwischen dem Outpost, der Ihre primäre DB-Instance hostet, und dem Outpost, der Ihre Standby-DB-Instance hostet, wirkt sich direkt auf die Schreiblatenz aus. Halten Sie die RTT-Latenz zwischen den AWS-Outposts im niedrigen einstelligen Millisekundenbereich. Wir empfehlen nicht mehr als 5 Millisekunden, Ihre Anforderungen können jedoch variieren.

  Die Nettoauswirkung auf die Netzwerklatenz finden Sie in den Amazon-CloudWatch-Metriken für `WriteLatency`. Weitere Informationen finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS.
- Die Verfügbarkeit der Verbindung zwischen den Outposts beeinflusst die Gesamtverfügbarkeit Ihrer DB-Instances. Sorgen Sie für redundante Netzwerkkonnektivität zwischen den Outposts.

Voraussetzungen

Für Multi-AZ-Bereitstellungen in RDS on Outposts gelten folgende Voraussetzungen:

- Richten Sie mindestens zwei Outposts ein, die über lokale Verbindungen verbunden und verschiedenen Availability Zones in einer AWS-Region angefügt sind.
- Stellen Sie sicher, dass Ihre DB-Subnetzgruppen Folgendes enthalten:
  - Mindestens zwei Subnetze in mindestens zwei Availability Zones in einer bestimmten AWS-Region.
  - Subnetze nur in Outposts.

  - Mindestens zwei Subnetze in mindestens zwei Outposts innerhalb derselben Virtual Private Cloud (VPC).
- Verknüpfen Sie die VPC Ihrer DB-Instance mit allen Ihren lokalen Gateway-Routing-Tabellen. Diese Verknüpfung ist notwendig, da die Replikation über Ihr lokales Netzwerk unter Verwendung der lokalen Gateways Ihrer Outposts läuft.

  Angenommen, Ihre VPC enthält Subnet-A in Outpost-A und Subnet-B in Outpost-B. Outpost-A verwendet LocalGateway-A (LGW-A) und Outpost-B verwendet LocalGateway-B (LGW-B). LGW-A verfügt über RouteTable-A und LGW-B über RouteTable-B. Sie sollten sowohl RouteTable-A als auch RouteTable-B für den Replikationsdatenverkehr verwenden. Verknüpfen Sie dazu Ihre VPC sowohl mit RouteTable-A als auch mit RouteTable-B.

  Weitere Informationen zum Erstellen einer Verknüpfung finden Sie im AWS CLI-Befehl create-local-gateway-route-table-vpc-association von Amazon-EC2.
- Stellen Sie sicher, dass Ihre Outposts das kundeneigene IP (CoIP)-Routing verwenden. Jede Routing-Tabelle muss außerdem jeweils mindestens einen Adresspool haben. Amazon RDS weist jeweils eine zusätzliche IP-Adresse für die primäre und die Standby-DB-Instance für die Datensynchronisation zu.
- Stellen Sie sicher, dass das AWS-Konto, das die RDS-DB-Instances besitzt, Eigentümer der lokalen Gateway-Routing-Tabellen und CoIP-Pools ist. Oder stellen Sie sicher, dass es Teil einer Resource-Access-Manager-Freigabe mit Zugriff auf die lokalen Gateway-Routing-Tabellen und CoIP-Pools ist.
- Stellen Sie sicher, dass die IP-Adressen in Ihren CoIP-Pools von einem lokalen Outpost-Gateway an die anderen weitergeleitet werden können.
- Vergewissern Sie sich, dass die CIDR-Blöcke der VPC (z. B. 10.0.0.0/4) und die CIDR-Blöcke Ihres CoIP-Pools keine IP-Adressen der Klasse E enthalten (240.0.0.0/4). RDS verwendet diese IP-Adressen intern.
- Stellen Sie sicher, dass Sie ausgehenden und damit verbundenen eingehenden Datenverkehr korrekt eingerichtet haben.

  RDS on Outposts stellt eine Virtual Private Network (VPN)-Verbindung zwischen der primären und der Standby-DB-Instance her. Damit dies ordnungsgemäß funktioniert, muss Ihr lokales Netzwerk ausgehenden und damit verbundenen eingehenden Datenverkehr für Internet Security Association and Key Management Protocol (ISAKMP) zulassen. Dazu verwendet es User Datagram Protocol (UDP) Port 500 und IP Security (IPSec) Network Address Translation Traversal (NAT-T) und den UDP-Port 4500.

Weitere Informationen zu CoIPs finden Sie unter Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts in diesem Leitfaden und unter Kundeneigene IP-Adressen im AWS Outposts-Benutzerhandbuch.

Arbeiten mit API-Operationen für Amazon-EC2-Berechtigungen

Unabhängig davon, ob Sie CoIPs für Ihre DB-Instance in AWS Outposts verwenden, benötigt RDS Zugriff auf Ihre CoIP-Pool-Ressourcen. RDS kann die folgenden EC2-Berechtigungs-API-Operationen für COIPs in Ihrem Namen für Multi-AZ-Bereitstellungen aufrufen:

- `CreateCoipPoolPermission` – Wenn Sie eine Multi-AZ-DB-Instance in RDS on Outposts erstellen
- `DeleteCoipPoolPermission` – Wenn Sie eine Multi-AZ-DB-Instance in RDS on Outposts löschen

Diese API-Operationen gewähren internen RDS-Konten die Berechtigung oder heben die Berechtigung auf, Elastic-IP-Adressen aus dem in der Berechtigung angegebenen CoIP-Pool zuzuweisen. Sie können diese IP-Adressen mit der API-Operation `DescribeCoipPoolUsage` anzeigen. Weitere Informationen zu CoIPs finden Sie unter Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts und unter Kundeneigene IP-Adressen im AWS Outposts-Benutzerhandbuch.

RDS kann die folgenden EC2-Berechtigungs-API-Operationen für lokale Gateway-Routing-Tabellen in Ihrem Namen für Multi-AZ-Bereitstellungen aufrufen:

- `CreateLocalGatewayRouteTablePermission` – Wenn Sie eine Multi-AZ-DB-Instance in RDS on Outposts erstellen
- `DeleteLocalGatewayRouteTablePermission` – Wenn Sie eine Multi-AZ-DB-Instance in RDS on Outposts löschen

Diese API-Operationen gewähren internen RDS-Konten die Berechtigung, Ihren lokalen Gateway-Routing-Tabellen interne RDS-VPCs zuzuordnen, oder heben diese Berechtigung auf. Sie können diese Zuordnungen zwischen Routing-Tabelle und VPC mit den API-Operationen `DescribeLocalGatewayRouteTableVpcAssociations` anzeigen.

Erstellen von DB-Instances für Amazon RDS in AWS Outposts

Das Erstellen einer Amazon RDS in AWS Outposts-DB-Instance ist dem Erstellen einer Amazon-RDS-DB-Instance in der AWS Cloud ähnlich. Sie müssen jedoch eine DB-Subnetzgruppe angeben, die mit Ihrem Outpost verknüpft ist.

Eine auf dem Amazon-VPC-Service basierende Virtual Private Cloud (VPC) kann alle Availability Zones in einer AWS-Region umfassen. Sie können jede VPC in der AWS-Region auf Ihren Outpost ausweiten, indem Sie ein Outpost-Subnetz hinzufügen. Um ein Outpost-Subnetz zu einer VPC hinzuzufügen, geben Sie beim Erstellen des Subnetzes den Amazon-Ressourcennamen (ARN) des Outpost an.

Bevor Sie eine RDS unter Outposts-DB-Instance erstellen, können Sie eine DB-Subnetzgruppe erstellen, die ein einzelnes, mit Ihrem Outpost verknüpftes Subnetz enthält. Wenn Sie eine RDS unter Outposts-DB-Instance erstellen, geben Sie diese DB-Subnetzgruppe an. Sie können optional auch eine neue DB-Subnetzgruppe erstellen, wenn Sie Ihre DB-Instance erstellen.

Weitere Informationen zur Konfiguration von AWS Outposts finden Sie im AWS Outposts Benutzerhandbuch.

Konsole

Erstellen einer DB-Subnetzgruppe

Erstellen Sie eine DB-Subnetzgruppe mit einem einzelnen, mit Ihrem Outpost verknüpften Subnetz.

Sie können optional auch eine neue DB-Subnetzgruppe erstellen, wenn Sie Ihre DB-Instance erstellen. Wenn Sie dies möchten, überspringen Sie diesen Vorgang.

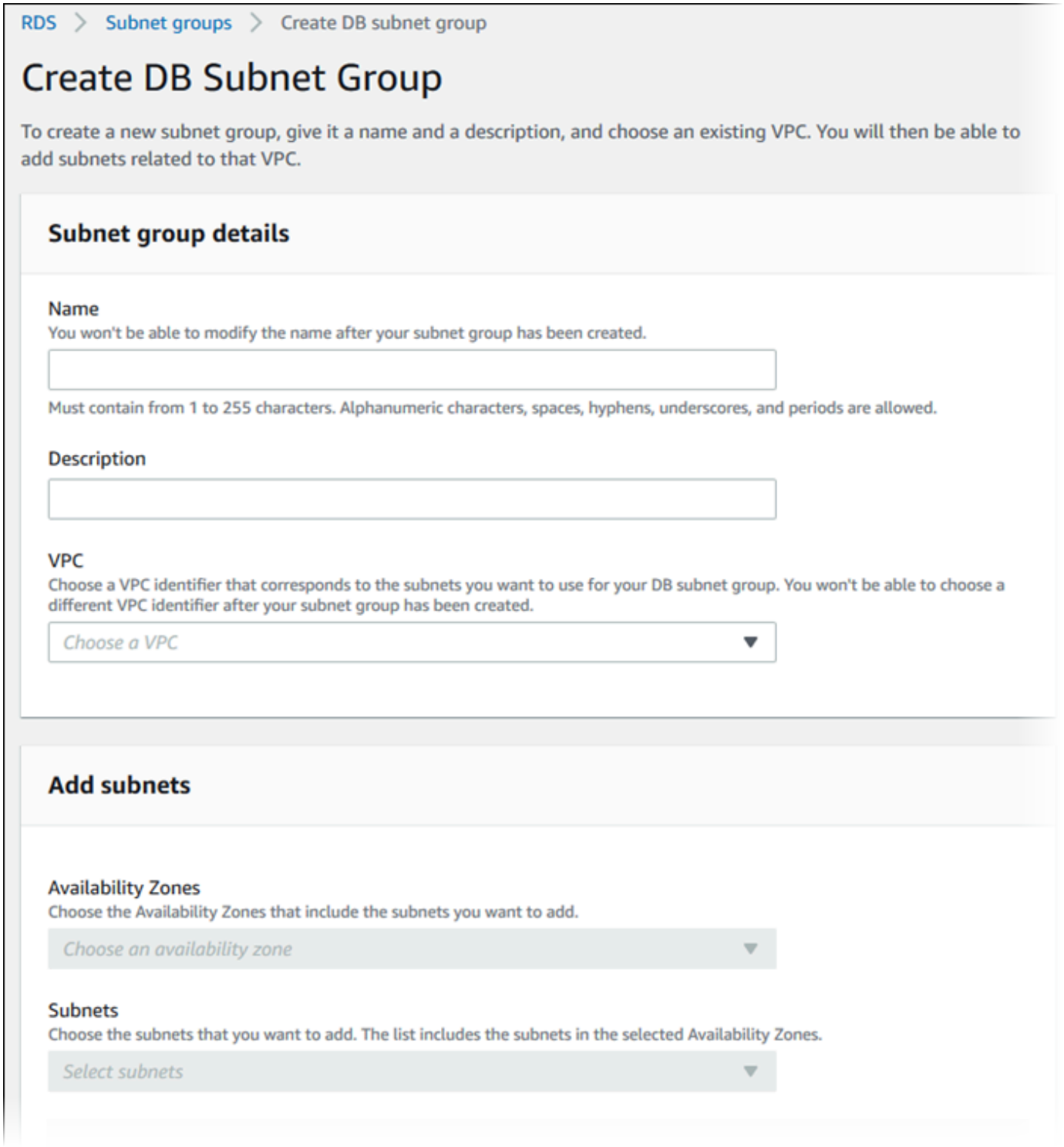
> **Note**
>
> Um eine DB-Subnetzgruppe für die AWS Cloud Cloud zu erstellen, müssen Sie mindestens zwei Subnetzte angeben.

Eine DB-Susnetzgruppe für Ihren Outpost erstellen Sie wie folgt:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie in der oberen rechten Ecke der Amazon-RDS-Konsole die AWS-Region aus, in der Sie die DB-Subnetzgruppe erstellen möchten.
3. Wählen Sie Subnet groups (Subnetzgruppen) und Create DB Subnet Group (DB-Subnetzgruppe erstellen) aus.

   Die Seite Create DB subnet group (DB-Subnetzgruppe erstellen) wird angezeigt.



4. Geben Sie im Feld Name den Namen Ihrer DB-Subnetzgruppe ein.
5. Wählen Sie für Description (Beschreibung) eine Beschreibung für die DB-Subnetzgruppe.
6. Wählen Sie für VPC die VPC, die Sie die DB-Subnetzgruppe erstellen.
7. Wählen Sie für Availability Zones die Availability Zone für Ihren Outpost aus.

8. Wählen Sie für Subnetze das Subnetz aus, das von RDS unter Outposts verwendet werden soll.
9. Wählen Sie Create (Erstellen) aus, um die DB-Subnetzgruppe zu erstellen.

Erstellen einer RDS unter Outposts-DB-Instance-Seite.

Erstellen Sie die DB-Instance und wählen Sie den Outpost für Ihre DB-Instance aus.

So erstellen Sie eine RDS unter Outposts-DB-Instance über die Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon-RDS-Konsole die AWS-Region aus, der der Outpost, auf dem Sie die DB-Instance erstellen möchten, angefügt ist.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.

   Die AWS Management Console erkennt verfügbare Outposts, die Sie konfiguriert haben, und zeigt die Option On-Premise im Abschnitt Speicherort der Datenbank an.

   > **Note**
   >
   > Wenn Sie keine Outposts konfiguriert haben, wird entweder der Abschnitt Speicherort der Datenbank nicht angezeigt, oder ist die RDS unter Outposts im Abschnitt Wählen Sie eine On-Premise-Erstellungsmethode nicht verfügbar.

5. Für Speicherort der Datenbank, wählen Sie Lokal aus.
6. Wählen sie für On-premises creation method (On-Premise-Erstellungsmethode) RDS on Outposts (RDS unter Outposts).
7. Geben Sie die Einstellungen für Outposts Konnektivität ein. Diese Einstellungen gelten für den Outpost, der die VPC verwendet, in der sich die DB-Subnetzgruppe für Ihre DB-Instance befindet. Ihre VPC hier muss auf dem Amazon VPC-Service basieren.

   a. Wählen Sie für Virtual Private Cloud (VPC) die VPC, die die DB-Subnetzgruppe für die DB-Instance enthält.
   b. Für VPC Security Group (VPC-Sicherheitsgruppe) wählen Sie die Amazon VPC-Sicherheitsgruppe für Ihre DB-Instance aus.
   c. Für die Subnetzgruppe verwenden Sie die DB-Subnetzgruppe für Ihre DB-Instance.

Sie können eine vorhandene DB-Susnetzgruppe auswählen, die mit dem Outpost verknüpft ist, z. B. wenn Sie die Prozedur in Erstellen einer DB-Subnetzgruppe ausgeführt haben.

Sie können auch eine neue DB-Subnetzgruppe für den Outpost erstellen.

8. Für Multi-AZ deployment (Multi-AZ-Bereitstellung) wählen Sie Create a standby instance (recommended for production usage) (Standby-Instance erstellen (empfohlen für die Produktion)) aus, um eine Standby-DB-Instance in einem anderen Outpost zu erstellen.

> **Note**
>
> Diese Option ist für Microsoft SQL Server nicht verfügbar.
> Wenn Sie sich entscheiden, eine Multi-AZ-Bereitstellung zu erstellen, können Sie keine Backups in Ihrem Outpost speichern.

9. Gehen Sie unter Backup wie folgt vor:

   a. Wählen Sie für Backup-Ziel eine der nachstehenden Optionen aus:

   - AWS Cloud, um automatisierte Backups und manuelle Snapshots in der übergeordneten AWS-Region zu speichern.
   - Outposts (lokal) um lokale Backups zu erstellen.

   > **Note**
   >
   > Wenn Sie Backups in Ihrem Outpost speichern möchten, muss Ihr Outpost über Amazon-S3-Fähigkeit verfügen. Weitere Informationen finden Sie unter Verwenden von Amazon S3 on Outposts.
   > Lokale Backups werden für Multi-AZ-Bereitstellungen oder Lesereplikate nicht unterstützt.

   b. Wählen Sie Automatische Backups aktivieren, um point-in-time Snapshots Ihrer DB-Instance zu erstellen.

   Wenn Sie automatische Backups aktivieren, können Sie Werte für Backup retention period (Aufbewahrungszeitraum für Backups) und Backup window (Fenster Backup) auswählen oder die Standardwerte beibehalten.

10. Geben Sie nach Ihrem Bedarf weitere DB-Instance-Einstellungen ein.

Informationen zu den einzelnen Einstellungen beim Erstellen einer DB-Instance finden Sie unter Einstellungen für DB-Instances.

11. Wählen Sie Create database (Datenbank erstellen) aus.

    Der Bereich Databases (Datenbanken) wird angezeigt. Ein Banner weist Sie darauf hin, dass Ihre DB-Instance erstellt wird, und zeigt die Schaltfläche Details zu den Anmeldeinformationen anzeigen an.

Anzeigen von DB-Instance-Details

Nachdem Sie die DB-Instance erstellt haben, können Sie Anmeldeinformationen und andere Details dafür anzeigen.

So zeigen Sie DB-Instance-Details an

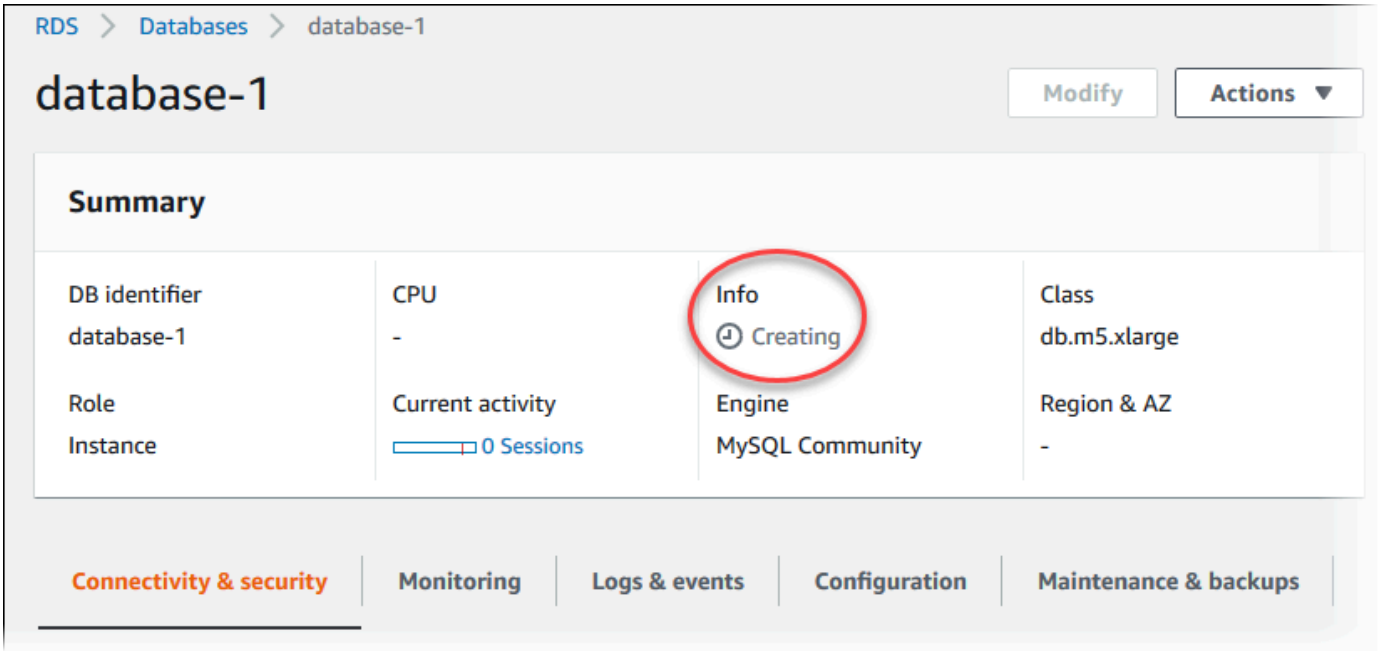
1. Um den Namen und das Passwort des Hauptbenutzers für die DB-Instance anzuzeigen, wählen Sie View credential details (Details zu Anmeldeinformationen anzeigen) in Datenbanken aus.

   Sie können unter Verwendung dieser Anmeldeinformationen eine Verbindung zur DB-Instance als Masterbenutzer herstellen.

   > **Important**
   >
   > Sie können dieses Passwort für den Hauptbenutzer nicht erneut anzeigen. Wenn Sie es nicht notieren, müssen Sie es möglicherweise ändern. Um das Passwort für den Master-Benutzer zu ändern, nachdem die DB-Instance verfügbar wurde, ändern Sie die DB-Instance entsprechend. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

2. Wählen Sie in Databases (Datenbanken) den Namen der neuen DB-Instance aus.

   In der RDS-Konsole werden die Details der neuen DB-Instance angezeigt. Die DB-Instance wird mit dem Status Creating (Wird erstellt) angezeigt, bis die Erstellung abgeschlossen ist und sie verwendet werden kann. Wenn sich der Status in Available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Klasse der DB-Instance und dem dieser zugeteilten Speicher kann es einige Minuten dauern, bis die neue DB-Instance verfügbar ist.



Wenn die DB-Instance verfügbar ist, können Sie diese auf dieselbe Weise verwalten, wie Sie RDS-DB-Instances in der AWS Cloud verwalten.

AWS CLI

Bevor Sie mit dem AWS CLI eine neue DB-Instance in einem Outpost erstellen, erstellen Sie zunächst eine DB-Subnetzgruppe zur Verwendung durch RDS auf Outposts.

Eine DB-Susnetzgruppe für Ihren Outpost erstellen Sie wie folgt:

- Verwenden Sie den create-db-subnet-group-Befehl. Geben Sie für `--subnet-ids` die Subnetzgruppe im Outpost an, die von RDS unter Outposts verwendet werden soll.

  Für Linux, macOSoder Unix:

  ```
  aws rds create-db-subnet-group \
      --db-subnet-group-name myoutpostdbsubnetgr \
      --db-subnet-group-description "DB subnet group for RDS on Outposts" \
      --subnet-ids subnet-abc123
  ```

  Windows:

  ```
  aws rds create-db-subnet-group ^
      --db-subnet-group-name myoutpostdbsubnetgr ^
  ```

```
    --db-subnet-group-description "DB subnet group for RDS on Outposts" ^
    --subnet-ids subnet-abc123
```

So erstellen Sie eine RDS unter Outposts-DB-Instance über die AWS CLI

- Verwenden Sie den create-db-instance-Befehl. Geben Sie eine Availability Zone für den Outpost, eine Amazon VPC-Sicherheitsgruppe, die dem Outpost zugeordnet ist, und die DB-Subnetzgruppe an, die Sie für den Outpost erstellt haben. Sie können die folgenden Optionen einschließen:
  - `--db-instance-identifier`
  - `--db-instance-class`
  - `--engine` Die Datenbank-Engine. Verwenden Sie einen der folgenden Werte:
    - MySQL — Geben Sie `mysql` ein.
    - PostgreSQL — Geben Sie `postgres` ein.
    - Microsoft SQL Server – Geben Sie `sqlserver-ee`, `sqlserver-se` oder `sqlserver-web` an.
  - `--availability-zone`
  - `--vpc-security-group-ids`
  - `--db-subnet-group-name`
  - `--allocated-storage`
  - `--max-allocated-storage`
  - `--master-username`
  - `--master-user-password`
  - `--multi-az | --no-multi-az` – (Optional) Ob eine Standby-DB-Instance in einer anderen Availability Zone erstellt werden soll. Der Standardwert ist `--no-multi-az`.

    Die `--multi-az` ist für SQL Server nicht verfügbar.
  - `--backup-retention-period`
  - `--backup-target` – (Optional) Wo werden automatisierte Backups und manuelle Snapshots gespeichert. Verwenden Sie einen der folgenden Werte:
    - `outposts` – Speichern Sie sie lokal auf Ihrem Outpost.
    - `region` – Speichern Sie sie in der übergeordneten AWS-Region. Dies ist der Standardwert.

Wenn Sie die Option `--multi-az` verwenden, können Sie `outposts` nicht für `--backup-target` nutzen. Darüber hinaus kann die DB-Instance keine Lesereplikate enthalten, wenn Sie `outposts` als `--backup-target` verwenden.

- `--storage-encrypted`
- `--kms-key-id`

Example

Das folgende Beispiel erstellt eine MySQL-DB-Instance namens `myoutpostdbinstance` mit Backups, die auf Ihrem Outpost gespeichert sind.

Für Linux, macOSoder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myoutpostdbinstance \
    --engine-version 8.0.17 \
    --db-instance-class db.m5.large \
    --engine mysql \
    --availability-zone us-east-1d \
    --vpc-security-group-ids outpost-sg \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3 \
    --backup-target outposts \
    --storage-encrypted \
    --kms-key-id mykey
```

Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myoutpostdbinstance ^
    --engine-version 8.0.17 ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --availability-zone us-east-1d ^
    --vpc-security-group-ids outpost-sg ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
```

```
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3 ^
    --backup-target outposts ^
    --storage-encrypted ^
    --kms-key-id mykey
```

Informationen zu den einzelnen Einstellungen beim Erstellen einer DB-Instance finden Sie unter Einstellungen für DB-Instances.

RDS-API

Um eine neue DB-Instance in einem Outpost mit der RDS-API zu erstellen, erstellen Sie zunächst eine DB-Subnetzgruppe zur Verwendung durch RDS unter Outposts, indem Sie die CreateDBSubnetGroup-Operation aufrufen. Geben Sie für `SubnetIds` die Subnetzgruppe im Outpost an, die von RDS unter Outposts verwendet werden soll.

Rufen Sie als Nächstes die Operation CreateDBInstance mit den folgenden Parametern auf. Geben Sie eine Availability Zone für den Outpost, eine Amazon VPC-Sicherheitsgruppe, die dem Outpost zugeordnet ist, und die DB-Subnetzgruppe an, die Sie für den Outpost erstellt haben.

- `AllocatedStorage`
- `AvailabilityZone`
- `BackupRetentionPeriod`
- `BackupTarget`

  Wenn Sie eine Multi-AZ-Bereitstellung für eine DB-Instance erstellen, können Sie `outposts` nicht als `BackupTarget` verwenden. Darüber hinaus kann die DB-Instance keine Lesereplikate enthalten, wenn Sie `outposts` als `BackupTarget` verwenden.
- `DBInstanceClass`
- `DBInstanceIdentifier`
- `VpcSecurityGroupIds`
- `DBSubnetGroupName`
- `Engine`
- `EngineVersion`

- `MasterUsername`
- `MasterUserPassword`
- `MaxAllocatedStorage` (optional)
- `MultiAZ` (optional)
- `StorageEncrypted`
- `KmsKeyID`

Informationen zu den einzelnen Einstellungen beim Erstellen einer DB-Instance finden Sie unter Einstellungen für DB-Instances.

Read Replicas für Amazon RDS erstellen auf AWS Outposts

Amazon RDS on AWS Outposts verwendet die integrierten Replikationsfunktionen der MySQL- und PostgreSQL-DB-Engines, um eine Read Replica aus einer Quell-DB-Instance zu erstellen. Die Quell-DB-Instance wird zur primären DB-Instance. In der primären DB-Instance ausgeführte Updates werden asynchron in das Lesereplikat kopiert. Sie können die Arbeitslast für Ihre primären DB-Instance reduzieren, indem Sie Leseabfragen aus Ihren Anwendungen an das Lesereplikat weiterleiten. Mit Lesereplikaten können Sie die Kapazitätseinschränkungen einer einzelnen DB-Instance für leseintensive Datenbank-Workloads elastisch erweitern.

Wenn Sie ein Lesereplikat aus einer DB-Instance von RDS on Outposts erstellen, verwendet das Lesereplikat eine kundeneigene IP-Adresse (CoIP-Adresse). Weitere Informationen finden Sie unter Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts.

Für Lesereplikate in RDS on Outposts gelten folgende Einschränkungen:

- Sie können keine Lesereplikate für RDS für SQL Server auf DB-Instances von RDS on Outposts erstellen.
- Regionsübergreifende Lesereplikate werden in RDS on Outposts nicht unterstützt.
- Kaskadierende Lesereplikate werden in RDS on Outposts nicht unterstützt.
- Für die Quell-DB-Instance von RDS on Outposts kann es keine lokalen Backups geben. Das Backup-Ziel für die Quell-DB-Instance muss Ihre AWS-Region sein.
- Sie benötigen kundeneigene IP-Pools (CoIP-Pools). Weitere Informationen finden Sie unter Kundeneigene IP-Adressen für Amazon RDS on AWS Outposts.
- Read Replicas auf RDS on Outposts können nur in derselben Virtual Private Cloud (VPC) wie die Quell-DB-Instance erstellt werden.
- Read Replicas auf RDS on Outposts können sich auf demselben Outpost oder einem anderen Outpost in derselben VPC wie die Quell-DB-Instance befinden.
- Sie können keine Read Replicas für DB-Instances erstellen, die mit verschlüsselt sind. AWS KMS External Key Store (XKS)

Sie können mit der, oder RDS-API eine Read Replica aus einer RDS on Outposts DB-Instance erstellen. AWS Management Console AWS CLI Weitere Informationen über Lesereplikate finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Konsole

Zum Erstellen einer Read Replica aus einer Quell-DB-Instance

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie als Quelle für eine Read Replica verwenden möchten.
4. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.
5. Geben Sie unter DB instance identifier (DB-Instance-Kennung) einen Namen für das Lesereplikat ein.
6. Geben Sie die Einstellungen für Outposts Konnektivität ein. Diese Einstellungen gelten für den Outpost, der die Virtual Private Cloud (VPC) verwendet, in der sich die DB-Subnetzgruppe für Ihre DB-Instance befindet. Ihre VPC hier muss auf dem Amazon VPC-Service basieren.
7. Wählen Sie die DB instance class (DB-Instance-Klasse) aus. Wir empfehlen Ihnen, dieselbe oder eine größere DB-Instance-Klasse und denselben Speichertyp wie bei der Quell-DB-Instance für das Lesereplikat zu verwenden.
8. Für Multi-AZ deployment (Multi-AZ-Bereitstellung) wählen Sie Create a standby instance (recommended for production usage) (Standby-Instance erstellen (empfohlen für die Produktion)) aus, um eine Standby-DB-Instance in einer anderen Availability Zone zu erstellen.

   Das Erstellen Ihres Lesereplikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist.
9. (Optional) Legen Sie unter Connectivity (Konnektivität) Werte für Subnet Group (Subnetzgruppe) und Availability Zone fest.

   Wenn Sie Werte sowohl für Subnet group (Subnetzgruppe) als auch Availability Zone angeben, wird das Lesereplikat auf einem Outpost erstellt, der der Availability Zone in der DB-Subnetzgruppe zugeordnet ist.

   Wenn Sie einen Wert für Subnet Group (Subnetzgruppe) und No preference (Keine Präferenz) für Availability Zone angeben, wird das Lesereplikat auf einem zufälligen Outpost in der DB-Subnetzgruppe erstellt.
10. Wählen Sie für AWS KMS keydie AWS KMS key Kennung des KMS-Schlüssels.

    Das Lesereplikat muss verschlüsselt sein.

11. Wählen Sie je nach Bedarf andere Optionen aus.
12. Wählen Sie Read Replica erstellen aus.

Nachdem die Read Replica erstellt wurde, können Sie sie auf der Seite „ Datenbanken “ in der RDS-Konsole sehen. Es zeigt Replica in der Spalte Rolle.

AWS CLI

Verwenden Sie den Befehl -replica, um eine Read Replica aus einer MySQL- oder PostgreSQL-DB-Instance zu erstellen. AWS CLI create-db-instance-read

Sie können steuern, wo das Lesereplikat erstellt wird, indem Sie die Optionen `--db-subnet-group-name` und `--availability-zone` angeben:

- Wenn Sie beide Optionen `--db-subnet-group-name` und `--availability-zone` angeben, wird das Lesereplikat auf einem Outpost erstellt, der der Availability Zone in der DB-Subnetzgruppe zugeordnet ist.
- Wenn Sie die Option `--db-subnet-group-name` angeben und die Option `--availability-zone` nicht festlegen, wird das Lesereplikat auf einem zufälligen Outpost in der DB-Subnetzgruppe erstellt.
- Wenn Sie keine der beiden Optionen angeben, wird das Lesereplikat auf demselben Outpost wie die Quell-DB-Instance von RDS on Outposts erstellt.

Im folgenden Beispiel wird ein Replikat erstellt und der Speicherort des Lesereplikats durch Angabe der Optionen `--db-subnet-group-name` und `--availability-zone` festgelegt.

Example

Für, oder: Linux macOS Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --availability-zone us-west-2a
```

Windows:

```
aws rds create-db-instance-read-replica ^
```

```
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --availability-zone us-west-2a
```

RDS-API

Um eine Read Replica aus einer Quell-MySQL- oder PostgreSQL-DB-Instance zu erstellen, rufen Sie den Amazon InstanceReadReplica RDS-API-Vorgang CreateDB mit den folgenden erforderlichen Parametern auf:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Sie können steuern, wo das Lesereplikat erstellt wird, indem Sie die Parameter `DBSubnetGroupName` und `AvailabilityZone` angeben:

- Wenn Sie beide Parameter `DBSubnetGroupName` und `AvailabilityZone` angeben, wird das Lesereplikat auf einem Outpost erstellt, der der Availability Zone in der DB-Subnetzgruppe zugeordnet ist.
- Wenn Sie den Parameter `DBSubnetGroupName` angeben und den Parameter `AvailabilityZone` nicht festlegen, wird das Lesereplikat auf einem zufälligen Outpost in der DB-Subnetzgruppe erstellt.
- Wenn Sie keinen der beiden Parameer angeben, wird das Lesereplikat auf demselben Outpost wie die Quell-DB-Instance von RDS on Outposts erstellt.

Überlegungen zum Wiederherstellen von DB-Instances in Amazon RDS on AWS Outposts

Wenn Sie eine DB-Instance in Amazon RDS on AWS Outposts wiederherstellen, können Sie den Speicherort für automatisierte Backups und manuelle Snapshots der wiederhergestellten DB-Instance generell auswählen.

- Beim Wiederherstellen von einem manuellen DB-Snapshot können Sie Backups entweder in der übergeordneten AWS-Region oder lokal auf Ihrem Outpost speichern.
- Beim Wiederherstellen von einem automatisierten Backup (zeitpunktbezogene Wiederherstellung) haben Sie weniger Möglichkeiten:

- Wenn Sie aus der übergeordneten AWS-Region wiederherstellen, können Sie Backups entweder in der AWS-Region oder auf Ihrem Outpost speichern.
- Wenn Sie von Ihrem Outpost wiederherstellen, können Sie Backups nur auf Ihrem Outpost speichern.

Amazon RDS-Proxy

Durch die Verwendung von Amazon RDS Proxy können Sie Ihren Anwendungen erlauben, Datenbankverbindungen zu bündeln und gemeinsam zu nutzen, um ihre Skalierbarkeit zu verbessern. RDS Proxy macht Anwendungen widerstandsfähiger gegenüber Datenbankfehlern, indem er automatisch eine Verbindung zu einer Standby-DB-Instance herstellt, während Anwendungsverbindungen erhalten bleiben. Mithilfe von RDS Proxy können Sie auch die AWS Identity and Access Management (IAM-) Authentifizierung für Datenbanken erzwingen und Anmeldeinformationen sicher in AWS Secrets Manager speichern.

Mit RDS-Proxy können Sie unvorhersehbare Spitzen des Datenbankverkehrs bewältigen. Andernfalls können diese Überlastungen zu Problemen führen, da Verbindungen zu viele Abonnenten haben oder neue Verbindungen schnell hergestellt werden. RDS-Proxy richtet einen Datenbankverbindungspool ein und verwendet Verbindungen in diesem Pool wieder. Dieser Ansatz vermeidet den Speicher- und CPU-Overhead, der jedes Mal beim Öffnen einer neuen Datenbankverbindung erforderlich wäre. Um eine Datenbank vor Überbelegung zu schützen, können Sie die Anzahl der erstellten Datenbankverbindungen kontrollieren.

RDS Proxy stellt Anwendungsverbindungen, die nicht sofort vom Verbindungspool aus bedient werden können, in die Warteschlange oder drosselt sie. Obwohl Latenzen zunehmen können, kann Ihre Anwendung weiter skalieren, ohne dass die Datenbank abrupt ausfällt oder überfordert wird. Wenn Verbindungsanforderungen die von Ihnen angegebenen Grenzwerte überschreiten, lehnt RDS Proxy Anwendungsverbindungen ab (d. h. die Last wird abgeworfen). Gleichzeitig wird eine vorhersehbare Leistung für die Last aufrechterhalten, die RDS mit der verfügbaren Kapazität bewältigen kann.

Sie können den Overhead für die Verarbeitung von Anmeldeinformationen reduzieren und für jede neue Verbindung eine sichere Verbindung herstellen. Einige dieser Arbeiten kann RDS Proxy im Auftrag der Datenbank verarbeiten.

RDS-Proxy ist mit den Engine-Versionen, die es unterstützt, vollständig kompatibel. Sie können RDS-Proxy für die meisten Anwendungen ohne Codeänderungen aktivieren.

Themen

- Verfügbarkeit von Regionen und Versionen
- Kontingente und Einschränkungen für RDS Proxy
- Planen des Verwendungsortes von RDS-Proxy

- RDSKonzepte und Terminologie von Proxys
- Erste Schritte mit RDS Proxy
- Einen RDS Proxy verwalten
- Arbeiten mit Amazon RDS Proxy-Endpunkten
- Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch
- Mit RDS Proxy-Ereignissen arbeiten
- Fehlersuche für RDS Proxy
- Verwenden von RDS Proxy mit AWS CloudFormation

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen über die Verfügbarkeit von Versionen und Regionen von Amazon RDS mit RDS Proxy finden Sie unter Unterstützte Regionen und DB-Engines für Amazon RDS Proxy.

Kontingente und Einschränkungen für RDS Proxy

Die folgenden Einschränkungen gelten für RDS Proxy:

- Jede AWS-Konto ID ist auf 20 Proxys begrenzt. Wenn für Ihre Anwendung mehr Proxys erforderlich sind, fordern Sie eine Erhöhung über die Seite Servicekontingente in der an. AWS Management Console Wählen Sie auf der Seite Service Quotas Amazon Relational Database Service (Amazon RDS) aus und suchen Sie nach Proxys, um eine Kontingenterhöhung zu beantragen. AWS kann Ihr Kontingent oder die ausstehende Prüfung Ihrer Anfrage automatisch um erhöhen. Support
- Jeder Proxy kann bis zu 200 zugehörige Secrets Manager-Secrets haben. Somit kann jeder Proxy jederzeit eine Verbindung mit bis zu 200 verschiedenen Benutzerkonten herstellen.
- Jeder Proxy hat einen Standardendpunkt. Sie können auch bis zu 20 Proxy-Endpunkte für jeden Proxy hinzufügen. Sie können diese Endpoints erstellen, anzeigen, ändern und löschen.
- Für RDS DB-Instances in Replikationskonfigurationen können Sie einen Proxy nur der Schreiber-DB-Instance und nicht einer Read Replica zuordnen.
- Ihr RDS Proxy muss sich in derselben VPC wie die Datenbank befinden. Obwohl die Datenbank öffentlich zugänglich sein kann, kann der Proxy dies nicht sein. Wenn Sie beispielsweise

Prototypen für Ihre Datenbank auf einem lokalen Host erstellen, können Sie keine Verbindung zu Ihrem Proxy herstellen, es sei denn, Sie haben die erforderlichen Netzwerkanforderungen für die Verbindung zum Proxy eingerichtet. Dies liegt daran, dass sich Ihr lokaler Host außerhalb der VPC des Proxys befindet.

- Sie können RDS Proxy nicht mit einer VPC verwenden, für deren Tenancy `dedicated` festgelegt wurde.
- Wenn Sie den RDS-Proxy mit einem verwenden, für den die IAM-Authentifizierung aktiviert ist, überprüfen Sie die Benutzerauthentifizierung. Benutzer, die eine Verbindung über einen Proxy herstellen, müssen sich mit ihren Anmeldedaten authentifizieren. Details zu Secrets Manager und zur IAM-Unterstützung in RDS Proxy finden Sie unter Datenbankanmeldedaten AWS Secrets Manager für RDS Proxy einrichten und Konfiguration der IAM-Authentifizierung für RDS Proxy.
- Sie können RDS Proxy nicht mit benutzerdefiniertem DNS verwenden, wenn Sie die SSL-Hostnamenvalidierung nutzen.
- Jeder Proxy kann einer einzelnen Ziel-DB-Instance zugeordnet werden. Sie können jedoch mehrere Proxys derselben DB-Instance zuordnen.
- Jede Anweisung mit einer Textgröße über 16 KB bewirkt, dass der Proxy die Sitzung in der aktuellen Verbindung fixiert.
- In bestimmten Regionen gelten Einschränkungen für Availability-Zones (AZ), die Sie bei der Erstellung des Proxys berücksichtigen müssen. Die Region USA Ost (Nord-Virginia) unterstützt RDS-Proxy nicht in der Availability Zone `use1-az3`. Die Region USA West (Nordkalifornien) unterstützt RDS-Proxy nicht in der Availability Zone `usw1-az2`. Achten Sie beim Auswählen von Subnetzen während der Proxy-Erstellung darauf, dass Sie keine Subnetze in den oben genannten Availability Zones auswählen.
- Derzeit unterstützt RDS Proxy keine globalen Bedingungskontextschlüssel.

  Weitere Informationen über globale Bedingungskontextschlüssel finden Sie unter Globale AWS - Bedingungskontextschlüssel im IAM-Benutzerhandbuch.
- Sie können RDS Proxy nicht mit RDS Custom for SQL Server verwenden.
- Um Änderungen an Datenbankparametergruppen an Ihrem Proxy widerzuspiegeln, ist ein Instance-Neustart erforderlich, auch wenn Sie sich dafür entscheiden, Ihre Änderungen sofort zu übernehmen. Für Parameter auf Clusterebene ist ein clusterweiter Neustart erforderlich.
- Ihr Proxy erstellt den `rdsproxyadmin` DB-Benutzer automatisch, wenn Sie ein Proxyziel registrieren. Dies ist ein geschützter Benutzer, der für die Proxyfunktionalität unerlässlich ist. Sie sollten jegliche Manipulation des `rdsproxyadmin` Benutzers vermeiden. Das Löschen oder

Ändern des `rdsproxyadmin` Benutzers oder seiner Berechtigungen kann dazu führen, dass der Proxy für Ihre Anwendung vollständig nicht verfügbar ist.

Weitere Informationen zu den Einschränkungen für jede DB-Engine finden Sie in den folgenden Abschnitten:

- Weitere Einschränkungen für RDS für MariaDB
- Weitere Einschränkungen für RDS für Microsoft SQL Server
- Weitere Einschränkungen für RDS für MySQL
- Weitere Einschränkungen für RDS für PostgreSQL

Weitere Einschränkungen für RDS für MariaDB

Die folgenden zusätzlichen Einschränkungen gelten für RDS-Proxy mit Datenbanken von RDS für MariaDB:

- Derzeit führen alle Proxys Listening auf Port 3306 für MariaDB durch. Die Proxys stellen weiterhin eine Verbindung mit Ihrer Datenbank her, indem Sie den Port verwenden, den Sie in den Datenbankeinstellungen angegeben haben.
- Sie können RDS Proxy nicht mit selbstverwalteten MariaDB-Datenbanken in Amazon-Instances verwenden. EC2
- Sie können RDS Proxy nicht mit einer DB-Instance von RDS für MariaDB verwenden, bei welcher der Parameter `read_only` in der DB-Parametergruppe auf 1 festgelegt wurde.
- RDS Proxy unterstützt den komprimierten MariaDB-Modus nicht. Es unterstützt z. B. nicht die Komprimierung, die von den Optionen `--compress` oder `-C` des `mysql`-Befehls verwendet wird.
- Einige SQL-Anweisungen und -Funktionen können den Verbindungsstatus ändern, ohne dass Pinning verursacht wird. Informationen zum aktuellen Fixierungsverhalten finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys.
- MariaDB unterstützt das MariaDB-Plug-In `auth_ed25519` nicht.
- RDS Proxy unterstützt Transport Layer Security (TLS) Version 1.3 für MariaDB-Datenbanken nicht.
- Datenbankverbindungen, die einen Befehl `GET DIAGNOSTIC` verarbeiten, geben möglicherweise ungenaue Informationen zurück, wenn der RDS-Proxy dieselbe Datenbankverbindung für eine weitere Abfrage wiederverwendet. Dies kann passieren, wenn der RDS-Proxy Datenbankverbindungen multiplexiert. Weitere Informationen finden Sie unter Überblick über die RDS Proxy-Konzepte.

- RDS Proxy unterstützt derzeit die `caching_sha2_password` Option `ClientPasswordAuthType` für MariaDB nicht.

> **Important**
>
> Legen Sie in der Initialisierungsabfrage bei Proxys, die mit MariaDB-Datenbanken verknüpft sind, den Konfigurationsparameter `sql_auto_is_null` nicht auf `true` oder einen Wert ungleich Null fest. Dies kann zu einem falschen Anwendungsverhalten führen.

Weitere Einschränkungen für RDS für Microsoft SQL Server

Die folgenden zusätzlichen Einschränkungen gelten für RDS-Proxy mit Datenbanken von RDS für Microsoft SQL Server:

- Die Anzahl der Secrets von Secrets Manager, die Sie für einen Proxy erstellen müssen, hängt von der Sortierung ab, die Ihre DB-Instance verwendet. Angenommen, Ihre DB-Instance verwendet eine Sortierung, die die Groß-/Kleinschreibung berücksichtigt. Wenn Ihre Anwendung sowohl „Admin“ als auch „admin“ akzeptiert, benötigt Ihr Proxy zwei separate Secrets. Weitere Informationen über die Sortierung in SQL Server finden Sie in der Dokumentation für Microsoft SQL Server.
- RDS-Proxy unterstützt keine Verbindungen, die Active Directory verwenden.
- Sie können die IAM-Authentifizierung nicht mit Clients verwenden, die keine Tokeneigenschaften unterstützen. Weitere Informationen finden Sie unter Überlegungen zum Herstellen einer Verbindung mit einem Proxy mit Microsoft SQL Server.
- Die Ergebnisse von `@@IDENTITY`, `@@ROWCOUNT` und `SCOPE_IDENTITY` sind nicht immer genau. Rufen Sie zur Umgehung dieses Problems ihre Werte in derselben Sitzungsanweisung ab, um sicherzustellen, dass sie die richtigen Informationen zurückgeben.
- Wenn die Verbindung mehrere aktive Ergebnismengen (MARS) verwendet, führt RDS-Proxy die Initialisierungsabfragen nicht aus. Weitere Informationen zu MARS finden Sie in der Dokumentation für Microsoft SQL Server.
- Derzeit unterstützt RDS Proxy kein RDS für SQL Server-DB-Instances, die auf der Hauptversion SQL Server 2022 ausgeführt werden.
- RDS Proxy unterstützt RDS nicht für SQL Server-DB-Instances, die auf der Hauptversion SQL Server 2014 ausgeführt werden.

- RDS Proxy unterstützt keine Client-Anwendungen, die nicht mehrere Antwortnachrichten in einem TLS-Eintrag verarbeiten können.

Weitere Einschränkungen für RDS für MySQL

Die folgenden zusätzlichen Einschränkungen gelten für RDS-Proxy mit Datenbanken von RDS für MySQL:

- Die RDS-Proxyunterstützung für die `caching_sha2_password` Authentifizierung erfordert eine sichere Verbindung (TLS).
- `caching_sha2_password`Es ist bekannt, dass es bei der RDS-Proxyunterstützung für Kompatibilitätsprobleme mit bestimmten Go-SQL-Treiberversionen gibt.
- Bei Verwendung des MySQL 8.4 C-Treibers kann die `mysql_stmt_bind_named_param` API fehlerhafte Pakete bilden, wenn die Anzahl der Parameter die Anzahl der Platzhalter in vorbereiteten Anweisungen übersteigt. Dies führt zu falschen Antworten. Weitere Informationen finden Sie im MySQL-Fehlerbericht.
- Derzeit führen alle Proxys Listening auf Port 3306 für MySQL durch. Die Proxys stellen weiterhin eine Verbindung mit Ihrer Datenbank her, indem Sie den Port verwenden, den Sie in den Datenbankeinstellungen angegeben haben.
- Sie können RDS Proxy nicht mit selbstverwalteten MySQL-Datenbanken in EC2 Instanzen verwenden.
- Sie können RDS Proxy nicht mit einer RDS-for-MySQL-DB-Instance verwenden, bei welcher der Parameter `read_only` in der DB-Parametergruppe auf 1 gesetzt wurde.
- RDS Proxy unterstützt den komprimierten MySQL-Modus nicht. Es unterstützt z. B. nicht die Komprimierung, die von den Optionen `--compress` oder `-C` des `mysql`-Befehls verwendet wird.
- Datenbankverbindungen, die einen Befehl `GET DIAGNOSTIC` verarbeiten, geben möglicherweise ungenaue Informationen zurück, wenn der RDS-Proxy dieselbe Datenbankverbindung für eine weitere Abfrage wiederverwendet. Dies kann passieren, wenn der RDS-Proxy Datenbankverbindungen multiplexiert.
- Einige SQL-Anweisungen und Funktionen `SET LOCAL` können z. B. den Verbindungsstatus ändern, ohne dass es zu einer Fixierung kommt. Informationen zum aktuellen Fixierungsverhalten finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys.
- Die Verwendung der `ROW_COUNT()` Funktion in einer Abfrage mit mehreren Anweisungen wird nicht unterstützt.

- RDS Proxy unterstützt keine Client-Anwendungen, die nicht mehrere Antwortnachrichten in einem TLS-Datensatz verarbeiten können.
- RDS Proxy unterstützt die dualen MySQL-Passwörter nicht.
- RDS Proxy funktioniert möglicherweise nicht wie erwartet, wenn Sie den `init_connect` Parameter in Ihrer RDS-DB-Parametergruppe so konfigurieren, dass Sitzungsstatusvariablen festgelegt werden. Stellen Sie stattdessen die Initialisierungsabfrage für Ihren Proxy so ein, dass Sitzungsinitialisierungsanweisungen ausgeführt werden, wenn Sie den Proxy für die Verbindung mit Ihrer Datenbank verwenden.

> **Important**
>
> Setzen Sie in der Initialisierungsabfrage bei Proxys, die mit MySQL-Datenbanken verknüpft sind, den Konfigurationsparameter `sql_auto_is_null` nicht auf `true` oder einen Wert ungleich Null. Dies kann zu einem falschen Anwendungsverhalten führen.

Weitere Einschränkungen für RDS für PostgreSQL

Die folgenden zusätzlichen Einschränkungen gelten für RDS-Proxy mit Datenbanken von RDS für PostgreSQL:

- RDS Proxy unterstützt keine Sitzungs-Pinning-Filter für PostgreSQL.
- Derzeit führen alle Proxys Listening auf Port 5432 für PostgreSQL durch.
- Für PostgreSQL unterstützt RDS Proxy derzeit nicht das Abbrechen einer Abfrage von einem Client durch Ausgeben von `CancelRequest`. Dies ist beispielsweise der Fall, wenn Sie eine lange andauernde Abfrage in einer interaktiven psql-Sitzung mithilfe von Strg+C abbrechen.
- Die Ergebnisse der PostgreSQL-Funktion lastval sind nicht immer genau. Verwenden Sie zur Umgehung die INSERT-Anweisung mit der Klausel `RETURNING`.
- RDS Proxy unterstützt derzeit den Streaming-Replikationsmodus nicht.
- Bei RDS for PostgreSQL 16 wirken sich Änderungen am `scram_iterations` Wert ausschließlich auf den Authentifizierungsprozess zwischen dem Proxy und der Datenbank aus. Insbesondere wenn Sie dies konfigurieren `ClientPasswordAuthTypescram-sha-256`, haben alle am `scram_iterations` Wert vorgenommenen Anpassungen keinen Einfluss client-to-proxy auf die Passwortauthentifizierung. Stattdessen ist der Iterationswert für die client-to-proxy Kennwortauthentifizierung auf 4096 festgelegt.

- Die `default` Datenbank muss existieren.
- Wenn Sie die Benutzerrolle mit ändern`SET ROLE`, verwenden `ALTER ROLE` nachfolgende Verbindungen als dieser Benutzer zum Proxy diese Rolleneinstellung möglicherweise nicht, wenn bei diesen Verbindungen gepinnt wird. Um dies zu verhindern, verwenden Sie bei der Verwendung eines Proxys `SET ROLE` in der Initialisierungsabfrage des Proxys. Weitere Informationen finden Sie unter Initialisierungsabfrage in. Einen Proxy für Amazon RDS — erstellen

> **Important**
>
> Wenn Sie bei vorhandenen Proxys mit PostgreSQL-Datenbanken die Datenbankauthentifizierung so ändern, dass nur `SCRAM` verwendet wird, ist der Proxy für bis zu 60 Sekunden nicht verfügbar. Um das Problem zu vermeiden, führen Sie einen der folgenden Schritte aus:
>
> - Stellen Sie sicher, dass die Datenbank sowohl die `SCRAM`- als auch die `MD5`-Authentifizierung zulässt.
> - Wenn Sie nur die `SCRAM`-Authentifizierung verwenden möchten, erstellen Sie einen neuen Proxy, migrieren Sie Ihren Anwendungsdatenverkehr auf den neuen Proxy und löschen Sie dann den zuvor mit der Datenbank verknüpften Proxy.

Planen des Verwendungsortes von RDS-Proxy

Sie können ermitteln, welche Ihrer DB-Instances, Cluster und Anwendungen möglicherweise am meisten von der Verwendung von RDS Proxy profitieren. Berücksichtigen Sie dazu folgende Faktoren:

- Alle DB-Instances, auf denen gelegentlich der Fehler „zu viele Verbindungen“ auftritt, sind gut geeignet für die Zuordnung zu einem Proxy. Dies ist oft durch einen hohen Wert der -`ConnectionAttempts` CloudWatch Metrik gekennzeichnet. Der Proxy ermöglicht es Anwendungen, viele Clientverbindungen zu öffnen, während der Proxy eine geringere Anzahl langlebiger Verbindungen mit der DB-Instance verwaltet.
- Bei DB-Instance, die kleinere AWS Instance-Klassen wie T2 oder T3 verwenden, kann die Verwendung eines Proxys dazu beitragen, out-of-memory Bedingungen zu vermeiden. Sie kann auch dazu beitragen, den CPU-Overhead für das Herstellen von Verbindungen zu reduzieren. Diese Bedingungen können auftreten, wenn es um eine große Anzahl von Verbindungen geht.

- Sie können bestimmte Amazon- CloudWatch Metriken überwachen, um festzustellen, ob sich eine DB-Instanceein DB- bestimmten Arten von Grenzwerten nähert. Diese Grenzwerte gelten für die Anzahl der Verbindungen und den Arbeitsspeicher, der mit der Verbindungsverwaltung verbunden ist. Sie können auch bestimmte CloudWatch Metriken überwachen, um festzustellen, ob eine DB-Instance ein DB- viele kurzlebige Verbindungen verarbeitet. Das Öffnen und Beenden solcher Verbindungen kann einen Leistungs-Overhead für Ihre Datenbank verursachen. Informationen zu den zu überwachenden Metriken finden Sie unter Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch.
- AWS Lambda-Funktionen können auch gute Kandidaten für die Verwendung eines Proxys sein. Diese Funktionen stellen häufig kurze Datenbankverbindungen her, die von dem von RDS Proxy angebotenen Verbindungspooling profitieren. Sie können alle IAM-Authentifizierungen nutzen, die Sie bereits für Lambda-Funktionen haben, anstatt Datenbankanmeldeinformationen im Lambda-Anwendungscode zu verwalten.
- Anwendungen, die in der Regel eine große Anzahl von Datenbankverbindungen öffnen und schließen und über keine integrierten Mechanismen für das Verbindungspooling verfügen, bieten sich für die Verwendung eines Proxys an.
- Anwendungen, die eine große Anzahl von Verbindungen über lange Zeiträume offen halten, sind in der Regel gute Kandidaten für die Verwendung eines Proxys. Anwendungen in Branchen wie Software as a Service (SaaS) oder E-Commerce minimieren häufig die Latenz für Datenbankanfragen, da sie Verbindungen offen lassen. Mit RDS Proxy kann eine Anwendung mehr Verbindungen offen halten als möglich, wenn eine direkte Verbindung mit der DB-Instancedem DB hergestellt wird.
- Möglicherweise haben Sie die IAM-Authentifizierung und Secrets Manager aufgrund der Komplexität der Einrichtung einer solchen Authentifizierung für alle DB-Instances nicht übernommen. Wenn dies der Fall ist, können Sie die vorhandenen Authentifizierungsmethoden beibehalten und die Authentifizierung an einen Proxy delegieren. Der Proxy kann die Authentifizierungsrichtlinien für Clientverbindungen für bestimmte Anwendungen erzwingen. Sie können alle IAM-Authentifizierungen nutzen, die Sie bereits für Lambda-Funktionen haben, anstatt Datenbankanmeldeinformationen im Lambda-Anwendungscode zu verwalten.
- RDS-Proxy kann dazu beitragen, Anwendungen widerstandsfähiger und transparenter gegenüber Datenbankausfällen zu machen. RDS-Proxy umgeht Domain Name System (DNS)-Caches, um die Failover-Zeiten für Multi-AZ-DB-Instances von Amazon RDS um bis zu 66 % zu reduzieren. RDS-Proxy leitet den Datenverkehr außerdem automatisch an eine neue Datenbank-Instance weiter, wobei Anwendungsverbindungen erhalten bleiben. Dadurch werden Failovers für Anwendungen transparenter.

RDSKonzepte und Terminologie von Proxys

Sie können das Verbindungsmanagement für Ihre Amazon RDS DB-Instances und mithilfe von RDS Proxy vereinfachen.

RDSDer Proxy verarbeitet den Netzwerkverkehr zwischen der Client-Anwendung und der Datenbank. Es tut dies auf aktive Weise, indem es das Datenbankprotokoll erfasst Anschließend passt er sein Verhalten an die SQL Operationen Ihrer Anwendung und die Ergebnismengen aus der Datenbank an.

RDSDer Proxy reduziert den Arbeitsspeicher und den CPU Overhead für die Verbindungsverwaltung in Ihrer Datenbank. Die Datenbank benötigt weniger Speicher und CPU Ressourcen, wenn Anwendungen viele Verbindungen gleichzeitig öffnen. Es erfordert auch keine Logik in Ihren Anwendungen, um Verbindungen zu schließen und wieder zu öffnen, die für eine lange Zeit inaktiv bleiben. Ebenso erfordert es weniger Anwendungslogik, um Verbindungen im Falle eines Datenbankproblems wiederherzustellen.

Die Infrastruktur für RDS Proxy ist hochverfügbar und wird über mehrere Availability Zones (AZs) bereitgestellt. Die Berechnung, der Arbeitsspeicher und der Speicher für RDS Proxy sind unabhängig vom . Diese Trennung hilft, den Overhead auf Ihren Datenbankservern zu senken, sodass sie ihre Ressourcen für die Bereitstellung von Datenbank-Workloads einsetzen können. Die RDS Proxy-Rechenressourcen sind serverlos und werden automatisch auf der Grundlage Ihrer Datenbank-Arbeitslast skaliert.

Themen

- Überblick über die RDS Proxy-Konzepte
- Verbindungspooling
- RDSProxy-Sicherheit
- Failover
- Transaktionen

Überblick über die RDS Proxy-Konzepte

RDSProxy verwaltet die Infrastruktur für das Verbindungspooling und die anderen Funktionen, die in den folgenden Abschnitten beschrieben werden. Die Proxys werden in der RDS Konsole auf der Seite Proxys dargestellt.

Jeder Proxy verarbeitet Verbindungen zu einer einzelnen RDSDB-Instance, einem . Der Proxy bestimmt automatisch die aktuelle Writer-Instance für RDSMulti-AZ-DB-Instances oder Cluster, die von .

Die Verbindungen, die ein Proxy offen hält und für Ihre Datenbankanwendungen verfügbar hält, bilden den Verbindungspool.

Standardmäßig kann RDS Proxy eine Verbindung nach jeder Transaktion in Ihrer Sitzung wiederverwenden. Diese Wiederverwendung auf Transaktionsebene wird als Multiplexing bezeichnet. Wenn RDS Proxy vorübergehend eine Verbindung aus dem Verbindungspool entfernt, um sie wiederzuverwenden, wird dieser Vorgang als Ausleihen der Verbindung bezeichnet. Wenn dies sicher ist, gibt RDS Proxy diese Verbindung wieder an den Verbindungspool zurück.

In einigen Fällen kann RDS Proxy nicht sicher sein, ob es sicher ist, eine Datenbankverbindung außerhalb der aktuellen Sitzung wiederzuverwenden. In diesen Fällen bleibt die Sitzung auf derselben Verbindung erhalten, bis die Sitzung beendet ist. Dieses Fallback-Verhalten wird als Fixierung (Pinning) bezeichnet.

Ein Proxy hat einen Standard-Endpunkt. Sie stellen eine Verbindung zu diesem Endpunkt her, wenn Sie mit einer Amazon RDS DB-Instance — einem Amazon — arbeiten. Sie tun dies, anstatt eine Verbindung mit dem Lese-/Schreib-Endpunkt herzustellen, der direkt mit der Instance verbunden wird. Für DB-Cluster können Sie auch zusätzliche Lese-/Schreib- und schreibgeschützte Endpunkte erstellen. Weitere Informationen finden Sie unter Überblick über Proxy-Endpunkte.

Sie können beispielsweise weiterhin eine Verbindung zum Clusterendpunkt für Verbindungen mit Lese-/Schreibzugriff ohne Verbindungspooling herstellen. Sie können weiterhin eine Verbindung zum Leser-Endpunkt für schreibgeschützte Verbindungen mit Lastenausgleich herstellen. Sie können weiterhin eine Verbindung zu den Instance-Endpunkten für die Diagnose und Behebung von Fehlern in bestimmten DB-Instances innerhalb eines Clusters herstellen. Wenn Sie andere AWS Dienste verwenden, z. B. AWS Lambda um eine Verbindung zu RDS Datenbanken herzustellen, ändern Sie deren Verbindungseinstellungen, sodass der Proxy-Endpunkt verwendet wird. Sie geben beispielsweise den Proxy-Endpunkt an, um Lambda-Funktionen den Zugriff auf Ihre Datenbank zu ermöglichen und gleichzeitig die RDS Proxy-Funktionalität zu nutzen.

Jeder Proxy enthält eine Zielgruppe. Diese Zielgruppe verkörpert den , zu dem der Proxy eine Verbindung herstellen kann. Der mit einem Proxy verknüpfte wird als Ziele dieses Proxys bezeichnet. Der Einfachheit halber erstellt Proxy beim Erstellen eines RDS Proxys über die Konsole auch die entsprechende Zielgruppe und registriert die zugehörigen Ziele automatisch.

Eine Engine-Familie ist eine verwandte Gruppe von Datenbank-Engines, die dasselbe DB-Protokoll verwenden. Sie wählen die Engine-Familie für jeden Proxy, den Sie erstellen.

Verbindungspooling

Jeder Proxy führt das Verbindungspooling separat für die Writer- und Reader-Instance der zugehörigen RDSDatenbank durch. Das Verbindungspooling ist eine Optimierung, die den Overhead reduziert, der mit dem Öffnen und Beenden von Verbindungen und dem Vorhandensein vieler aktiver Verbindungen gleichzeitig verbunden ist. Dieser Overhead umfasst den Arbeitsspeicher, der für die Verarbeitung jeder neuen Verbindung erforderlich ist. Außerdem ist das Schließen jeder Verbindung und das Öffnen einer neuen Verbindung mit CPU Mehraufwand verbunden. Beispiele hierfür sind Handshaking (Transport LayerSecurity/Secure Sockets Layer (TLS/SSL), Authentifizierung, Verhandlungsmöglichkeiten usw. Das Verbindungspooling vereinfacht Ihre Anwendungslogik. Sie müssen keinen Anwendungscode schreiben, um die Anzahl gleichzeitig geöffneter Verbindungen zu minimieren.

Jeder Proxy führt darüber hinaus Verbindungsmultiplexing aus, auch bekannt als Wiederverwendung von Verbindungen. Beim Multiplexing führt der RDS Proxy alle Operationen für eine Transaktion mithilfe einer zugrunde liegenden Datenbankverbindung aus. RDSkann dann eine andere Verbindung für die nächste Transaktion verwenden. Sie können viele gleichzeitige Verbindungen zum Proxy öffnen und der Proxy hält eine geringere Anzahl von Verbindungen zur DB-Instance oder zum Cluster offen. Dadurch wird der Speicher-Overhead für Verbindungen auf dem Datenbankserver weiter minimiert. Diese Technik reduziert auch die Wahrscheinlichkeit des Fehlers "zu viele Verbindungen".

RDSProxy-Sicherheit

RDSDer Proxy verwendet die vorhandenen RDS Sicherheitsmechanismen wieTLS/SSLund AWS Identity and Access Management (IAM). Allgemeine Informationen zu diesen Sicherheitsfunktionen finden Sie unter Sicherheit in Amazon RDS. Machen Sie sich außerdem mit der Funktionsweise von RDS in Bezug auf Authentifizierung, Autorisierung und andere Sicherheitsbereiche vertraut.

RDSDer Proxy kann als zusätzliche Sicherheitsebene zwischen Client-Anwendungen und der zugrunde liegenden Datenbank dienen. Sie können beispielsweise mit TLS 1.3 eine Verbindung zum Proxy herstellen, auch wenn die zugrunde liegende DB-Instance eine ältere Version von unterstütztTLS. Sie können mithilfe einer IAM Rolle eine Verbindung zum Proxy herstellen. Dies gilt auch dann, wenn der Proxy mithilfe der nativen Benutzer- und Passwortauthentifizierungsmethode eine Verbindung mit der Datenbank herstellt. Mit dieser Technik können Sie hohe Authentifizierungsanforderungen für Datenbankanwendungen ohne einen kostspieligen Migrationsaufwand für die DB-Instances selbst erzwingen.

Sie speichern die vom RDS Proxy verwendeten Datenbankanmeldeinformationen in AWS Secrets Manager. Jeder Datenbankbenutzer für den , auf den ein Proxy zugreift, muss über ein entsprechendes Geheimnis in Secrets Manager verfügen. Sie können auch die IAM Authentifizierung für Benutzer von RDS Proxy einrichten. Auf diese Weise können Sie die IAM Authentifizierung für den Datenbankzugriff erzwingen, auch wenn die Datenbanken die systemeigene Passwortauthentifizierung verwenden. Es wird empfohlen, diese Sicherheitsfunktionen zu verwenden, anstatt Datenbankanmeldeinformationen in Ihren Anwendungscode einzubetten.

TLS/SSLmit RDS Proxy verwenden

Sie können mit dem SSL ProtokollTLS/eine Verbindung zum RDS Proxy herstellen.

> **Note**
>
> RDSDer Proxy verwendet Zertifikate von AWS Certificate Manager (ACM). Wenn Sie RDS Proxy verwenden, müssen Sie keine RDS Amazon-Zertifikate herunterladen oder Anwendungen aktualisieren, die RDS Proxy-Verbindungen verwenden.

Um dies TLS für alle Verbindungen zwischen dem Proxy und Ihrer Datenbank durchzusetzen, können Sie beim Erstellen oder Ändern eines Proxys in der die Einstellung Transport Layer Security erforderlich angeben AWS Management Console.

RDSDer Proxy kann auch sicherstellen, dass Ihre SitzungTLS/SSLzwischen Ihrem Client und dem RDS Proxy-Endpunkt verwendet. Damit RDS Proxy dies tun kann, geben Sie die Anforderung auf der Clientseite an. SSLSitzungsvariablen werden nicht für SSL Verbindungen zu einer Datenbank über RDS Proxy festgelegt.

- Geben Sie RDSfür My SQL die Anforderung auf der Clientseite mit dem `--ssl-mode` Parameter an, wenn Sie den `mysql` Befehl ausführen.
- Geben Sie bei Amazon RDS Postgre SQL `sslmode=require` als Teil der `conninfo` Zeichenfolge anSQL, wenn Sie den Befehl ausführen. `psql`

RDSDer Proxy unterstützt die TLS Protokollversionen 1.0, 1.1, 1.2 und 1.3. Sie können eine Verbindung zum Proxy herstellen, indem Sie eine höhere Version von verwenden TLS als die, die Sie in der zugrunde liegenden Datenbank verwenden.

Standardmäßig stellen Client-Programme eine verschlüsselte Verbindung mit dem RDS Proxy her. Weitere Steuerungsmöglichkeiten stehen über `--ssl-mode` diese Option zur Verfügung. Auf der Client-Seite unterstützt RDS Proxy alle SSL Modi.

Für den Client sind die SSL Modi die folgenden:

PREFERRED

SSList die erste Wahl, aber sie ist nicht erforderlich.

DISABLED

Nein SSL ist erlaubt.

REQUIRED

ErzwingenSSL.

VERIFY_CA

Erzwingen SSL und verifizieren Sie die Zertifizierungsstelle (CA).

VERIFY_IDENTITY

Erzwingen SSL und verifizieren Sie die Zertifizierungsstelle und den CA-Hostnamen.

Bei Verwendung eines Clients mit `--ssl-mode` `VERIFY_CA` oder `VERIFY_IDENTITY`, geben Sie die Option `--ssl-ca` an, die auf eine CA im `.pem`-Format verweist. Laden Sie für die zu verwendende `.pem` Datei alle Stammzertifizierungsstellen PEMs von Amazon Trust Services herunter und platzieren Sie sie in einer einzigen `.pem` Datei.

RDSDer Proxy verwendet Wildcard-Zertifikate, die sowohl für eine Domain als auch für ihre Subdomains gelten. Wenn Sie den `mysql` Client für die Verbindung mit dem SSL Modus verwenden`VERIFY_IDENTITY`, müssen Sie derzeit den My SQL `mysql` 8.0-kompatiblen Befehl verwenden.

Failover

Failover ist eine Funktion mit hoher Verfügbarkeit, die eine Datenbank-Instance durch eine andere ersetzt, wenn die ursprüngliche Instance nicht verfügbar ist. Ein Failover kann aufgrund eines Problems mit einer Datenbank-Instance auftreten. Er kann aber auch Teil normaler Wartungsverfahren sein, z. B. während eines Datenbank-Upgrades. Failover gilt für RDS DB-Instances in einer Multi-AZ-Konfiguration.

Wenn Sie eine Verbindung über einen Proxy herstellen, sind Ihre Anwendungen widerstandsfähiger gegenüber Datenbank-Failovers. Wenn die ursprüngliche DB-Instance nicht mehr verfügbar ist, stellt der RDS Proxy eine Verbindung zur Standby-Datenbank her, ohne dass inaktive Anwendungsverbindungen unterbrochen werden. Dies trägt dazu bei, den Failover-Prozess zu beschleunigen und zu vereinfachen. Dadurch wird Ihre Anwendung weniger beeinträchtigt als ein typischer Neustart oder ein Datenbankproblem.

Ohne RDS Proxy bedeutet ein Failover einen kurzen Ausfall. Während des Ausfalls können Sie im Failover keine Schreibvorgänge an der Datenbank ausführen. Alle vorhandenen Datenbankverbindungen werden unterbrochen und Ihre Anwendung muss sie erneut öffnen. Die Datenbank wird für neue Verbindungen und Schreiboperationen verfügbar, wenn eine schreibgeschützte DB-Instance anstelle einer nicht verfügbaren Instance hochgestuft wird.

Während eines DB-Failovers akzeptiert der RDS Proxy weiterhin Verbindungen mit derselben IP-Adresse und leitet Verbindungen automatisch zur neuen primären DB-Instance weiter. Clients, die eine Verbindung über RDS einen Proxy herstellen, sind nicht anfällig für Folgendes:

- Verzögerungen bei der Übertragung von Domain Name System (DNS) beim Failover.
- Lokales DNS Caching.
- Verbindungszeitüberschreitungen.
- Unsicherheit darüber, welche DB-Instance der aktuelle Writer ist.
- Warten auf eine Abfrageantwort eines früheren Writers, die nicht verfügbar wurde, ohne Verbindungen zu schließen.

Bei Anwendungen, die ihren eigenen Verbindungspool verwalten, bedeutet die Verwendung eines RDS Proxys, dass die meisten Verbindungen auch bei Failovers oder anderen Störungen bestehen bleiben. Nur Verbindungen, die sich mitten in einer Transaktion oder SQL Abrechnung befinden, werden storniert. RDS Der Proxy akzeptiert sofort neue Verbindungen. Wenn der Datenbank-Writer nicht verfügbar ist, stellt RDS Proxy eingehende Anfragen in eine Warteschlange.

Für Anwendungen, die keine eigenen Verbindungspools verwalten, bietet RDS Proxy schnellere Verbindungsraten und mehr offene Verbindungen. Es reduziert den teuren Overhead, der mit dem häufigen Herstellen neuer Verbindungen über die Datenbank verbunden ist. Dies geschieht durch die Wiederverwendung von Datenbankverbindungen, die im RDS Proxy-Verbindungspool verwaltet werden. Dieser Ansatz ist besonders wichtig für TLS Verbindungen, bei denen die Einrichtungskosten erheblich sind.

Transaktionen

Alle Anweisungen innerhalb einer einzelnen Transaktion verwenden immer dieselbe zugrunde liegende Datenbankverbindung. Die Verbindung wird für eine andere Sitzung verfügbar, wenn die Transaktion beendet wird. Die Verwendung der Transaktion als Granularitätseinheit hat folgende Auswirkungen:

- Die Wiederverwendung von Verbindungen kann nach jeder einzelnen Anweisung erfolgen, wenn die SQL `autocommit` Einstellung RDSfür My SQL aktiviert ist.
- Ist die Einstellung `autocommit` deaktiviert, beginnt im Gegensatz dazu die erste Anweisung, die Sie in einer Sitzung ausgeben, eine neue Transaktion. Nehmen wir beispielsweise an, Sie geben eine Folge von`SELECT`, `INSERTUPDATE`, und anderen Anweisungen in der Datenmanipulationssprache (DML) ein. In diesem Fall wird die Wiederverwendung von Verbindungen erst vorgenommen, wenn Sie einen `COMMIT`, `ROLLBACK` ausgeben oder die Transaktion anderweitig beenden.
- Die Eingabe einer Anweisung zur Datendefinitionssprache (DDL) führt dazu, dass die Transaktion beendet wird, nachdem diese Anweisung abgeschlossen ist.

RDSDer Proxy erkennt anhand des Netzwerkprotokolls, das von der Datenbank-Client-Anwendung verwendet wird, wenn eine Transaktion endet. Die Transaktionserkennung stützt sich nicht auf Schlüsselwörter wie `COMMIT` oder solche, die im Text der SQL Anweisung `ROLLBACK` vorkommen.

In einigen Fällen erkennt der RDS Proxy möglicherweise eine Datenbankanforderung, sodass es nicht praktikabel ist, Ihre Sitzung auf eine andere Verbindung zu verschieben. In diesen Fällen wird das Multiplexing für diese Verbindung während der verbleibenden Sitzung deaktiviert. Dieselbe Regel gilt, wenn der RDS Proxy nicht sicher sein kann, ob Multiplexing für die Sitzung praktikabel ist. Diese Operation wird als Fixierung (Pinning) bezeichnet. Hinweise zum Erkennen und Minimieren von Fixierungen finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys.

Erste Schritte mit RDS Proxy

Verwenden Sie die Informationen auf den folgenden Seiten, um die entsprechenden Sicherheitsoptionen einzurichten, zu verwalten Amazon RDS-Proxy und festzulegen. Die Sicherheitsoptionen steuern, wer auf jeden Proxy zugreifen kann und wie jeder Proxy eine Verbindung zu DB-Instances herstellt.

Wenn Sie RDS Proxy noch nicht kennen, empfehlen wir, den Seiten in der Reihenfolge zu folgen, in der sie angezeigt werden.

Themen

- Netzwerkvoraussetzungen für RDS Proxy einrichten
- Datenbankanmeldedaten AWS Secrets Manager für RDS Proxy einrichten
- Konfiguration der IAM-Authentifizierung für RDS Proxy
- Einen Proxy für Amazon RDS — erstellen
- Einen Proxy anzeigen
- Verbinden mit einer Datenbank über RDS Proxy

Netzwerkvoraussetzungen für RDS Proxy einrichten

Für die Verwendung von RDS Proxy benötigen Sie eine gemeinsame Virtual Private Cloud (VPC) zwischen Ihrer RDS-DB-Instance, dem und dem RDS-Proxy. Diese VPC sollte über mindestens zwei Subnetze verfügen, die sich in verschiedenen Availability Zones befinden. Ihr Konto kann entweder der Eigentümer dieser Subnetze sein oder sie mit anderen Konten teilen. Informationen zur VPC-Sharing finden Sie unter Arbeiten mit Shared VPCs.

Ihre Client-Anwendungsressourcen wie Amazon EC2, Lambda oder Amazon ECS können sich in derselben VPC wie der Proxy befinden. Sie können sich auch in einer vom Proxy getrennten VPC befinden. Wenn Sie eine Verbindung mit RDS-DB-Instances hergestellt haben, verfügen Sie bereits über die erforderlichen Netzwerkressourcen.

Themen

- Abrufen von Informationen zu Ihren Subnetzen
- Planen der Kapazität von IP-Adressen

Abrufen von Informationen zu Ihren Subnetzen

Um einen Proxy zu erstellen, müssen Sie die Subnetze und die VPC angeben, in denen der Proxy betrieben wird. Das folgende Linux-Beispiel zeigt AWS CLI Befehle, die die Subnetze VPCs und Subnetze untersuchen, die Ihrem gehören. AWS-Konto Insbesondere übergeben Sie das Subnetz IDs als Parameter, wenn Sie einen Proxy mit der CLI erstellen.

```
aws ec2 describe-vpcs
```

```
aws ec2 describe-internet-gateways
aws ec2 describe-subnets --query '*[].[VpcId,SubnetId]' --output text | sort
```

Das folgende Linux-Beispiel zeigt AWS CLI Befehle zur Bestimmung des Subnetzes, das einem bestimmten einer bestimmten RDS-DB-Instance IDs entspricht. Suchen Sie die VPC-ID für die DB-Instance. Untersuchen Sie die VPC, um ihre Subnetze zu finden. Das folgende Linux-Beispiel zeigt, wie das geht.

```
$ #From the DB instance, trace through the DBSubnetGroup and Subnets to find the subnet
 IDs.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]||[0]||[0]||[Subnets]||[0]||[*].SubnetIdentifier' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
...
```

```
$ #From the DB instance, find the VPC.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]||[0]||[0].VpcId' --output text
```

```
my_vpc_id
```

```
$ aws ec2 describe-subnets --filters Name=vpc-id,Values=my_vpc_id --query '*[].
[SubnetId]' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
subnet_id_4
subnet_id_5
subnet_id_6
```

Planen der Kapazität von IP-Adressen

Ein RDS-Proxy passt seine Kapazität basierend auf der Größe und Anzahl der bei ihm registrierten DB-Instances automatisch nach Bedarf an. Bestimmte Operationen erfordern möglicherweise auch zusätzliche Proxykapazität, wie z. B. die Erhöhung der Größe einer registrierten Datenbank

oder interne Wartungsvorgänge des RDS-Proxys. Bei diesen Vorgängen benötigt Ihr Proxy möglicherweise mehr IP-Adressen, um die zusätzliche Kapazität bereitzustellen. Mit diesen zusätzlichen Adressen kann Ihr Proxy skaliert werden, ohne Ihre Workload zu beeinträchtigen. Ein Mangel an freien IP-Adressen in Ihren Subnetzen verhindert, dass ein Proxy hochskaliert wird. Dies kann zu höheren Abfragelatenzen oder Verbindungsfehlern bei Clients führen. RDS benachrichtigt Sie durch das Ereignis `RDS-EVENT-0243`, wenn in Ihren Subnetzen nicht genügend freie IP-Adressen vorhanden sind. Weitere Informationen zu diesem Ereignis finden Sie unter Mit RDS Proxy-Ereignissen arbeiten.

Reservieren Sie die folgende Mindestanzahl an freien IP-Adressen in Ihren Subnetzen für Ihren Proxy, basierend auf der Größe der DB-Instance-Klassen.

DB-Instance-Klasse	Mindestanzahl freier IP-Adressen
db.*.xlarge oder kleiner	10
db.*.2xlarge	15
db.*.4xlarge	25
db.*.8xlarge	45
db.*.12xlarge	60
db.*.16xlarge	75
db.*.24xlarge	110

Bei dieser empfohlenen Anzahl von IP-Adressen handelt es sich um Schätzungen für einen Proxy, der nur den Standardendpunkt verwendet. Ein Proxy mit zusätzlichen Endpunkten oder Read Replicas benötigt möglicherweise mehr freie IP-Adressen. Wir empfehlen, für jeden weiteren Endpunkt drei weitere IP-Adressen zu reservieren. Es wird empfohlen, für jede Read Replica zusätzliche IP-Adressen zu reservieren, wie in der Tabelle angegeben, basierend auf der Größe der Read Replica.

> **Note**
>
> RDS Proxy verwendet nicht mehr als 215 IP-Adressen in einer VPC.

Datenbankanmeldedaten AWS Secrets Manager für RDS Proxy einrichten

RDS Proxy in Amazon RDS verwendet AWS Secrets Manager , um Datenbankanmeldedaten sicher zu speichern und zu verwalten. Anstatt Anmeldeinformationen in Ihre Anwendung einzubetten, verknüpfen Sie einen Proxy mit einem Secrets Manager Manager-Geheimnis, das die erforderlichen Authentifizierungsdetails enthält. Sie erstellen ein separates Secrets Manager Manager-Geheimnis für jedes Datenbankbenutzerkonto, mit dem sich der Proxy auf dem verbindet.

Themen

- Geheimnisse zur Verwendung mit RDS Proxy erstellen

Geheimnisse zur Verwendung mit RDS Proxy erstellen

Bevor Sie einen Proxy erstellen, müssen Sie zunächst mindestens einen geheimen Schlüssel erstellen, in dem Ihre Datenbankanmeldedaten gespeichert werden.

Konsole

So erstellen Sie ein Secret

1. Öffnen Sie die Secrets Manager Manager-Konsole unter https://console.aws.amazon.com/secretsmanager/.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Wählen Sie Credentials for Amazon RDS database aus.
4. Geben Sie einen Benutzernamen und ein Passwort ein. Die von Ihnen eingegebenen Anmeldeinformationen müssen mit den Anmeldeinformationen eines Datenbankbenutzers übereinstimmen, der in der zugehörigen RDS-Datenbank vorhanden ist. RDS Proxy verwendet diese Anmeldeinformationen, um im Namen von Anwendungen Verbindungen zur Datenbank zu authentifizieren und herzustellen.

   Bei Nichtübereinstimmung können Sie das Geheimnis so aktualisieren, dass es mit dem Datenbankkennwort übereinstimmt. Solange Sie den geheimen Schlüssel nicht aktualisieren, schlagen Versuche fehl, eine Verbindung über den Proxy herzustellen, aber Verbindungen, die andere gültige geheime Schlüssel verwenden, funktionieren trotzdem.

> **Note**
>
> Für RDS for SQL Server erfordert RDS Proxy unabhängig von den Sortierungseinstellungen der DB-Instance einen geheimen Schlüssel in Secrets Manager, bei dem Groß- und Kleinschreibung beachtet wird. Wenn Ihre Anwendung Benutzernamen mit unterschiedlichen Groß- und Kleinschreibung zulässt, z. B. „Admin“ und „admin“, müssen Sie für jeden Benutzer separate geheime Schlüssel erstellen. RDS Proxy unterstützt keine Benutzernamenauthentifizierung zwischen Client und Proxy, bei der Groß- und Kleinschreibung nicht berücksichtigt wird.
> Weitere Informationen über die Sortierung in SQL Server finden Sie in der Dokumentation für Microsoft SQL Server.

5. Wählen Sie für Datenbank die Amazon RDS-Datenbank aus, auf die das Secret zugreifen soll.
6. Geben Sie weitere Einstellungen für das Geheimnis ein und wählen Sie dann Speichern. Umfassende Anweisungen finden Sie im AWS Secrets Manager Benutzerhandbuch unter Ein AWS Secrets Manager Geheimnis erstellen.

AWS CLI

Wenn Sie über den einen Proxy erstellen AWS CLI, geben Sie die Amazon-Ressourcennamen (ARNs) der entsprechenden Geheimnisse an. Diesen Vorgang führen Sie für alle DB-Benutzerkonten aus, auf die der Proxy zugreifen kann. In der AWS Management Console wählen Sie die Geheimnisse anhand ihrer aussagekräftigen Namen aus.

- Verwenden Sie den Befehl create-secret, um ein Secrets Manager-Geheimnis für die Verwendung mit RDS Proxy zu erstellen:

```
aws secretsmanager create-secret \
  --name "secret_name" \
  --description "secret_description" \
  --region region_name \
  --secret-string '{"username":"db_user","password":"db_user_password"}'
```

- Sie können auch einen benutzerdefinierten Schlüssel erstellen, um Ihr Secrets Manager Manager-Geheimnis zu verschlüsseln. Der folgende Befehl erstellt einen Beispielschlüssel.

```
aws kms create-key --description "test-key" --policy '{
  "Id":"kms-policy",
```

```
"Version":"2012-10-17",
"Statement":
  [
    {
      "Sid":"Enable IAM User Permissions",
      "Effect":"Allow",
      "Principal":{"AWS":"arn:aws:iam::account_id:root"},
      "Action":"kms:*","Resource":"*"
    },
    {
      "Sid":"Allow access for Key Administrators",
      "Effect":"Allow",
      "Principal":
        {
          "AWS":
            ["$USER_ARN","arn:aws:iam:account_id::role/Admin"]
        },
      "Action":
        [
          "kms:Create*",
          "kms:Describe*",
          "kms:Enable*",
          "kms:List*",
          "kms:Put*",
          "kms:Update*",
          "kms:Revoke*",
          "kms:Disable*",
          "kms:Get*",
          "kms:Delete*",
          "kms:TagResource",
          "kms:UntagResource",
          "kms:ScheduleKeyDeletion",
          "kms:CancelKeyDeletion"
        ],
      "Resource":"*"
    },
    {
      "Sid":"Allow use of the key",
      "Effect":"Allow",
      "Principal":{"AWS":"$ROLE_ARN"},
      "Action":["kms:Decrypt","kms:DescribeKey"],
      "Resource":"*"
    }
  ]
```

```
}'
```

Mit den folgenden Befehlen werden beispielsweise Secrets Manager Manager-Geheimnisse für zwei Datenbankbenutzer erstellt:

```
aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}'

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}'
```

Verwenden Sie die folgenden Befehle, um diese mit Ihrem benutzerdefinierten AWS KMS Schlüssel verschlüsselten Geheimnisse zu erstellen:

```
aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}' \
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}' \
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id
```

Verwenden Sie den Befehl list-secrets, um die Geheimnisse zu sehen, die Ihrem AWS Konto gehören:

```
aws secretsmanager list-secrets
```

Wenn Sie mit der CLI einen Proxy erstellen, übergeben Sie die Amazon-Ressourcennamen (ARNs) eines oder mehrerer Secrets an den `--auth` Parameter. Das folgende Beispiel zeigt, wie Sie einen Bericht erstellen, der nur den Namen und den ARN jedes Geheimnisses enthält, das Ihrem AWS Konto gehört. In diesem Beispiel wird der `--output table` Parameter verwendet, der in AWS CLI Version 2 verfügbar ist. Wenn Sie AWS CLI Version 1 verwenden, verwenden Sie `--output text` stattdessen.

```
aws secretsmanager list-secrets --query '*[].[Name,ARN]' --output table
```

Verwenden Sie den get-secret-valueBefehl, um zu überprüfen, ob der geheime Schlüssel die richtigen Anmeldeinformationen im richtigen Format enthält. *your_secret_name*Ersetzen Sie es durch den Kurznamen oder den ARN des Geheimnisses.

```
aws secretsmanager get-secret-value --secret-id your_secret_name
```

Die Ausgabe enthält eine Zeile mit einem JSON-codierten Wert, der dem folgenden ähnelt:

```
...
"SecretString": "{\"username\":\"your_username\",\"password\":\"your_password\"}",
...
```

Konfiguration der IAM-Authentifizierung für RDS Proxy

Um die AWS Identity and Access Management (IAM-) Authentifizierung für RDS Proxy in Amazon RDS einzurichten, erstellen und konfigurieren Sie eine IAM-Richtlinie, die die erforderlichen Berechtigungen gewährt. RDS Proxy dient AWS Secrets Manager der sicheren Verwaltung von Datenbankanmeldedaten, sodass Anwendungen sich über den Proxy authentifizieren können, ohne die Anmeldeinformationen direkt verarbeiten zu müssen.

In diesem Thema werden die Schritte zur Konfiguration der IAM-Authentifizierung für den RDS-Proxy beschrieben, einschließlich der Erstellung der erforderlichen IAM-Richtlinie und deren Anfügung an eine IAM-Rolle.

> **Tip**
>
> Dieses Verfahren ist nur erforderlich, wenn Sie Ihre eigene IAM-Rolle erstellen möchten. Andernfalls kann RDS die erforderliche Rolle automatisch erstellen, wenn Sie den Proxy einrichten, sodass Sie diese Schritte überspringen können.

Voraussetzungen

Bevor Sie die IAM-Authentifizierung für den RDS-Proxy einrichten, stellen Sie sicher, dass Sie über Folgendes verfügen:

- AWS Secrets Manager— Mindestens ein gespeichertes Geheimnis, das Datenbankanmeldedaten enthält. Anweisungen zum Erstellen von Geheimnissen finden Sie unterthe section called “Datenbankanmeldedaten einrichten”.

- IAM-Berechtigungen — Eine IAM-Rolle oder ein IAM-Benutzer mit Berechtigungen zum Erstellen und Verwalten von IAM-Richtlinien, -Rollen und -Geheimnissen. AWS Secrets Manager

Eine IAM-Richtlinie für den Secrets Manager Manager-Zugriff erstellen

Damit RDS Proxy Datenbankanmeldedaten von Secrets Manager abrufen kann, erstellen Sie eine IAM-Rolle mit einer Richtlinie, die die erforderlichen Berechtigungen gewährt.

Konsole

Um eine Rolle für den Zugriff auf Ihre Geheimnisse zur Verwendung mit Ihrem Proxy zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Erstellen Sie eine Berechtigungsrichtlinie für die Rolle. Allgemeine Schritte finden Sie unter IAM-Richtlinien erstellen (Konsole).

   Fügen Sie diese Richtlinie in den JSON-Editor ein und nehmen Sie die folgenden Änderungen vor:

   - Ersetzen Sie die Vorgabe durch Ihre eigene Konto-ID.
   - `us-east-2`Ersetzen Sie es durch die Region, in der sich der Proxy befinden wird.
   - Ersetzen Sie die geheimen Namen durch die Namen, die Sie erstellt haben. Weitere Informationen finden Sie unter Angabe von KMS-Schlüsseln in IAM-Richtlinienanweisungen.
   - Ersetzen Sie die KMS-Schlüssel-ID durch die, mit der Sie die Secrets Manager Manager-Geheimnisse verschlüsselt haben, entweder den Standardschlüssel oder Ihren eigenen Schlüssel.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
```

```
                "arn:aws:secretsmanager:us-
east-2:111122223333:secret:secret_name_1",
                "arn:aws:secretsmanager:us-
east-2:111122223333:secret:secret_name_2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:111122223333:key/key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-
east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

3. Erstellen Sie die Rolle und fügen Sie ihr die Berechtigungsrichtlinie hinzu. Allgemeine Schritte finden Sie unter Eine Rolle erstellen, um Berechtigungen an einen AWS Dienst zu delegieren.

   Wählen Sie für den Entitätstyp Vertrauenswürdig die Option AWS Service aus. Wählen Sie unter Anwendungsfall die Option RDS aus und wählen Sie für den Anwendungsfall RDS — Rolle zur Datenbank hinzufügen aus.

4. Wählen Sie für Berechtigungsrichtlinien die Richtlinie aus, die Sie erstellt haben.

5. Geben Sie unter Vertrauenswürdige Entitäten auswählen die folgende Vertrauensrichtlinie für die Rolle ein:

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

AWS CLI

Um die Rolle mithilfe von zu erstellen AWS CLI, senden Sie die folgende Anfrage:

```
aws iam create-role \
  --role-name my_role_name \
  --assume-role-policy-document '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"Service":
["rds.amazonaws.com"]},"Action":"sts:AssumeRole"}]}'
```

Hängen Sie dann die Richtlinie an die Rolle an:

```
aws iam put-role-policy \
  --role-name my_role_name \
  --policy-name secret_reader_policy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
                }
            }
```

```
        }
    ]
}'
```

Nachdem die IAM-Rolle und die Berechtigungen konfiguriert sind, können Sie jetzt einen Proxy erstellen und ihn dieser Rolle zuordnen. Auf diese Weise kann der Proxy Datenbankanmeldedaten sicher aus Ihren Anwendungen abrufen AWS Secrets Manager und die IAM-Authentifizierung für Ihre Anwendungen aktivieren. Detaillierte Anweisungen finden Sie unter the section called “Einen Proxy erstellen”.

Einen Proxy für Amazon RDS — erstellen

Sie können einen Proxy einer DB-Instance von RDS für MariaDB, RDS für Microsoft SQL Server, RDS für MySQL oder RDS für PostgreSQL zuordnen.

Konsole

So erstellen Sie einen Proxy

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Wählen Sie Create proxy (Proxy erstellen).
4. Konfigurieren Sie die folgenden Einstellungen für Ihren Proxy.

Einstellung	Beschreibung
Motorenfamilie	Das Datenbank-Netzwerkprotokoll, das der Proxy erkennt, wenn er den Netzwerkverkehr zur und von der Datenbank interpretiert. **Note** Um RDS für PostgreSQL zu verwenden, stellen Sie sicher, dass die `postgres` Datenbank in Ihrer Instance beibehalten wird. Siehe Problembehandlung bei gelöschter `postgres` Datenbank.

Einstellung	Beschreibung
Proxy-ID	Ein Name, der innerhalb Ihrer AWS Konto-ID und Ihrer aktuellen AWS Region eindeutig ist.
Timeout für Client-Verbindungen im Leerlauf	Der Proxy schließt eine Client-Verbindung, wenn sie für einen bestimmten Zeitraum inaktiv bleibt. Standardmäßig sind dies 1.800 Sekunden (30 Minuten). Eine Verbindung befindet sich im Leerlauf, wenn die Anwendung nicht innerhalb der angegebenen Zeit nach Abschluss der vorherigen Anfrage eine neue Anfrage einreicht. Der Proxy hält die zugrunde liegende Datenbankverbindung geöffnet und gibt sie an den Verbindungspool zurück, sodass sie für neue Clientverbindungen verfügbar ist. Um veraltete Verbindungen proaktiv zu entfernen, reduzieren Sie den Timeout für inaktive Client-Verbindungen. Um die Verbindungskosten bei Arbeitslastspitzen zu minimieren, erhöhen Sie den Timeout.
Datenbank	Der , auf den über diesen Proxy zugegriffen werden soll. Die Liste enthält nur DB-Instances und Cluster mit kompatiblen Datenbank-Engines, Engine-Versionen und anderen Einstellungen. Wenn die Liste leer ist, erstellen Sie eine neue DB-Instance oder einen neuen Cluster, mit der/dem RDS Proxy kompatibel ist. Eine Schritt-für-Schritt-Anleitung hierzu finden Sie unter Erstellen einer Amazon RDS-DB-Instance. Versuchen Sie dann erneut, den Proxy zu erstellen.
Verbindungspool, maximale Anzahl an Verbindungen	Ein Wert zwischen 1 und 100, um den Prozentsatz des `max_connections` Limits zu definieren, den der RDS-Proxy verwenden kann. Wenn Sie nur einen Proxy mit dieser DB-Instance oder diesem Cluster verwenden möchten, legen Sie diesen Wert auf 100 fest. Weitere Informationen darüber, wie RDS Proxy diese Einstellung verwendet, finden Sie unterthe section called “MaxConnectionsPercent”.

Einstellung	Beschreibung
Filter für das Anheften von Sitzungen	Hindert den RDS-Proxy daran, bestimmte erkannte Sitzungszustände zu fixieren, wodurch die Standardsicherheitsmaßnahmen für Multiplexverbindungen umgangen werden. Derzeit unterstützt PostgreSQL diese Einstellung nicht, und die einzig verfügbare Option ist. `EXCLUDE_VARIABLE_SETS` Ihre Aktivierung kann dazu führen, dass sich Sitzungsvariablen von einer Verbindung auf andere auswirken, was zu Fehlern oder Korrektheitsproblemen führen kann, wenn Abfragen auf Sitzungsvariablen basieren, die außerhalb der aktuellen Transaktion gesetzt wurden. Verwenden Sie diese Option erst, nachdem Sie sich vergewissert haben, dass Ihre Anwendungen Datenbankverbindungen sicher gemeinsam nutzen können. Die folgenden Muster gelten als sicher: • `SET`Anweisungen, bei denen der effektive Wert der Sitzungsvariablen nicht geändert wird. Mit anderen Worten, es gibt keine Änderung an der Sitzungsvariablen. • Sie ändern den Wert der Sitzungsvariablen und führen eine Anweisung in derselben Transaktion aus. Weitere Informationen finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys.
Timeout bei Verbindungsausleihe	Wenn Sie erwarten, dass der Proxy alle verfügbaren Datenbankverbindungen verwendet, legen Sie die Wartezeit fest, bis ein Timeout-Fehler zurückgegeben wird. Sie können bis zu fünf Minuten angeben. Diese Einstellung gilt nur, wenn der Proxy die maximale Anzahl von Verbindungen erreicht hat und alle verwendet werden.

Einstellung	Beschreibung
Initialisierungsabfrage	(Optional) Fügen Sie eine Initialisierungsabfrage hinzu, oder ändern Sie die aktuelle. Sie können eine oder mehrere SQL-Anweisungen für die Ausführung durch den Proxy beim Öffnen jeder neuen Datenbankverbindung festlegen. Die Einstellung wird normalerweise zusammen mit SET Anweisungen verwendet , um sicherzustellen, dass jede Verbindung identische Einstellu ngen hat. Stellen Sie sicher, dass die von Ihnen hinzugefügte Abfrage gültig ist. Um mehrere Variablen in eine einzelne SET Anweisung aufzunehmen, verwenden Sie Kommatrennzeichen. Zum Beispiel: `SET variable1 =value1, variable2 =value2` Verwenden Sie für mehrere Anweisungen Semikola als Trennzeichen. **Important** Da Sie im Rahmen der Zielgruppenkonfiguration auf die Initialisierungsabfrage zugreifen können, ist sie nicht durch Authentifizierungs- oder kryptografische Methoden geschützt. Jeder, der Zugriff auf Ihre Proxy-Zielgruppenkonfiguration hat, kann die Initialisierungsab frage einsehen. Sie sollten dieser Option keine vertrauli chen Daten wie Passwörter oder langlebige Verschlüs selungsschlüssel hinzufügen.
AWS Identity and Access Management (IAM) -Rolle	Eine IAM-Rolle mit der Berechtigung zum Zugriff auf die Secrets Manager Manager-Geheimnisse, die die Anmeldeinformation en für Datenbankbenutzerkonten darstellen, die der Proxy verwenden kann. Alternativ können Sie eine neue IAM-Rolle aus dem erstellen. AWS Management Console

Einstellung	Beschreibung
Secrets Manager Geheimnisse	Wählen Sie mindestens ein Secrets Manager Manager-Geheimnis, das Datenbank-Benutzeranmeldedaten enthält, die es dem Proxy ermöglichen, auf den zuzugreifen.
Art der Client-Authentifizierung	Die Art der Authentifizierung, die der Proxy für Verbindungen von Clients verwendet. Die ausgewählte Option gilt für alle Secrets-Manager-Secrets, die Sie diesem Proxy zuordnen. Wenn Sie für jedes Geheimnis einen anderen Client-Authentifizierungstyp angeben müssen, erstellen Sie Ihren Proxy stattdessen mithilfe der AWS CLI oder der API.
IAM-Authentifizierung	Gibt an, ob die IAM-Authentifizierung für Verbindungen zu Ihrem Proxy erforderlich, zugelassen oder verboten werden soll. Die Option „zulassen“ ist nur für Proxys für RDS für SQL Server gültig. Die ausgewählte Option gilt für alle Secrets-Manager-Secrets, die Sie diesem Proxy zuordnen. Wenn Sie für jedes Geheimnis eine andere IAM-Authentifizierung angeben müssen, erstellen Sie Ihren Proxy stattdessen mithilfe der AWS CLI oder der API.
Transport Layer Security erforderlich	Erzwingt TLS/SSL für alle Client-Verbindungen. Der Proxy verwendet dieselbe Verschlüsselungseinstellung für seine Verbindung zur zugrunde liegenden Datenbank, unabhängig davon, ob die Client-Verbindung verschlüsselt oder unverschlüsselt ist.
Subnets	Dieses Feld ist bereits mit allen Subnetzen gefüllt, die mit Ihrer VPC verknüpft sind. Sie können alle Subnetze entfernen, die für den Proxy nicht benötigt werden, müssen jedoch mindestens zwei Subnetze belassen.

Einstellung	Beschreibung
VPC Security Group (VPC-Sicherheitsgruppe)	Wählen Sie eine vorhandene VPC-Sicherheitsgruppe oder erstellen Sie eine neue aus der AWS Management Console. Konfigurieren Sie die eingehenden Regeln so, dass Ihre Anwendungen auf den Proxy zugreifen können, und die Regeln für ausgehenden Datenverkehr, um den Datenverkehr von Ihren Datenbankzielen zuzulassen. **Note** Die Sicherheitsgruppe muss Verbindungen vom Proxy zur Datenbank zulassen. Sie dient sowohl für den Eingang von Ihren Anwendungen zum Proxy als auch für den Ausgang vom Proxy zur Datenbank. Wenn Sie beispielsweise dieselbe Sicherheitsgruppe sowohl für die Datenbank als auch für den Proxy verwenden, stellen Sie sicher, dass die Ressourcen innerhalb dieser Sicherheitsgruppe miteinander kommunizieren können. Wenn Sie eine gemeinsam genutzte VPC verwenden , vermeiden Sie es, die Standardsicherheitsgruppe für die VPC oder eine mit einem anderen Konto verknüpfte Sicherheitsgruppe zu verwenden. Wählen Sie stattdessen eine Sicherheitsgruppe aus, die zu Ihrem Konto gehört. Wenn keine vorhanden ist, erstellen Sie eine. Weitere Informationen finden Sie unter Arbeiten mit geteilten Inhalten VPCs. RDS stellt einen Proxy in mehreren Availability Zones bereit, um eine hohe Verfügbarkeit zu gewährleisten. Um die AZ-übergr eifende Kommunikation zu ermöglichen, muss die Network Access Control List (ACL) für Ihr Proxy-Subnetz Egress am Engine-Port und Ingress an allen Ports zulassen. Weitere Informationen zum Netzwerk ACLs finden Sie unter Steuern des Datenverkehrs zu Subnetzen über das Netzwerk. ACLs Wenn die Netzwerk-ACL für Ihren Proxy und Ihr Ziel identisch sind,

Einstellung	Beschreibung
	müssen Sie eine TCP-Protokoll-Ingress-Regel hinzufügen, bei der die Quelle auf die VPC-CIDR festgelegt ist. Sie müssen auch eine Engine-Port-spezifische TCP-Protokollausgangsregel hinzufügen, bei der das Ziel auf die VPC-CIDR festgelegt ist.
Aktivieren Sie die erweiterte Protokollierung	Aktivieren Sie diese Einstellung, um Proxykompatibilitäts- oder Leistungsprobleme zu beheben. Wenn diese Option aktiviert ist, protokolliert RDS Proxy detaillierte Leistungsinformationen, um Sie beim Debuggen des SQL-Verhaltens oder der Leistung und Skalierbarkeit von Proxyverbindungen zu unterstützen. Aktivieren Sie diese Einstellung nur zum Debuggen und stellen Sie sicher, dass angemessene Sicherheitsmaßnahmen zum Schutz vertraulicher Informationen in den Protokollen getroffen wurden. Um den Overhead zu minimieren, deaktiviert RDS Proxy diese Einstellung 24 Stunden nach der Aktivierung automatisch. Verwenden Sie es vorübergehend, um bestimmte Probleme zu beheben.

5. Wählen Sie Create proxy (Proxy erstellen).

AWS CLI

Um einen Proxy mit dem zu erstellen AWS CLI, rufen Sie den create-db-proxyBefehl mit den folgenden erforderlichen Parametern auf:

- `--db-proxy-name`
- `--engine-family`
- `--role-arn`
- `--auth`
- `--vpc-subnet-ids`

Bei `--engine-family`-Wert ist die Groß- und Kleinschreibung zu beachten.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-db-proxy \
    --db-proxy-name proxy_name \
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } \
    --auth ProxyAuthenticationConfig_JSON_string \
    --role-arn iam_role \
    --vpc-subnet-ids space_separated_list \
    [--vpc-security-group-ids space_separated_list] \
    [--require-tls | --no-require-tls] \
    [--idle-client-timeout value] \
    [--debug-logging | --no-debug-logging] \
    [--tags comma_separated_list]
```

Windows:

```
aws rds create-db-proxy ^
    --db-proxy-name proxy_name ^
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } ^
    --auth ProxyAuthenticationConfig_JSON_string ^
    --role-arn iam_role ^
    --vpc-subnet-ids space_separated_list ^
    [--vpc-security-group-ids space_separated_list] ^
    [--require-tls | --no-require-tls] ^
    [--idle-client-timeout value] ^
    [--debug-logging | --no-debug-logging] ^
    [--tags comma_separated_list]
```

Nachstehend finden Sie ein Beispiel für den JSON-Wert der `--auth`-Option. In diesem Beispiel wird auf jedes Geheimnis ein anderer Client-Authentifizierungstyp angewendet.

```
[
  {
    "Description": "proxy description 1",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret/1234abcd-12ab-34cd-56ef-1234567890ab",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_SCRAM_SHA_256"
  },
```

```
  {
    "Description": "proxy description 2",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret/1234abcd-12ab-34cd-56ef-1234567890cd",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_MD5"

  },

  {
    "Description": "proxy description 3",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122221111:secret/1234abcd-12ab-34cd-56ef-1234567890ef",
    "IAMAuth": "REQUIRED"
  }

]
```

Tip

Wenn Sie noch nicht wissen, welches Subnetz für den `--vpc-subnet-ids` Parameter verwendet werden IDs soll, finden Sie unter Netzwerkvoraussetzungen für RDS Proxy einrichten Beispiele, wie Sie sie finden können.

Note

Diese Sicherheitsgruppe muss den Zugriff auf die Datenbank zulassen, mit welcher der Proxy eine Verbindung herstellt. Dieselbe Sicherheitsgruppe wird für eingehenden Datenverkehr von Ihren Anwendungen zum Proxy und für ausgehenden Datenverkehr vom Proxy zur Datenbank verwendet. Angenommen, Sie verwenden dieselbe Sicherheitsgruppe für Ihre Datenbank und Ihren Proxy. Stellen Sie in diesem Fall sicher, dass Sie angeben, dass Ressourcen in dieser Sicherheitsgruppe mit anderen Ressourcen in derselben Sicherheitsgruppe kommunizieren können.
Wenn Sie eine freigegebene VPC verwenden, können Sie die Standardsicherheitsgruppe für die VPC oder eine zu einem anderen Konto gehörende Gruppe nicht verwenden. Wählen Sie eine Sicherheitsgruppe aus, die zu Ihrem Konto gehört. Wenn keine vorhanden ist,

erstellen Sie eine. Weitere Informationen zu dieser Einschränkung finden Sie unter Arbeiten mit Shared VPCs.

Um die richtigen Verknüpfungen für den Proxy zu erstellen, verwenden Sie auch den register-db-proxy-targetsBefehl. Angabe des `default`-Zielgruppennamens. RDS Proxy erstellt automatisch eine Zielgruppe mit diesem Namen, wenn Sie jeden Proxy erstellen.

```
aws rds register-db-proxy-targets
    --db-proxy-name value
    [--target-group-name target_group_name]
    [--db-instance-identifiers space_separated_list]  # rds db instances, or
    [--db-cluster-identifiers cluster_id]        # rds db cluster (all instances)
```

RDS-API

Um einen RDS-Proxy zu erstellen, rufen Sie den Amazon RDS-API-Vorgang Create aufDBProxy. Sie übergeben einen Parameter mit der AuthConfigDatenstruktur.

RDS Proxy erstellt automatisch eine Zielgruppe mit dem Namen `default`, wenn Sie die einzelnen Proxys erstellen. Sie ordnen der Zielgruppe einen zu, indem Sie die Funktion DBProxyZiele registrieren aufrufen.

Einen Proxy anzeigen

Nachdem Sie einen oder mehrere RDS-Proxys erstellt haben, können Sie sie in der AWS Management Console AWS CLI, oder der RDS-API anzeigen und verwalten. Sie können ihre Konfigurationsdetails überprüfen, die Leistung überwachen und festlegen, welche Proxys nach Bedarf geändert oder gelöscht werden sollen.

Damit Datenbankanwendungen den Datenverkehr über einen Proxy weiterleiten können, müssen Sie den Proxyendpunkt in der Verbindungszeichenfolge angeben.

Konsole

Um einen Proxy in der Konsole anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Wählen Sie den Proxynamen aus, um dessen Details anzuzeigen.
4. Auf der Detailseite zeigt der Abschnitt Zielgruppen, wie der Proxy mit einem bestimmten der RDS-DB-Instance verknüpft ist. Sie können zur Standardzielgruppenseite navigieren, um einen tieferen Überblick über diese Zuordnung zu erhalten, einschließlich der bei der Proxyerstellung definierten Konfigurationseinstellungen. Zu diesen Einstellungen gehören der maximale Verbindungsprozentsatz, das Timeout für die Verbindungsausleihe, die Engine-Familie und die Sitzungs-Pinning-Filter.

CLI

Verwenden Sie den describe-db-proxiesBefehl, um Ihren Proxy mit der CLI anzuzeigen. Standardmäßig gibt die Anfrage alle Proxys zurück, die Ihrem AWS Konto gehören. Um Details für einen einzelnen Proxy anzuzeigen, geben Sie seinen Namen mit dem Parameter `--db-proxy-name` an.

```
aws rds describe-db-proxies [--db-proxy-name proxy_name]
```

Verwenden Sie die folgenden Befehle, um weitere mit dem Proxy verknüpfte Informationen anzuzeigen.

```
aws rds describe-db-proxy-target-groups  --db-proxy-name proxy_name

aws rds describe-db-proxy-targets --db-proxy-name proxy_name
```

Verwenden Sie die folgende Befehlsfolge, um weitere Details zu den Elementen anzuzeigen, die mit dem Proxy verknüpft sind:

1. Führen Sie den Befehl aus, um eine Liste der Proxys abzurufen. describe-db-proxies
2. Um Verbindungsparameter wie den maximalen Prozentsatz der Verbindungen anzuzeigen, die der Proxy verwenden kann, führen describe-db-proxy-target Sie -groups aus. `--db-proxy-name` Verwenden Sie den Namen des Proxys als Parameterwert.
3. Führen Sie den Befehl aus, um die Details des anzuzeigen, der mit der zurückgegebenen Zielgruppe verknüpft ist describe-db-proxy-targets.

RDS-API

Verwenden Sie den Vorgang DBProxiesDescribe, um Ihre Proxys mithilfe der RDS-API anzuzeigen. Sie gibt Werte des DBProxyDatentyps zurück.

Um Einzelheiten zu den Verbindungseinstellungen für den Proxy anzuzeigen, verwenden Sie die Proxybezeichner aus diesem Rückgabewert mit der DBProxy TargetGroups Operation Describe. Sie gibt Werte des DBProxyTargetGroupDatentyps zurück.

Verwenden Sie den Vorgang Describe DBProxy Targets, um die mit dem Proxy verknüpfte RDS-Instance oder den Aurora-DB-Cluster zu sehen. Sie gibt Werte des DBProxyTarget-Datentyps zurück.

Verbinden mit einer Datenbank über RDS Proxy

Die Art und Weise, über einen Proxy oder eine Verbindung zur Datenbank eine Verbindung zu einer RDS-DB-Instance herzustellen, ist im Allgemeinen dieselbe. Weitere Informationen finden Sie unter Überblick über Proxy-Endpunkte.

Themen

- Herstellen einer Verbindung mit einem Proxy mithilfe der systemeigenen Authentifizierung
- Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung
- Überlegungen zum Herstellen einer Verbindung mit einem Proxy mit Microsoft SQL Server
- Überlegungen zum Herstellen einer Verbindung mit einem Proxy mit PostgreSQL

Herstellen einer Verbindung mit einem Proxy mithilfe der systemeigenen Authentifizierung

Gehen Sie wie folgt vor, um mithilfe der systemeigenen Authentifizierung eine Verbindung zu einem Proxy herzustellen:

1. Suchen Sie den Proxy-Endpunkt. Im finden Sie den AWS Management Console Endpunkt auf der Detailseite für den entsprechenden Proxy. Mit dem AWS CLI können Sie den describe-db-proxiesBefehl verwenden. Im folgenden Beispiel wird gezeigt, wie dies geschieht.

```
# Add --output text to get output as a simple tab-separated list.
$ aws rds describe-db-proxies --query '*[*].
{DBProxyName:DBProxyName,Endpoint:Endpoint}'
[
    [
```

```
        {
            "Endpoint": "the-proxy.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy"
        },
        {
            "Endpoint": "the-proxy-other-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-other-secret"
        },
        {
            "Endpoint": "the-proxy-rds-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-rds-secret"
        },
        {
            "Endpoint": "the-proxy-t3.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-t3"
        }
    ]
]
```

2. Geben Sie den Endpunkt als Hostparameter in der Verbindungszeichenfolge für Ihre Client-Anwendung an. Geben Sie beispielsweise den Proxy-Endpunkt als Wert für die Option `mysql -h` oder `psql -h` an.
3. Geben Sie denselben Datenbankbenutzernamen und dasselbe Passwort an, die Sie normalerweise verwenden.

Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung

Wenn Sie die IAM-Authentifizierung für RDS Proxy verwenden, müssen Sie die Benutzer Ihrer Datenbank auf die Authentifizierung mit ihrem regulären Benutzernamen und Passwort festlegen. Die IAM-Authentifizierung gilt für den Abruf von Benutzername und Passwort aus Secrets Manager durch RDS Proxy. Die Verbindung von RDS Proxy zur zugrunde liegenden Datenbank erfolgt nicht über IAM.

Um mithilfe der IAM-Authentifizierung eine Verbindung zum RDS-Proxy herzustellen, verwenden Sie dasselbe allgemeine Verbindungsverfahren wie für die IAM-Authentifizierung mit einem DB-Cluster einer RDS-DB-Instance. Allgemeine Informationen zur Verwendung von IAM finden Sie unter Sicherheit in Amazon RDS.

Zu den Hauptunterschieden bei der IAM-Nutzung für RDS Proxy gehören die folgenden:

- Sie konfigurieren nicht jeden einzelnen Datenbankbenutzer mit einem Autorisierungs-Plugin. Die Datenbankbenutzer haben weiterhin normale Benutzernamen und Passwörter innerhalb der Datenbank. Sie richten Secrets Manager-Geheimnisse ein, die diese Benutzernamen und Passwörter enthalten, und autorisieren RDS Proxy, die Anmeldeinformationen von Secrets Manager abzurufen.

  Die IAM-Authentifizierung gilt für die Verbindung zwischen Ihrem Clientprogramm und dem Proxy. Der Proxy authentifiziert sich dann bei der Datenbank mit den Anmeldeinformationen aus Benutzername und Passwort, die von Secrets Manager abgerufen wurden.
- Anstelle des Instance-, Cluster- oder Leser-Endpunkts geben Sie den Proxy-Endpunkt an. Weitere Informationen zum Proxy-Endpunkt finden Sie unter Herstellen einer Verbindung zu Ihrem DB-Instance- mithilfe der IAM-Authentifizierung.
- Im Anwendungsfall mit direkter Datenbank-IAM-Authentifizierung wählen Sie Datenbankbenutzer selektiv aus und konfigurieren sie so, dass sie mit einem speziellen Authentifizierungs-Plug-In identifiziert werden. Sie können dann mit IAM-Authentifizierung eine Verbindung zu diesen Benutzern herstellen.

  Im Proxy-Anwendungsfall müssen Sie dem Proxy die Geheimnisse bereitstellen, die den Benutzernamen und das Passwort eines Benutzers enthalten (native Authentifizierung). Anschließend stellen Sie mithilfe der IAM-Authentifizierung eine Verbindung mit dem Proxy her. Hierzu generieren Sie ein Authentifizierungstoken mit dem Proxy-Endpunkt und nicht mit dem Datenbankendpunkt. Sie verwenden auch einen Benutzernamen, der mit einem der Benutzernamen für die von Ihnen angegebenen Geheimnisse übereinstimmt.
- Sie müssen Transport Layer Security (TLS)/Secure Sockets Layer (SSL) verwenden, wenn Sie mit IAM-Authentifizierung eine Verbindung zu einem Proxy herstellen.

Sie können einem bestimmten Benutzer Zugriff auf den Proxy gewähren, indem Sie die IAM-Richtlinie ändern. Ein Beispiel folgt.

```
"Resource": "arn:aws:rds-db:us-east-2:1234567890:dbuser:prx-ABCDEFGHIJKL01234/db_user"
```

Überlegungen zum Herstellen einer Verbindung mit einem Proxy mit Microsoft SQL Server

Für die Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung verwenden Sie nicht das Passwortfeld. Stattdessen geben Sie die entsprechende Tokeneigenschaft für jeden

Datenbanktreibertyp im Tokenfeld an. Verwenden Sie beispielsweise die `accessToken`-Eigenschaft für JDBC oder die `sql_copt_ss_access_token`-Eigenschaft für ODBC. Oder verwenden Sie die `AccessToken` Eigenschaft für den SqlClient .NET-Treiber. Sie können die IAM-Authentifizierung nicht mit Clients verwenden, die keine Tokeneigenschaften unterstützen.

Unter bestimmten Bedingungen kann ein Proxy eine Datenbankverbindung nicht gemeinsam nutzen und fixiert stattdessen die Verbindung von Ihrer Client-Anwendung zum Proxy an eine dedizierte Datenbankverbindung. Weitere Informationen über diese Bedingungen finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys.

Überlegungen zum Herstellen einer Verbindung mit einem Proxy mit PostgreSQL

Wenn Sie einen neuen PostgreSQL-Datenbankbenutzer für die Verbindung mit dem RDS-Proxy erstellen, stellen Sie sicher, dass Sie die `CONNECT` Benutzerberechtigung für die Datenbank gewähren. Ohne dies kann der Benutzer keine Verbindung herstellen. Weitere Informationen finden Sie unter the section called “Hinzufügen eines neuen Datenbankbenutzers zu einer SQL Postgre-Datenbank bei Verwendung von Proxy RDS”.

Wenn ein Client eine Verbindung zu einer PostgreSQL-Datenbank herstellt, sendet er eine Startnachricht. Diese Nachricht enthält Paare von Parameternamen und Wertzeichenfolgen. Weitere Informationen finden Sie unter `StartupMessage` in den PostgreSQL-Nachrichtenformaten in der PostgreSQL-Dokumentation.

Wenn Sie eine Verbindung über einen RDS-Proxy herstellen, kann die Startnachricht die folgenden derzeit bekannten Parameter enthalten:

- `user`
- `database`

Die Startmeldung kann auch die folgenden zusätzlichen Laufzeitparameter enthalten:

- `application_name`
- `client_encoding`
- `DateStyle`
- `TimeZone`
- `extra_float_digits`
- `search_path`

Weitere Informationen zum PostgreSQL-Messaging finden Sie im Frontend/Backend-Protokoll in der PostgreSQL-Dokumentation.

Wenn Sie für PostgreSQL JDBC verwenden, empfehlen wir Folgendes, um Pinning zu vermeiden:

- Stellen Sie den JDBC-Verbindungsparameter `assumeMinServerVersion` mindestens auf `9.0` ein, um Pinning zu vermeiden. Dadurch wird verhindert, dass der JDBC-Treiber beim Verbindungsstart, wenn er ausgeführt wird, einen zusätzlichen Roundtrip durchführt. `SET extra_float_digits = 3`
- Setzen Sie den JDBC-Verbindungsparameter `ApplicationName` auf *`any/your-application-name`*, um Pinning zu vermeiden. Dadurch wird verhindert, dass der JDBC-Treiber beim Starten der Verbindung einen zusätzlichen Roundtrip ausführt, wenn er ausführ `SET application_name = "PostgreSQL JDBC Driver"`. Beachten Sie, dass der JDBC-Parameter `ApplicationName`, der PostgreSQL-`StartupMessage`-Parameter aber `application_name` ist.

Weitere Informationen finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys. Weitere Informationen zum Herstellen einer Verbindung mit JDBC finden Sie unter Verbinden mit der Datenbank in der PostgreSQL-Dokumentation.

Einen RDS Proxy verwalten

Dieser Abschnitt enthält Informationen zur Verwaltung des RDS Proxybetriebs und der Konfiguration. Diese Verfahren helfen Ihrer Anwendung, Datenbankverbindungen möglichst effizient zu nutzen und eine maximale Wiederverwendung der Verbindung zu erzielen. Je mehr Sie die Vorteile der Wiederverwendung von Verbindungen nutzen können, desto mehr CPU Speicherplatz können Sie sparen. Dies reduziert wiederum die Latenz für Ihre Anwendung und ermöglicht es der Datenbank, mehr Ressourcen für die Verarbeitung von Anwendungsanforderungen zu verwenden.

Themen

- Ändern eines RDS Proxy
- Hinzufügen eines neuen Datenbankbenutzers bei Verwendung von RDS Proxy
- Überlegungen zur RDS-Proxyverbindung
- Vermeiden Sie das Anheften eines RDS-Proxys
- Löschen eines RDS Proxys

Ändern eines RDS Proxy

Sie können bestimmte Einstellungen ändern, die einem Proxy zugeordnet sind, nachdem Sie den Proxy erstellt haben. Dazu ändern Sie den Proxy selbst, die zugehörige Zielgruppe oder beides. Jedem Proxy ist eine Zielgruppe zugeordnet.

AWS Management Console

> **Important**
>
> Die Werte in den Feldern Client authentication type (Client-Authentifizierungstyp) und IAM authentication (IAM-Authentifizierung) gelten für alle Secrets-Manager-Secrets, die diesem Proxy zugeordnet sind. Um für jedes Geheimnis unterschiedliche Werte anzugeben, ändern Sie Ihren Proxy, indem Sie stattdessen die AWS CLI oder die API verwenden.

So ändern Sie die Einstellungen für einen Proxy:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Wählen Sie in der Liste der Proxys den Proxy aus, dessen Einstellungen Sie ändern möchten, oder gehen Sie zur Detailseite.
4. Wählen Sie für Actions (Aktionen) die Option Modify (Ändern) aus.
5. Geben Sie die zu ändernden Eigenschaften ein, oder wählen Sie sie aus. Sie können folgende Formen angeben:

   - Proxy-Identifikator - Benennen Sie den Proxy um, indem Sie einen neuen Identifikator eingeben.
   - Zeitüberschreitung für Client-Verbindungsleerlauf— Geben Sie einen Zeitraum für das Timeout für die Clientleerlauf ein.
   - IAM-Rolle - Ändern Sie die IAM-Rolle, die zum Abrufen der Geheimnisse von Secrets Manager verwendet wird.
   - Secrets Manager-Secrets— Hinzufügen oder entfernen Sie Secrets Manager Geheimnisse. Diese Secrets entsprechen Datenbankbenutzernamen und Passwörtern.
   - Client authentication type (Client-Authentifizierungstyp) – (nur PostgreSQL) Ändern Sie den Authentifizierungstyp für Client-Verbindungen zum Proxy.

- IAM authentication (IAM-Authentifizierung) – Verlangen oder unterbinden Sie die IAM-Authentifizierung für Verbindungen mit dem Proxy.
- Require Transport Layer Security— Aktivieren oder deaktivieren Sie die Anforderung für Transport Layer Security (TLS).
- VPC-Sicherheitsgruppe - Fügen Sie VPC-Sicherheitsgruppen hinzu oder entfernen Sie sie, damit der Proxy sie verwenden kann.
- Aktivieren der erweiterten Protokollierung— Aktivieren oder deaktivieren Sie die erweiterte Protokollierung.

6. Wählen Sie Ändern aus.

Wenn Sie die Einstellungen, die Sie ändern möchten, in der Liste nicht gefunden haben, verwenden Sie das folgende Verfahren, um die Zielgruppe für den Proxy zu aktualisieren. Die Zielgruppe, die einem Proxy zugeordnet ist, steuert die Einstellungen für die physischen Datenbankverbindungen. Jedem Proxy ist eine Zielgruppe mit dem Namen `default` zugeordnet, die automatisch zusammen mit dem Proxy erstellt wird. Sie können die Standardzielgruppe nicht umbenennen.

Sie können die Zielgruppe nur über die Proxy-Detailseite ändern, nicht über die Liste auf der Seite Proxies (Proxys).

So ändern Sie die Einstellungen für eine Proxy-Zielgruppe:

1. Gehen Sie auf der Seite Proxys zur Detailseite für einen Proxy.
2. Wählen Sie für Target groups (Zielgruppen) den `default`-Link aus. Derzeit haben alle Proxys eine einzelne Zielgruppe mit dem Namen `default`.
3. Wählen Sie auf der Detailseite für die Standard-Zielgruppe die Option Modify (Ändern).
4. Wählen Sie neue Einstellungen für die Eigenschaften aus, die Sie ändern können:
   - Datenbank – Wählen Sie eine andere RDS-DB-Instance oder Cluster aus.
   - Maximale Verbindungen des Verbindungspools - Legen Sie fest, wie viel Prozent der maximal verfügbaren Verbindungen der Proxy nutzen kann.
   - Vortrags-Anheftungsfilter – Wählen Sie optional einen Vortrags-Anheftungsfilter aus. Dadurch werden die standardmäßigen Sicherheitsmaßnahmen für das Multiplexing von Datenbankverbindungen über Client-Verbindungen hinweg umgangen. Derzeit wird die Einstellung für PostgreSQL nicht unterstützt. Die einzige Wahl ist. `EXCLUDE_VARIABLE_SETS`

Die Aktivierung dieser Einstellung kann dazu führen, dass sich Sitzungsvariablen einer Verbindung auf andere Verbindungen auswirken. Dies kann zu Fehlern oder Problemen mit der Korrektheit führen, wenn Ihre Abfragen von Sitzungsvariablenwerten abhängen, die außerhalb der aktuellen Transaktion festgelegt wurden. Erwägen Sie, diese Option zu verwenden, nachdem Sie sich vergewissert haben, dass Ihre Anwendungen Datenbankverbindungen über mehrere Client-Verbindungen gemeinsam nutzen können.

Die folgenden Muster können als sicher angesehen werden:

- `SET`-Anweisungen, bei denen der effektive Wert der Sitzungsvariablen nicht geändert wird, d. h. dass keine Änderung an der Sitzungsvariablen vorgenommen wird.
- Sie ändern den Wert der Sitzungsvariablen und führen eine Anweisung in derselben Transaktion aus.

Weitere Informationen finden Sie unter Vermeiden Sie das Anheften eines RDS-Proxys.

- Connection borrow timeout— Passen Sie das Timeout-Intervall der Verbindung an. Diese Einstellung gilt, wenn für den Proxy bereits die maximale Anzahl von Verbindungen verwendet wird. Diese Einstellung legt fest, wie lange der Proxy wartet, bis eine Verbindung verfügbar ist, bevor ein Timeout-Fehler zurückgegeben wird.
- Initialisierungsabfrage. (Optional) Fügen Sie eine Initialisierungsabfrage hinzu, oder ändern Sie die aktuelle. Sie können eine oder mehrere SQL-Anweisungen für die Ausführung durch den Proxy beim Öffnen jeder neuen Datenbankverbindung festlegen. Die Einstellung wird in der Regel zusammen mit `SET` Anweisungen verwendet, um sicherzustellen, dass jede Verbindung identische Einstellungen hat. Stellen Sie sicher, dass die von Ihnen hinzugefügte Abfrage gültig ist. Um mehrere Variablen in eine einzelne `SET` Anweisung aufzunehmen, verwenden Sie Kommatrennzeichen. Zum Beispiel:

```
SET variable1=value1, variable2=value2
```

Verwenden Sie für mehrere Anweisungen Semikola als Trennzeichen.

Sie können bestimmte Eigenschaften, z. B. die Zielgruppenkennung und die Datenbank-Engine, nicht ändern.

5. Wählen Sie Modify target group (Zielgruppe ändern).

AWS CLI

Um einen Proxy mit dem zu ändern AWS CLI, verwenden Sie die Befehle modify-db-proxy-target-group modify-db-proxy, und deregister-db-proxy-targets. register-db-proxy-targets

Mit dem Befehl `modify-db-proxy` können Sie Eigenschaften wie die folgenden ändern:

- Der Satz von Secrets Manager-Secrets, die vom Proxy verwendet werden.
- Ob TLS erforderlich ist.
- Das Timeout für Client-Leerlauf.
- Ob zusätzliche Informationen aus SQL-Anweisungen zum Debuggen protokolliert werden sollen.
- Die IAM-Rolle, die zum Abrufen von Secrets Manager-Secrets verwendet wird.
- Die vom Proxy verwendeten Sicherheitsgruppen.

Das folgende Beispiel zeigt, wie ein vorhandener Proxy umbenannt wird.

```
aws rds modify-db-proxy --db-proxy-name the-proxy --new-db-proxy-name the_new_name
```

Um verbindungsbezogene Einstellungen zu ändern oder die Zielgruppe umzubenennen, verwenden Sie den Befehl `modify-db-proxy-target-group`. Derzeit haben alle Proxys eine einzelne Zielgruppe mit dem Namen `default`. Wenn Sie mit dieser Zielgruppe arbeiten, geben Sie den Namen des Proxys und `default` für den Namen der Zielgruppe an. Sie können die Standardzielgruppe nicht umbenennen.

Das folgende Beispiel zeigt, wie Sie zuerst die `MaxIdleConnectionsPercent`-Einstellung für einen Proxy überprüfen und dann mithilfe der Zielgruppe ändern.

```
aws rds describe-db-proxy-target-groups --db-proxy-name the-proxy

{
    "TargetGroups": [
        {
            "Status": "available",
            "UpdatedDate": "2019-11-30T16:49:30.342Z",
            "ConnectionPoolConfig": {
                "MaxIdleConnectionsPercent": 50,
                "ConnectionBorrowTimeout": 120,
                "MaxConnectionsPercent": 100,
```

```
                "SessionPinningFilters": []
            },
            "TargetGroupName": "default",
            "CreatedDate": "2019-11-30T16:49:27.940Z",
            "DBProxyName": "the-proxy",
            "IsDefault": true
        }
    ]
}

aws rds modify-db-proxy-target-group --db-proxy-name the-proxy --target-group-name
 default --connection-pool-config '
{ "MaxIdleConnectionsPercent": 75 }'

{
    "DBProxyTargetGroup": {
        "Status": "available",
        "UpdatedDate": "2019-12-02T04:09:50.420Z",
        "ConnectionPoolConfig": {
            "MaxIdleConnectionsPercent": 75,
            "ConnectionBorrowTimeout": 120,
            "MaxConnectionsPercent": 100,
            "SessionPinningFilters": []
        },
        "TargetGroupName": "default",
        "CreatedDate": "2019-11-30T16:49:27.940Z",
        "DBProxyName": "the-proxy",
        "IsDefault": true
    }
}
```

Mit den Befehlen `deregister-db-proxy-targets` und `register-db-proxy-targets` ändern Sie, welchen RDS-DB-Instances der Proxy über die Zielgruppe zugeordnet ist. Derzeit kann jeder Proxy eine Verbindung zu einem herstellen. Die Zielgruppe verfolgt die Verbindungsdetails für alle RDS-DB-Instances in einer Multi-AZ-Konfiguration, .

Das folgende Beispiel beginnt mit einem Proxy, der einem Aurora MySQL-Cluster mit dem Namen zugeordnet is `cluster-56-2020-02-25-1399`. Das Beispiel zeigt, wie der Proxy so geändert wird, dass er eine Verbindung zu einem anderen Cluster namens herstellen kan `provisioned-cluster`.

Wenn Sie mit einer RDS-DB-Instance arbeiten, geben Sie die `--db-instance-identifier-` Option an.

Im folgenden Beispiel wird ein Aurora MySQL-Proxy geändert. Ein Aurora PostgreSQL-Proxy hat Port 5432.

```
aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": [
        {
            "Endpoint": "instance-9814.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-9814"
        },
        {
            "Endpoint": "instance-8898.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-8898"
        },
        {
            "Endpoint": "instance-1018.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-1018"
        },
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "cluster-56-2020-02-25-1399"
        },
        {
            "Endpoint": "instance-4330.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-4330"
        }
    ]
}

aws rds deregister-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 cluster-56-2020-02-25-1399

aws rds describe-db-proxy-targets --db-proxy-name the-proxy
```

```
{
    "Targets": []
}

aws rds register-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 provisioned-cluster

{
    "DBProxyTargets": [
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "provisioned-cluster"
        },
        {
            "Endpoint": "gkldje.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "gkldje"
        },
        {
            "Endpoint": "provisioned-1.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "provisioned-1"
        }
    ]
}
```

RDS-API

Um einen Proxy mithilfe der RDS-API zu ändern, verwenden Sie die Operationen ModifyDBProxy, Modify DBProxy TargetGroup, Deregister DBProxy Targets und Register DBProxy Targets.

Mit `ModifyDBProxy` können Sie Eigenschaften wie die folgenden ändern:

- Der Satz von Secrets Manager-Secrets, die vom Proxy verwendet werden.
- Ob TLS erforderlich ist.
- Das Timeout für Client-Leerlauf.
- Ob zusätzliche Informationen aus SQL-Anweisungen zum Debuggen protokolliert werden sollen.
- Die IAM-Rolle, die zum Abrufen von Secrets Manager-Secrets verwendet wird.

- Die vom Proxy verwendeten Sicherheitsgruppen.

Mit `ModifyDBProxyTargetGroup` können Sie verbindungsbezogene Einstellungen ändern. Derzeit haben alle Proxys eine einzelne Zielgruppe mit dem Namen `default`. Wenn Sie mit dieser Zielgruppe arbeiten, geben Sie den Namen des Proxys und `default` für den Namen der Zielgruppe an. Sie können die Standardzielgruppe nicht umbenennen.

Mit `DeregisterDBProxyTargets` und ändern Sie`RegisterDBProxyTargets`, mit welchem der RDS-DB-Instance der Proxy über seine Zielgruppe verknüpft ist. Derzeit kann jeder Proxy eine Verbindung mit einer RDS-DB-Instance herstellen. Die Zielgruppe verfolgt die Verbindungsdetails für die RDS-DB-Instances in einer Multi-AZ-Konfiguration.

Hinzufügen eines neuen Datenbankbenutzers bei Verwendung von RDS Proxy

In einigen Fällen können Sie einer RDSDB-Instance oder einem , der einem Proxy zugeordnet ist, einen neuen Datenbankbenutzer hinzufügen. Fügen Sie dazu ein Secrets Manager-Secret hinzu, um die Anmeldeinformationen für diesen Benutzer zu speichern, oder verwenden Sie ein Secret neu. Führen Sie dazu die folgenden Schritte aus:

1. Erstellen Sie ein neues Secrets-Manager-Secret, indem Sie das unter beschriebene Verfahren verwende Datenbankanmeldedaten AWS Secrets Manager für RDS Proxy einrichten.
2. Aktualisieren Sie die IAM Rolle, um dem RDS Proxy Zugriff auf das neue Secrets Manager Manager-Geheimnis zu gewähren. Aktualisieren Sie dazu den Abschnitt Ressourcen der IAM Rollenrichtlinie.
3. Ändern Sie den RDS Proxy, um das neue Secrets Manager Manager-Geheimnis unter Secrets Manager-Geheimnisse hinzuzufügen.
4. Wenn der neue Benutzer an die Stelle eines vorhandenen Benutzers tritt, aktualisieren Sie die Anmeldeinformationen, die im Secrets Manager-Secret des Proxys für den vorhandenen Benutzer gespeichert sind.

Hinzufügen eines neuen Datenbankbenutzers zu einer SQL Postgre-Datenbank bei Verwendung von Proxy RDS

Wenn Sie Ihrer SQL Postgre-Datenbank einen neuen Benutzer hinzufügen, wenn Sie den folgenden Befehl ausgeführt haben:

```
REVOKE CONNECT ON DATABASE postgres FROM PUBLIC;
```

Gewähren Sie dem Benutzer `rdsproxyadmin` die `CONNECT`-Berechtigung, damit der Benutzer Verbindungen in der Zieldatenbank überwachen kann.

```
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

Sie können auch anderen Zieldatenbankbenutzern die Durchführung von Zustandsprüfungen ermöglichen, indem Sie im obigen Befehl `rdsproxyadmin` auf den Datenbankbenutzer ändern.

Ändern des Passworts für einen Datenbankbenutzer bei Verwendung von Proxy RDS

In einigen Fällen können Sie das Passwort für einen Datenbankbenutzer in einem RDS ändern, der einem Proxy zugeordnet ist. Aktualisieren Sie dazu das entsprechende Secrets Manager-Secret mit dem neuen Passwort.

Überlegungen zur RDS-Proxyverbindung

Konfigurieren der Verbindungseinstellungen

Um das Verbindungs-Pooling von RDS Proxy anzupassen, können Sie die folgenden Einstellungen ändern:

- IdleClientTimeout
- MaxConnectionsPercent
- MaxIdleConnectionsPercent
- ConnectionBorrowTimeout

IdleClientTimeout

Sie können angeben, wie lange eine Client-Verbindung inaktiv sein kann, bevor der Proxy sie schließt. Der Standardwert ist 1.800 Sekunden (30 Minuten).

Eine Client-Verbindung gilt als inaktiv, wenn die Anwendung innerhalb der angegebenen Zeit nach Abschluss der vorherigen Anforderung keine neue Anforderung absendet. Die zugrunde liegende Datenbankverbindung bleibt offen und wird an den Verbindungspool zurückgegeben. Somit ist sie für neue Clientverbindungen verfügbar. Wenn Sie möchten, dass der Proxy proaktiv veraltete

Verbindungen entfernt und so das Timeout für inaktive Client-Verbindungen verringert. Wenn Ihr Workload häufig Verbindungen mit dem Proxy herstellt, erhöhen Sie den Timeout für inaktive Client-Verbindungen, um die Kosten für den Verbindungsaufbau zu sparen.

Diese Einstellung wird durch das Feld Timeout für die Verbindung bei inaktiven Clients in der RDS-Konsole und durch die `IdleClientTimeout` Einstellung in der AWS CLI und der API dargestellt. Informationen dazu, wie Sie den Wert des Felds Idle client connection timeout (Zeitüberschreitung bei Client-Verbindungsinaktivität) in der RDS-Konsole ändern, finden Sie unter AWS Management Console. Informationen zum Ändern des Werts der `IdleClientTimeout` Einstellung finden Sie im CLI-Befehl modify-db-proxyoder in der API-Operation Modify DBProxy.

MaxConnectionsPercent

Sie können die Anzahl der Verbindungen beschränken, die ein RDS Proxy mit der Zieldatenbank herstellen kann. Sie geben das Limit als Prozentsatz der maximal für Ihre Datenbank verfügbaren Verbindungen an. Diese Einstellung wird durch das Feld Maximale Anzahl an Verbindungen für den Verbindungspool in der RDS-Konsole und durch die `MaxConnectionsPercent` Einstellung in der AWS CLI und der API dargestellt.

Der Wert `MaxConnectionsPercent` wird als Prozentsatz der Einstellung `max_connections` für die RDS-DB-Instance ausgedrückt, die von der Zielgruppe verwendet wird. Der Proxy erstellt nicht alle diese Verbindungen im Voraus. Diese Einstellung ermöglicht es dem Proxy, diese Verbindungen so einzurichten, wie es der Workload benötigt.

Beispielsweise legt RDS Proxy für ein registriertes Datenbankziel, für das `max_connections` auf 1000 und `MaxConnectionsPercent` auf 95 festgelegt ist, 950 Verbindungen als Obergrenze für gleichzeitige Verbindungen mit diesem Datenbankziel fest.

Ein häufiger Nebeneffekt, der auftritt, wenn Ihre Workload die maximale Anzahl zulässiger Datenbankverbindungen erreicht, ist ein Anstieg der allgemeinen Abfragelatenz in Verbindung mit einer Erhöhung der Metrik `DatabaseConnectionsBorrowLatency`. Sie können die aktuell verwendeten Datenbankverbindungen und die Gesamtzahl der zulässigen Datenbankverbindungen überwachen, indem Sie die Metriken `DatabaseConnections` und `MaxDatabaseConnectionsAllowed` vergleichen.

Beachten Sie für das Festlegen dieses Parameters die folgenden bewährten Methoden:

- Sorgen Sie für ausreichend Verbindungsspielraum für Änderungen des Workload-Musters. Es wird empfohlen, den Parameter mindestens 30 % über Ihrer zuletzt überwachten maximalen Nutzung einzustellen. Da RDS Proxy die Kontingente für Datenbankverbindungen auf mehrere Knoten

neu verteilt, können interne Kapazitätsänderungen mindestens 30 % Spielraum für zusätzliche Verbindungen erfordern, um erhöhte Ausleihlatenzen zu vermeiden.

- RDS Proxy reserviert eine bestimmte Anzahl von Verbindungen für die aktive Überwachung, um schnelles Failover, Weiterleitung von Datenverkehr und interne Operationen zu unterstützen. Die Metrik `MaxDatabaseConnectionsAllowed` umfasst diese reservierten Verbindungen nicht. Sie stellt die Anzahl der Verbindungen dar, die für den Workload verfügbar sind, und kann niedriger sein als der aus der Einstellung `MaxConnectionsPercent` abgeleitete Wert.

  Empfohlene Mindestwerte für `MaxConnectionsPercent`

  - db.t3.small: 30
  - db.t3.medium oder höher: 20

Informationen dazu, wie Sie den Wert des Felds Connection pool maximum connections (Max. Verbindungen Verbindungspool) in der RDS-Konsole ändern, finden Sie unter AWS Management Console. Informationen zum Ändern des Werts der `MaxConnectionsPercent` Einstellung finden Sie im modify-db-proxy-targetCLI-Befehl -group oder in der API-Operation Modify DBProxy TargetGroup.

Informationen zu Datenbankverbindungslimits finden Sie unter Maximale Anzahl von Datenbankverbindungen.

MaxIdleConnectionsPercent

Sie können die Anzahl inaktiver Datenbankverbindungen festlegen, die RDS Proxy im Verbindungspool behalten kann. Standardmäßig betrachtet der RDS-Proxy eine Datenbankverbindung in seinem Pool als inaktiv, wenn fünf Minuten lang keine Aktivität auf der Verbindung stattgefunden hat.

Der `MaxIdleConnectionsPercent` Wert wird als Prozentsatz der `max_connections` Einstellung für die Zielgruppe der RDS-DB-Instance ausgedrückt. Der Standardwert ist 50 Prozent von `MaxConnectionsPercent` und die Obergrenze ist der Wert `MaxConnectionsPercent`. Wenn beispielsweise 80 ist`MaxConnectionsPercent`, dann `MaxIdleConnectionsPercent` ist der Standardwert 40. Wenn der Wert von `MaxConnectionsPercent` nicht angegeben ist, `MaxIdleConnectionsPercent` ist für RDS für SQL Server der Wert 5 und für alle anderen Engines der Standardwert 50.

Bei einem hohen Wert lässt der Proxy einen hohen Prozentsatz an ungenutzten Datenbankverbindungen offen. Bei einem niedrigen Wert schließt der Proxy einen hohen Prozentsatz

von inaktiven Datenbankverbindungen. Wenn Ihre Workloads unvorhersehbar sind, sollten Sie erwägen, einen hohen Wert für `MaxIdleConnectionsPercent` festzulegen. Dies bedeutet, dass RDS-Proxy Aktivitätsspitzen verarbeiten kann, ohne viele neue Datenbankverbindungen zu öffnen.

Diese Einstellung wird durch die `MaxIdleConnectionsPercent` Einstellung von `DBProxyTargetGroup` in der AWS CLI und der API dargestellt. Informationen zum Ändern des Werts der `MaxIdleConnectionsPercent` Einstellung finden Sie im modify-db-proxy-targetCLI-Befehl -group oder in der API-Operation Modify DBProxy TargetGroup.

Informationen zu Datenbankverbindungslimits finden Sie unter Maximale Anzahl von Datenbankverbindungen.

ConnectionBorrowTimeout

Sie können angeben, wie lange RDS Proxy warten soll, bis eine Datenbankverbindung im Verbindungspool verfügbar ist, bevor ein Timeout-Fehler zurückgegeben wird. Standardmäßig sind 120 Sekunden festgelegt. Diese Einstellung greift, wenn die maximale Anzahl von Verbindungen erreicht ist und daher keine Verbindungen im Verbindungspool verfügbar sind. Dies gilt auch, wenn keine geeignete Datenbankinstanz zur Bearbeitung der Anfrage verfügbar ist, z. B. wenn ein Failover-Vorgang ausgeführt wird. Mit dieser Einstellung können Sie die beste Wartezeit für Ihre Anwendung festlegen, ohne das Abfragetimeout in Ihrem Anwendungscode zu ändern.

Diese Einstellung wird durch das Feld Timeout für die Verbindungsausleihe in der RDS-Konsole oder durch die `ConnectionBorrowTimeout` Einstellung `DBProxyTargetGroup` in der AWS CLI oder API dargestellt. Informationen dazu, wie Sie den Wert des Felds Connection borrow timeout (Zeitüberschreitung für die Verbindung) in der RDS-Konsole ändern, finden Sie unter AWS Management Console. Informationen zum Ändern des Werts der `ConnectionBorrowTimeout` Einstellung finden Sie im modify-db-proxy-targetCLI-Befehl -group oder in der API-Operation Modify DBProxy TargetGroup.

Client- und Datenbankverbindungen

Verbindungen von Ihrer Anwendung zum RDS-Proxy werden als Client-Verbindungen bezeichnet. Verbindungen von einem Proxy zur Datenbank sind Datenbankverbindungen. Bei Verwendung von RDS-Proxy werden Client-Verbindungen am Proxy beendet, während Datenbankverbindungen innerhalb des RDS-Proxys verwaltet werden.

Anwendungsseitiges Verbindungspooling kann den Vorteil bieten, dass der wiederkehrende Verbindungsaufbau zwischen Ihrer Anwendung und dem RDS-Proxy reduziert wird.

Berücksichtigen Sie die folgenden Konfigurationsaspekte, bevor Sie einen anwendungsseitigen Verbindungspool implementieren:

- Maximale Lebensdauer der Client-Verbindung: RDS Proxy erzwingt eine maximale Lebensdauer von Client-Verbindungen von 24 Stunden. Dieser Wert kann nicht konfiguriert werden. Konfigurieren Sie Ihren Pool mit einer maximalen Verbindungsdauer von weniger als 24 Stunden, um unerwartete Verbindungsabbrüche beim Client zu vermeiden.
- Timeout im Leerlauf der Client-Verbindung: Der RDS-Proxy erzwingt eine maximale Leerlaufzeit für Client-Verbindungen. Konfigurieren Sie Ihren Pool mit einem Timeout für inaktive Verbindungen, der niedriger ist als die Einstellung für das Leerlaufzeitlimit für Client-Verbindungen für den RDS-Proxy, um unerwartete Verbindungsabbrüche zu vermeiden.

Die maximale Anzahl von Client-Verbindungen, die in Ihrem anwendungsseitigen Verbindungspool konfiguriert sind, muss nicht auf die Einstellung max_connections für den RDS-Proxy beschränkt sein.

Das Pooling von Client-Verbindungen führt zu einer längeren Lebensdauer der Client-Verbindung. Wenn es bei Ihren Verbindungen zum Pinning kommt, kann das Pooling von Client-Verbindungen die Effizienz des Multiplexings verringern. Clientverbindungen, die im anwendungsseitigen Verbindungspool angeheftet, aber inaktiv sind, behalten weiterhin eine Datenbankverbindung bei und verhindern, dass die Datenbankverbindung von anderen Clientverbindungen wiederverwendet wird. Überprüfen Sie Ihre Proxyprotokolle, um zu überprüfen, ob es bei Ihren Verbindungen zu Pinning kommt.

> **Note**
>
> RDS-Proxy schließt Datenbankverbindungen nach 24 Stunden, wenn sie nicht mehr verwendet werden. Der Proxy führt diese Aktion unabhängig vom Wert der Einstellung für maximale Leerlaufverbindungen aus.

Vermeiden Sie das Anheften eines RDS-Proxys

Multiplexing ist effizienter, wenn Datenbankanforderungen nicht auf Statusinformationen aus früheren Anforderungen angewiesen sind. In diesem Fall kann RDS Proxy eine Verbindung zum Abschluss jeder Transaktion wiederverwenden. Beispiele für solche Zustandsinformationen sind die meisten Variablen und Konfigurationsparameter, die Sie durch `SET`-oder `SELECT`-Anweisungen ändern

können. SQL-Transaktionen auf einer Clientverbindung können standardmäßig zwischen zugrunde liegenden Datenbankverbindungen Multiplexing durchführen.

Ihre Verbindungen zum Proxy können einen Status eingeben, der als Pinning (Fixieren) bezeichnet wird. Wenn eine Verbindung angeheftet wird, verwendet jede spätere Transaktion dieselbe zugrunde liegende Datenbankverbindung, bis die Sitzung beendet ist. Andere Clientverbindungen können diese Datenbankverbindung auch erst dann wieder verwenden, wenn die Sitzung beendet ist. Die Sitzung wird beendet, wenn die Clientverbindung unterbrochen wird.

RDS Proxy heftet automatisch eine Clientverbindung an eine bestimmte DB-Verbindung an, wenn eine Sitzungsstatusänderung erkannt wird, die für andere Sitzungen nicht geeignet ist. Das Fixieren verringert die Effektivität der Wiederverwendung der Verbindung. Wenn alle oder fast alle Verbindungen fixiert sind, können Sie Ihren Anwendungscode oder Ihre Workload ändern, um dafür zu sorgen, dass Fixierungen weniger erforderlich sind.

Ihre Anwendung ändert beispielsweise eine Sitzungsvariable oder einen Konfigurationsparameter. In diesem Fall können sich spätere Anweisungen darauf verlassen, dass die neue Variable oder der neue Parameter wirksam ist. Wenn also RDS Proxy Anforderungen verarbeitet, um Sitzungsvariablen oder Konfigurationseinstellungen zu ändern, wird diese Sitzung an die DB-Verbindung fixiert. Auf diese Weise bleibt der Sitzungsstatus für alle späteren Transaktionen in derselben Sitzung gültig.

Bei Datenbank-Engines gilt diese Regel nicht für alle Parameter, die Sie festlegen können. RDS Proxy verfolgt bestimmte Anweisungen und Variablen. Daher fixiert RDS Proxy die Sitzung nicht, wenn Sie sie ändern. In diesem Fall verwendet RDS Proxy nur die Verbindung für andere Sitzungen erneut, die dieselben Werte für diese Einstellungen haben. Einzelheiten darüber, welche Daten RDS-Proxy für eine Datenbank-Engine verfolgt, finden Sie im Folgenden:

- Welche Daten RDS-Proxy für Datenbanken von RDS für SQL Server verfolgt
- Welche Daten RDS-Proxy für Datenbanken von RDS für MariaDB und RDS für MySQL verfolgt

Welche Daten RDS-Proxy für Datenbanken von RDS für SQL Server verfolgt

RDS Proxy verfolgt die folgenden SQL Server-Anweisungen:

- `USE`
- `SET ANSI_NULLS`
- `SET ANSI_PADDING`
- `SET ANSI_WARNINGS`

- SET ARITHABORT
- SET CONCAT_NULL_YIELDS_NULL
- SET CURSOR_CLOSE_ON_COMMIT
- SET DATEFIRST
- SET DATEFORMAT
- SET LANGUAGE
- SET LOCK_TIMEOUT
- SET NUMERIC_ROUNDABORT
- SET QUOTED_IDENTIFIER
- SET TEXTSIZE
- SET TRANSACTION ISOLATION LEVEL

Welche Daten RDS-Proxy für Datenbanken von RDS für MariaDB und RDS für MySQL verfolgt

RDS Proxy verfolgt die folgenden MariaDB- und MySQL-Anweisungen:

- DROP DATABASE
- DROP SCHEMA
- USE

RDS Proxy verfolgt die folgenden MySQL- und MariaDB-Variablen:

- AUTOCOMMIT
- AUTO_INCREMENT_INCREMENT
- CHARACTER SET (or CHAR SET)
- CHARACTER_SET_CLIENT
- CHARACTER_SET_DATABASE
- CHARACTER_SET_FILESYSTEM
- CHARACTER_SET_CONNECTION
- CHARACTER_SET_RESULTS
- CHARACTER_SET_SERVER

- `COLLATION_CONNECTION`
- `COLLATION_DATABASE`
- `COLLATION_SERVER`
- `INTERACTIVE_TIMEOUT`
- `NAMES`
- `NET_WRITE_TIMEOUT`
- `QUERY_CACHE_TYPE`
- `SESSION_TRACK_SCHEMA`
- `SQL_MODE`
- `TIME_ZONE`
- `TRANSACTION_ISOLATION (or TX_ISOLATION)`
- `TRANSACTION_READ_ONLY (or TX_READ_ONLY)`
- `WAIT_TIMEOUT`

> **Note**
>
> RDS Proxy verfolgt Änderungen an den `TRANSACTION_READ_ONLY` Variablen `TRANSACTION_ISOLATION` und, wenn Sie sie im Sitzungsbereich festlegen. Wenn Sie sie jedoch für den nächsten Transaktionsbereich festlegen, verbindet der RDS-Proxy die Verbindungen. Dieses Verhalten gilt unabhängig davon, ob Sie eine `SET` Anweisung oder eine `SET TRANSACTION` Anweisung verwenden, um diese Werte zu konfigurieren.

Minimieren des Fixierens

Die Leistungsoptimierung für RDS Proxy beinhaltet den Versuch, die Wiederverwendung von Verbindungen auf Transaktionsebene (Multiplexing) zu maximieren, indem das Fixieren minimiert wird.

Sie können das Fixieren wie folgt minimieren:

- Vermeiden Sie unnötige Datenbankanforderungen, die Anheften (Pinning) verursachen könnten.
- Legen Sie Variablen und Konfigurationseinstellungen konsistent über alle Verbindungen hinweg fest. Auf diese Weise verwenden spätere Sitzungen häufiger Verbindungen, die über diese speziellen Einstellungen verfügen.

Wenn für PostgreSQL jedoch eine Variable festgelegt wird, wird die Sitzung durch Pinning fixiert.

- Wenden Sie bei einer MySQL-Engine-Familiendatenbank einen Sitzungs-Pinning-Filter auf den Proxy an. Sie können bestimmte Arten von Operationen vom Fixieren der Sitzung ausnehmen, wenn Sie wissen, dass dies den korrekten Betrieb Ihrer Anwendung nicht beeinträchtigt.
- Sehen Sie sich anhand der CloudWatch Amazon-Metrik `DatabaseConnectionsCurrentlySessionPinned` an, wie häufig das Anheften erfolgt. Hinweise zu diesen und anderen CloudWatch-Metriken finden Sie unter Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch.
- Wenn Sie `SET`-Anweisungen verwenden, um eine identische Initialisierung für jede Clientverbindung durchzuführen, können Sie dies tun, während Sie das Multiplexing auf Transaktionsebene beibehalten. In diesem Fall verschieben Sie die Anweisungen, die den ursprünglichen Sitzungsstatus einrichten, in die Initialisierungsabfrage, die von einem Proxy verwendet wird. Diese Eigenschaft ist eine Zeichenfolge, die eine oder mehrere SQL-Anweisungen enthält, die durch Semikola getrennt sind.

  Beispielsweise können Sie eine Initialisierungsabfrage für einen Proxy definieren, der bestimmte Konfigurationsparameter festlegt. RDS Proxy wendet dann diese Einstellungen an, wenn eine neue Verbindung für diesen Proxy eingerichtet wird. Sie können die entsprechenden `SET`-Anweisungen aus Ihrem Anwendungscode entfernen, damit sie das Multiplexing auf Transaktionsebene nicht beeinträchtigen.

  Metriken zur Häufigkeit des Pinnings für einen Proxy finden Sie unter Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch.

Bedingungen, die für alle Engine-Familien zum Pinning führen

Der Proxy fixiert die Sitzung an der aktuellen Verbindung in den folgenden Situationen an, in denen Multiplexing unerwartetes Verhalten verursachen kann:

- Jede Anweisung mit einer Textgröße über 16 KB bewirkt, dass der Proxy die Sitzung fixiert.

Bedingungen, die das Fixieren für RDS für Microsoft SQL Server verursachen

Bei RDS für SQL Server verursachen die folgenden Interaktionen eine Fixierung:

- Verwendung mehrerer aktiver Ergebnissätze (MARS). Weitere Informationen zu MARS finden Sie in der SQL-Server-Dokumentation.
- Verwendung der Kommunikation mit verteiltem Transaktionskoordinator (DTC).
- Erstellen von temporären Tabellen, Transaktionen, Cursor oder vorbereiteten Anweisungen.
- Verwenden der folgenden SET-Anweisungen:
  - `SET ANSI_DEFAULTS`
  - `SET ANSI_NULL_DFLT`
  - `SET ARITHIGNORE`
  - `SET DEADLOCK_PRIORITY`
  - `SET FIPS_FLAGGER`
  - `SET FMTONLY`
  - `SET FORCEPLAN`
  - `SET IDENTITY_INSERT`
  - `SET NOCOUNT`
  - `SET NOEXEC`
  - `SET OFFSETS`
  - `SET PARSEONLY`
  - `SET QUERY_GOVERNOR_COST_LIMIT`
  - `SET REMOTE_PROC_TRANSACTIONS`
  - `SET ROWCOUNT`
  - `SET SHOWPLAN_ALL`, `SHOWPLAN_TEXT` und `SHOWPLAN_XML`
  - `SET STATISTICS`
  - `SET XACT_ABORT`

Bedingungen, die das Fixieren für RDS für MariaDB und RDS für MySQL verursachen

Für MariaDB und MySQL führen die folgenden Interaktionen auch zum Pinning:

- Die expliziten MySQL-Anweisungen `LOCK TABLE`, `LOCK TABLES` oder `FLUSH TABLES WITH READ LOCK` bewirken, dass der Proxy ein Pinning der Sitzung vornimmt.
- Durch Erstellen benannter Sperren mit `GET_LOCK` wird bewirkt, dass der Proxy ein Pinning der Sitzung vornimmt.

- Durch das Setzen einer Benutzer- oder Systemvariablen (mit einigen Ausnahmen) wird die Sitzung an den Proxy gebunden. Wenn dadurch die Wiederverwendung von Verbindungen erheblich eingeschränkt wird, können Sie `SET` Vorgänge so konfigurieren, dass das Anheften vermieden wird. Passen Sie dazu die Eigenschaft Sitzungsanheftungsfilter an. Weitere Informationen erhalten Sie unter Einen Proxy für Amazon RDS — erstellen und Ändern eines RDS Proxy.
- Beim Erstellen einer temporären Tabelle fixiert der Proxy die Sitzung. Auf diese Weise wird der Inhalt der temporären Tabelle während der gesamten Sitzung beibehalten, unabhängig von den Transaktionsgrenzen.
- Das Aufrufen der `FOUND_ROWS` Funktionen `ROW_COUNT` und führt manchmal zu einer Fixierung.
- Vorbereitete Anweisungen bewirken, dass der Proxy die Sitzung fixiert. Diese Regel bestimmt, ob die vorbereitete Anweisung SQL-Text oder das Binärprotokoll verwendet.
- RDS Proxy pingt keine Verbindungen an, wenn Sie SET LOCAL verwenden.
- Das Aufrufen von gespeicherten Prozeduren und gespeicherten Funktionen verursacht kein Pinning. RDS Proxy erkennt keine Änderungen des Sitzungsstatus, die aus solchen Aufrufen resultieren. Stellen Sie sicher, dass Ihre Anwendung den Sitzungsstatus in gespeicherten Routinen nicht ändert, wenn Sie darauf angewiesen sind, dass dieser Sitzungsstatus transaktionsübergreifend bestehen bleibt. Beispielsweise ist RDS Proxy derzeit nicht mit einer gespeicherten Prozedur kompatibel, die eine temporäre Tabelle erstellt, die für alle Transaktionen beibehalten wird.

Wenn Sie über eingehende Kenntnisse über das Verhalten Ihrer Anwendung verfügen, können Sie das Pinning-Verhalten für bestimmte Anwendungsanweisungen überspringen. Dazu wählen Sie beim Erstellen des Proxys die Option Sitzungs-Pinning-Filter. Derzeit können Sie das Sitzungs-Pinning für das Festlegen von Sitzungsvariablen und Konfigurationseinstellungen deaktivieren.

Bedingungen, die das Fixieren für RDS für PostgreSQL verursachen

Für PostgreSQL verursachen die folgenden Interaktionen eine Fixierung:

- `SET`Befehle verwenden.
- Verwendung von `EXECUTE` Befehlen `PREPARE` `DISCARDDEALLOCATE`,, oder zur Verwaltung vorbereiteter Anweisungen.
- Erstellen temporärer Sequenzen, Tabellen oder Ansichten.
- Cursor deklarieren.
- Der Sitzungsstatus wird verworfen.

- Abhören auf einem Benachrichtigungskanal.
- Laden eines Bibliotheksmoduls wie`auto_explain`.
- Manipulieren von Sequenzen mit Funktionen wie `nextval` und`setval`.
- Interaktion mit Sperren mithilfe von Funktionen wie `pg_advisory_lock` und`pg_try_advisory_lock`.

> **Note**
>
> RDS Proxy legt keine Sicherheitssperren auf Transaktionsebene fest`pg_advisory_xact_lock`, insbesondere nicht auf `pg_advisory_xact_lock_shared`, `pg_try_advisory_xact_lock`, und`pg_try_advisory_xact_lock_shared`.

- Einstellung eines Parameters oder Zurücksetzen eines Parameters auf seine Standardwerte. Insbesondere die Verwendung von `set_config` Befehlen `SET` und zum Zuweisen von Standardwerten zu Sitzungsvariablen.
- Das Aufrufen von gespeicherten Prozeduren und gespeicherten Funktionen verursacht kein Pinning. RDS Proxy erkennt keine Änderungen des Sitzungsstatus, die aus solchen Aufrufen resultieren. Stellen Sie sicher, dass Ihre Anwendung den Sitzungsstatus in gespeicherten Routinen nicht ändert, wenn Sie darauf angewiesen sind, dass dieser Sitzungsstatus transaktionsübergreifend beibehalten wird. Beispielsweise ist RDS Proxy derzeit nicht mit einer gespeicherten Prozedur kompatibel, die eine temporäre Tabelle erstellt, die für alle Transaktionen beibehalten wird.
- Der Sitzungsstatus wird verworfen. Wenn Sie Verbindungspooling-Bibliotheken verwenden, deren `DISCARD ALL` Abfrage als Reset-Abfrage konfiguriert ist, fixiert RDS Proxy Ihre Client-Verbindung bei der Freigabe. Dies verringert die Multiplexing-Effizienz des Proxys und kann zu unerwarteten Ergebnissen führen, da der `DISCARD ALL` Befehl die Sitzungsstatusverwaltung beeinträchtigen kann.

Löschen eines RDS Proxys

Sie können einen Proxy löschen, wenn Sie ihn nicht mehr benötigen. Oder Sie können einen Proxy löschen, wenn Sie die zugehörige DB-Instance oder den zugehörigen Cluster außer Betrieb nehmen.

AWS Management Console

So löschen Sie einen Proxy:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Wählen Sie den zu löschenden Proxy aus der Liste aus.
4. Wählen Sie Delete Proxy (Proxy löschen).

AWS CLI

Verwenden Sie den AWS CLI Befehl, um einen DB-Proxy zu löschen delete-db-proxy. Verwenden Sie auch den deregister-db-proxy-targetsBefehl, um zugehörige Verknüpfungen zu entfernen.

```
aws rds delete-db-proxy --name proxy_name
```

```
aws rds deregister-db-proxy-targets
    --db-proxy-name proxy_name
    [--target-group-name target_group_name]
    [--target-ids comma_separated_list]       # or
    [--db-instance-identifiers instance_id]       # or
    [--db-cluster-identifiers cluster_id]
```

RDS API

Um einen DB-Proxy zu löschen, rufen Sie die RDS API Amazon-Funktion D aufeleteDBProxy. Um verwandte Elemente und Verknüpfungen zu löschen, rufen Sie auch die Funktionen D eleteDBProxy TargetGroup und D eregisterDBProxy Targets auf.

Arbeiten mit Amazon RDS Proxy-Endpunkten

RDS-Proxy-Endpunkte bieten flexible und effiziente Möglichkeiten zur Verwaltung von Datenbankverbindungen, wodurch Skalierbarkeit, Verfügbarkeit und Sicherheit verbessert werden. Mit Proxy-Endpunkten können Sie:

- Vereinfachen Sie die Überwachung und Fehlerbehebung — Verwenden Sie mehrere Endpunkte, um Verbindungen von verschiedenen Anwendungen unabhängig voneinander zu verfolgen und zu verwalten.

Themen

- Überblick über Proxy-Endpunkte
- Limits für Proxy-Endpunkte
- Proxy-Endpunkte für Multi-AZ-DB-Cluster
- Zugriff auf RDS-Datenbanken über VPCs
- Erstellen eines Proxy-Endpunktes
- Anzeigen von Proxy-Endpunkten
- Ändern eines Proxy-Endpunkts
- Löschen eines Proxy-Endpunkts

Überblick über Proxy-Endpunkte

Arbeiten mit RDS-Proxy-Endpunkten umfassen die gleichen Arten von Verfahren wie bei RDS-Instance-Endpunkten. Wenn Sie mit RDS-Endpunkten nicht vertraut sind, finden Sie weitere Informationen unter Herstellen einer Verbindung mit einer DB-Instance, auf der die MySQL-Datenbank-Engine läuft und Herstellen einer Verbindung mit einer DB-Instance mit der PostgreSQL-Datenbank-Engine.

Wenn Sie einen Proxy-Endpunkt erstellen, können Sie ihn einer anderen Virtual Private Cloud (VPC) als der VPC des Proxys zuordnen. Auf diese Weise können Sie von einer anderen VPC aus eine Verbindung zum Proxy herstellen, z. B. von einer VPC, die von einer anderen Anwendung in Ihrer Organisation verwendet wird.

Informationen zu Limits im Zusammenhang mit Proxy-Endpunkten finden Sie unter Limits für Proxy-Endpunkte.

RDS-Proxyprotokolle stellen jedem Eintrag den Namen des zugehörigen Proxy-Endpunkts voran. Dies kann entweder der Name sein, den Sie für einen benutzerdefinierten Endpunkt angegeben haben, oder der spezielle Name `default` für den Standard-Lese-/Schreibendpunkt des Proxys.

Jeder Proxy-Endpunkt hat seinen eigenen Satz von Metriken. CloudWatch Überwachen Sie Metriken für alle Proxy-Endpunkte, einen bestimmten Endpunkt oder alle Lese-/Schreib- oder schreibgeschützten Endpunkte eines Proxys. Weitere Informationen finden Sie unter Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch.

Ein Proxy-Endpunkt verwendet denselben Authentifizierungsmechanismus wie der zugehörige Proxy. Für RDS Proxy richtet automatisch Berechtigungen und Autorisierungen für den benutzerdefinierten Endpunkt ein, die mit den Eigenschaften des zugehörigen Proxys übereinstimmen.

Limits für Proxy-Endpunkte

Für RDS-Proxy-Endpunkte gelten die folgenden Einschränkungen:

- Der RDS-Proxy-Standardendpunkt kann nicht geändert werden.
- Die maximale Anzahl von benutzerdefinierten Endpunkten für einen Proxy beträgt 20. Daher kann ein Proxy bis zu 21 Endpunkte haben: den Standard-Endpunkt plus 20, die Sie erstellen.
- Wenn Sie zusätzliche Endpunkte mit einem Proxy verknüpfen, bestimmt RDS Proxy automatisch, welche DB-Instances in Ihrem Cluster für jeden Endpunkt verwendet werden.

Wenn Sie einen Proxy erstellen, erstellt RDS automatisch einen VPC-Endpunkt für die sichere Kommunikation zwischen Anwendungen und der Datenbank. Der VPC-Endpunkt ist sichtbar und kann von der Amazon VPC-Konsole aus aufgerufen werden.

Durch das Hinzufügen eines neuen Proxy-Endpunkts wird ein AWS PrivateLink Schnittstellenendpunkt bereitgestellt. Wenn Sie Ihrem Proxy einen oder mehrere Endpunkte hinzufügen, fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Preise für RDS-Proxys.

Proxy-Endpunkte für Multi-AZ-DB-Cluster

Standardmäßig hat der Endpunkt, mit dem Sie eine Verbindung herstellen, wenn Sie RDS-Proxy mit einem Multi-AZ-DB-Cluster verwenden, Lese-/Schreibfähigkeit. Infolgedessen sendet dieser Endpunkt alle Anforderungen an die Writer-Instance des Clusters. Alle diese Verbindungen werden auf den `max_connections`-Wert für die Writer-Instance angerechnet. Wenn Ihr Proxy mit einem Multi-AZ-DB-Cluster verbunden ist, können Sie zusätzliche Lese-/Schreib- oder schreibgeschützte Endpunkte für diesen Proxy erstellen.

Sie können einen schreibgeschützten Endpunkt mit Ihrem Proxy für schreibgeschützte Abfragen verwenden. Sie verwenden diesen, wie Sie den Reader-Endpunkt für einen Multi-AZ-DB-Cluster nutzen. Auf diese Weise können Sie die Leseskalierbarkeit eines Multi-AZ-DB-Clusters mit einer oder mehreren Reader-DB-Instances nutzen. Sie können mehr gleichzeitige Abfragen ausführen und mehr gleichzeitige Verbindungen herstellen, indem Sie einen schreibgeschützten Endpunkt verwenden und Ihrem Multi-AZ-DB-Cluster nach Bedarf mehr Reader-DB-Instances hinzufügen. Diese Reader-

Endpunkte tragen dazu bei, die Leseskalierbarkeit Ihrer abfrageintensiven Anwendungen zu verbessern. Reader-Endpunkte helfen auch, die Verfügbarkeit Ihrer Verbindungen zu verbessern, wenn eine Reader-DB-Instance in Ihrem Cluster nicht verfügbar ist.

Reader-Endpunkte für Multi-AZ-DB-Cluster

Mit RDS Proxy können Sie Reader-Endpunkte erstellen und verwenden. Diese Endpunkte funktionieren jedoch nur für Proxys, die mit Multi-AZ-DB-Clustern verbunden sind. Wenn Sie die RDS-CLI oder API verwenden, sehen Sie möglicherweise das `TargetRole`-Attribut mit einem Wert von `READ_ONLY`. Sie können solche Proxys nutzen, indem Sie das Ziel eines Proxys von einer RDS-DB-Instance in einen Multi-AZ-DB-Cluster ändern.

Sie können schreibgeschützte Endpunkte mit der Bezeichnung Reader-Endpunkte erstellen und eine Verbindung herstellen, wenn Sie RDS-Proxy mit Multi-AZ-DB-Clustern verwenden.

Wie Reader-Endpunkte die Verfügbarkeit von Anwendungen unterstützen

In einigen Fällen ist möglicherweise eine Reader-Instance in Ihrem Cluster nicht verfügbar. In diesen Fällen können Verbindungen, die einen Reader-Endpunkt eines DB-Proxys verwenden, schneller wiederhergestellt werden als solche, die den Reader-Endpunkt des Multi-AZ-DB-Clusters verwenden. RDS-Proxy leitet Verbindungen nur an die verfügbare Reader-Instance in dem Cluster weiter. Es gibt keine Verzögerung aufgrund von DNS-Caching, wenn eine Instance nicht verfügbar ist.

Wenn die Verbindung Multiplexing durchführt, leitet RDS-Proxy nachfolgende Abfragen ohne Unterbrechung Ihrer Anwendung an eine andere Reader-Instance weiter. Wenn eine Reader-Instance den Status „Nicht verfügbar“ aufweist, werden alle Client-Verbindungen mit diesem Instance-Endpunkt geschlossen.

Wenn die Verbindung fixiert ist, gibt die nächste Abfrage der Verbindung einen Fehler zurück. Ihre Anwendung kann jedoch sofort wieder eine Verbindung zu demselben Proxy-Endpunkt herstellen. RDS-Proxy leitet die Verbindung zu einer anderen Reader-DB-Instance weiter, die sich im Status `available` befindet. Wenn Sie die Verbindung manuell wiederherstellen, überprüft RDS-Proxy nicht die Replikationsverzögerung zwischen der alten und neuen Reader-Instance.

Wenn Ihr Multi-AZ-DB-Cluster keine verfügbaren Reader-Instances hat, versucht RDS-Proxy, eine Verbindung mit einem Reader-Endpunkt herzustellen, sobald dieser verfügbar ist. Wenn innerhalb des Zeitraums der Zeitüberschreitung für die Verbindung keine Reader-Instance verfügbar wird, schlägt der Verbindungsversuch fehl. Wenn eine Reader-Instance verfügbar wird, ist der Verbindungsversuch erfolgreich.

Wie Reader-Endpunkte bei der Skalierbarkeit von Abfragen unterstützen

Reader-Endpunkte für einen Proxy helfen auf folgende Weisen bei der Skalierbarkeit von Abfragen für Multi-AZ-DB-Cluster:

- Wo praktisch, verwendet RDS-Proxy dieselbe Reader-DB-Instance für alle Abfragen unter Verwendung einer bestimmten Reader-Endpunktverbindung. Auf diese Weise kann eine Reihe von verwandten Abfragen in denselben Tabellen das Caching, die Planoptimierung usw. für eine bestimmte DB-Instance nutzen.
- Wenn eine Reader-DB-Instance nicht verfügbar ist, hängt die Auswirkung auf Ihre Anwendung davon ab, ob die Sitzung Multiplexing durchführt oder fixiert ist. Wenn die Sitzung Multiplexing durchführt, leitet RDS Proxy leitet alle nachfolgenden Abfragen an eine andere Reader-DB-Instance weiter, ohne dass Sie etwas unternehmen müssen. Wenn die Sitzung fixiert ist, bekommt Ihre Anwendung einen Fehler und muss sich erneut verbinden. Sie können sich sofort wieder mit dem Reader-Endpunkt verbinden und RDS Proxy leitet die Verbindung zu einer verfügbaren Reader-DB-Instance. Weitere Informationen zum Multiplexing und Pinning für Proxy-Sitzungen finden Sie unter Überblick über die RDS Proxy-Konzepte.

Zugriff auf RDS-Datenbanken über VPCs

Standardmäßig befinden sich die Komponenten Ihres -RDS-Technologie-Stacks alle in derselben Amazon-VPC. Nehmen wir zum Beispiel an, dass eine Anwendung, die auf einer EC2 Amazon-Instance ausgeführt wird, eine Verbindung zu einer Amazon RDS-DB-Instance herstellt. In diesem Fall müssen sich der Anwendungsserver und die Datenbank beide innerhalb derselben VPC befinden.

Mit RDS Proxy können Sie den Zugriff auf eine Amazon RDS-DB-Instance in einer VPC von Ressourcen in einer anderen VPC aus einrichten, z. B. EC2 Instances. Zum Beispiel könnte Ihre Organisation mehrere Anwendungen haben, die auf dieselben Datenbankressourcen zugreifen. Jede Anwendung könnte sich in einer eigenen VPC befinden.

Um den VPC-übergreifenden Zugriff zu aktivieren, erstellen Sie einen neuen Endpunkt für den Proxy. Der Proxy selbst befindet sich in derselben VPC wie die Amazon-RDS-DB-Instance. Der VPC-übergreifende Endpunkt befindet sich jedoch zusammen mit den anderen Ressourcen wie den Instances in der anderen VPC. EC2 Der VPC-übergreifende Endpunkt ist Subnetzen und Sicherheitsgruppen aus derselben VPC wie die EC2 und anderen Ressourcen zugeordnet. Mit diesen Zuordnungen können Sie von den Anwendungen aus eine Verbindung zum Endpunkt herstellen, die aufgrund der VPC-Einschränkungen sonst nicht auf die Datenbank zugreifen können.

In den folgenden Schritten wird erläutert, wie Sie einen VPC-übergreifenden Endpunkt erstellen und darauf über RDS Proxy zugreifen:

1. Erstellen Sie zwei VPCs oder wählen Sie zwei aus VPCs , die Sie bereits für die Arbeit mit RDS verwenden. Jede VPC sollte über eigene zugeordnete Netzwerkressourcen wie ein Internet-Gateway, Routing-Tabellen, Subnetze und Sicherheitsgruppen verfügen. Wenn Sie nur eine VPC haben, können Sie Erste Schritte mit Amazon RDS für die Schritte zum Einrichten einer anderen VPC zur erfolgreichen Verwendung von RDS konsultieren. Sie können auch Ihre bestehende VPC in der EC2 Amazon-Konsole überprüfen, um zu sehen, welche Arten von Ressourcen miteinander verbunden werden sollen.
2. Erstellen Sie einen DB-Proxy, der mit der Amazon-RDS-DB-Instance verknüpft ist, mit dem/der Sie eine Verbindung herstellen möchten. Folgen Sie dem Verfahren unter Einen Proxy für Amazon RDS — erstellen.
3. Auf der Seite Einzelheiten für Ihren Proxy in der RDS-Konsole unter dem Abschnitt Proxy-Endpunkte wählen Sie Endpunkt erstellen. Folgen Sie dem Verfahren unter Erstellen eines Proxy-Endpunktes.
4. Wählen Sie aus, ob der VPC-übergreifende Endpunkt zum Lesen/Schreiben oder schreibgeschützt sein soll.
5. Anstatt den Standard der gleichen VPC wie für die Amazon-RDS-DB-Instance zu akzeptieren, wählen Sie eine andere VPC. Diese VPC muss sich in derselben AWS Region befinden wie die VPC, in der sich der Proxy befindet.
6. Anstatt nun die Standardeinstellungen für Subnetze und Sicherheitsgruppen von derselben VPC wie für die Amazon-RDS-Instance zu akzeptieren, treffen Sie eine neue Auswahl. Wählen Sie diese basierend auf den Subnetzen und Sicherheitsgruppen aus der von Ihnen ausgewählten VPC.
7. Sie müssen keine der Einstellungen für die Secrets Manager-Secrets ändern. Dieselben Anmeldeinformationen funktionieren für alle Endpunkte für Ihren Proxy, unabhängig davon, in welcher VPC sich jeder Endpunkt befindet.
8. Warten Sie, bis der neue Endpunkt den Status verfügbar erreicht.
9. Notieren Sie sich den vollständigen Endpunktnamen. Dies ist der Wert, der auf *`Region_name`*`.rds.amazonaws.com` endet, den Sie als Teil der Verbindungszeichenfolge für Ihre Datenbankanwendung angeben.
10. Greifen Sie auf den neuen Endpunkt von einer Ressource in derselben VPC wie der Endpunkt zu. Eine einfache Möglichkeit, diesen Prozess zu testen, besteht darin, eine neue EC2 Instanz in dieser VPC zu erstellen. Melden Sie sich dann bei der EC2 Instanz an und führen Sie die `psql`

Befehle `mysql` oder aus, um mithilfe des Endpunktwerts in Ihrer Verbindungszeichenfolge eine Verbindung herzustellen.

Erstellen eines Proxy-Endpunktes

Gehen Sie folgendermaßen vor, um einen Proxy-Endpunkt zu erstellen:

Konsole

So erstellen Sie einen Proxy-Endpunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Klicken Sie auf den Namen des Proxys, für den Sie einen neuen Endpunkt erstellen möchten.

   Die Detailseite für diesen Proxy wird angezeigt.
4. Im Abschnitt Proxy-Endpunkte wählen Sie Proxy-Endpunkt erstellen.

   Das Fenster Proxy-Endpunkt erstellen wird angezeigt.
5. Für Name des Proxy-Endpunkts geben Sie einen beschreibenden Namen Ihrer Wahl ein.
6. Für Ziel-Rolle wählen Sie aus, ob der Endpunkt lese-/schreibgeschützt oder schreibgeschützt sein soll.

   Verbindungen, die Lese-/Schreibendpunkte verwenden, können alle Arten von Vorgängen ausführen, z. B. Anweisungen in der Datendefinitionssprache (DDL), Anweisungen zur Datenmanipulationssprache (DML) und Abfragen. Diese Endpunkte stellen immer eine Verbindung zur primären Instance des her. Sie können Endpunkte mit Lese-/Schreibzugriff für allgemeine Datenbankoperationen verwenden, wenn Sie nur einen einzelnen Endpunkt in Ihrer Anwendung verwenden. Sie können Lese-/Schreibendpunkte auch für Verwaltungsvorgänge, Online-Transaktionsverarbeitungsanwendungen () und extract-transform-load (OLTP) Jobs verwenden. ETL

   Verbindungen, die einen schreibgeschützten Endpunkt verwenden, können nur Abfragen durchführen. RDSDer Proxy kann für jede Verbindung zum Endpunkt eine der Reader-Instanzen verwenden. Auf diese Weise kann eine abfrageintensive Anwendung die Clustering-Fähigkeit eines Multi-AZ-DB-Clusters nutzen. Diese schreibgeschützten Verbindungen verursachen keinen

Overhead für die primäre Instance des Clusters. Auf diese Weise verlangsamen Ihre Berichts- und Analyseabfragen nicht die Schreibvorgänge Ihrer OLTP Anwendungen.

7. Wählen Sie für Virtual Private Cloud (VPC) die Standardeinstellung für den Zugriff auf den Endpunkt von denselben EC2 Instanzen oder anderen Ressourcen aus, die normalerweise für den Zugriff auf den Proxy oder die zugehörige Datenbank verwendet werden. Um den VPC Querzugriff für diesen Proxy einzurichten, wählen Sie einen VPC anderen als den Standard. Weitere Hinweise zum VPC Querzugriff finden Sie unterZugriff auf RDS-Datenbanken über VPCs.
8. Bei Subnetzen füllt RDS Proxy standardmäßig dieselben Subnetze aus wie der zugehörige Proxy. Um den Zugriff auf den Endpunkt auf nur einen Teil VPC des Adressbereichs zu beschränken, mit dem eine Verbindung hergestellt werden kann, entfernen Sie ein oder mehrere Subnetze.
9. Als VPCSicherheitsgruppe können Sie eine vorhandene Sicherheitsgruppe auswählen oder eine neue erstellen. RDSDer Proxy füllt standardmäßig dieselbe oder mehrere Sicherheitsgruppen wie der zugehörige Proxy aus. Wenn die Regeln für eingehenden und ausgehenden Datenverkehr für den Proxy für diesen Endpunkt geeignet sind, behalten Sie die Standardauswahl bei.

   Wenn Sie eine neue Sicherheitsgruppe erstellen möchten, geben Sie auf dieser Seite einen Namen für die Sicherheitsgruppe an. Bearbeiten Sie dann später die Sicherheitsgruppeneinstellungen von der EC2 Konsole aus.
10. Wählen Sie Proxy-Endpunkt erstellen.

AWS CLI

Verwenden Sie den AWS CLI create-db-proxy-endpointBefehl, um einen Proxy-Endpunkt zu erstellen.

Nutzen Sie die folgenden erforderlichen Parameter:

- `--db-proxy-name` *`value`*
- `--db-proxy-endpoint-name` *`value`*
- `--vpc-subnet-ids` *`list_of_ids`*. Trennen Sie das Subnetz IDs durch Leerzeichen. Sie geben nicht die ID von VPC sich selbst an.

Sie können auch die folgenden optionalen Parameter angeben:

- `--target-role { READ_WRITE | READ_ONLY }`. Dieser Parameter lautet standardmäßig `READ_WRITE`. Wenn der Proxy einem Multi-AZ-DB-Cluster zugeordnet ist, , der nur eine Writer-DB-Instance enthält, können Sie nicht angeben`READ_ONLY`. Weitere Informationen zur beabsichtigten Verwendung von schreibgeschützten Endpunkten mit , finden Sie unter. Reader-Endpunkte für Multi-AZ-DB-Cluster
- `--vpc-security-group-ids` *`value`*. Trennen Sie die Sicherheitsgruppe IDs durch Leerzeichen. Wenn Sie diesen Parameter weglassen, verwendet RDS Proxy die Standardsicherheitsgruppe für. VPC RDSDer Proxy bestimmt VPC anhand des SubnetzesIDs, das Sie für den `--vpc-subnet-ids` Parameter angeben.

Example

Im folgenden Beispiel wird ein Proxy-Endpunkt mit dem Namen erstell `my-endpoint`.

Für LinuxmacOS, oderUnix:

```
aws rds create-db-proxy-endpoint \
  --db-proxy-name my-proxy \
  --db-proxy-endpoint-name my-endpoint \
  --vpc-subnet-ids subnet_id subnet_id subnet_id ... \
  --target-role READ_ONLY \
  --vpc-security-group-ids security_group_id ]
```

Windows:

```
aws rds create-db-proxy-endpoint ^
  --db-proxy-name my-proxy ^
  --db-proxy-endpoint-name my-endpoint ^
  --vpc-subnet-ids subnet_id_1 subnet_id_2 subnet_id_3 ... ^
  --target-role READ_ONLY ^
  --vpc-security-group-ids security_group_id
```

RDS API

Verwenden Sie die Aktion RDS API C Endpoint, um einen reateDBProxy Proxy-Endpunkt zu erstellen.

Anzeigen von Proxy-Endpunkten

Gehen Sie wie folgt vor, um vorhandene Proxy-Endpunkte anzuzeigen:

Konsole

Anzeigen der Details für einen Proxy-Endpunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Wählen Sie in der Liste den Proxy aus, dessen Endpunkt Sie anzeigen möchten. Klicken Sie auf den Proxy-Namen, um die Detailseite anzuzeigen.
4. Wählen Sie im Abschnitt Proxy-Endpunkte den Endpunkt aus, den Sie anzeigen möchten. Klicken Sie auf seinen Namen, um die Detailseite aufzurufen.
5. Untersuchen Sie die Parameter, für deren Werten Sie sich interessieren. Sie können Eigenschaften wie die Folgenden überprüfen:

   - Ob der Endpunkt lese-/schreibgeschützt oder schreibgeschützt ist.
   - Die Endpunkt-Adresse, die Sie in einer Datenbank-Verbindungszeichenfolge verwenden.
   - DieVPC, Subnetze und Sicherheitsgruppen, die dem Endpunkt zugeordnet sind.

AWS CLI

Verwenden Sie den Befehl, um einen oder mehrere Proxy-Endpunkte anzuzeigen. AWS CLI describe-db-proxy-endpoints

Sie können die folgenden optionalen Parameter angeben:

- `--db-proxy-endpoint-name`
- `--db-proxy-name`

Das folgende Beispiel beschreibt den `my-endpoint`-Proxy-Endpunkt.

Example

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-proxy-endpoints \
  --db-proxy-endpoint-name my-endpoint
```

Windows:

```
aws rds describe-db-proxy-endpoints ^
  --db-proxy-endpoint-name my-endpoint
```

RDS API

Verwenden Sie die Operation RDS API D Endpoints, um einen oder mehrere escribeDBProxy Proxyendpunkte zu beschreiben.

Ändern eines Proxy-Endpunkts

Gehen Sie wie folgt vor, um Ihre Proxy-Endpunkte zu ändern:

Konsole

So ändern Sie einen oder mehrere Proxy-Endpunkte

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Proxies (Proxys).
3. Wählen Sie in der Liste den Proxy aus, dessen Endpunkt Sie ändern möchten. Klicken Sie auf den Proxy-Namen, um die Detailseite anzuzeigen.
4. Wählen Sie m Abschnitt Proxy-Endpunkte den Endpunkt aus, den Sie ändern möchten. Sie können ihn in der Liste auswählen oder auf seinen Namen klicken, um die Detailseite anzuzeigen.
5. Auf der Seite mit den Proxy-Details unter dem Abschnitt Proxy-Endpunkte wählen Sie Bearbeiten. Oder wählen Sie auf der Seite mit den Details zum Proxy-Endpunkt für Aktionen die Option Bearbeiten aus.
6. Ändern Sie wie gewünscht die Werte der Parameter.
7. Wählen Sie Save Changes.

AWS CLI

Verwenden Sie den AWS CLI modify-db-proxy-endpointBefehl mit den folgenden erforderlichen Parametern, um einen Proxy-Endpunkt zu ändern:

- `--db-proxy-endpoint-name`

Geben Sie Änderungen in den Endpunkteigenschaften an, indem Sie einen oder mehrere der folgenden Parameter verwenden:

- `--new-db-proxy-endpoint-name`
- `--vpc-security-group-ids`. Trennen Sie die Sicherheitsgruppe IDs durch Leerzeichen.

Das folgende Beispiel benennt den `my-endpoint`-Proxy-Endpunkt in `new-endpoint-name`.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint \
  --new-db-proxy-endpoint-name new-endpoint-name
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint ^
  --new-db-proxy-endpoint-name new-endpoint-name
```

RDS API

Verwenden Sie den Vorgang RDS API M Endpoint, um einen odifyDBProxy Proxy-Endpunkt zu ändern.

Löschen eines Proxy-Endpunkts

Gehen Sie folgendermaßen vor, um einen Endpunkt für Ihren Proxy zu löschen:

> **Note**
>
> Sie können den Standard-Proxy-Endpunkt, den RDS Proxy automatisch für jeden Proxy erstellt, nicht löschen.
> Wenn Sie einen Proxy löschen, löscht RDS Proxy automatisch alle zugehörigen Endpunkte.

Konsole

Um einen Proxy-Endpunkt zu löschen, verwenden Sie AWS Management Console

1. Wählen Sie im Navigationsbereich Proxies (Proxys).
2. Wählen Sie in der Liste den Proxy aus, dessen Endpunkt Sie löschen möchten. Klicken Sie auf den Proxy-Namen, um die Detailseite anzuzeigen.
3. Im Abschnitt Proxy-Endpunkte den Endpunkt aus, den Sie löschen möchten. Sie können einen oder mehrere Endpunkte in der Liste auswählen oder auf den Namen eines einzelnen Endpunkts klicken, um die Detailseite anzuzeigen.
4. Auf der Seite mit den Proxy-Details unter dem Abschnitt Proxy-Endpunkte, wählen Sie Löschen. Oder wählen Sie auf der Detailseite des Proxy-Endpunkts für Aktionen die Option Löschen aus.

AWS CLI

Um einen Proxy-Endpunkt zu löschen, führen Sie den delete-db-proxy-endpointBefehl mit den folgenden erforderlichen Parametern aus:

- `--db-proxy-endpoint-name`

Der folgende Befehl löscht den Proxy-Endpunkt mit dem Namen `my-endpoint`.

Für LinuxmacOS, oderUnix:

```
aws rds delete-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint
```

Windows:

```
aws rds delete-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint
```

RDS API

Um einen Proxyendpunkt mit dem zu löschen RDSAPI, führen Sie den Vorgang D eleteDBProxy Endpoint aus. Geben Sie den Namen des Proxy-Endpunkts für den `DBProxyEndpointName`-Parameter an.

Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch

Sie können den RDS-Proxy mithilfe von Amazon überwachen CloudWatch. CloudWatch sammelt und verarbeitet Rohdaten aus den Proxys in lesbare Metriken near-real-time. Um diese Metriken in der CloudWatch Konsole zu finden, wählen Sie Metrics, dann RDS und dann Per-Proxy-Metriken aus. Weitere Informationen finden Sie unter Verwenden von CloudWatch Amazon-Metriken im CloudWatch Amazon-Benutzerhandbuch.

> **Note**
>
> RDS veröffentlicht diese Metriken für jede zugrunde liegende EC2 Amazon-Instance, die einem Proxy zugeordnet ist. Ein einzelner Proxy kann von mehr als einer EC2 Instance bedient werden. Verwenden Sie CloudWatch Statistiken, um die Werte für einen Proxy für alle zugehörigen Instanzen zu aggregieren.
> Einige dieser Metriken sind möglicherweise erst nach der ersten erfolgreichen Verbindung durch einen Proxy sichtbar.

In den RDS Proxy-Protokollen wird jedem Eintrag der Name des zugehörigen Proxy-Endpunkts vorangestellt. Dieser Name kann der Name sein, den Sie für einen benutzerdefinierten Endpunkt angegeben haben, oder der spezielle Name `default` für den Standardendpunkt eines Proxys, der read/write Anfragen ausführt.

Alle RDS Proxy-Metriken befinden sich in der Gruppe `proxy`.

Jeder Proxy-Endpunkt hat seine eigenen CloudWatch Metriken. Sie können die Nutzung jedes Proxy-Endpunkts unabhängig überwachen. Weitere Informationen zu den Proxy-Endpunkten finden Sie unter Arbeiten mit Amazon RDS Proxy-Endpunkten.

Sie können die Werte für jede Metrik mit einem der folgenden Dimensionssätze aggregieren. Zum Beispiel können Sie, indem Sie den `ProxyName`-Dimensionssatz verwenden, den gesamten Datenverkehr für einen bestimmten Proxy analysieren. Durch die Verwendung der anderen Dimensionssätze können Sie die Metriken auf verschiedene Arten aufteilen. Sie können die Metriken auf der Grundlage der verschiedenen Endpunkte oder Zieldatenbanken jedes Proxys oder auf der Grundlage des schreibgeschützten Datenverkehrs zu jeder Datenbank aufteilen. read/write

- -Dimensionssatz 1: `ProxyName`
- -Dimensionssatz 2: `ProxyName`, `EndpointName`

- -Dimensionssatz 3: `ProxyName`, `TargetGroup`, `Target`
- -Dimensionssatz 4: `ProxyName`, `TargetGroup`, `TargetRole`

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
`AvailabilityPercentage`	Der Prozentsatz der Zeit, für die die Zielgruppe in der durch die Dimension angegebenen Rolle verfügbar war. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute	Dimension set 4
`ClientConnections`	Die aktuelle Anzahl von Clientverbindungen. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute	Dimension set 1, Dimension set 2
`ClientConnectionsClosed`	Die Anzahl der geschlossenen Clientverbindungen. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2
`ClientConnectionsInSetup`	Die aktuelle Anzahl der geöffneten Client-Verbindungen, deren Einrichtung jedoch noch nicht	1 Minute	Dimension set 1, Dimension set 2

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
	abgeschlossen ist. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist Summe.		
`ClientConnectionsNoTLS`	Die aktuelle Anzahl von Clientverbindungen ohne TLS (Transport Layer Security). Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute	Dimension set 1, Dimension set 2
`ClientConnectionsReceived`	Die Anzahl der empfangenen Clientverbindungsanforderungen. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2
`ClientConnectionsSetupFailedAuth`	Die Anzahl der Clientverbindungsversuche, die aufgrund einer fehlerhaften Authentifizierungs- oder TLS-Konfiguration fehlgeschlagen sind. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
`ClientConnectionsSetupSucceeded`	Die Anzahl der Clientverbindungen, die erfolgreich mit einem Authentifizierungsmechanismus mit oder ohne TLS hergestellt wurden. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2
`ClientConnectionsTLS`	Die aktuelle Anzahl von Clientverbindungen mit TLS. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute	Dimension set 1, Dimension set 2
`DatabaseConnectionRequests`	Die Anzahl der Anforderungen zum Erstellen einer Datenbankverbindung. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionRequestsWithTLS`	Die Anzahl der Anforderungen zum Erstellen einer Datenbankverbindung mit TLS. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 3, Dimension set 4

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
DatabaseConnections	Die aktuelle Anzahl von Datenbank verbindungen. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute	Dimension set 1, Dimension set 3, Dimension set 4
DatabaseConnectionsBorrowLatency	Die Zeit in Mikrosekunden, die der überwachte Proxy benötigt, um eine Datenbankverbindung zu erhalten. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute und höher	Dimension set 1, Dimension set 2
DatabaseConnectionsCurrentlyBorrowed	Die aktuelle Anzahl von Datenbank verbindungen im Ausleihstatus. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute	Dimension set 1, Dimension set 3, Dimension set 4
DatabaseConnectionsCurrentlyInTransaction	Die aktuelle Anzahl der Datenbank verbindungen in einer Transaktion. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute	Dimension set 1, Dimension set 3, Dimension set 4

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
DatabaseConnectionsCurrentlySessionPinned	Die aktuelle Anzahl von Datenbankverbindungen, die derzeit aufgrund von Operationen in Clientanforderungen, die den Sitzungsstatus ändern, angeheftet, bzw. fixiert, werden. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute	Dimension set 1, Dimension set 3, Dimension set 4
DatabaseConnectionsSetupFailed	Die Anzahl der fehlgeschlagenen Datenbankverbindungsanforderungen. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute und höher	Dimension set 1, Dimension set 3, Dimension set 4
DatabaseConnectionsSetupSucceeded	Die Anzahl der erfolgreich aufgebauten Datenbankverbindungen mit oder ohne TLS. Die nützlichste Statistik für diese Metrik ist Sum.	1 Minute und höher	Dimension set 1, Dimension set 3, Dimension set 4

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
`DatabaseConnectionsWithTLS`	Die aktuelle Anzahl von Datenbankverbindungen mit TLS. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute	Dimension set 1, Dimension set 3, Dimension set 4
`MaxDatabaseConnectionsAllowed`	Die maximal zulässige Anzahl von Datenbankverbindungen. Diese Metrik wird jede Minute gemeldet. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute	Dimension set 1, Dimension set 3, Dimension set 4
`QueryDatabaseResponseLatency`	Die Zeit in Mikrosekunden, die die Datenbank benötigt hat, um auf die Abfrage zu antworten. Die nützlichste Statistik für diese Metrik ist `Average`.	1 Minute und höher	Dimension set 1, Dimension set 2, Dimension set 3, Dimension set 4

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
`QueryRequests`	Die Anzahl der empfangenen Abfragen. Eine Abfrage, die mehrere Anweisungen enthält, wird als eine Abfrage gezählt. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2
`QueryRequestsNoTLS`	Die Anzahl der Abfragen, die von Nicht-TLS-Verbindungen empfangen wurden. Eine Abfrage, die mehrere Anweisungen enthält, wird als eine Abfrage gezählt. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2
`QueryRequestsTLS`	Die Anzahl der Abfragen, die von TLS-Verbindungen empfangen wurden. Eine Abfrage, die mehrere Anweisungen enthält, wird als eine Abfrage gezählt. Die nützlichste Statistik für diese Metrik ist `Sum`.	1 Minute und höher	Dimension set 1, Dimension set 2

Metrik	Beschreibung	Gültiger Zeitraum	CloudWatch Dimensionssatz
`QueryResponseLatency`	Die Zeit in Mikrosekunden zwischen dem Abrufen einer Abfrageanforderung und dem darauf antwortenden Proxy. Die nützlichste Statistik für diese Metrik ist `Average`.	1 Minute und höher	Dimension set 1, Dimension set 2

Sie finden Protokolle der RDS-Proxyaktivitäten CloudWatch unter AWS Management Console. Jeder Proxy hat einen Eintrag auf der Seite Log groups (Protokollgruppen) .

> **Important**
>
> Diese Protokolle sind zur menschlichen Verwendung zur Fehlerbehebung und nicht für den programmatischen Zugriff bestimmt. Das Format und der Inhalt der Protokolle können geändert werden.
> Insbesondere enthalten ältere Protokolle keine Präfixe, die den Endpunkt für jede Anfrage angeben. In neueren Protokollen wird jedem Eintrag der Name des zugehörigen Proxy-Endpunkts vorangestellt. Dieser Name kann der Name sein, den Sie für einen benutzerdefinierten Endpunkt angegeben haben, oder der spezielle Name `default` für Anforderungen mit dem Standard-Endpunkt eines Proxys.

Mit RDS Proxy-Ereignissen arbeiten

Ein Ereignis weist auf eine Änderung in einer Umgebung hin, z. B. in einer AWS Umgebung, einem Dienst oder einer Anwendung von einem Software-as-a-Service (SaaS) -Partner. Es kann sich auch um eine Ihrer eigenen benutzerdefinierten Anwendungen oder Dienste handeln. Amazon RDS Amazon generiert beispielsweise ein Ereignis, wenn Sie einen RDS Proxy erstellen oder ändern. Amazon RDS liefert Ereignisse nahezu EventBridge in Echtzeit an Amazon. Im Folgenden finden Sie

eine Liste von RDS Proxy-Ereignissen, die Sie abonnieren können, sowie ein Beispiel für ein RDS Proxy-Ereignis.

Weitere Informationen zum Arbeiten mit Ereignissen finden Sie in den folgenden Abschnitten:

- Anweisungen zum Anzeigen von Ereignissen mithilfe von AWS Management Console AWS CLI RDSAPI, oder finden Sie unter RDSAmazon-Ereignisse anzeigen.
- Informationen zur Konfiguration von Amazon RDS Amazon für das Senden von Ereignissen an EventBridge finden Sie unter Erstellen einer Regel, die bei einem Amazon RDS-Ereignis ausgelöst wird.

RDSProxy-Ereignisse

Die folgende Tabelle zeigt die Ereigniskategorie und eine Liste von Ereignissen, wenn ein RDS Proxy der Quelltyp ist.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS- EVENT -0204	RDSmodifizierter DB-Proxy *name*.	None
Konfigurationsänderung	RDS- EVENT -0207	RDShat den Endpunkt des DB-Proxys geändert *name*.	None
Konfigurationsänderung	RDS- EVENT -0213	RDShat das Hinzufügen der DB-Instance erkannt und sie automatisch der Zielgruppe des DB-Proxys hinzugefügt *name*.	None
Konfigurationsänderung	RDS- EVENT -0214	RDShat das Löschen der DB-Instance erkannt *name* und hat es automatisch aus der Zielgruppe entfernt *name* des DB-Proxys *name*.	None

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Konfigurationsänderung	RDS- EVENT -0215	RDShat das Löschen des DB-Clusters erkannt *name* und hat es automatisch aus der Zielgruppe entfernt *name* des DB-Proxys *name*.	None
Erstellung	RDS- EVENT -0203	RDShat einen DB-Proxy erstellt *name*.	None
Erstellung	RDS- EVENT -0206	RDSEndpunkt erstellt *name* für DB-Proxy *name*.	None
Löschung	RDS- EVENT -0205	RDShat den DB-Proxy gelöscht *name*.	None
Löschung	RDS- EVENT -0208	RDSgelöschter Endpunkt *name* für DB-Proxy *name*.	None
Ausfall	RDS- EVENT -0243	RDSKapazität für den Proxy konnte nicht bereitgestellt werden *name* weil in Ihren Subnetzen nicht genügend IP-Adressen verfügbar sind: *name*. Um das Problem zu beheben, stellen Sie sicher, dass Ihre Subnetze über die Mindestanzahl ungenutzter IP-Adressen verfügen, wie in der RDS Proxy-Dokumentation empfohlen.	Informationen zur Bestimmung der empfohlenen Anzahl für Ihre Instance-Klasse finden Sie unter Planen der Kapazität von IP-Adressen.

Kategorie	RDS-Ereignis-ID	Fehlermeldung	Hinweise
Ausfall	RDS- -0275 EVENT	RDShat einige Verbindungen zum DB-Proxy gedrosselt *name*. Die Anzahl der gleichzeitigen Verbindungsanfragen vom Client zum Proxy hat das Limit überschritten.	None

Im Folgenden finden Sie ein Beispiel für ein RDS Proxy-Ereignis im JSON Format. Das Ereignis zeigt, dass der Endpunkt mit dem Namen `my-endpoint` des angegebenen RDS Proxys RDS geändert wurde`my-rds-proxy`. Die Ereignis-ID lautet RDS - EVENT -0207.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Proxy Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
     "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PROXY",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "RDS modified endpoint my-endpoint of DB Proxy my-rds-proxy.",
    "SourceIdentifier": "my-endpoint",
    "EventID": "RDS-EVENT-0207"
  }
}
```

Fehlersuche für RDS Proxy

Im Folgenden finden Sie Ideen zur Fehlerbehebung für einige häufig auftretende Probleme mit dem RDS-Proxy sowie Informationen zu CloudWatch Protokollen für RDS Proxy.

In den RDS Proxy-Protokollen wird jedem Eintrag der Name des zugehörigen Proxy-Endpunkts vorangestellt. Dieser Name kann derjenige sein, den Sie für einen benutzerdefinierten Endpunkt angegeben haben. Es kann sich auch um den speziellen Namen `default` für den Standardendpunkt eines Proxys handeln, der Lese-/Schreibanforderungen ausführt. Weitere Informationen zu den Proxy-Endpunkten finden Sie unter Arbeiten mit Amazon RDS Proxy-Endpunkten.

Themen

- Überprüfen der Konnektivität für einen Proxy
- Häufige Probleme und Lösungen
- Behebung von RDS-Proxyproblemen mit RDS for MySQL
- Behebung von RDS-Proxyproblemen mit RDS for PostgreSQL

Überprüfen der Konnektivität für einen Proxy

Sie können die folgenden Befehle verwenden, um zu überprüfen, ob alle Komponenten wie Proxy-, Datenbank- und Recheninstanzen in der Verbindung miteinander kommunizieren können.

Untersuchen Sie den Proxy selbst mithilfe des describe-db-proxiesBefehls. Untersuchen Sie auch die zugehörige Zielgruppe mit dem Befehl describe-db-proxy-target-groups. Überprüfen Sie, ob die Details der Ziele mit der RDS-DB-Instance übereinstimmen, die Sie dem Proxy zuordnen möchten. Verwenden Sie Befehle wie die folgenden.

```
aws rds describe-db-proxies --db-proxy-name $DB_PROXY_NAME
aws rds describe-db-proxy-target-groups --db-proxy-name $DB_PROXY_NAME
```

Um zu überprüfen, ob der Proxy eine Verbindung zur zugrunde liegenden Datenbank herstellen kann, überprüfen Sie mithilfe des describe-db-proxy-targetsBefehls die in den Zielgruppen angegebenen Ziele. Verwenden Sie einen Befehl wie den folgenden.

```
aws rds describe-db-proxy-targets --db-proxy-name $DB_PROXY_NAME
```

Die Ausgabe des describe-db-proxy-targetsBefehls enthält ein `TargetHealth` Feld. Sie können die Felder `State`, `Reason` und `Description` in `TargetHealth` überprüfen, um festzustellen, ob der Proxy mit der zugrunde liegenden DB-Instance kommunizieren kann.

- Der Wert `State` für `AVAILABLE` gibt an, dass der Proxy eine Verbindung mit der DB-Instance herstellen kann.
- Der Wert `State` für `UNAVAILABLE` weist auf ein temporäres oder dauerhaftes Verbindungsproblem hin. Überprüfen Sie in diesem Fall die Felder `Reason` und `Description`. Wenn beispielsweise der `Reason` den Wert `PENDING_PROXY_CAPACITY` hat, versuchen Sie erneut , eine Verbindung herzustellen, nachdem der Proxy seinen Skalierungsvorgang beendet hat. Wenn `Reason` den Wert `UNREACHABLE`, `CONNECTION_FAILED` oder `AUTH_FAILURE` hat, verwenden Sie die Erläuterung aus dem Feld `Description`, um das Problem leichter zu diagnostizieren.
- Das Feld `State` hat möglicherweise für kurze Zeit einen Wert von `REGISTERING`, bevor Sie zu `AVAILABLE` oder `UNAVAILABLE` wechseln.

Wenn der folgende Netcat-Befehl (`nc`) erfolgreich ist, können Sie von der EC2 Instanz oder einem anderen System aus, auf dem Sie angemeldet sind, auf den Proxy-Endpunkt zugreifen. Dieser Befehl meldet einen Fehler, wenn Sie sich nicht in derselben VPC wie der Proxy und die zugehörige Datenbank befinden. Möglicherweise können Sie sich direkt bei der Datenbank anmelden, ohne sich in derselben VPC zu befinden. Sie können sich jedoch nur am Proxy anmelden, wenn Sie sich in derselben VPC befinden.

```
nc -zx MySQL_proxy_endpoint 3306

nc -zx PostgreSQL_proxy_endpoint 5432
```

Sie können die folgenden Befehle verwenden, um sicherzustellen, dass Ihre EC2 Instanz über die erforderlichen Eigenschaften verfügt. Insbesondere muss die VPC für die EC2 Instance mit der VPC für den der identisch sein, mit dem der Proxy eine Verbindung herstellt.

```
aws ec2 describe-instances --instance-ids your_ec2_instance_id
```

Untersuchen Sie die für den Proxy verwendeten Secrets Manager-Secrets.

```
aws secretsmanager list-secrets
aws secretsmanager get-secret-value --secret-id your_secret_id
```

Stellen Sie sicher, dass das von angezeigte `SecretString` Feld als JSON-Zeichenfolge codiert `get-secret-value` ist, die die Felder und enthält. `username` `password` Das folgende Beispiel zeigt das Format des `SecretString`-Feldes.

```
{
  "ARN": "some_arn",
  "Name": "some_name",
  "VersionId": "some_version_id",
  "SecretString": '{"username":"some_username","password":"some_password"}',
  "VersionStages": [ "some_stage" ],
  "CreatedDate": some_timestamp
}
```

Häufige Probleme und Lösungen

In diesem Abschnitt werden einige häufig auftretende Probleme und mögliche Lösungen bei der Verwendung von RDS Proxy beschrieben.

Überprüfen Sie nach dem Ausführen des `aws rds describe-db-proxy-targets` CLI-Befehls Folgendes`Proxy does not have any registered credentials`, falls in der `TargetHealth` Beschreibung angegeben:

- Es sind Anmeldeinformationen registriert, damit der Benutzer auf den Proxy zugreifen kann.
- Die IAM-Rolle für den Zugriff auf das vom Proxy verwendete Secrets Manager-Geheimnis ist gültig.

Beim Erstellen eines DB-Proxys oder beim Herstellen einer Verbindung mit einem DB-Proxy können die folgenden RDS-Ereignisse auftreten.

Kategorie	RDS-Ereignis-ID	Beschreibung
Ausfall	RDS-EVENT-0243	RDS konnte keine Kapazität für den Proxy bereitstellen, da in Ihren Subnetzen nicht genügend IP-Adressen verfügbar sind. Stellen Sie sicher, dass Ihre Subnetze die Mindestanzahl unbenutzt er IP-Adressen aufweisen

Kategorie	RDS-Ereignis-ID	Beschreibung
		, um das Problem zu lösen. Informationen zur Bestimmung der empfohlenen Anzahl für Ihre Instance-Klasse finden Sie unter Planen der Kapazität von IP-Adressen.
Ausfall	RDS-EVENT-0275	RDS hat einige Verbindungen zum DB-Proxy gedrosselt. *name* Die Anzahl der gleichzeitigen Verbindungsanfragen vom Client zum Proxy hat das Limit überschritten.

Beim Erstellen eines neuen Proxys oder beim Herstellen einer Verbindung mit einem Proxy können die folgenden Probleme auftreten.

Fehler	Ursachen oder Problemumgehungen
`403: The security token included in the request is invalid`	Wählen Sie eine vorhandene IAM-Rolle aus, anstatt eine neue zu erstellen.

Behebung von RDS-Proxyproblemen mit RDS for MySQL

Beim Herstellen einer Verbindung mit einem MySQL-Proxy können die folgenden Probleme auftreten.

Fehler	Ursachen oder Problemumgehungen
`ERROR 1040 (HY000): Connections`	Die Rate der Verbindungsanforderungen vom Client zum Proxy hat den Grenzwert überschritten.

Fehler	Ursachen oder Problemumgehungen
`rate limit exceeded (limit_value )`	
`ERROR 1040 (HY000): IAM authentication rate limit exceeded`	Die Anzahl der gleichzeitigen Anforderungen mit IAM-Authentifizierung vom Client an den Proxy hat den Grenzwert überschritten.
`ERROR 1040 (HY000): Number simultane ous connectio ns exceeded (limit_value )`	Die Anzahl der gleichzeitigen Verbindungsanforderungen vom Client zum Proxy hat den Grenzwert überschritten.
`ERROR 1045 (28000): Access denied for user 'DB_USER'@'%' (usi password: YES)`	Das vom Proxy verwendete Secrets Manager-Geheimnis stimmt nicht mit dem Benutzernamen und dem Passwort eines vorhandenen Datenbank benutzers überein. Aktualisieren Sie entweder die Anmeldeinformation en im Secrets Manager-Geheimnis oder stellen Sie sicher, dass der Datenbankbenutzer vorhanden ist und über das gleiche Passwort wie im Geheimnis verfügt.
`ERROR 1105 (HY000): Unknown error`	Es ist ein unbekannter Fehler aufgetreten.
`ERROR 1231 (42000): Variable ''charact er_set_cl ient'' can't be set to the value of value`	Der für den `character_set_client` -Parameter gesetzte Wert ist ungültig. Beispielsweise ist der Wert `ucs2` ungültig, da er den MySQL-Server zum Absturz bringen kann.

Fehler	Ursachen oder Problemumgehungen
`ERROR 3159 (HY000): This RDS Proxy requires TLS connections.`	Sie haben die Einstellung Transport Layer Security erfordern im Proxy aktiviert, aber Ihre Verbindung enthielt den Parameter `ssl-mode=DISABLED` im MySQL-Client. Führen Sie eine der folgenden Aufgaben aus: • Deaktivieren Sie die Einstellung Transport Layer Security erfordern. • Stellen Sie eine Verbindung mit der Datenbank mit der Mindestei nstellung `ssl-mode=REQUIRED` im MySQL-Client her.
`ERROR 2026 (HY000): SSL connection error: Internal Server Error`	Der TLS-Handshake an den Proxy ist fehlgeschlagen. Einige mögliche Gründe sind: • SSL ist erforderlich, aber der Server unterstützt dies nicht. • Es ist ein interner Serverfehler aufgetreten. • Es ist ein fehlerhafter Handshake aufgetreten.
`ERROR 9501 (HY000): Timed-out waiting to acquire database connection`	Auf dem Proxy ist beim Herstellen einer Datenbankverbindung eine Zeitüberschreitung aufgetreten. Einige mögliche Gründe sind: • Der Proxy kann keine Datenbankverbindung herstellen, da die maximale Anzahl an Verbindungen erreicht wurde. • Der Proxy kann keine Datenbankverbindung herstellen, da die Datenbank nicht verfügbar ist.

Behebung von RDS-Proxyproblemen mit RDS for PostgreSQL

Beim Herstellen einer Verbindung mit einem PostgreSQL-Proxy können die folgenden Probleme auftreten.

Fehler	Ursache	Lösung
`ERROR 28000: IAM authentication is allowed only with SSL connections.`	Der Benutzer hat versucht, mithilfe der IAM-Authentifizier ung mit der Einstellung `sslmode=disable` im	Der Benutzer muss eine Verbindung mit der Datenbank mit der Mindesteinstellung `sslmode=require` im

Fehler	Ursache	Lösung
	PostgreSQL-Client eine Verbindung zur Datenbank herzustellen.	PostgreSQL-Client herstellen. Weitere Informationen finden Sie in der PostgreSQL SSL Support-Dokumentation.
`ERROR 28000: This RDS proxy has no credentials for the role` *`role_name`* `. Check the credentials for this role and try again.`	Es gibt kein Secrets Manager-Secret für diese Rolle.	Fügen Sie ein Secrets Manager-Secret diese Rolle hinzu. Weitere Informationen finden Sie unter Konfiguration der IAM-Authentifizierung für RDS Proxy.
`ERROR 28000: RDS supports only IAM, MD5, or SCRAM authentication.`	Der Datenbank-Client, der für die Verbindung mit dem Proxy verwendet wird, nutzt einen Authentifizierungsmechanismus, der derzeit vom Proxy nicht unterstützt wird.	Wenn Sie keine IAM-Authentifizierung verwenden, verwenden Sie die MD5 oder SCRAM-Passwortauthentifizierung.
`ERROR 28000: A user name is missing from the connection startup packet. Provide a user name for this connection.`	Der Datenbankclient, der zum Herstellen einer Verbindung mit dem Proxy verwendet wird, sendet beim Versuch, eine Verbindung herzustellen, keinen Benutzernamen.	Stellen Sie sicher, dass Sie beim Einrichten einer Verbindung zum Proxy mit dem PostgreSQL-Client Ihrer Wahl einen Benutzernamen definieren.
`ERROR 28000: IAM is allowed only with SSL connections.`	Ein Client hat versucht, eine Verbindung über die IAM-Authentifizierung herzustellen, aber SSL war nicht aktiviert.	Aktivieren Sie SSL im PostgreSQL-Client.

Fehler	Ursache	Lösung
`ERROR 28000: This RDS Proxy requires TLS connections.`	Der Benutzer hat die Option Transport Layer Security erfordern aktiviert, hat aber versucht, auf dem PostgreSQL-Client eine Verbindung mit `sslmode=disable` herzustellen.	Führen Sie einen der folgenden Schritte aus, um diesen Fehler zu beheben: • Deaktivieren Sie die Option Transport Layer Security erfordern des Proxys. • Herstellen einer Verbindung mit der Datenbank mit der Mindesteinstellung `sslmode=allow` im PostgreSQL-Client.
`ERROR 28P01: IAM authentication failed for user` *`user_name`* `. Check the IAM token for this user and try again.`	Dies kann folgende Ursachen haben: • Der Client hat den falschen IAM-Benutzernamen angegeben. • Der Client hat ein falsches IAM-Autorisierungstoken für den Benutzer angegeben. • Der Client verwendet eine IAM-Richtlinie, die nicht über die erforderlichen Berechtigungen verfügt. • Der Client hat ein abgelaufenes IAM-Autorisierungstoken für den Benutzer bereitgestellt.	Gehen Sie folgendermaßen vor, um diesen Fehler zu beheben: 1. Bestätigen Sie, dass der bereitgestellte IAM-Benutzer vorhanden ist. 2. Vergewissern Sie sich, dass das IAM-Autorisierungstoken dem bereitgestellten IAM-Benutzer gehört. 3. Bestätigen Sie, dass die IAM-Richtlinie über ausreichende Berechtigungen für RDS verfügt. 4. Überprüfen Sie die Gültigkeit des verwendeten IAM-Autorisierungstokens.

Fehler	Ursache	Lösung
`ERROR 28P01: The password that was provided for the role` *`role_name`* `is wrong.`	Das Passwort für diese Rolle stimmt nicht mit dem Secrets Manager-Secret überein.	Überprüfen Sie den Schlüssel für diese Rolle in Secrets Manager, um festzuste llen, ob das Passwort mit dem in Ihrem PostgreSQL-Client verwendeten Passwort übereinstimmt.
`ERROR 28P01: The IAM authentication failed for the role` *`role_name`* `. Check the IAM token for this role and try again.`	Es gibt ein Problem mit dem IAM-Token, das für die IAM-Authentifizierung verwendet wird.	Generieren Sie ein neues Authentifizierungstoken und verwenden Sie es in einer neuen Verbindung.
`ERROR 0A000: Feature not supported: RDS Proxy supports only version 3.0 of the PostgreSQL messaging protocol.`	Der PostgreSQL-Client, mit dem eine Verbindung zum Proxy hergestellt wird, verwendet ein Protokoll, das älter als 3.0 ist.	Verwenden Sie einen neueren PostgreSQL-Client, der das 3.0-Messaging-Prot okoll unterstützt. Wenn Sie die `psql` PostgreSQL-CLI verwenden, verwenden Sie Version 7.4 oder höher.
`ERROR 0A000: Feature not supported: RDS Proxy currently doesn't support streaming replication mode.`	Der PostgreSQL-Client, der für die Verbindung mit dem Proxy verwendet wird, versucht, den Streaming-Replikationsmodus zu verwenden, der derzeit vom RDS-Proxy nicht unterstützt wird.	Deaktivieren Sie den Streaming-Replikationsmodus im PostgreSQL-Client, der für die Verbindung verwendet wird.

Fehler	Ursache	Lösung
`ERROR 0A000: Feature not supported: RDS Proxy currently doesn't support the option` *`option_name`* `.`	Über die Startmeldung fordert der PostgreSQL-Client, der für die Verbindung mit dem Proxy verwendet wird, eine Option an, die derzeit vom RDS-Proxy nicht unterstützt wird.	Deaktivieren Sie die Option, die in der obigen Nachricht als nicht unterstützt angezeigt wird, in dem PostgreSQL-Client, der für die Verbindung verwendet wird.
`ERROR 53300: The IAM authentication failed because of too many competing requests.`	Die Anzahl der gleichzeitigen Anforderungen mit IAM-Authentifizierung vom Client an den Proxy hat den Grenzwert überschritten.	Reduzieren Sie die Rate, mit der Verbindungen mit IAM-Authentifizierung von einem PostgreSQL-Client hergestellt werden.
`ERROR 53300: The maximum number of client connections to the proxy exceeded` *`number_value`* `.`	Die Anzahl der gleichzeitigen Verbindungsanforderungen vom Client zum Proxy hat den Grenzwert überschritten.	Reduzieren Sie die Anzahl der aktiven Verbindungen von PostgreSQL-Clients zu diesem RDS-Proxy.
`ERROR 53300: Rate of connection to proxy exceeded` *`number_value`* `.`	Die Rate der Verbindungsanforderungen vom Client zum Proxy hat den Grenzwert überschritten.	Reduzieren Sie die Rate, mit der Verbindungen von einem PostgreSQL-Client hergestellt werden.
`ERROR XX000: Unknown error.`	Es ist ein unbekannter Fehler aufgetreten.	Wenden Sie sich an den AWS Support, um das Problem zu untersuchen.

Fehler	Ursache	Lösung
`ERROR 08000: Timed-out waiting to acquire database connection.`	Der Proxy hat beim Warten auf den Aufbau einer Datenbankverbindung innerhalb der in der Einstellung angegebenen Dauer eine Zeitüberschreitung festgestellt. `ConnectionBorrowTimeout` Einige mögliche Gründe sind: • Der Proxy kann keine Datenbankverbindung herstellen, da die maximale Anzahl an Verbindungen erreicht wurde. • Der Proxy kann keine Datenbankverbindung herstellen, weil die Datenbank nicht verfügbar ist oder wenn der Verbindungsaufbau zur Datenbank länger dauert als konfiguriert. `ConnectionBorrowTimeout`	Folgende Lösungen sind möglich: • Vermeiden Sie das Anheften von Proxyverbindungen. Siehe Vermeiden Sie das Anheften eines RDS-Proxys. • Überprüfung `ConnectionBorrowTimeout` und `MaxConnectionsPercent` Einstellungen. Siehe Überlegungen zur RDS-Proxyverbindung. • Überprüfen Sie die Verfügbarkeit des Ziels. Weitere Informationen finden Sie unter `AvailabilityPercentage` in Überwachung von RDS-Proxy-Metriken mit Amazon CloudWatch.

Fehler	Ursache	Lösung
`ERROR XX000: Request returned an error: database_error .`	Die vom Proxy hergestellte Datenbankverbindung hat einen Fehler zurückgegeben.	Die Lösung hängt vom spezifischen Datenbank fehler ab. Ein Beispiel is: `Request returned an error: database "your-database-name" does not exist`. Das bedeutet, dass der angegebene Datenbankname nicht auf dem Datenbankserver existiert. Oder es bedeutet, dass der als Datenbankname verwendet e Benutzername (wenn kein Datenbankname angegeben ist) auf dem Server nicht vorhanden ist.

Problembehandlung bei gelöschter **`postgres`** Datenbank

Wenn Sie die `postgres` Datenbank versehentlich aus Ihrer Instance löschen, müssen Sie die Datenbank wiederherstellen, um die Konnektivität zu Ihrer Instance wiederherzustellen. Führen Sie die folgenden Befehle in Ihrer DB-Instance aus:

```
CREATE DATABASE postgres;
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

Verwenden von RDS Proxy mit AWS CloudFormation

Sie können RDS Proxy mit AWS CloudFormation nutzen. Auf diese Weise können Sie Gruppen verwandter Ressourcen erstellen. Eine solche Gruppe kann einen Proxy enthalten, der eine Verbindung mit einer neu erstellten DB-Instance von Amazon RDS herstellen kann. Die RDS-Proxy-Unterstützung in AWS CloudFormation umfasst zwei neue Registrierungstypen: `DBProxy` und `DBProxyTargetGroup`.

Die folgende Auflistung zeigt eine AWS CloudFormation-Beispielvorlage für RDS Proxy.

```
Resources:
 DBProxy:
   Type: AWS::RDS::DBProxy
   Properties:
     DBProxyName: CanaryProxy
     EngineFamily: MYSQL
     RoleArn:
      Fn::ImportValue: SecretReaderRoleArn
     Auth:
       - {AuthScheme: SECRETS, SecretArn: !ImportValue ProxySecret, IAMAuth: DISABLED}
     VpcSubnetIds:
       Fn::Split: [",", "Fn::ImportValue": SubnetIds]

 ProxyTargetGroup:
   Type: AWS::RDS::DBProxyTargetGroup
   Properties:
     DBProxyName: CanaryProxy
     TargetGroupName: default
     DBInstanceIdentifiers:
       - Fn::ImportValue: DBInstanceName
   DependsOn: DBProxy
```

Weitere Informationen zu den Ressourcen in diesem Beispiel finden Sie unter DBProxy und DBProxyTargetGroup.

Weitere Informationen zu den Ressourcen, die Sie mit AWS CloudFormation erstellen können, finden Sie in der Referenz zu RDS-Ressourcentypen.

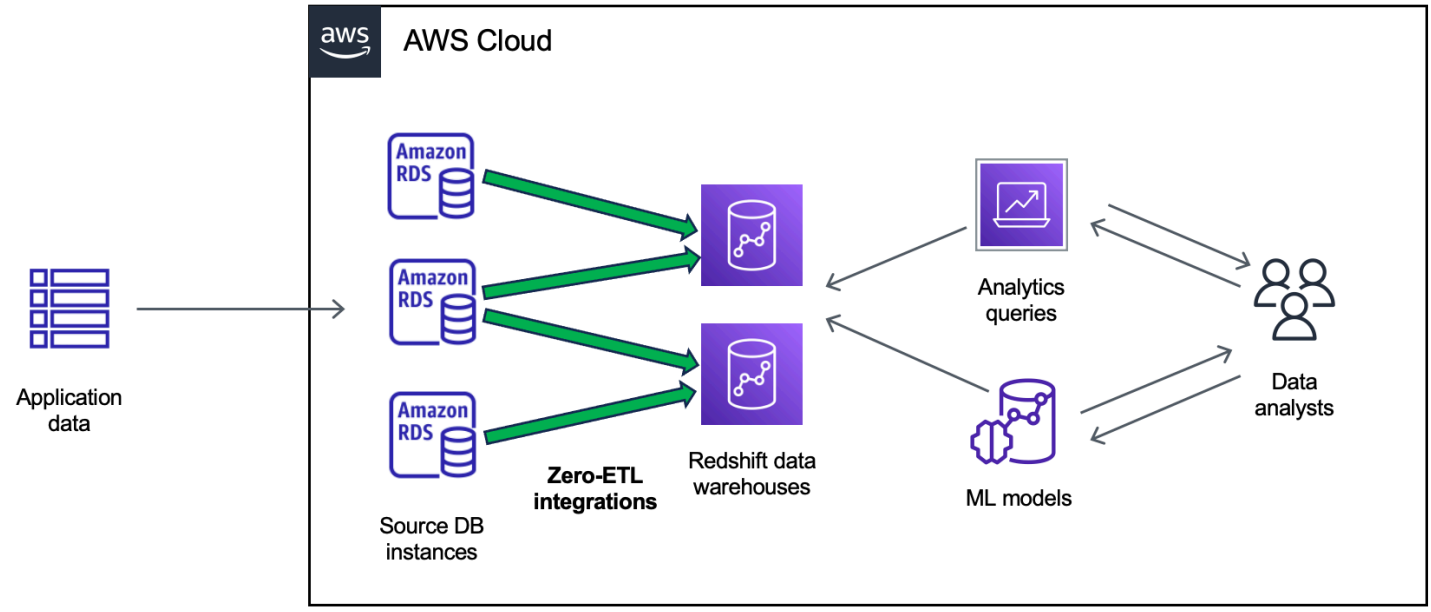
Amazon RDS Zero-ETL-Integrationen

Es handelt sich um eine vollständig verwaltete Lösung, mit der Sie Transaktionsdaten in Ihrem Analyseziel verfügbar machen können, nachdem sie in einen der RDS-Datenbank geschrieben wurden. Beim Extrahieren, Transformieren und Laden (ETL) werden Daten aus mehreren Quellen in einem großen, zentralen Data Warehouse kombiniert.

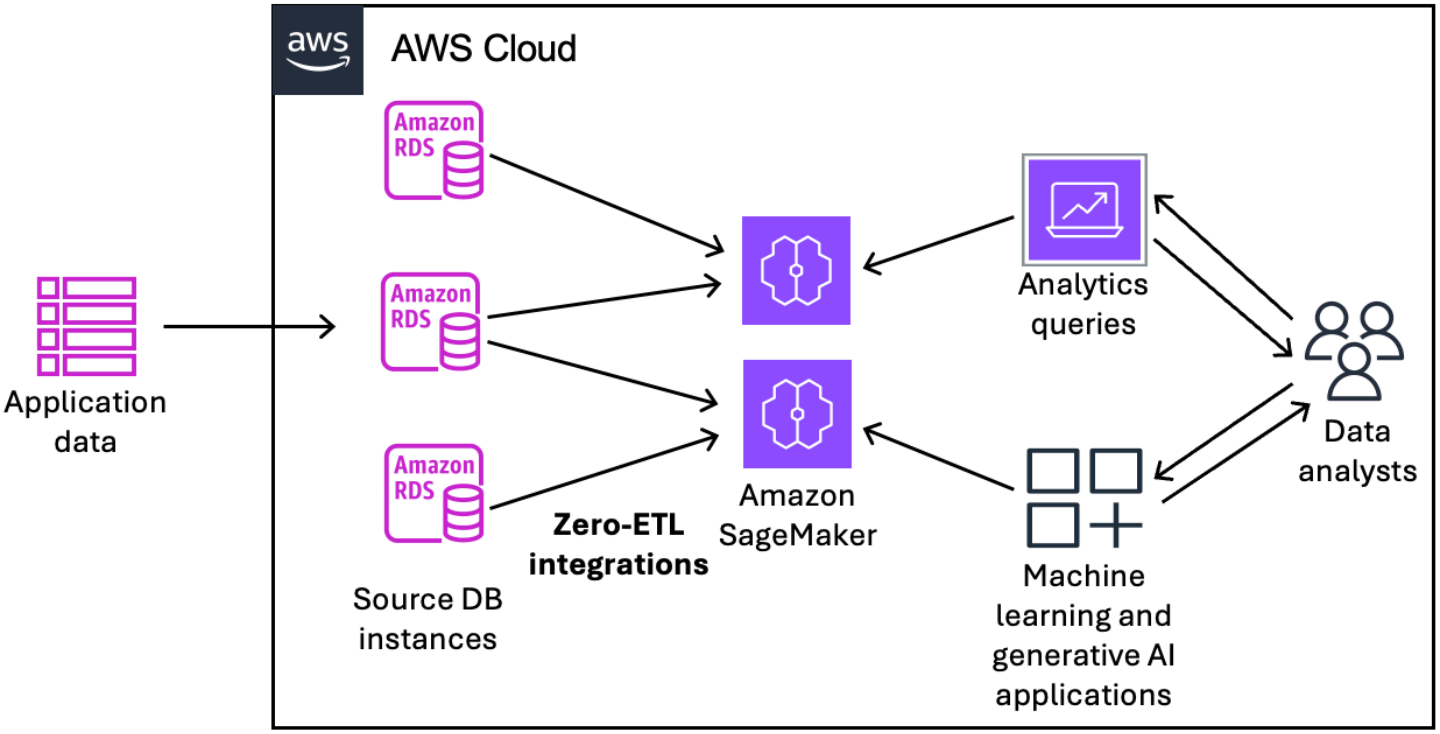
Eine Zero-ETL-Integration macht die Daten in Ihrem in Ihrer RDS-Datenbank in Amazon Redshift oder einem Amazon SageMaker AI Lakehouse nahezu in Echtzeit verfügbar. Sobald sich diese Daten im Ziel-Data Warehouse oder Data Lake befinden, können Sie Ihre Analyse-, ML- und KI-Workloads mithilfe der integrierten Funktionen wie maschinelles Lernen, materialisierte Ansichten, gemeinsame Nutzung von Daten, Verbundzugriff auf mehrere Datenspeicher und Data Lakes sowie Integrationen mit Amazon SageMaker AI und anderen unterstützen. QuickSight AWS-Services

Um eine Zero-ETL-Integration zu erstellen, geben Sie einen für die RDS-Datenbank als Quelle und ein unterstütztes Data Warehouse oder Lakehouse als Ziel an. Die Integration repliziert Daten aus der Quelldatenbank in das Ziel-Data Warehouse oder Lakehouse.

Das folgende Diagramm veranschaulicht diese Funktionalität für die Zero-ETL-Integration mit Amazon Redshift:



Das folgende Diagramm veranschaulicht diese Funktionalität für die Zero-ETL-Integration mit einem Lakehouse: Amazon SageMaker AI



Die Integration überwacht den Zustand der Datenpipeline und behebt nach Möglichkeit Probleme. Sie können Integrationen aus mehreren von mehreren RDS-Datenbanken in ein einziges Ziel-Data Warehouse oder Lakehouse erstellen, sodass Sie Erkenntnisse über mehrere Anwendungen hinweg gewinnen können.

Themen

- Vorteile
- Die wichtigsten Konzepte
- Einschränkungen
- Kontingente
- Unterstützte Regionen
- Erste Schritte mit Amazon RDS Zero-ETL-Integrationen
- Erstellen von Null-ETL-Integrationen von Amazon RDS mit Amazon Redshift
- Erstellen von Amazon RDS Zero-ETL-Integrationen mit einem Lakehouse Amazon SageMaker
- Datenfilterung für Amazon RDS Zero-ETL-Integrationen
- Hinzufügen von Daten zu einem einer RDS-Quelldatenbank und deren Abfrage
- Amazon RDS Zero-ETL-Integrationen anzeigen und überwachen
- Änderung der Amazon RDS Zero-ETL-Integrationen

- Löschen von Amazon RDS Zero-ETL-Integrationen
- Fehlerbehebung bei Amazon RDS Zero-ETL-Integrationen

Vorteile

RDS Zero-ETL-Integrationen bieten die folgenden Vorteile:

- Sie helfen Ihnen dabei, ganzheitliche Erkenntnisse aus mehreren Datenquellen zu gewinnen.
- Eliminieren Sie die Erfordernis zur Erstellung und Verwaltung komplexer Daten-Pipelines, die Extract, Transform, Load (ETL)-Operationen ausführen. Null-ETL-Integrationen beseitigen die Herausforderungen, die mit dem Aufbau und der Verwaltung von Pipelines einhergehen, indem sie diese für Sie bereitstellen und verwalten.
- Sie reduzieren den Betriebsaufwand und die Kosten, sodass Sie sich ganz auf die Verbesserung Ihrer Anwendungen konzentrieren können.
- Sie können die Analyse- und ML-Funktionen des Zielorts nutzen, um Erkenntnisse aus Transaktions- und anderen Daten zu gewinnen und effektiv auf kritische, zeitkritische Ereignisse zu reagieren.

Die wichtigsten Konzepte

Wenn Sie mit Null-ETL-Integrationen beginnen, sollten Sie die folgenden Konzepte berücksichtigen:

Integration

Eine vollständig verwaltete Datenpipeline, die automatisch Transaktionsdaten und Schemas aus einem einer RDS-Datenbank in ein Data Warehouse oder einen Katalog repliziert.

Der der RDS-Datenbank, aus dem Daten repliziert werden. Sie können eine Single-AZ- oder Multi-AZ-DB-Instance oder einen Multi-AZ-DB-Cluster angeben.

Ziel

Das Data Warehouse oder Lakehouse, in das die Daten repliziert werden. Es gibt zwei Arten von Data Warehouse: ein bereitgestelltes Cluster-Data-Warehouse und ein Serverless-Data-Warehouse. Ein bereitgestelltes Cluster-Data-Warehouse ist eine Sammlung von Datenverarbeitungsressourcen, den sogenannten Knoten, die zu einer Gruppe, einem sogenannten Cluster, zusammengefasst werden. Ein Serverless-Data-Warehouse besteht aus

einer Arbeitsgruppe, die Datenverarbeitungsressourcen speichert, und einem Namespace, in dem die Datenbankobjekte und Benutzer gespeichert sind. Beide Data Warehouses verwenden eine Analyse-Engine und enthalten eine oder mehrere Datenbanken.

Ein Ziel-Lakehouse besteht aus Katalogen, Datenbanken, Tabellen und Ansichten. Weitere Informationen zur Lakehouse-Architektur finden Sie SageMaker Lakehouse componentsim Amazon SageMaker AI Unified Studio Benutzerhandbuch.

können mehrere Quelldatenbanken auf dasselbe Ziel schreiben.

Weitere Informationen finden Sie unter Architektur des Data-Warehouse-Systems im Entwicklerhandbuch zu Amazon Redshift.

Einschränkungen

Die folgenden Einschränkungen gelten für RDS Zero-ETL-Integrationen.

Themen

- Allgemeine Einschränkungen
- Einschränkungen von RDS für MySQL
- Einschränkungen von RDS für PostgreSQL
- Beschränkungen von RDS for Oracle
- Einschränkungen für Amazon Redshift
- Amazon SageMaker AI Einschränkungen bei Lakehouse

Allgemeine Einschränkungen

- Der muss sich in derselben Region wie das Ziel befinden.
- Sie können einen Datenbank-DB-Cluster nicht umbenennen, wenn er über bestehende Integrationen verfügt.
- Sie können nicht mehrere Integrationen zwischen derselben Quell- und Zieldatenbank erstellen.
- Sie können keinen löschen, der über bestehende Integrationen verfügt. Sie müssen zuerst alle zugehörigen Integrationen löschen.
-

- Sie können eine Integration nicht löschen, wenn die Quelldatenbank gestoppt ist.
- Wenn Ihr die Quelle einer blauen/grünen Bereitstellung ist, können die blauen und grünen Umgebungen während des Switchovers keine vorhandenen Zero-ETL-Integrationen enthalten. Sie müssen zuerst die Integration löschen und umstellen. Anschließend erstellen Sie die Integration neu.
- Sie können keine Integration für eine Quelldatenbank erstellen, für die eine andere Integration aktiv erstellt wird.
- Wenn Sie zum ersten Mal eine Integration erstellen oder wenn eine Tabelle erneut synchronisiert wird, kann das Seeding von Daten von der Quelle zum Ziel je nach Größe der Quelldatenbank 20 bis 25 Minuten oder länger dauern. Diese Verzögerung kann zu einer erhöhten Replikatverzögerung führen.
- Einige Datentypen werden nicht unterstützt. Weitere Informationen finden Sie unter the section called “Datentypunterschiede”.
- Systemtabellen, temporäre Tabellen und Ansichten werden nicht in Ziel-Warehouses repliziert.
- Das Ausführen von DDL-Befehlen (z. B.`ALTER TABLE`) für eine Quelltabelle kann eine Resynchronisierung der Tabelle auslösen, sodass die Tabelle während der Resynchronisierung nicht mehr abgefragt werden kann. Weitere Informationen finden Sie unter the section called “Eine oder mehrere meiner Amazon-Redshift-Tabellen erfordern eine erneute Synchronisation”.

Einschränkungen von RDS für MySQL

- In Ihrer Quelldatenbank muss eine unterstützte Version von RDS for MySQL ausgeführt werden. Eine Liste der unterstützten Versionen finden Sie unter the section called “Null-ETL-Integrationen ”.
- Zero-ETL-Integrationen werden nicht sowohl auf der primären Instance als auch auf einer Read Replica-Instance in derselben Region unterstützt. AWS
- Null-ETL-Integrationen benötigen die MySQL-Binärprotokollierung (Binlog), um laufende Datenänderungen zu erfassen. Verwenden Sie keine binlog-basierte Datenfilterung, da dies zu Dateninkonsistenzen zwischen der Quell- und der Zieldatenbank führen kann.
- Null-ETL-Integrationen werden nur für Datenbanken unterstützt, die für die Verwendung der InnoDB-Speicher-Engine konfiguriert sind.
- Fremdschlüsselverweise mit vordefinierten Tabellenaktualisierungen werden nicht unterstützt. Insbesondere werden `ON UPDATE` Regeln mit `CASCADESET NULL`, und `SET DEFAULT` -Aktionen nicht unterstützt. `ON DELETE` Der Versuch, eine Tabelle mit solchen Verweisen in einer anderen Tabelle zu erstellen oder zu aktualisieren, führt zu einem Fehlschlag der Tabelle.

Einschränkungen von RDS für PostgreSQL

- Bei der Quelldatenbank muss es sich um eine RDS für PostgreSQL-Instance handeln, auf der Version 15.7+, 16.3+ oder 17.1+ ausgeführt wird. Frühere Versionen werden nicht unterstützt.
- Sie können keine Zero-ETL-Integration von einer RDS for PostgreSQL-Read Replica-Instance aus erstellen.
- Nicht protokollierte PostgreSQL-Tabellen und materialisierte Ansichten werden nicht nach Amazon Redshift repliziert.
- Die Replikation bestimmter PostgreSQL-Datentypen, wie Geometriedatentypen und Daten mit mehr als 64 KB, wird aufgrund von Einschränkungen in Amazon Redshift nicht unterstützt. Weitere Informationen zu den Datentypunterschieden zwischen RDS for PostgreSQL und Amazon Redshift finden Sie the section called “” im Abschnitt Datentypunterschiede.
- Sie können kein Hauptversions-Upgrade für die Quell-Instance von RDS for PostgreSQL durchführen, solange sie über eine aktive Zero-ETL-Integration verfügt. Um die Quell-Instance zu aktualisieren, müssen Sie zunächst alle vorhandenen Zero-ETL-Integrationen löschen. Nachdem das Upgrade der Hauptversion abgeschlossen ist, können Sie die Zero-ETL-Integrationen neu erstellen.
- Wenn Sie deklarative Partitionierungstransaktionen auf der Quell-DB-Instance durchführen, gehen alle betroffenen Tabellen in den Status „Fehlgeschlagen“ über und sind nicht mehr zugänglich.

Beschränkungen von RDS for Oracle

- Bei der Quelldatenbank muss es sich um eine RDS for Oracle-Instance handeln, auf der Version 19c Enterprise Edition oder Standard Edition 2, Release-Update vom Juli 2019 oder höher ausgeführt wird. Frühere Versionen werden nicht unterstützt.
- Sie können keine Zero-ETL-Integration von einer Read Replica-Instance von RDS for Oracle aus erstellen.
- Sie können eine Tenant-Datenbank nicht umbenennen, wenn in dieser Tenant-Datenbank eine Zero-ETL-Integration vorhanden ist.
- Eine Tenant-Datenbank kann nur eine Zero-ETL-Integration haben.
- Wenn Sie eine Zero-ETL-Integration von RDS for Oracle erstellen, können Sie diese Integration nicht ändern. Um eine Integration zu ändern, müssen Sie die Integration löschen und dann eine neue erstellen.

- RDS for Oracle und Amazon Redshift weisen einige Datentypunterschiede auf. Weitere Informationen finden Sie the section called “RDS für Oracle” im Abschnitt Unterschiede zwischen den Datentypen.

Einschränkungen für Amazon Redshift

Eine Liste der Einschränkungen von Amazon Redshift im Zusammenhang mit Zero-ETL-Integrationen finden Sie unter Überlegungen zur Verwendung von Zero-ETL-Integrationen mit Amazon Redshift im Amazon Redshift Management Guide.

Amazon SageMaker AI Einschränkungen bei Lakehouse

Im Folgenden finden Sie eine Einschränkung für Amazon SageMaker AI Lakehouse Zero-ETL-Integrationen.

- Katalognamen sind auf eine Länge von 19 Zeichen begrenzt.

Kontingente

Ihr Konto hat die folgenden Kontingente für RDS Zero-ETL-Integrationen. Jedes Kontingent gilt pro Region, sofern nicht anders angegeben.

Name	Standard	Beschreibung
Integrationen	100	Die Gesamtzahl der Integrationen innerhalb eines AWS-Konto.
Integrationen pro Ziel	50	Die Anzahl der Integrationen, die Daten an ein einzelnes Ziel-Data Warehouse oder Lakehouse senden.
Integrationen pro Quell-Instance	5	

Darüber hinaus legt das Ziel-Warehouse bestimmte Beschränkungen für die Anzahl der Tabellen fest, die in jeder DB-Instance oder jedem Clusterknoten zulässig sind. Weitere Informationen zu Amazon Redshift-Kontingenten und -Limits finden Sie unter Kontingente und Limits in Amazon Redshift im Amazon Redshift Management Guide.

Unterstützte Regionen

RDS Zero-ETL-Integrationen sind in einer Teilmenge von verfügbar. AWS-Regionen Eine Liste der unterstützten -Regionen finden Sie unter the section called “Null-ETL-Integrationen ”.

Erste Schritte mit Amazon RDS Zero-ETL-Integrationen

Bevor Sie eine Zero-ETL-Integration erstellen, konfigurieren Sie Ihren für die RDS-Datenbank und Ihr Data Warehouse mit den erforderlichen Parametern und Berechtigungen. Während der Einrichtung führen Sie die folgenden Schritte aus:

1. Erstellen einer benutzerdefinierten DB--Parametergruppe.
2. Erstellen Sie einen .
3. Erstellen Sie ein Ziel-Data Warehouse für Amazon Redshift oder Erstellen Sie ein Amazon SageMaker AI-Ziel-Lakehouse.

Nachdem Sie diese Aufgaben abgeschlossen haben, fahren Sie mit oder fortthe section called “Erstellen von Zero-ETL-Integrationen mit Amazon Redshift”. the section called “Erstellung von Zero-ETL-Integrationen mit einem Lakehouse Amazon SageMaker”

> **Tip**
>
> Sie können RDS diese Einrichtungsschritte für Sie ausführen lassen, während Sie die Integration erstellen, anstatt sie manuell durchzuführen. Um sofort mit der Erstellung einer Integration zu beginnen, siehe the section called “Erstellen von Zero-ETL-Integrationen mit Amazon Redshift”.

Für Schritt 3 können Sie je nach Bedarf wählen, ob Sie entweder ein Ziel-Data Warehouse (Schritt 3a) oder ein Ziel-Lakehouse (Schritt 3b) erstellen möchten:

- Wählen Sie ein Data Warehouse, wenn Sie herkömmliche Data Warehousing-Funktionen mit SQL-basierten Analysen benötigen.
- Entscheiden Sie sich für ein Amazon SageMaker AI Lakehouse, wenn Sie Funktionen für maschinelles Lernen benötigen und Lakehouse-Funktionen für Data-Science- und ML-Workflows verwenden möchten.

Schritt 1: Erstellen einer benutzerdefinierten DB--Parametergruppe

Amazon RDS Zero-ETL-Integrationen erfordern spezifische Werte für die DB-Parameter, die die Datenreplikation steuern. Die spezifischen Parameter hängen von Ihrer Quell-DB-Engine ab. Um diese Parameter zu konfigurieren, müssen Sie zuerst eine benutzerdefinierte DB-Parametergruppe erstellen und diese dann der Quelldatenbank zuordnen. Konfigurieren Sie die folgenden Parameterwerte je nach Ihrer Quell-DB-Engine. Anweisungen zum Erstellen einer Parametergruppe finden Sie unter the section called “DB-Parametergruppen”. Wir empfehlen, dass Sie alle Parameterwerte innerhalb derselben Anfrage konfigurieren, um Abhängigkeitsprobleme zu vermeiden.

RDS für MySQL:

- `binlog_format=ROW`
- `binlog_row_image=full`

Stellen Sie außerdem sicher, dass der `binlog_row_value_options`-Parameter nicht auf `PARTIAL_JSON` gesetzt ist. Wenn es sich bei der Quelldatenbank um einen Multi-AZ-DB-Cluster handelt, stellen Sie sicher, dass der `binlog_transaction_compression` Parameter nicht auf gesetzt ist. `ON`

Einige dieser Parameter (z. B.`binlog_format`) sind dynamisch, was bedeutet, dass Sie Änderungen an den Parametern vornehmen können, ohne einen Neustart auszulösen. Dies bedeutet, dass einige bestehende Sitzungen möglicherweise weiterhin den alten Wert des Parameters verwenden. Um zu verhindern, dass dies zu Problemen bei der Erstellung einer Zero-ETL-Integration führt, aktivieren Sie Performance Schema. Performance Schema stellt sicher, dass Zero-ETL-Vorabprüfungen ausgeführt werden, mit deren Hilfe fehlende Parameter frühzeitig im Prozess erkannt werden können.

RDS für PostgreSQL:

- `rds.logical_replication = 1`
- `rds.replica_identity_full = 1`
- `session_replication_role = origin`
- `wal_sender_timeout ≥ 20000 or = 0`
- `max_wal_senders ≥ 20`
- `max_replication_slots ≥ 20`

Für mehrere PostgreSQL-Integrationen wird pro Integration ein logischer Replikationsslot verwendet. Überprüfen Sie die `max_wal_senders` Parameter `max_replication_slots` und je nach Nutzung.

Für eine effiziente Datensynchronisierung bei Zero-ETL-Integrationen legen Sie das `rds.replica_identity_full` in Ihrer Quell-DB-Instance fest. Dadurch wird die Datenbank angewiesen, während der `DELETE` Operationen vollständige Zeilendaten `UPDATE` und nicht nur Primärschlüsselinformationen im Write-Ahead-Log (WAL) zu protokollieren. Zero-ETL benötigt vollständige Zeilendaten, auch wenn für alle replizierten Tabellen Primärschlüssel erforderlich sind. Um festzustellen, welche Daten bei Abfragen sichtbar sind, Amazon Redshift verwendet eine spezielle Anti-Join-Strategie, um Ihre Daten mit einer internen Tabelle zur Nachverfolgung von Löschvorgängen zu vergleichen. Durch das Protokollieren ganzer Zeilenbilder kann Amazon Redshift diese Anti-Joins effizient durchführen. Ohne vollständige Zeilendaten müsste Amazon Redshift zusätzliche Suchvorgänge durchführen, was die Leistung bei Vorgängen mit hohem Durchsatz in der von Amazon Redshift verwendeten Spalten-Engine beeinträchtigen könnte.

> **Important**
>
> Wenn Sie die Replikatidentität so einrichten, dass ganze Zeilen protokolliert werden, erhöht sich Ihr WAL-Volumen, was zu einer höheren Schreibverstärkung und I/O -nutzung führen kann, insbesondere bei umfangreichen Tabellen oder häufigen Aktualisierungen. Um auf diese Auswirkungen vorbereitet zu sein, sollten Sie Ihre Speicherkapazität und I/O Anforderungen planen, Ihr WAL-Wachstum überwachen und die Replikationsverzögerung bei schreibintensiven Workloads verfolgen.

RDS für Oracle:

Für RDS for Oracle sind keine Parameteränderungen erforderlich.

Schritt 2: Wählen oder erstellen Sie einen

Nachdem Sie eine benutzerdefinierte erstellt haben, wählen oder erstellen Sie einen . Dieser wird die Quelle der Datenreplikation zum Ziel-Data Warehouse sein. Anweisungen zum Erstellen eines Multi-AZ-DB-Clusters finden Sie unter. the section called “Erstellen eines Multi-AZ-DB-Clusters”

Auf der Datenbank muss eine unterstützte DB-Engine-Version ausgeführt werden. Eine Liste der unterstützten Versionen finden Sie unter the section called “Null-ETL-Integrationen ”.

Wenn Sie die Datenbank erstellen, ändern Sie unter Zusätzliche Konfiguration die in die benutzerdefinierte Parametergruppe, die Sie im vorherigen Schritt erstellt haben.

> **Note**
>
> Wenn Sie die Parametergruppe dem zuordnen, nachdem der bereits erstellt wurde, müssen Sie die Änderungen zu übernehmen, bevor Sie eine Zero-ETL-Integration erstellen können. Anleitungen Anweisungen finden Sie unter the section called “Neustarten einer DB-Instance” oder the section called “Neustarten von Multi-AZ-DB-Clustern”.

Stellen Sie außerdem sicher, dass automatische Backups in der Datenbank aktiviert sind. Weitere Informationen finden Sie unter the section called “Aktivieren von automatisierten Backups”.

Schritt 3a: Erstellen Sie ein Ziel-Data Warehouse

Nachdem Sie Ihren erstellt haben, müssen Sie ein Ziel-Data Warehouse erstellen und konfigurieren. Das Data Warehouse muss die folgenden Anforderungen erfüllen:

- Verwendung eines RA3 Knotentyps mit mindestens zwei Knoten oder Redshift Serverless.
- Es muss verschlüsselt sein (bei Verwendung eines bereitgestellten Clusters). Weitere Informationen finden Sie unter Datenbankverschlüsselung in Amazon Redshift.

Anweisungen zum Erstellen eines Data Warehouse finden Sie unter Erstellen eines Clusters für bereitgestellte Cluster oder Erstellen einer Arbeitsgruppe mit einem Namespace für Redshift Serverless.

Aktivieren Sie die Berücksichtigung von Groß- und Kleinschreibung im Data Warehouse

Damit die Integration erfolgreich ist, muss der Parameter für die Berücksichtigung von Groß- und Kleinschreibung (`enable_case_sensitive_identifier`) für das Data Warehouse aktiviert sein. Standardmäßig ist die Berücksichtigung von Groß- und Kleinschreibung auf allen bereitgestellten Clustern und Redshift-Serverless-Arbeitsgruppen deaktiviert.

Um die Berücksichtigung von Groß- und Kleinschreibung zu aktivieren, führen Sie je nach Data-Warehouse-Typ die folgenden Schritte aus:

- Bereitgestellter Cluster – Um die Berücksichtigung von Groß- und Kleinschreibung in einem bereitgestellten Cluster zu aktivieren, erstellen Sie eine benutzerdefinierte Parametergruppe mit aktiviertem `enable_case_sensitive_identifier`-Parameter. Ordnen Sie diese Parametergruppe dann dem Cluster zu. Anweisungen finden Sie unter Verwalten von Parametergruppen mit der Konsole oder Konfigurieren von Parameterwerten mit der AWS CLI.

  > **Note**
  >
  > Denken Sie daran, den Cluster neu zu starten, nachdem Sie ihm die benutzerdefinierte Parametergruppe zugeordnet haben.

- Serverless-Arbeitsgruppe – Um die Berücksichtigung von Groß- und Kleinschreibung in einer Redshift-Serverless-Arbeitsgruppe zu aktivieren, müssen Sie die AWS CLI verwenden. Die Amazon-Redshift-Konsole unterstützt derzeit nicht das Ändern von Redshift-Serverless-Parameterwerten. Senden Sie die folgende Anfrage zur Aktualisierung der Arbeitsgruppe:

  ```
  aws redshift-serverless update-workgroup \
    --workgroup-name target-workgroup \
    --config-parameters
   parameterKey=enable_case_sensitive_identifier,parameterValue=true
  ```

  Sie müssen eine Arbeitsgruppe nicht neu starten, nachdem Sie ihre Parameterwerte geändert haben.

Konfigurieren der Autorisierung für das Data Warehouse

Nachdem Sie ein Data Warehouse erstellt haben, müssen Sie den der Quell-RDS-Datenbank als autorisierte Integrationsquelle konfigurieren. Anweisungen finden Sie unter Konfigurieren der Autorisierung für Ihr Amazon-Redshift-Data-Warehouse.

Richten Sie eine Integration mit dem ein AWS SDKs

Anstatt jede Ressource manuell einzurichten, können Sie das folgende Python-Skript ausführen, um die erforderlichen Ressourcen automatisch für Sie einzurichten. Das Codebeispiel verwendet den AWS SDK für Python (Boto3), um einen als Quelle, eine RDS für MySQL-DB-Instance und ein Ziel-Data Warehouse zu erstellen, jeweils mit den erforderlichen Parameterwerten. Es wartet dann, bis die Datenbanken verfügbar sind, bevor eine Zero-ETL-Integration zwischen ihnen erstellt wird. Sie

können verschiedene Funktionen auskommentieren, je nachdem, welche Ressourcen Sie einrichten müssen.

Führen Sie die folgenden Befehle aus, um die erforderlichen Abhängigkeiten zu installieren:

```
pip install boto3
pip install time
```

Innerhalb des Skripts können Sie optional die Namen der Quell-, Ziel- und Parametergruppen ändern. Die letzte Funktion erstellt eine Integration, die `my-integration` nach der Einrichtung der Ressourcen benannt ist.

Python-Codebeispiel

```
import boto3
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default Region.

rds = boto3.client('rds')
redshift = boto3.client('redshift')
sts = boto3.client('sts')

source_db_name = 'my-source-db' # A name for the source database
source_param_group_name = 'my-source-param-group' # A name for the source parameter
 group
target_cluster_name = 'my-target-cluster' # A name for the target cluster
target_param_group_name = 'my-target-param-group' # A name for the target parameter
 group

def create_source_db(*args):
    """Creates a source RDS for MySQL DB instance"""

    response = rds.create_db_parameter_group(
        DBParameterGroupName=source_param_group_name,
        DBParameterGroupFamily='mysql8.0',
        Description='RDS for MySQL zero-ETL integrations'
    )
    print('Created source parameter group: ' + response['DBParameterGroup']
['DBParameterGroupName'])
```

```
    response = rds.modify_db_parameter_group(
        DBParameterGroupName=source_param_group_name,
        Parameters=[
            {
                'ParameterName': 'binlog_format',
                'ParameterValue': 'ROW',
                'ApplyMethod': 'pending-reboot'
            },
            {
                'ParameterName': 'binlog_row_image',
                'ParameterValue': 'full',
                'ApplyMethod': 'pending-reboot'
            }
        ]
    )
    print('Modified source parameter group: ' + response['DBParameterGroupName'])

    response = rds.create_db_instance(
        DBInstanceIdentifier=source_db_name,
        DBParameterGroupName=source_param_group_name,
        Engine='mysql',
        EngineVersion='8.0.32',
        DBName='mydb',
        DBInstanceClass='db.m5.large',
        AllocatedStorage=15,
        MasterUsername='username',
        MasterUserPassword='Password01**'
    )
    print('Creating source database: ' + response['DBInstance']
['DBInstanceIdentifier'])
    source_arn = (response['DBInstance']['DBInstanceArn'])
    create_target_cluster(target_cluster_name, source_arn, target_param_group_name)
    return(response)

def create_target_cluster(target_cluster_name, source_arn, target_param_group_name):
    """Creates a target Redshift cluster"""

    response = redshift.create_cluster_parameter_group(
        ParameterGroupName=target_param_group_name,
        ParameterGroupFamily='redshift-1.0',
        Description='RDS for MySQL zero-ETL integrations'
    )
    print('Created target parameter group: ' + response['ClusterParameterGroup']
['ParameterGroupName'])
```

```
    response = redshift.modify_cluster_parameter_group(
        ParameterGroupName=target_param_group_name,
        Parameters=[
            {
                'ParameterName': 'enable_case_sensitive_identifier',
                'ParameterValue': 'true'
            }
        ]
    )
    print('Modified target parameter group: ' + response['ParameterGroupName'])

    response = redshift.create_cluster(
        ClusterIdentifier=target_cluster_name,
        NodeType='ra3.4xlarge',
        NumberOfNodes=2,
        Encrypted=True,
        MasterUsername='username',
        MasterUserPassword='Password01**',
        ClusterParameterGroupName=target_param_group_name
    )
    print('Creating target cluster: ' + response['Cluster']['ClusterIdentifier'])

    # Retrieve the target cluster ARN
    response = redshift.describe_clusters(
        ClusterIdentifier=target_cluster_name
    )
    target_arn = response['Clusters'][0]['ClusterNamespaceArn']

    # Retrieve the current user's account ID
    response = sts.get_caller_identity()
    account_id = response['Account']

    # Create a resource policy granting access to source database and account ID
    response = redshift.put_resource_policy(
        ResourceArn=target_arn,
        Policy='''
        {
            \"Version\":\"2012-10-17\",
            \"Statement\":[
                {\"Effect\":\"Allow\",
                \"Principal\":{
                    \"Service\":\"redshift.amazonaws.com\"
                },
```

```
                \"Action\":[\"redshift:AuthorizeInboundIntegration\"],
                \"Condition\":{
                    \"StringEquals\":{
                        \"aws:SourceArn\":\"%s\"}
                    }
                },
                {\"Effect\":\"Allow\",
                \"Principal\":{
                    \"AWS\":\"arn:aws:iam::%s:root\"},
                \"Action\":\"redshift:CreateInboundIntegration\"}
            ]
        }
        ''' % (source_arn, account_id)
    )
    return(response)

def wait_for_db_availability(*args):
    """Waits for both databases to be available"""

    print('Waiting for source and target to be available...')

    response = rds.describe_db_instances(
        DBInstanceIdentifier=source_db_name
    )
    source_status = response['DBInstances'][0]['DBInstanceStatus']
    source_arn = response['DBInstances'][0]['DBInstanceArn']

    response = redshift.describe_clusters(
        ClusterIdentifier=target_cluster_name
    )
    target_status = response['Clusters'][0]['ClusterStatus']
    target_arn = response['Clusters'][0]['ClusterNamespaceArn']

    # Every 60 seconds, check whether the databases are available
    if source_status != 'available' or target_status != 'available':
        time.sleep(60)
        response = wait_for_db_availability(
            source_db_name, target_cluster_name)
    else:
        print('Databases available. Ready to create zero-ETL integration.')
        create_integration(source_arn, target_arn)
        return

def create_integration(source_arn, target_arn):
```

```
    """Creates a zero-ETL integration using the source and target databases"""

    response = rds.create_integration(
        SourceArn=source_arn,
        TargetArn=target_arn,
        IntegrationName='my-integration'
    )
    print('Creating integration: ' + response['IntegrationName'])

def main():
    """main function"""
    create_source_db(source_db_name, source_param_group_name)
    wait_for_db_availability(source_db_name, target_cluster_name)

if __name__ == "__main__":
    main()
```

Schritt 3b: Erstellen Sie einen AWS Glue Katalog für die Amazon SageMaker AI Zero-ETL-Integration

Wenn Sie eine Zero-ETL-Integration mit einem Amazon SageMaker AI Lakehouse erstellen, müssen Sie einen AWS Glue verwalteten Katalog in erstellen. AWS Lake Formation Der Zielkatalog muss ein von Amazon Redshift verwalteter Katalog sein. Um einen von Amazon Redshift verwalteten Katalog zu erstellen, erstellen Sie zunächst die `AWSServiceRoleForRedshift` serviceverknüpfte Rolle. Fügen Sie in der Lake Formation Formation-Konsole den `AWSServiceRoleForRedshift` als Administrator mit Schreibschutz hinzu.

Weitere Informationen zu den vorherigen Aufgaben finden Sie in den folgenden Themen.

- Informationen zum Erstellen eines von Amazon Redshift verwalteten Katalogs finden Sie unter Creating an Amazon Redshift managed catalog AWS Glue Data Catalog im AWS Lake Formation Developer Guide.
- Informationen zur serviceverknüpften Rolle für Amazon Redshift finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon Redshift im Amazon Redshift Management Guide.
- Informationen zu Administratorberechtigungen mit Lesezugriff für Lake Formation finden Sie unter Referenz zu Lake Formation Formation-Personas und IAM-Berechtigungen im Developer Guide.AWS Lake Formation

Konfigurieren Sie die Berechtigungen für den Zielkatalog AWS Glue

Bevor Sie einen Zielkatalog für die Zero-ETL-Integration erstellen, müssen Sie die Lake Formation Formation-Zielerstellungsrolle und die AWS Glue Datenübertragungsrolle erstellen. Verwenden Sie die Rolle Lake Formation Target Creation, um den Zielkatalog zu erstellen. Geben Sie bei der Erstellung des Zielkatalogs die Glue-Datenübertragungsrolle in das Feld IAM-Rolle im Abschnitt Zugriff von Engines ein.

Rolle bei der Schaffung von Zielen in der Lake Formation

Die Rolle zur Zielerstellung muss ein Lake Formation-Administrator sein und erfordert die folgenden Berechtigungen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "lakeformation:RegisterResource",
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "s3:PutEncryptionConfiguration",
                "iam:PassRole",
                "glue:CreateCatalog",
                "glue:GetCatalog",
                "s3:PutBucketTagging",
                "s3:PutLifecycleConfiguration",
                "s3:PutBucketPolicy",
                "s3:CreateBucket",
                "redshift-serverless:CreateNamespace",
                "s3:DeleteBucket",
                "s3:PutBucketVersioning",
                "redshift-serverless:CreateWorkgroup"
            ],
            "Resource": [
```

```
                "arn:aws:glue:*:111122223333:catalog",
                "arn:aws:glue:*:111122223333:catalog/*",
                "arn:aws:s3:::*",
                "arn:aws:redshift-serverless:*:111122223333:workgroup/*",
                "arn:aws:redshift-serverless:*:111122223333:namespace/*",
                "arn:aws:iam::111122223333:role/GlueDataCatalogDataTransferRole"
            ]
        }
    ]
}
```

Die Rolle zur Zielerstellung muss über die folgende Vertrauensstellung verfügen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "glue.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::111122223333:user/Username"
          },
          "Action": "sts:AssumeRole"
        }
    ]
}
```

Glue-Datenübertragungsrolle

Die Glue-Datenübertragungsrolle ist für MySQL-Katalogoperationen erforderlich und muss über die folgenden Berechtigungen verfügen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DataTransferRolePolicy",
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "glue:GetCatalog",
                "glue:GetDatabase"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Die Glue-Datenübertragungsrolle muss über die folgende Vertrauensstellung verfügen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "glue.amazonaws.com",
                    "redshift.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Nächste Schritte

Mit einem für die Quell-RDS-Datenbank und entweder einem Amazon Redshift Redshift-Ziel-Data Warehouse oder einem Amazon SageMaker AI Lakehouse können Sie eine Zero-ETL-Integration erstellen und Daten replizieren. Detaillierte Anweisungen finden Sie unter the section called “Erstellen von Zero-ETL-Integrationen mit Amazon Redshift”.

Erstellen von Null-ETL-Integrationen von Amazon RDS mit Amazon Redshift

Wenn Sie eine Amazon RDS Zero-ETL-Integration erstellen, geben Sie den der Quell-RDS-Datenbank und das Amazon Redshift Redshift-Ziel-Data Warehouse an. Sie können auch die Verschlüsselungseinstellungen anpassen und Tags hinzufügen. Amazon RDS erstellt eine Integration zwischen dem und seinem Ziel. Sobald die Integration aktiv ist, werden alle Daten, die Sie in den einfügen, in das konfigurierte Amazon Redshift Redshift-Ziel repliziert.

Voraussetzungen

Bevor Sie eine Zero-ETL-Integration erstellen, müssen Sie einen und ein Amazon Redshift Redshift-Ziel-Data Warehouse erstellen. Sie müssen auch die Replikation in das Data Warehouse zulassen, indem Sie den als autorisierte Integrationsquelle hinzufügen.

Anweisungen zum Ausführen der einzelnen Schritte finden Sie unter the section called “Erste Schritte mit Null-ETL-Integrationen”.

Erforderliche Berechtigungen

Bestimmte IAM-Berechtigungen sind erforderlich, um eine Null-ETL-Integration zu erstellen. Sie benötigen mindestens die Berechtigungen, um die folgenden Aktionen durchführen zu können:

- Erstellen Sie Zero-ETL-Integrationen für den DB-Cluster der Quell-RDS-Datenbank.
- Anzeigen und Löschen aller Null-ETL-Integrationen.
- Erstellen eingehender Integrationen in das Ziel-Data-Warehouse.

Die folgenden Beispielrichtlinien zeigen die Berechtigungen mit den geringsten Rechten, die für die Erstellung und Verwaltung von Integrationen erforderlich sind. Möglicherweise benötigen Sie genau diese Berechtigungen nicht, wenn Ihr Benutzer oder Ihre Rolle über umfassendere Berechtigungen verfügt, z. B. über eine `AdministratorAccess` verwaltete Richtlinie.

> **Note**
>
> Redshift Amazon Resource Names (ARNs) haben das folgende Format. Beachten Sie die Verwendung eines Schrägstrichs (`/`) anstelle eines Doppelpunkts (`:`) vor der Serverless-Namespace-UUID.
>
> - Bereitgestellter Cluster – `arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid`
> - Serverless – `arn:aws:redshift-serverless:{region}:{account-id}:namespace/namespace-uuid`

Beispielrichtlinie für Redshift-Ziel

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateIntegration",
      "Effect": "Allow",
      "Action": [
        "rds:CreateIntegration"
      ],
      "Resource": [
        "arn:aws:rds:us-east-1:123456789012:db:source-db",
        "arn:aws:rds:us-east-1:123456789012:integration:*"
      ]
    },
    {
      "Sid": "DescribeIntegrationDetails",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeIntegrations"
      ],
      "Resource": [
      "arn:aws:rds:us-east-1:123456789012:integration:*"
  ]
    },
```

```
    {
      "Sid": "ChangeIntegrationDetails",
      "Effect": "Allow",
      "Action": [
        "rds:DeleteIntegration",
        "rds:ModifyIntegration"
      ],
      "Resource": [
        "arn:aws:rds:us-east-1:123456789012:integration:*"
      ]
    },
    {
      "Sid": "AllowRedShiftIntegration",
      "Effect": "Allow",
      "Action": [
        "redshift:CreateInboundIntegration"
      ],
      "Resource": [
        "arn:aws:redshift:us-east-1:123456789012:namespace:namespace-uuid"
      ]
    }
  ]
}
```

Auswählen eines Ziel-Data-Warehouse in einem anderen Konto

Wenn Sie ein Amazon Redshift Redshift-Ziel-Data Warehouse angeben möchten, das sich in einem anderen befindet AWS-Konto, müssen Sie eine Rolle erstellen, die es Benutzern des aktuellen Kontos ermöglicht, auf Ressourcen im Zielkonto zuzugreifen. Weitere Informationen finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , dem Sie gehören.

Die Rolle muss über die folgenden Berechtigungen verfügen, die es dem Benutzer ermöglichen, verfügbare von Amazon Redshift bereitgestellte Cluster und Redshift-Serverless-Namespaces im Zielkonto einzusehen.

Erforderliche Berechtigungen und Vertrauensrichtlinie

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "redshift:DescribeClusters",
            "redshift-serverless:ListNamespaces"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Die Rolle muss über die folgende Vertrauensrichtlinie verfügen, die die Zielkonto-ID angibt.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Anweisungen zum Erstellen der Rolle finden Sie unter Erstellen einer Rolle mit benutzerdefinierten Vertrauensrichtlinien.

Erstellen von Null-ETL-Integrationen

Sie können eine Zero-ETL-Integration mithilfe der AWS Management Console, der oder der AWS CLI RDS-API erstellen.

> **Important**
>
> Zero-ETL-Integrationen unterstützen keine Aktualisierungs- oder Resynchronisierungsvorgänge. Wenn nach der Erstellung Probleme mit einer Integration auftreten, müssen Sie die Integration löschen und eine neue erstellen.

Standardmäßig löscht RDS für MySQL binäre Protokolldateien sofort. Da Zero-ETL-Integrationen auf Binärlogs angewiesen sind, um Daten von der Quelle zum Ziel zu replizieren, muss der Aufbewahrungszeitraum für die Quelldatenbank mindestens eine Stunde betragen. Sobald Sie eine Integration erstellen, überprüft Amazon RDS den Aufbewahrungszeitraum für binäre Protokolldateien für die ausgewählte Quelldatenbank. Wenn der aktuelle Wert 0 Stunden ist, ändert Amazon RDS ihn automatisch auf 1 Stunde. Andernfalls bleibt der Wert gleich.

RDS-Konsole

So erstellen Sie eine Null-ETL-Integration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Zero-ETL-Integrationen aus.
3. Wählen Sie Zero-ETL-Integration erstellen aus.
4. Geben Sie für Integrationskennung einen Namen für die Integration ein. Der Name kann bis zu 63 alphanumerische Zeichen umfassen und Bindestriche enthalten.

   > **Important**
   >
   > Katalognamen sind auf eine Länge von 19 Zeichen begrenzt. Stellen Sie sicher, dass Ihre Integrations-ID diese Anforderung erfüllt, wenn sie als Katalogname verwendet wird.

5. Wählen Sie Weiter aus.
6. Wählen Sie als Quelle den der RDS-Datenbank aus, aus dem die Daten stammen sollen.

   > **Note**
   >
   > RDS benachrichtigt Sie, wenn die DB--Parameter nicht korrekt konfiguriert sind. Wenn Sie diese Nachricht erhalten, können Sie entweder Reparieren wählen oder eine manuelle Konfiguration vornehmen. Anweisungen zur manuellen Behebung

finden Sie unter the section called “Schritt 1: Erstellen einer benutzerdefinierten DB--Parametergruppe”.
Das Ändern der DB--Parameter erfordert einen Neustart. Bevor Sie die Integration erstellen können, muss der Neustart abgeschlossen sein und die neuen Parameterwerte müssen erfolgreich auf den angewendet werden.

7. Sobald Ihr erfolgreich konfiguriert wurde, wählen Sie Weiter.
8. Gehen Sie bei Ziel wie folgt vor:

   1. (Optional) Um ein anderes AWS-Konto für das Amazon Redshift Redshift-Ziel zu verwenden, wählen Sie Anderes Konto angeben aus. Geben Sie dann den Namen einer IAM-Rolle mit Berechtigungen zur Anzeige Ihrer Data Warehouses ein. Anweisungen zum Erstellen der IAM-Rolle finden Sie unter the section called “Auswählen eines Ziel-Data-Warehouse in einem anderen Konto”.
   2. Wählen Sie für Amazon Redshift Data Warehouse das Ziel für replizierte Daten aus dem aus. Sie können einen bereitgestellten Amazon-Redshift-Cluster oder einen Redshift-Serverless-Namespace als Ziel auswählen.

   > **Note**
   >
   > RDS benachrichtigt Sie, wenn die Ressourcenrichtlinie oder die Einstellungen zur Berücksichtigung der Groß- und Kleinschreibung für das angegebene Data Warehouse nicht korrekt konfiguriert sind. Wenn Sie diese Nachricht erhalten, können Sie entweder Reparieren wählen oder eine manuelle Konfiguration vornehmen. Anweisungen zur manuellen Behebung finden Sie unter Aktivieren der Groß- und Kleinschreibung für Ihr Data Warehouse und Konfigurieren der Autorisierung für Ihr Data Warehouse im Amazon Redshift Management Guide.
   > Das Ändern der Groß- und Kleinschreibung für einen bereitgestellten Redshift-Cluster erfordert einen Neustart. Bevor Sie die Integration erstellen können, muss der Neustart abgeschlossen und der neue Parameterwert erfolgreich auf den Cluster angewendet werden.
   > Wenn sich Ihre gewählte Quelle und Ihr Ziel in verschiedenen AWS-Konten befinden, kann Amazon RDS diese Einstellungen nicht für Sie korrigieren. Sie müssen zu dem anderen Konto navigieren und diese manuell in Amazon Redshift korrigieren.

9. Sobald Ihr Ziel-Data Warehouse korrekt konfiguriert ist, wählen Sie Weiter.

10. (Optional) Fügen Sie unter Tags ein oder mehrere Tags zu der Integration hinzu. Weitere Informationen finden Sie unter the section called " RDS Ressourcen taggen".
11. Geben Sie für Verschlüsselung an, wie Ihre Integration verschlüsselt werden soll. Standardmäßig verschlüsselt RDS alle Integrationen mit einem. AWS-eigener Schlüssel Um stattdessen einen vom Kunden verwalteten Schlüssel auszuwählen, aktivieren Sie die Option Verschlüsselungseinstellungen anpassen und wählen Sie einen KMS-Schlüssel aus, der für die Verschlüsselung verwendet werden soll. Weitere Informationen finden Sie unter the section called "Verschlüsseln von Amazon RDS-Ressourcen".

    Fügen Sie optional einen Verschlüsselungskontext hinzu. Weitere Informationen finden Sie unter Verschlüsselungskontext im AWS Key Management Service -Entwicklerhandbuch.

    > **Note**
    >
    > Amazon RDS fügt zusätzlich zu den von Ihnen hinzugefügten Verschlüsselungskontextpaaren die folgenden Verschlüsselungskontextpaare hinzu:
    >
    > - `aws:redshift:integration:arn` - `IntegrationArn`
    > - `aws:servicename:id` - `Redshift`
    >
    > Dies reduziert die Gesamtzahl der Paare, die Sie hinzufügen können, von 8 auf 6 und trägt zur allgemeinen Zeichenbeschränkung der Grant-Beschränkung bei. Weitere Informationen finden Sie im AWS Key Management Service Developer Guide unter Using Grant Constraints.
12. Wählen Sie Weiter aus.
13. Überprüfen Sie Ihre Integrationseinstellungen und wählen Sie Null-ETL-Integration erstellen aus.

    Wenn die Erstellung fehlschlägt, finden Sie Informationen zur Fehlerbehebung unter the section called "Ich kann keine Null-ETL-Integration erstellen".

Der Status der Integration lautet während der Erstellung `Creating`. Das Ziel-Data-Warehouse von Amazon Redshift hat den Status `Modifying`. Während dieser Zeit können Sie das Data Warehouse nicht abfragen und keine Konfigurationsänderungen daran vornehmen.

Wenn die Integration erfolgreich erstellt wurde, ändern sich sowohl der Status der Integration als auch der Status des Ziel-Data-Warehouse von Amazon Redshift in `Active`.

AWS CLI

Um eine Zero-ETL-Integration mit dem zu erstellen AWS CLI, verwenden Sie den Befehl create-integration mit den folgenden Optionen:

> **Note**
>
> Denken Sie daran, dass Katalognamen auf 19 Zeichen begrenzt sind. Wählen Sie Ihren Integrationsnamen entsprechend aus, wenn er als Katalogname verwendet werden soll.

- `--integration-name` – Geben Sie einen Namen für die Integration an.
- `--source-arn`— Geben Sie den ARN des der RDS-Datenbank an, der die Quelle für die Integration sein wird.
- `--target-arn` – Geben Sie den ARN des Amazon Redshift Data Warehouse an, das das Ziel für die Integration sein soll.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-integration \
    --integration-name my-integration \
    --source-arn arn:aws:rds:{region}:{account-id}:my-db \
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

Windows:

```
aws rds create-integration ^
    --integration-name my-integration ^
    --source-arn arn:aws:rds:{region}:{account-id}:my-db ^
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

RDS-API

Verwenden Sie die `CreateIntegration`-Operation mit den folgenden Parametern, um mithilfe der Amazon-RDS-API eine Null-ETL-Integration zu erstellen:

> **Note**
>
> Katalognamen sind auf 19 Zeichen begrenzt. Stellen Sie sicher, dass Ihr IntegrationName Parameter diese Anforderung erfüllt, wenn er als Katalogname verwendet werden soll.

- `IntegrationName` – Geben Sie einen Namen für die Integration an.
- `SourceArn`— Geben Sie den ARN des der RDS-Datenbank an, der die Quelle für die Integration sein wird.
- `TargetArn` – Geben Sie den ARN des Amazon Redshift Data Warehouse an, das das Ziel für die Integration sein soll.

Verschlüsselung von Integrationen mit einem vom Kunden verwalteten Schlüssel

Wenn Sie AWS-eigener Schlüssel beim Erstellen einer Integration einen benutzerdefinierten KMS-Schlüssel anstelle eines angeben, muss die Schlüsselrichtlinie dem Amazon Redshift Redshift-Serviceprinzipal Zugriff auf die `CreateGrant` Aktion gewähren. Darüber hinaus muss es dem aktuellen Benutzer ermöglicht werden, die `CreateGrant` Aktionen `DescribeKey` und auszuführen.

Die folgende Beispielrichtlinie zeigt, wie die erforderlichen Berechtigungen in der Schlüsselrichtlinie bereitgestellt werden. Sie enthält Kontextschlüssel, um den Umfang der Berechtigungen weiter zu reduzieren.

Beispiel für eine Schlüsselrichtlinie

JSON

```
{
    "Version": "2012-10-17",
    "Id": "Key policy",
    "Statement": [
        {
            "Sid": "Enables IAM user permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
```

```
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allows the Redshift service principal to add a grant to a KMS
key",
            "Effect": "Allow",
            "Principal": {
                "Service": "redshift.amazonaws.com"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}": "{context-value}"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "Allows the current user or role to add a grant to a KMS key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/{role-name}"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}": "{context-value}",
                    "kms:ViaService": "rds.us-east-1.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
```

```
                }
            }
        },
        {
            "Sid": "Allows the current uer or role to retrieve information about
 a KMS key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/{role-name}"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*"
        }
    ]
}
```

Weitere Informationen finden Sie unter Erstellen einer Schlüsselrichtlinie im AWS Key Management Service -Entwicklerhandbuch.

Nächste Schritte

Nachdem Sie erfolgreich eine Null-ETL-Integration erstellt haben, müssen Sie eine Zieldatenbank in Ihrem Amazon-Redshift-Zielcluster oder Ihrer Zielarbeitsgruppe erstellen. Anschließend können Sie damit beginnen, Daten zum der Quell-RDS-Datenbank hinzuzufügen und sie in Amazon Redshift abzufragen. Anweisungen finden Sie unter Erstellen von Zieldatenbanken in Amazon Redshift.

Erstellen von Amazon RDS Zero-ETL-Integrationen mit einem Lakehouse Amazon SageMaker

Wenn Sie eine Amazon RDS Zero-ETL-Integration mit einem Amazon SageMaker Lakehouse erstellen, geben Sie den der Quell-RDS-Datenbank und den verwalteten AWS Glue Zielkatalog an. Sie können auch die Verschlüsselungseinstellungen anpassen und Tags hinzufügen. Amazon RDS erstellt eine Integration zwischen dem und seinem Ziel. Sobald die Integration aktiv ist, werden alle Daten, die Sie in den einfügen, in das konfigurierte Ziel repliziert.

Voraussetzungen

Bevor Sie eine Zero-ETL-Integration mit einem Amazon SageMaker Lakehouse erstellen, müssen Sie einen und einen verwalteten Zielkatalog erstellen. AWS Glue Sie müssen auch die Replikation in den Katalog zulassen, indem Sie den als autorisierte Integrationsquelle hinzufügen.

Anweisungen zum Ausführen der einzelnen Schritte finden Sie unter the section called “Erste Schritte mit Null-ETL-Integrationen”.

Erforderliche Berechtigungen

Bestimmte IAM-Berechtigungen sind erforderlich, um eine Zero-ETL-Integration mit einem Lakehouse zu erstellen. Amazon SageMaker Sie benötigen mindestens die Berechtigungen, um die folgenden Aktionen durchführen zu können:

- Erstellen Sie Zero-ETL-Integrationen für den DB-Cluster der Quell-RDS-Datenbank.
- Anzeigen und Löschen aller Null-ETL-Integrationen.
- Erstellen Sie eingehende Integrationen in den verwalteten Zielkatalog. AWS Glue
- Greifen Sie auf Amazon S3 S3-Buckets zu, die vom AWS Glue verwalteten Katalog verwendet werden.
- Verwenden Sie AWS KMS Schlüssel für die Verschlüsselung, wenn eine benutzerdefinierte Verschlüsselung konfiguriert ist.
- Registrieren Sie Ressourcen bei Lake Formation.
- Fügen Sie dem AWS Glue verwalteten Katalog eine Ressourcenrichtlinie hinzu, um eingehende Integrationen zu autorisieren.

Die folgende Beispielrichtlinie zeigt die Berechtigungen mit den geringsten Rechten, die zum Erstellen und Verwalten von Integrationen mit einem Lakehouse erforderlich sind. Amazon SageMaker Möglicherweise benötigen Sie genau diese Berechtigungen nicht, wenn Ihr Benutzer oder Ihre Rolle über umfassendere Berechtigungen verfügt, z. B. über eine `AdministratorAccess` verwaltete Richtlinie.

Darüber hinaus müssen Sie eine Ressourcenrichtlinie für den AWS Glue verwalteten Zielkatalog konfigurieren, um eingehende Integrationen zu autorisieren. Verwenden Sie den folgenden AWS CLI Befehl, um die Ressourcenrichtlinie anzuwenden.

AWS CLI Beispielbefehl zur Autorisierung eingehender Integrationen im Zielkatalog

```
aws glue put-resource-policy \
      --policy-in-json  '{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "glue.amazonaws.com"
        },
        "Action": [
            "glue:AuthorizeInboundIntegration"
        ],
        "Resource": ["arn:aws:glue:region:account_id:catalog/catalog_name"],
        "Condition": {
            "StringEquals": {
                "aws:SourceArn": "arn:aws:rds:region:account_id:db:source_name"
            }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "account_id"
        },
        "Action": ["glue:CreateInboundIntegration"],
        "Resource": ["arn:aws:glue:region:account_id:catalog/catalog_name"]
    }
    ]
}' \
      --region region
```

> **Note**
>
> Amazon Resource Names (ARNs) für den Glue-Katalog haben das folgende Format:
>
> - Kleberkatalog — `arn:aws:glue:{region}:{account-id}:catalog/catalog-name`

Auswahl eines AWS Glue verwalteten Zielkatalogs in einem anderen Konto

Wenn Sie einen AWS Glue verwalteten Zielkatalog angeben möchten, der sich in einem anderen AWS-Konto befindet, müssen Sie eine Rolle erstellen, die es Benutzern im aktuellen Konto ermöglicht, auf Ressourcen im Zielkonto zuzugreifen. Weitere Informationen finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , dem Sie gehören.

Die Rolle muss über die folgenden Berechtigungen verfügen, die es dem Benutzer ermöglichen, die verfügbaren AWS Glue Kataloge im Zielkonto einzusehen.

Erforderliche Berechtigungen und Vertrauensrichtlinie

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "glue:GetCatalog"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Die Rolle muss über die folgende Vertrauensrichtlinie verfügen, die die Zielkonto-ID angibt.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": "arn:aws:iam::111122223333:root"
```

```
        },
        "Action":"sts:AssumeRole"
      }
   ]
}
```

Anweisungen zum Erstellen der Rolle finden Sie unter Erstellen einer Rolle mit benutzerdefinierten Vertrauensrichtlinien.

Erstellung von Zero-ETL-Integrationen mit einem Lakehouse Amazon SageMaker

Sie können eine Zero-ETL-Integration mit einem Amazon SageMaker Lakehouse mithilfe der AWS Management Console, der oder der RDS-API erstellen. AWS CLI

> **Important**
>
> Zero-ETL-Integrationen mit einem Amazon SageMaker Lakehouse unterstützen keine Aktualisierungs- oder Resynchronisierungsvorgänge. Wenn nach der Erstellung Probleme mit einer Integration auftreten, müssen Sie die Integration löschen und eine neue erstellen.

Standardmäßig löscht RDS für MySQL binäre Protokolldateien sofort. Da Zero-ETL-Integrationen auf Binärlogs angewiesen sind, um Daten von der Quelle zum Ziel zu replizieren, muss der Aufbewahrungszeitraum für die Quelldatenbank mindestens eine Stunde betragen. Sobald Sie eine Integration erstellen, überprüft Amazon RDS den Aufbewahrungszeitraum für binäre Protokolldateien für die ausgewählte Quelldatenbank. Wenn der aktuelle Wert 0 Stunden ist, ändert Amazon RDS ihn automatisch auf 1 Stunde. Andernfalls bleibt der Wert gleich.

RDS-Konsole

Um eine Zero-ETL-Integration mit einem Lakehouse zu erstellen Amazon SageMaker

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Zero-ETL-Integrationen aus.
3. Wählen Sie Zero-ETL-Integration erstellen aus.

4. Geben Sie für Integrationskennung einen Namen für die Integration ein. Der Name kann bis zu 63 alphanumerische Zeichen umfassen und Bindestriche enthalten.
5. Wählen Sie Weiter aus.
6. Wählen Sie als Quelle den der RDS-Datenbank aus, aus dem die Daten stammen sollen.

> **Note**
>
> RDS benachrichtigt Sie, wenn die DB--Parameter nicht korrekt konfiguriert sind. Wenn Sie diese Nachricht erhalten, können Sie entweder Reparieren wählen oder eine manuelle Konfiguration vornehmen. Anweisungen zur manuellen Behebung finden Sie unter the section called “Schritt 1: Erstellen einer benutzerdefinierten DB--Parametergruppe”.
> Das Ändern der DB--Parameter erfordert einen Neustart. Bevor Sie die Integration erstellen können, muss der Neustart abgeschlossen sein und die neuen Parameterwerte müssen erfolgreich auf den angewendet werden.

7. Sobald Ihr erfolgreich konfiguriert wurde, wählen Sie Weiter.
8. Gehen Sie bei Ziel wie folgt vor:

   1. (Optional) Um ein anderes AWS-Konto für das Amazon SageMaker Lakehouse-Ziel zu verwenden, wählen Sie Anderes Konto angeben aus. Geben Sie dann den ARN einer IAM-Rolle mit Berechtigungen zum Anzeigen Ihrer AWS Glue Kataloge ein. Anweisungen zum Erstellen der IAM-Rolle finden Sie unter the section called “Auswahl eines AWS Glue verwalteten Zielkatalogs in einem anderen Konto”.
   2. Sie können einen vorhandenen AWS Glue verwalteten Katalog als Ziel wählen.
   3. Die Ziel-IAM-Rolle muss die Berechtigungen für den Zielkatalog beschreiben und über die folgenden Berechtigungen verfügen:

      JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "glue:GetCatalog",
            "Resource": [
```

```
                "arn:aws:glue:us-east-1:111122223333:catalog/*",
                "arn:aws:glue:us-east-1:111122223333:catalog"
            ]
        }
    ]
}
```

Die Ziel-IAM-Rolle muss über die folgende Vertrauensstellung verfügen:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "glue.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

4. Sie müssen der Ziel-IAM-Rolle Beschreibungsberechtigungen für den AWS Glue verwalteten Zielkatalog mit der Lake Formation-Administratorrolle gewähren, die in Schritt 3b: Erstellen Sie einen AWS Glue Katalog für die Amazon SageMaker AI Zero-ETL-Integration erstellt wurde.

> **Note**
>
> RDS benachrichtigt Sie, wenn die Ressourcenrichtlinien oder die Konfigurationseinstellungen für den angegebenen AWS Glue verwalteten Katalog nicht korrekt konfiguriert sind. Wenn Sie diese Nachricht erhalten, können Sie entweder Reparieren wählen oder eine manuelle Konfiguration vornehmen.
> Wenn sich Ihre gewählte Quelle und Ihr Ziel in verschiedenen AWS-Konten befinden, kann Amazon RDS diese Einstellungen nicht für Sie korrigieren. Sie müssen zu dem anderen Konto navigieren und die Fehler manuell korrigieren. SageMaker Unified Studio

9. Sobald Ihr AWS Glue verwalteter Zielkatalog korrekt konfiguriert ist, wählen Sie Weiter.

10. (Optional) Fügen Sie unter Tags ein oder mehrere Tags zu der Integration hinzu. Weitere Informationen finden Sie unter the section called " RDS Ressourcen taggen".
11. Geben Sie für Verschlüsselung an, wie Ihre Integration verschlüsselt werden soll. Standardmäßig verschlüsselt RDS alle Integrationen mit einem. AWS-eigener Schlüssel Um stattdessen einen vom Kunden verwalteten Schlüssel auszuwählen, aktivieren Sie die Option Verschlüsselungseinstellungen anpassen und wählen Sie einen KMS-Schlüssel aus, der für die Verschlüsselung verwendet werden soll. Weitere Informationen finden Sie unter the section called "Verschlüsseln von Amazon RDS-Ressourcen".

    Fügen Sie optional einen Verschlüsselungskontext hinzu. Weitere Informationen finden Sie unter Verschlüsselungskontext im AWS Key Management Service -Entwicklerhandbuch.

    > **Note**
    >
    > Amazon RDS fügt zusätzlich zu den von Ihnen hinzugefügten Verschlüsselungskontextpaaren die folgenden Verschlüsselungskontextpaare hinzu:
    >
    > - `aws:glue:integration:arn` - `IntegrationArn`
    > - `aws:servicename:id` - `glue`
    >
    > Dies reduziert die Gesamtzahl der Paare, die Sie hinzufügen können, von 8 auf 6 und trägt zur allgemeinen Zeichenbeschränkung der Grant-Beschränkung bei. Weitere Informationen finden Sie im AWS Key Management Service Developer Guide unter Using Grant Constraints.

12. Wählen Sie Weiter aus.
13. Überprüfen Sie Ihre Integrationseinstellungen und wählen Sie Null-ETL-Integration erstellen aus.

    Wenn die Erstellung fehlschlägt, finden Sie Informationen zur Fehlerbehebung unter the section called "Fehlerbehebung bei Null-ETL-Integrationen".

Die Integration hat den Status „`Creating`Wird gerade erstellt", und das Amazon SageMaker Ziel-Lakehouse hat den Status. `Modifying` Während dieser Zeit können Sie den Katalog nicht abfragen und keine Konfigurationsänderungen daran vornehmen.

Wenn die Integration erfolgreich erstellt wurde, ändern sich sowohl der Status der Integration als auch des Amazon SageMaker Ziel-Lakehouse auf`Active`.

AWS CLI

Um einen AWS Glue verwalteten Zielkatalog für die Zero-ETL-Integration mit dem vorzubereiten AWS CLI, müssen Sie zunächst den create-integration-resource-propertyBefehl mit den folgenden Optionen verwenden:

- `--resource-arn`— Geben Sie den ARN des AWS Glue verwalteten Katalogs an, der das Ziel für die Integration sein soll.
- `--target-processing-properties`— Geben Sie den ARN der IAM-Rolle an, um auf den AWS Glue verwalteten Zielkatalog zuzugreifen

```
aws glue create-integration-resource-property --region us-east-1
 --resource-arn arn:aws:glue:region:account_id:catalog/catalog_name \
 --target-processing-properties '{"RoleArn" : "arn:aws:iam::account_id:role/
TargetIamRole"}'
```

Um mithilfe von eine Zero-ETL-Integration mit einem Amazon SageMaker Lakehouse zu erstellen AWS CLI, verwenden Sie den Befehl create-integration mit den folgenden Optionen:

- `--integration-name` – Geben Sie einen Namen für die Integration an.
- `--source-arn`— Geben Sie den ARN des der RDS-Datenbank an, der die Quelle für die Integration sein wird.
- `--target-arn`— Geben Sie den ARN des AWS Glue verwalteten Katalogs an, der das Ziel für die Integration sein soll.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-integration \
    --integration-name my-sagemaker-integration \
    --source-arn arn:aws:rds:{region}:{account-id}:my-db \
    --target-arn arn:aws:glue:{region}:{account-id}:catalog/catalog-name
```

Windows:

```
aws rds create-integration ^
    --integration-name my-sagemaker-integration ^
```

```
    --source-arn arn:aws:rds:{region}:{account-id}:my-db ^
    --target-arn arn:aws:glue:{region}:{account-id}:catalog/catalog-name
```

RDS-API

Um mithilfe der Amazon RDS-API eine Zero-ETL-Integration zu erstellen, verwenden Sie den `CreateIntegration`Vorgang mit den folgenden Parametern: Amazon SageMaker

> **Note**
>
> Katalognamen sind auf 19 Zeichen begrenzt. Stellen Sie sicher, dass Ihr IntegrationName Parameter diese Anforderung erfüllt, wenn er als Katalogname verwendet werden soll.

- `IntegrationName` – Geben Sie einen Namen für die Integration an.
- `SourceArn`— Geben Sie den ARN des der RDS-Datenbank an, der die Quelle für die Integration sein wird.
- `TargetArn`— Geben Sie den ARN des AWS Glue verwalteten Katalogs an, der das Ziel für die Integration sein soll.

Verschlüsselung von Integrationen mit einem vom Kunden verwalteten Schlüssel

Wenn Sie AWS-eigener Schlüssel beim Erstellen einer Integration mit einen benutzerdefinierten KMS-Schlüssel und nicht einen angebenAmazon SageMaker, muss die Schlüsselrichtlinie dem SageMaker Unified Studio Dienstprinzipal Zugriff auf die Aktion gewähren. `CreateGrant` Darüber hinaus muss es dem aktuellen Benutzer ermöglicht werden, die `CreateGrant` Aktionen `DescribeKey` und auszuführen.

Die folgende Beispielrichtlinie zeigt, wie die erforderlichen Berechtigungen in der Schlüsselrichtlinie bereitgestellt werden. Sie enthält Kontextschlüssel, um den Umfang der Berechtigungen weiter zu reduzieren.

Beispiel für eine Schlüsselrichtlinie

JSON

```
{
```

```
    "Version": "2012-10-17",
    "Id": "Key policy",
    "Statement": [
        {
            "Sid": "EnablesIAMUserPermissions",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "AllowsGlueServicePrincipalAddGrant AWS KMS Key",
            "Effect": "Allow",
            "Principal": {
                "Service": "glue.amazonaws.com"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}":"{context-value}"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "AllowsCurrentUserRoleAddGrantKMSKey",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/{role-name}"
            },
            "Action": "kms:CreateGrant",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:{context-key}":"{context-value}",
```

```
                    "kms:ViaService": "rds.us-east-1.amazonaws.com"
                },
                "ForAllValues:StringEquals": {
                    "kms:GrantOperations": [
                        "Decrypt",
                        "GenerateDataKey",
                        "CreateGrant"
                    ]
                }
            }
        },
        {
            "Sid": "AllowsCurrentUserRoleRetrieveKMSKeyInformation",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/{role-name}"
            },
            "Action": "kms:DescribeKey",
            "Resource": "*"
        }
    ]
}
```

Weitere Informationen finden Sie unter Erstellen einer Schlüsselrichtlinie im AWS Key Management Service -Entwicklerhandbuch.

Nächste Schritte

Nachdem Sie erfolgreich eine Zero-ETL-Integration mit erstellt haben, können Sie damit beginnenAmazon SageMaker, Daten zum der Quell-RDS-Datenbank hinzuzufügen und sie in Ihrem Lakehouse abzufragen. Amazon SageMaker Die Daten werden automatisch repliziert und für Analysen und maschinelles Lernen zur Verfügung gestellt.

Datenfilterung für Amazon RDS Zero-ETL-Integrationen

Amazon RDS Zero-ETL-Integrationen unterstützen Datenfilterung, mit der Sie steuern können, welche Daten aus Ihrem der Amazon RDS-Quelldatenbank in Ihr Ziel-Data Warehouse repliziert werden. Anstatt die gesamte Datenbank zu replizieren, können Sie einen oder mehrere Filter anwenden, um bestimmte Tabellen selektiv ein- oder auszuschließen. Auf diese Weise können Sie die Speicher- und Abfrageleistung optimieren, indem sichergestellt wird, dass nur relevante Daten

übertragen werden. Derzeit ist die Filterung auf Datenbank- und Tabellenebene beschränkt. Das Filtern auf Spalten- und Zeilenebene wird nicht unterstützt.

Datenfilterung kann nützlich sein, wenn Sie:

- Verbinden Sie bestimmte Tabellen aus zwei oder mehr verschiedenen , und Sie benötigen keine vollständigen Daten aus einem der .
- Sparen Sie Kosten, indem Sie Analysen nur mit einer Teilmenge von Tabellen und nicht mit einer ganzen Flotte von Datenbanken durchführen.
- Filtern Sie vertrauliche Informationen wie Telefonnummern, Adressen oder Kreditkarteninformationen aus bestimmten Tabellen heraus.

Sie können Datenfilter zu einer Zero-ETL-Integration hinzufügen AWS Management Console, indem Sie die AWS Command Line Interface (AWS CLI) oder die Amazon RDS-API verwenden.

Wenn die Integration einen bereitgestellten Cluster als Ziel hat, muss sich der Cluster auf Patch 180 oder höher befinden, um die Datenfilterung verwenden zu können.

Themen

- Format eines Datenfilters
- Filterlogik
- Priorität filtern
- RDS für MySQL-Beispiele
- — Beispiele
- Beispiele für RDS für Oracle
- Hinzufügen von Datenfiltern zu einer Integration
- Datenfilter aus einer Integration entfernen

Format eines Datenfilters

Sie können mehrere Filter für eine einzelne Integration definieren. Jeder Filter schließt alle vorhandenen und future Datenbanktabellen ein oder schließt sie aus, die einem der Muster im Filterausdruck entsprechen. Amazon RDS Zero-ETL-Integrationen verwenden die Maxwell-Filtersyntax für die Datenfilterung.

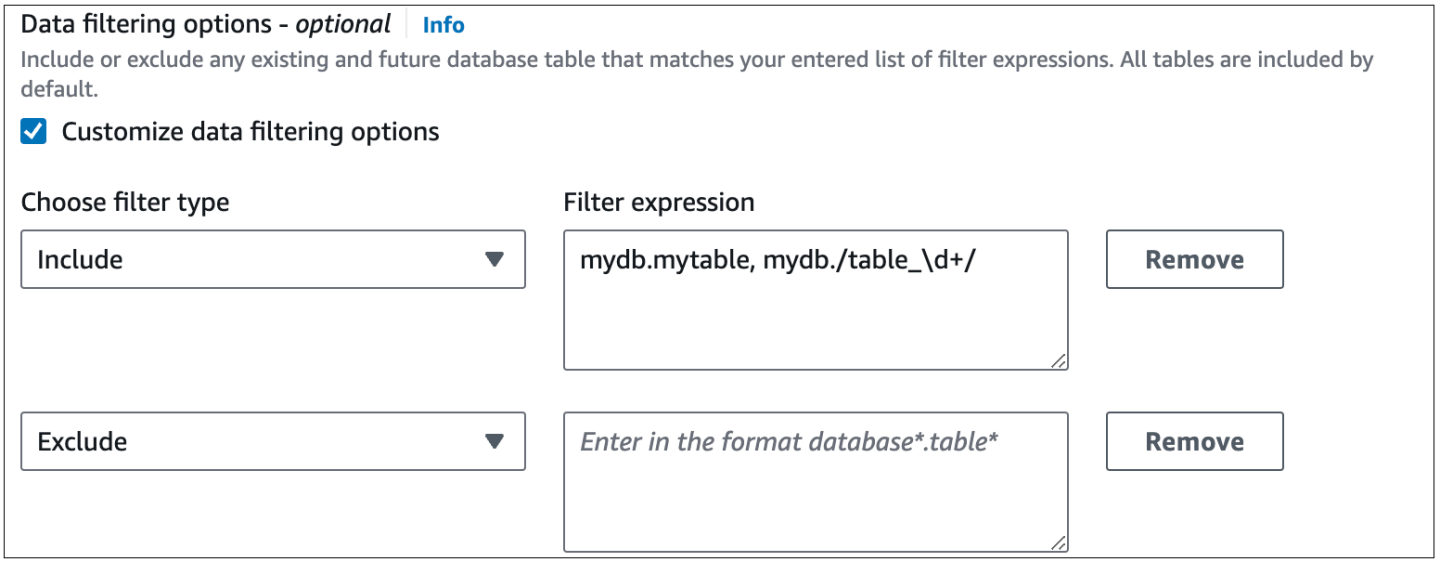
Jeder Filter besteht aus den folgenden Elementen:

Element	Beschreibung
Typ des Filters	Ein `Include` Filtertyp umfasst alle Tabellen, die einem der Muster im Filterausdruck entsprechen. Ein `Exclude` Filtertyp schließt alle Tabellen aus, die einem der Muster entsprechen.
Filterausdruck	Eine durch Kommas getrennte Liste von Mustern. Ausdrücke müssen die Maxwell-Filtersyntax verwenden.
Muster	Ein Filtermuster im Format *`database`*`.`*`table`* für RDS für MySQL oder *`database`*`.`*`schema`*`.`*`table`* für . Sie können Literalnamen angeben oder reguläre Ausdrücke definieren. **Note** Für RDS for MySQL werden reguläre Ausdrücke sowohl im Datenbank- als auch im Tabellennamen unterstützt. Für RDS for PostgreSQL werden reguläre Ausdrücke nur im Schema- und Tabellennamen unterstützt, nicht im Datenbanknamen. Sie können keine Filter oder Ablehnungslisten auf Spaltenebene einbeziehen. Eine einzelne Integration kann insgesamt maximal 99 Muster haben. In der Konsole können Sie Muster innerhalb eines einzelnen Filterausdrucks eingeben oder sie auf

Element	Beschreibung
	mehrere Ausdrücke verteilen. Ein einzelnes Muster darf nicht länger als 256 Zeichen sein.

> **Important**
>
> Wenn Sie einen RDS für die PostgreSQL-Quelldatenbank auswählen, müssen Sie mindestens ein Datenfiltermuster angeben. Das Muster muss mindestens eine einzelne Datenbank (*`database-name`*`.*.*`) für die Replikation in das Ziel-Data Warehouse enthalten.

Die folgende Abbildung zeigt die Struktur der RDS for MySQL-Datenfilter in der Konsole:



> **Important**
>
> Nehmen Sie keine personenbezogenen, vertraulichen oder sensiblen Informationen in Ihre Filtermuster auf.

Datenfilter im AWS CLI

Wenn Sie den verwenden AWS CLI , um einen Datenfilter hinzuzufügen, unterscheidet sich die Syntax geringfügig von der der Konsole. Sie müssen jedem Muster einzeln einen Filtertyp

(`Include`oder`Exclude`) zuweisen, sodass Sie nicht mehrere Muster unter einem Filtertyp gruppieren können.

In der Konsole können Sie beispielsweise die folgenden durch Kommas getrennten Muster unter einer einzigen `Include` Anweisung gruppieren:

RDS für MySQL

```
mydb.mytable, mydb./table_\d+/
```

```
mydb.myschema.mytable, mydb.myschema./table_\d+/
```

Wenn Sie den verwenden AWS CLI, muss derselbe Datenfilter jedoch das folgende Format haben:

RDS für MySQL

```
'include: mydb.mytable, include: mydb./table_\d+/'
```

```
'include: mydb.myschema.mytable, include: mydb.myschema./table_\d+/'
```

Filterlogik

Wenn Sie in Ihrer Integration keine Datenfilter angeben, geht Amazon RDS von einem Standardfilter von aus`include:*.*`, der alle Tabellen in das Ziel-Data Warehouse repliziert. Wenn Sie jedoch mindestens einen Filter hinzufügen, wechselt die Standardlogik zu`exclude:*.*`, wodurch standardmäßig alle Tabellen ausgeschlossen werden. Auf diese Weise können Sie explizit definieren, welche Datenbanken und Tabellen in die Replikation einbezogen werden sollen.

Wenn Sie beispielsweise den folgenden Filter definieren:

```
'include: db.table1, include: db.table2'
```

Amazon RDS bewertet den Filter wie folgt:

```
'exclude:*.*, include: db.table1, include: db.table2'
```

Daher repliziert Amazon RDS nur `table1` und `table2` von der angegebenen Datenbank in db das Ziel-Data Warehouse.

Priorität filtern

Amazon RDS wertet Datenfilter in der von Ihnen angegebenen Reihenfolge aus. In der AWS Management Console verarbeitet es Filterausdrücke von links nach rechts und von oben nach unten. Ein zweiter Filter oder ein individuelles Muster, das dem ersten folgt, kann ihn überschreiben.

Wenn der erste Filter beispielsweise aktiviert ist`Include books.stephenking`, schließt er nur die `stephenking` Tabelle aus der `books` Datenbank ein. Wenn Sie jedoch einen zweiten Filter hinzufügen`Exclude books.*`, überschreibt dieser den ersten Filter. Dadurch wird verhindert, dass Tabellen aus dem `books` Index in das Ziel-Data Warehouse repliziert werden.

Wenn Sie mindestens einen Filter angeben, beginnt die Logik mit der Standardannahme, wodurch automatisch alle Tabellen `exclude:*.*` von der Replikation ausgeschlossen werden. Es hat sich bewährt, Filter vom breitesten bis zum spezifischsten Filter zu definieren. Beginnen Sie mit einer oder mehreren `Include` Anweisungen, um die zu replizierenden Daten anzugeben, und fügen Sie dann `Exclude` Filter hinzu, um bestimmte Tabellen selektiv zu entfernen.

Das gleiche Prinzip gilt für Filter, die Sie mit dem definieren. AWS CLIAmazon RDS wertet diese Filtermuster in der Reihenfolge aus, in der Sie sie angeben, sodass ein Muster möglicherweise ein Muster überschreibt, das Sie zuvor angegeben haben.

RDS für MySQL-Beispiele

Die folgenden Beispiele zeigen, wie Datenfilterung für RDS für MySQL-Beispiele Zero-ETL-Integrationen:

- Schließt alle Datenbanken und alle Tabellen ein:

```
'include: *.*'
```

- Schließt alle Tabellen in die `books` Datenbank ein:

```
'include: books.*'
```

- Schließt alle Tabellen mit den folgenden Namen aus`mystery`:

```
'include: *.*, exclude: *.mystery'
```

- Schließt zwei spezifische Tabellen in die `books` Datenbank ein:

```
'include: books.stephen_king, include: books.carolyn_keene'
```

- Schließt alle Tabellen in die `books` Datenbank ein, außer denen, die die Teilzeichenfolge `mystery` enthalten:

```
'include: books.*, exclude: books./.*mystery.*/'
```

- Schließt alle Tabellen in die `books` Datenbank ein, außer denen, die beginnen mit`mystery`:

```
'include: books.*, exclude: books./mystery.*/'
```

- Schließt alle Tabellen in die `books` Datenbank ein, mit Ausnahme der Tabellen, die wie folgt enden`mystery`:

```
'include: books.*, exclude: books./.*mystery/'
```

- Schließt alle Tabellen in der `books` Datenbank ein, die mit beginnen`table_`, mit Ausnahme der benannten`table_stephen_king`. Zum Beispiel, `table_movies` oder `table_books` würde repliziert werden, aber nicht`table_stephen_king`.

```
'include: books./table_.*/, exclude: books.table_stephen_king'
```

— Beispiele

Die folgenden Beispiele zeigen, wie Datenfilterung für funktioniert:

- Schließt alle Tabellen in die Datenbank ein: `books`

```
'include: books.*.*'
```

- Schließt alle `mystery` in der `books` Datenbank benannten Tabellen aus:

```
'include: books.*.*, exclude: books.*.mystery'
```

- Nehmen Sie eine Tabelle innerhalb der `books` Datenbank in das `mystery` Schema und eine Tabelle innerhalb der `employee` Datenbank in das `finance` Schema auf:

```
'include: books.mystery.stephen_king, include: employee.finance.benefits'
```

- Schließt alle Tabellen in die `books` Datenbank und das `science_fiction` Schema ein, mit Ausnahme derjenigen, die die Teilzeichenfolge `king` enthalten:

```
'include: books.science_fiction.*, exclude: books.*./.*king.*/
```

- Schließt alle Tabellen in die `books` Datenbank ein, außer denen, deren Schemaname beginnt mit`sci`:

```
'include: books.*.*, exclude: books./sci.*/.*'
```

- Schließt alle Tabellen in der `books` Datenbank ein, mit Ausnahme der Tabellen im `mystery` Schema, die auf Folgendes enden`king`:

```
'include: books.*.*, exclude: books.mystery./.*king/'
```

- Schließt alle Tabellen in der `books` Datenbank ein, die mit beginnen`table_`, mit Ausnahme der benannten`table_stephen_king`. Zum Beispiel werden `table_movies` im `fiction` Schema und `table_books` im `mystery` Schema repliziert, aber `table_stephen_king` in keinem der Schemas:

```
'include: books.*./table_.*/, exclude: books.*.table_stephen_king'
```

Beispiele für RDS für Oracle

Die folgenden Beispiele zeigen, wie Datenfilterung für RDS for Oracle Zero-ETL-Integrationen funktioniert:

- Schließt alle Tabellen in die Bücherdatenbank ein:

```
'include: books.*.*'
```

- Schließt alle Tabellen mit dem Namen Mystery aus der Bücherdatenbank aus:

```
'include: books.*.*, exclude: books.*.mystery'
```

- Nehmen Sie eine Tabelle innerhalb der Bücherdatenbank in das Mystery-Schema und eine Tabelle in der Mitarbeiterdatenbank in das Finanzschema auf:

```
'include: books.mystery.stephen_king, include: employee.finance.benefits'
```

- Schließt alle Tabellen des Mystery-Schemas in die Bücherdatenbank ein:

```
'include: books.mystery.*'
```

> **Note**
>
> Die Verwendung regulärer Ausdrücke im Filterwert für Datenbanknamen, Schema oder Tabellennamen wird in RDS for Oracle-Integrationen nicht unterstützt.

Hinzufügen von Datenfiltern zu einer Integration

Sie können die Datenfilterung mithilfe der AWS Management Console AWS CLI, der oder der Amazon RDS-API konfigurieren.

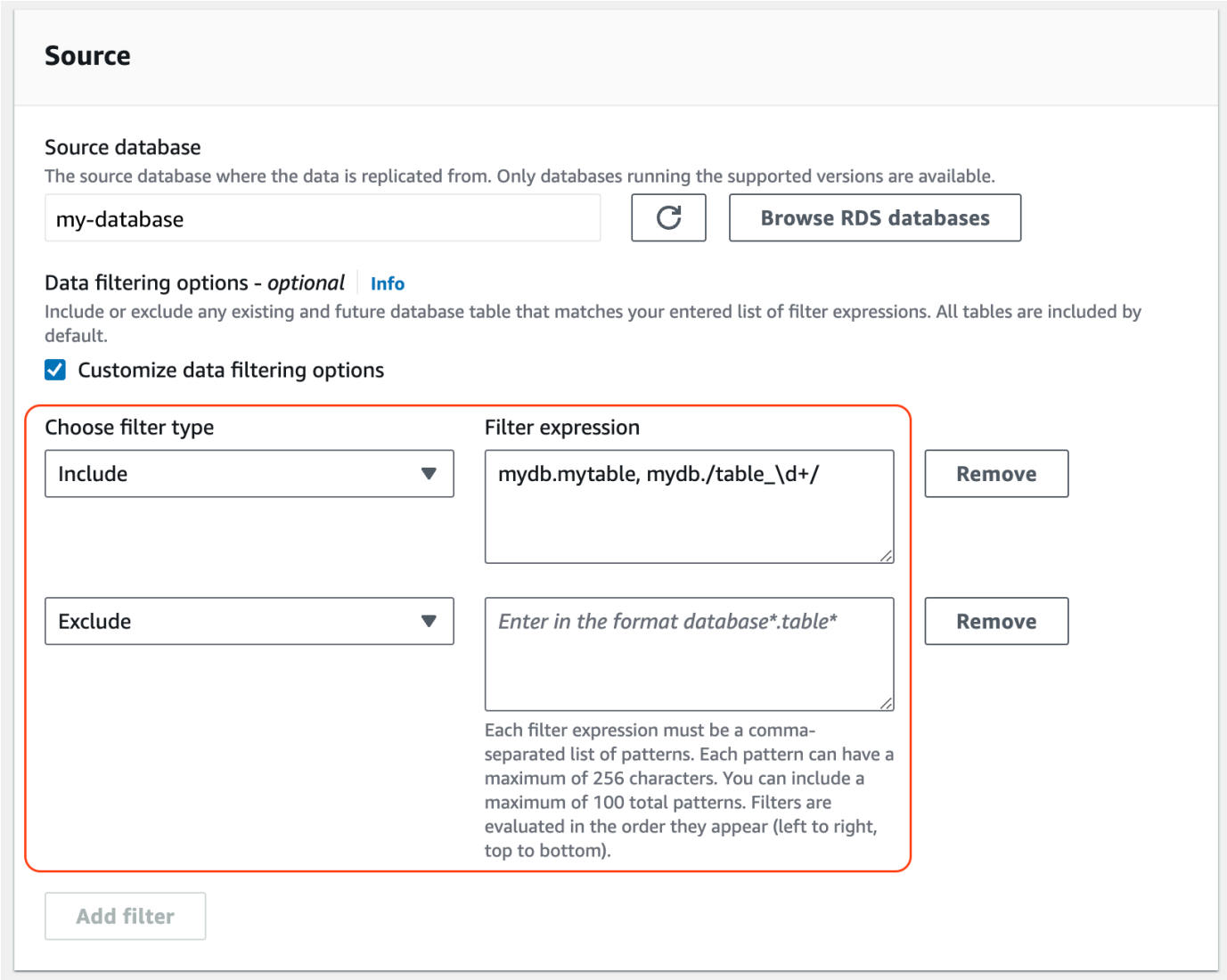
> **Important**
>
> Wenn Sie einen Filter hinzufügen, nachdem Sie eine Integration erstellt haben, behandelt Amazon RDS ihn so, als ob er schon immer existiert hätte. Es entfernt alle Daten im Ziel-Data Warehouse, die nicht den neuen Filterkriterien entsprechen, und synchronisiert alle betroffenen Tabellen erneut.

RDS-Konsole

Um Datenfilter zu einer Zero-ETL-Integration hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Zero-ETL-Integrationen aus. Wählen Sie die Integration aus, zu der Sie Datenfilter hinzufügen möchten, und klicken Sie dann auf Ändern.
3. Fügen Sie unter Quelle eine oder mehrere `Include` `Exclude` AND-Anweisungen hinzu.

   Die folgende Abbildung zeigt ein Beispiel für Datenfilter für eine MySQL-Integration:



4. Wenn Sie mit den Änderungen zufrieden sind, wählen Sie Weiter und Änderungen speichern aus.

AWS CLI

Um einer Zero-ETL-Integration mithilfe von Datenfilter hinzuzufügen AWS CLI, rufen Sie den Befehl modify-integration auf. Geben Sie den `--data-filter` Parameter zusätzlich zur Integrations-ID mit einer kommagetrennten Liste von und Maxwell-Filtern an. `Include` `Exclude`

Example

Im folgenden Beispiel werden Filtermuster zu hinzugefügt. `my-integration`

Für LinuxmacOS, oderUnix:

```
aws rds modify-integration \
    --integration-identifier my-integration \
    --data-filter 'include: foodb.*, exclude: foodb.tbl, exclude: foodb./table_\d+/'
```

Windows:

```
aws rds modify-integration ^
    --integration-identifier my-integration ^
    --data-filter 'include: foodb.*, exclude: foodb.tbl, exclude: foodb./table_\d+/'
```

RDS-API

Um eine Zero-ETL-Integration mithilfe der RDS-API zu ändern, rufen Sie den Vorgang auf ModifyIntegration. Geben Sie die Integrations-ID an und stellen Sie eine durch Kommas getrennte Liste von Filtermustern bereit.

Datenfilter aus einer Integration entfernen

Wenn Sie einen Datenfilter aus einer Integration entfernen, bewertet Amazon RDS die verbleibenden Filter neu, als ob der entfernte Filter nie existiert hätte. Anschließend repliziert es alle zuvor ausgeschlossenen Daten, die jetzt die Kriterien erfüllen, in das Ziel-Data Warehouse. Dadurch wird eine Neusynchronisierung aller betroffenen Tabellen ausgelöst.

Hinzufügen von Daten zu einem einer RDS-Quelldatenbank und deren Abfrage

Um die Erstellung einer Zero-ETL-Integration abzuschließen, die Daten von Amazon RDS Amazon Redshift repliziert, müssen Sie eine Datenbank im Zielziel erstellen.

Stellen Sie für Verbindungen mit Amazon Redshift eine Verbindung zu Ihrem Amazon Redshift Redshift-Cluster oder Ihrer Arbeitsgruppe her und erstellen Sie eine Datenbank mit einem Verweis auf Ihre Integrations-ID. Anschließend können Sie Daten zu Ihrem für die Quell-RDS-Datenbank hinzufügen und sehen, wie sie in Amazon Redshift oder repliziert werden. Amazon SageMaker

Themen

- Eine Zieldatenbank erstellen
- Daten zum der Quelldatenbank hinzufügen

- Abfragen Ihrer Amazon RDS in Amazon Redshift
- Datentypunterschiede zwischen RDS und Amazon Redshift-Datenbanken
-

Eine Zieldatenbank erstellen

Bevor Sie mit der Replikation von Daten in Amazon Redshift beginnen können, müssen Sie nach dem Erstellen einer Integration eine Datenbank in Ihrem Ziel-Data Warehouse erstellen. Diese Datenbank muss einen Verweis auf die Integrations-ID enthalten. Sie können die Amazon-Redshift-Konsole oder Query Editor v2 verwenden, um die Datenbank zu erstellen.

Anweisungen zum Erstellen einer Zieldatenbank finden Sie unter Erstellen einer Zieldatenbank in Amazon Redshift.

Daten zum der Quelldatenbank hinzufügen

Nachdem Sie Ihre Integration konfiguriert haben, können Sie den der Quell-RDS-Datenbank mit Daten füllen, die Sie in Ihr Data Warehouse replizieren möchten.

> **Note**
>
> Es gibt Unterschiede zwischen den Datentypen in Amazon RDS, und dem Target Analytics Warehouse. Eine Tabelle mit Datentypzuordnungen finden Sie unter the section called “Datentypunterschiede”.

Stellen Sie zunächst mit dem MySQL Ihrer Wahl eine Verbindung zum her. Detaillierte Anweisungen finden Sie unter the section called “Verbinden mit einer DB-Instance, auf der MySQL ausgeführt wird”.

Erstellen Sie dann eine Tabelle und fügen Sie eine Zeile mit Beispieldaten ein.

> **Important**
>
> Stellen Sie sicher, dass die Tabelle über einen Primärschlüssel verfügt. Andernfalls kann sie nicht in das Ziel-Data-Warehouse repliziert werden.

RDS für MySQL

Im folgenden Beispiel wird das MySQL Workbench-Hilfsprogramm verwendet.

```
CREATE DATABASE my_db;

USE my_db;

CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL, Author
 VARCHAR(50) NOT NULL,
Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));

INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977, 'Supernatural
 fiction');
```

RDS für PostgreSQL

Das folgende Beispiel verwendet das interaktive `psql` PostgreSQL-Terminal. Wenn Sie eine Verbindung zur Datenbank herstellen, geben Sie den Datenbanknamen an, den Sie replizieren möchten.

```
psql -h mydatabase.123456789012.us-east-2.rds.amazonaws.com -p 5432 -U username -
d named_db;

named_db=> CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL,
 Author VARCHAR(50) NOT NULL,
Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));

named_db=> INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977,
 'Supernatural fiction');
```

RDS für Oracle

Im folgenden Beispiel wird SQL*Plus verwendet, um eine Verbindung zu Ihrer RDS for Oracle-Datenbank herzustellen.

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))'

SQL> CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL, Author
 VARCHAR(50) NOT NULL,
```

```
Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));

SQL> INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977,
 'Supernatural fiction');
```

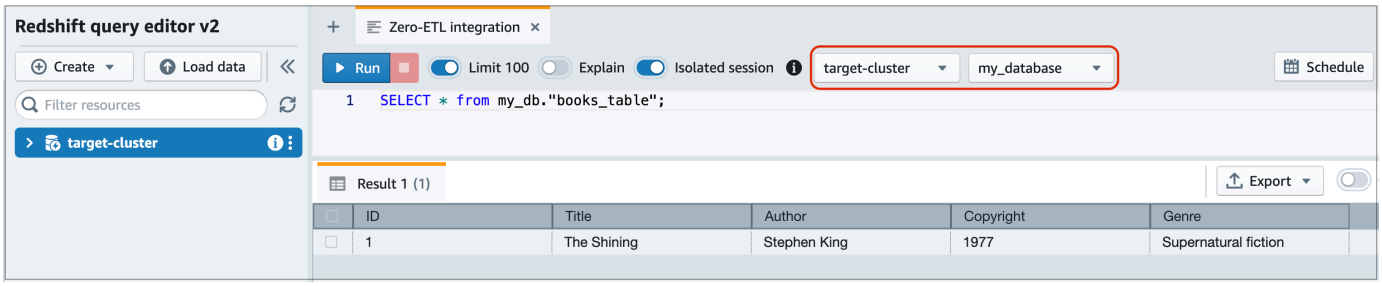
Abfragen Ihrer Amazon RDS in Amazon Redshift

Nachdem Sie dem der RDS-Datenbank Daten hinzugefügt haben, werden sie in die Zieldatenbank repliziert und können abgefragt werden.

So fragen Sie die replizierten Daten ab

1. Navigieren Sie zur Amazon Redshift-Konsole und wählen Sie im linken Navigationsbereich die Option Query Editor v2 aus.
2. Stellen Sie eine Verbindung mit Ihrem Cluster oder Ihrer Arbeitsgruppe her und wählen Sie Ihre aus der Integration erstellte Datenbank im Drop-down-Menü aus (destination_database in diesem Beispiel). Anweisungen zum Erstellen einer Zieldatenbank finden Sie unter Erstellen einer Zieldatenbank in Amazon Redshift.
3. Verwenden Sie eine SELECT-Anweisung, um Ihre Daten abzufragen. In diesem Beispiel können Sie den folgenden Befehl ausführen, um alle Daten aus der Tabelle auszuwählen, die Sie im der Quell-RDS-Datenbank erstellt haben:

   ```
   SELECT * from my_db."books_table";
   ```

   - *my_db* ist der Name des RDS-Datenbankschemas.
   - *books_table* ist der Name der RDS-Tabelle.

Sie können die Daten auch mit einem Befehlszeilenclient abfragen. Zum Beispiel:

```
destination_database=# select * from my_db."books_table";
```

```
 ID |       Title |        Author |   Copyright |                 Genre |  txn_seq |
 txn_id
----+-------------+---------------+-------------+-----------------------+----------
+--------+
  1 | The Shining |  Stephen King |        1977 |   Supernatural fiction |        2 |
 12192
```

> **Note**
>
> Um zwischen Groß- und Kleinschreibung zu unterscheiden, verwenden Sie doppelte Anführungszeichen (" ") für Schema-, Tabellen- und Spaltennamen. Weitere Informationen finden Sie unter enable_case_sensitive_identifier.

Datentypunterschiede zwischen RDS und Amazon Redshift-Datenbanken

Die folgenden Tabellen zeigen die Zuordnungen von RDS für MySQL, RDS für PostgreSQL und RDS für Oracle. Sie zu den entsprechenden Zieldatentypen. Amazon RDS unterstützt derzeit nur diese Datentypen für Zero-ETL-Integrationen.

Wenn eine Tabelle in Ihrem einen nicht unterstützten Datentyp enthält, ist die Tabelle nicht mehr synchron und kann vom Zielziel nicht mehr verwendet werden. Das Streaming von der Quelle zum Ziel wird fortgesetzt, aber die Tabelle mit dem nicht unterstützten Datentyp ist nicht verfügbar. Um die Tabelle zu reparieren und sie im Zielziel verfügbar zu machen, müssen Sie die grundlegende Änderung manuell rückgängig machen und dann die Integration aktualisieren, indem Sie Folgendes ausführen. `ALTER DATABASE...INTEGRATION REFRESH`

> **Note**
>
> Sie können Zero-ETL-Integrationen nicht mit einem Lakehouse aktualisieren. Amazon SageMaker Löschen Sie stattdessen die Integration und versuchen Sie erneut, sie zu erstellen.

Themen

- RDS für MySQL
-

- RDS für Oracle

RDS für MySQL

RDS für MySQL-Datentyp	Zieldatentyp	Beschreibung	Einschränkungen
INT	INTEGER	4-Byte-Ganzzahl mit Vorzeichen	Keine
SMALLINT	SMALLINT	2-Byte-Ganzzahl mit Vorzeichen	Keine
TINYINT	SMALLINT	2-Byte-Ganzzahl mit Vorzeichen	Keine
MEDIUMINT	INTEGER	4-Byte-Ganzzahl mit Vorzeichen	Keine
BIGINT	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
INT UNSIGNED	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
TINYINT UNSIGNED	SMALLINT	2-Byte-Ganzzahl mit Vorzeichen	Keine
MEDIUMINT UNSIGNED	INTEGER	4-Byte-Ganzzahl mit Vorzeichen	Keine
BIGINT UNSIGNED	DECIMAL(20,0)	Genauer Zahlenwert mit wählbarer Genauigkeit	Keine
DEZIMAL (p, s) = NUMERISCH (p, s)	DECIMAL (p,s)	Genauer Zahlenwert	Eine Genauigkeit von mehr als 38 und eine

RDS für MySQL-Datentyp	Zieldatentyp	Beschreibung	Einschränkungen
		mit wählbarer Genauigkeit	Skalierung von mehr als 37 werden nicht unterstützt
DEZIMAL (p, s) OHNE VORZEICHEN = NUMERISCH (p, s) OHNE VORZEICHEN	DECIMAL (p,s)	Genauer Zahlenwert mit wählbarer Genauigkeit	Eine Genauigkeit von mehr als 38 und eine Skalierung von mehr als 37 werden nicht unterstützt
FLOAT4/REAL	REAL	Gleitkommazahl mit einfacher Genauigkeit	Keine
FLOAT4/REAL UNSIGNIERT	REAL	Gleitkommazahl mit einfacher Genauigkeit	Keine
DOUBLE/REAL/FLOAT8	DOUBLE PRECISION	Double (Gleitkommazahl mit doppelter Genauigkeit)	Keine
DOUBLE/REAL/FLOAT8 UNSIGNIERT	DOUBLE PRECISION	Double (Gleitkommazahl mit doppelter Genauigkeit)	Keine
BIT (n)	VARBYTE(8)	Binärwert mit variabler Länge	Keine
BINARY(n)	VARBYTE (n)	Binärwert mit variabler Länge	Keine

RDS für MySQL-Datentyp	Zieldatentyp	Beschreibung	Einschränkungen
VARBINARY (n)	VARBYTE (n)	Binärwert mit variabler Länge	Keine
CHAR(n)	VARCHAR (n)	Zeichenkettenwert mit variabler Länge	Keine
VARCHAR (n)	VARCHAR (n)	Zeichenkettenwert mit variabler Länge	Keine
TEXT	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
TINYTEXT	VARCHAR(255)	Zeichenkettenwert mit variabler Länge bis zu 255 Zeichen	Keine
MEDIUMTEXT	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
LONGTEXT	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine

RDS für MySQL-Datentyp	Zieldatentyp	Beschreibung	Einschränkungen
ENUM	VARCHAR(1020)	Zeichenkettenwert mit variabler Länge bis zu 1.020 Zeichen	Keine
SET	VARCHAR(1020)	Zeichenkettenwert mit variabler Länge bis zu 1.020 Zeichen	Keine
DATUM	DATUM	Kalenderdatum (Jahr, Monat, Tag)	Keine
DATETIME	TIMESTAMP (ZEITSTEMPEL)	Datum und Uhrzeit (ohne Zeitzone)	Keine
TIMESTAMP(p)	TIMESTAMP (ZEITSTEMPEL)	Datum und Uhrzeit (ohne Zeitzone)	Keine
TIME	VARCHAR(18)	Zeichenkettenwert mit variabler Länge bis zu 18 Zeichen	Keine
JAHR	VARCHAR(4)	Zeichenkettenwert mit variabler Länge bis zu 4 Zeichen	Keine

RDS für MySQL-Datentyp	Zieldatentyp	Beschreibung	Einschränkungen
JSON	SUPER	Semistrukturierte Daten oder Dokumente als Werte	Keine

Zero-ETL-Integrationen für RDS for PostgreSQL unterstützen keine benutzerdefinierten Datentypen oder Datentypen, die durch Erweiterungen erstellt wurden.

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
Array	SUPER	Semistrukturierte Daten oder Dokumente als Werte	Keine
bigint	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
große Seriennummer	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
etwas variierend (n)	VARBYTE (n)	Binärwert mit variabler Länge bis zu 16.777.216 Byte	Keine
Bit (n)	VARBYTE (n)	Binärwert mit variabler Länge bis zu 16.777.216 Byte	Keine
Bit, Bit variierend	VARBYTE (16777216)	Binärwert mit variabler Länge	Keine

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
		bis zu 16.777.216 Byte	
boolesch	BOOLEAN	Logischer boolescher Wert (wahr/falsch)	Keine
bytea	VARBYTE (16777216)	Binärwert mit variabler Länge bis zu 16.777.216 Byte	Keine
char (n)	CHAR(n)	Zeichenkettenwert mit fester Länge bis zu 65.535 Byte	Keine
Zeichen variierend (n)	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
cid	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
Apfelwein	VARCHAR (19)	Zeichenkettenwert mit variabler Länge bis zu 19 Zeichen	Keine
date	DATUM	Kalenderdatum (Jahr, Monat, Tag)	Werte über 294.276 n. Chr. werden nicht unterstützt

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
double precision	DOUBLE PRECISION	Gleitkommazahlen mit doppelter Genauigkeit	Subnormale Werte werden nicht vollständig unterstützt
gtsvector	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
inet	VARCHAR (19)	Zeichenkettenwert mit variabler Länge bis zu 19 Zeichen	Keine
Ganzzahl	INTEGER	4-Byte-Ganzzahl mit Vorzeichen	Keine
int2vector	SUPER	Semistrukturierte Daten oder Dokumente als Werte.	Keine
Intervall	INTERVAL	Dauer der Zeit	Es werden nur INTERVAL-Typen unterstützt, die entweder ein Jahr pro Monat oder einen Tag bis Sekunde angeben.

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
json	SUPER	Semistrukturierte Daten oder Dokumente als Werte	Keine
jsonb	SUPER	Semistrukturierte Daten oder Dokumente als Werte	Keine
Json-Pfad	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
macaddr	VARCHAR (17)	Zeichenkettenwert mit variabler Länge bis zu 17 Zeichen	Keine
macaddr8	VARCHAR (23)	Zeichenkettenwert mit variabler Länge bis zu 23 Zeichen	Keine
money	DEZIMAL (20,3)	Betrag in der Währung	Keine

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
Name	VARCHAR (64)	Zeichenkettenwert mit variabler Länge bis zu 64 Zeichen	Keine
numeric(p,s)	DECIMAL (p,s)	Benutzerdefinierter Wert mit fester Genauigkeit	• `NaN`Werte werden nicht unterstützt • Genauigkeit und Maßstab müssen explizit definiert werden und dürfen nicht größer als 38 (Genauigkeit) und 37 (Skala) sein • Negative Skala wird nicht unterstützt
OID	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
Oid-Vektor	SUPER	Semistrukturierte Daten oder Dokumente als Werte.	Keine

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
pg_brin_bloom_summary	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
pg_dependencies	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
pg_lsn	VARCHAR (17)	Zeichenkettenwert mit variabler Länge bis zu 17 Zeichen	Keine
pg_mcv_list	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
pg_ndistinct	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
pg_node_tree	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
pg_snapshot	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
real	REAL	Gleitkommazahl mit einfacher Genauigkeit	Subnormale Werte werden nicht vollständig unterstützt
Vorläufer	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
smallint	SMALLINT	2-Byte-Ganzzahl mit Vorzeichen	Keine
kleine Seriennummer	SMALLINT	2-Byte-Ganzzahl mit Vorzeichen	Keine
serial	INTEGER	4-Byte-Ganzzahl mit Vorzeichen	Keine
text	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
tid	VARCHAR (23)	Zeichenkettenwert mit variabler Länge bis zu 23 Zeichen	Keine
Zeit [(p)] ohne Zeitzone	VARCHAR (19)	Zeichenkettenwert mit variabler Länge bis zu 19 Zeichen	`Infinity`und `-Infinity` Werte werden nicht unterstützt
Zeit [(p)] mit Zeitzone	VARCHAR (22)	Zeichenkettenwert mit variabler Länge bis zu 22 Zeichen	`Infinity`und `-Infinity` Werte werden nicht unterstützt
Zeitstempel [(p)] ohne Zeitzone	TIMESTAMP (ZEITSTEMPEL)	Datum und Uhrzeit (ohne Zeitzone)	• `Infinity`und `-Infinity` Werte werden nicht unterstützt • Werte, die größer als `9999-12-31` nicht unterstützt sind • B.C.-Werte werden nicht unterstützt

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
Zeitstempel [(p)] mit Zeitzone	TIMESTAMPTZ	Datum und Uhrzeit (mit Zeitzone)	• `Infinity`und `-Infinity` Werte werden nicht unterstützt • Werte, die größer als `9999-12-31` nicht unterstützt sind • B.C.-Werte werden nicht unterstützt
tsquery	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
tsvector	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine
txid_snapshot	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine

RDS für PostgreSQL	Amazon-Redshift-Datentyp	Beschreibung	Einschränkungen
uuid	VARCHAR (36)	36 Zeichen lange Zeichenfolge mit variabler Länge	Keine
xid	BIGINT	8-Byte-Ganzzahl mit Vorzeichen	Keine
xid8	DEZIMAL (20, 0)	Dezimalzahl mit fester Genauigkeit	Keine
xml	VARCHAR(65535)	Zeichenkettenwert mit variabler Länge bis zu 65.535 Zeichen	Keine

RDS für Oracle

Datentypen werden nicht unterstützt

Die folgenden RDS for Oracle-Datentypen werden von Amazon Redshift nicht unterstützt:

- `ANYDATA`
- `BFILE`
- `REF`
- `ROWID`
- `UROWID`
- `VARRAY`
- `SDO_GEOMETRY`
- Benutzerdefinierte Datentypen

Unterschiede zwischen den Datentypen

Die folgende Tabelle zeigt die Unterschiede zwischen den Datentypen, die sich auf eine Zero-ETL-Integration auswirken, wenn RDS for Oracle die Quelle und Amazon Redshift das Ziel ist.

Datentyp RDS für Oracle	Amazon-Redshift-Datentyp
BINARY_FLOAT	FLOAT4
BINARY_DOUBLE	FLOAT8
BINARY	VARCHAR (Länge)
FLOAT (P)	Wenn die Genauigkeit =< 24 ist, dann FLOAT4. Wenn die Genauigkeit > 24 ist, dann FLOAT8.
NUMBER (P,S)	Wenn die Skala => 0 und =< 37 ist, dann NUMERIC (p, s). Wenn die Skala => 38 und =< 127 ist, dann VARCHAR (Länge). Wenn die Skala 0 ist: • Und die Genauigkeit ist dann INT2 kleiner oder gleich 2. • Und die Genauigkeit ist dann INT2 größer als 2 und kleiner oder gleich 4. • Und die Genauigkeit ist dann INT4 größer als 4 und kleiner oder gleich 9. • Und die Genauigkeit ist größer als 9, dann NUMERISCH (p, s). • Und die Genauigkeit ist größer oder gleich der Skala, dann NUMERIC (p, s). Wenn die Skala kleiner als 0 ist, dann INT8.
DATUM	Wenn die Skala je nach Redshift-Zielspaltentyp => 0 und =< 6 ist, dann eine der folgenden:

Datentyp RDS für Oracle	Amazon-Redshift-Datentyp
	• TIMESTAMP (s) • TIMESTAMPTZ (s) – Wenn der Quell-Zeitstempel einen Zonen-Offset enthält (wie in SQL Server oder Oracle), wird er beim Einfügen/Aktualisieren in UTC konvertiert. Wenn sie keinen Offset enthält, wird die Zeit bereits in UTC berücksichtigt. Wenn die Skala => 7 und =< 9 ist, dann VARCHAR (37).
INTERVAL_YEAR TO MONTH	Wenn die Länge 1—65.535 ist, dann VARCHAR (Länge in Byte). Wenn die Länge 65.536—2.147.483.647 ist, dann VARCHAR (65535).
INTERVAL_DAY TO SECOND	Wenn die Länge 1—65.535 ist, dann VARCHAR (Länge in Byte). Wenn die Länge 65.536—2.147.483.647 ist, dann VARCHAR (65535).

Datentyp RDS für Oracle	Amazon-Redshift-Datentyp
TIMESTAMP (ZEITSTEMPEL)	Wenn die Skala je nach Redshift-Zielspaltentyp => 0 und =< 6 ist, dann eine der folgenden: • TIMESTAMP (s) • TIMESTAMPTZ (s) – Wenn der Quell-Zeitstempel einen Zonen-Offset enthält (wie in SQL Server oder Oracle), wird er beim Einfügen/Aktualisieren in UTC konvertiert. Wenn sie keinen Offset enthält, wird die Zeit bereits in UTC berücksichtigt. Wenn die Skala => 7 und =< 9 ist, dann VARCHAR (37).
TIMESTAMP WITH TIME ZONE	Wenn die Länge 1—65.535 ist, dann VARCHAR (Länge in Byte). Wenn die Länge 65.536—2.147.483.647 ist, dann VARCHAR (65535).
TIMESTAMP WITH LOCAL TIME ZONE	Wenn die Länge 1—65.535 ist, dann VARCHAR (Länge in Byte). Wenn die Länge 65.536—2.147.483.647 ist, dann VARCHAR (65535).
CHAR	Wenn die Länge 1—65.535 ist, dann VARCHAR (Länge in Byte). Wenn die Länge 65.536—2.147.483.647 ist, dann VARCHAR (65535).

Datentyp RDS für Oracle	Amazon-Redshift-Datentyp
VARCHAR2	Wenn die Länge größer als 4.000 Byte ist, dann VARCHAR (maximale LOB-Größe). Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB. Wenn die Länge 4.000 Byte oder weniger beträgt, dann VARCHAR (Länge in Byte).
NCHAR	Wenn die Länge 1—65.535 ist, dann NVARCHAR (Länge in Byte). Wenn die Länge 65.536—2.147.483.647 ist, dann NVARCHAR (65535).
NVARCHAR2	Wenn die Länge mehr als 4.000 Byte beträgt, dann NVARCHAR (maximale LOB-Größe). Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB. Wenn die Länge 4.000 Byte oder weniger beträgt, dann NVARCHAR (Länge in Byte).
RAW	VARCHAR (Länge)
REAL	FLOAT8
BLOB	VARCHAR (maximale LOB-Größe * 2) Die maximale LOB-Größe darf 31 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB.

Datentyp RDS für Oracle	Amazon-Redshift-Datentyp
CLOB	VARCHAR (maximale LOB-Größe) Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB.
NCLOB	NVARCHAR (maximale LOB-Größe) Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB.
LONG	VARCHAR (maximale LOB-Größe) Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB.
LONG RAW	VARCHAR (maximale LOB-Größe * 2) Die maximale LOB-Größe darf 31 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB.
XMLTYPE	VARCHAR (maximale LOB-Größe) Die maximale LOB-Größe darf 63 KB nicht überschreiten. Amazon Redshift unterstützt keine Größe von VARCHARs mehr als 64 KB.

Amazon Redshift ist von PostgreSQL abgeleitet und teilt daher aufgrund der gemeinsamen PostgreSQL-Architektur mehrere Funktionen mit RDS for PostgreSQL . Zero-ETL-Integrationen nutzen diese Ähnlichkeiten, um die Datenreplikation von RDS für PostgreSQL nach Amazon Redshift zu optimieren, indem sie Datenbanken nach Namen zuordnen und die gemeinsame Datenbank, das Schema und die Tabellenstruktur nutzen.

Beachten Sie bei der Verwaltung von RDS für PostgreSQL Zero-ETL-Integrationen die folgenden Punkte:

- Die Isolierung wird auf Datenbankebene verwaltet.
- Die Replikation erfolgt auf Datenbankebene.
- RDS for PostgreSQL werden Amazon Redshift Redshift-Datenbanken namentlich zugeordnet, wobei Daten in die entsprechende umbenannte Redshift-Datenbank fließen, wenn das Original umbenannt wird.

Trotz ihrer Ähnlichkeiten weisen Amazon Redshift und RDS for PostgreSQL Aurora PostgreSQL Unterschiede auf. In den folgenden Abschnitten werden die Antworten des Amazon Redshift Redshift-Systems für gängige DDL-Operationen beschrieben.

Themen

- Datenbankoperationen
- Schemaoperationen
- Tabellenoperationen

Datenbankoperationen

Die folgende Tabelle zeigt die Systemantworten für Datenbank-DDL-Operationen.

DDL-Vorgang	Reaktion des Redshift-Systems
`CREATE DATABASE`	Keine Operation
`DROP DATABASE`	Amazon Redshift löscht alle Daten in der Redshift-Zieldatenbank.
`RENAME DATABASE`	Amazon Redshift löscht alle Daten in der ursprünglichen Zieldatenbank und synchroni siert die Daten in der neuen Zieldatenbank erneut. Wenn die neue Datenbank nicht existiert, müssen Sie sie manuell erstellen. Anweisungen finden Sie unter Erstellen einer Zieldatenbank in Amazon Redshift.

Schemaoperationen

Die folgende Tabelle zeigt die Systemantworten für Schema-DDL-Operationen.

DDL-Vorgang	Reaktion des Redshift-Systems
`CREATE SCHEMA`	Keine Operation
`DROP SCHEMA`	Amazon Redshift löscht das ursprüngliche Schema.
`RENAME SCHEMA`	Amazon Redshift löscht das ursprüngliche Schema und synchronisiert dann die Daten im neuen Schema erneut.

Tabellenoperationen

Die folgende Tabelle zeigt die Systemantworten für Tabellen-DDL-Operationen.

DDL-Vorgang	Reaktion des Redshift-Systems
`CREATE TABLE`	Amazon Redshift erstellt die Tabelle. Einige Operationen führen dazu, dass die Tabellenerstellung fehlschlägt, z. B. das Erstellen einer Tabelle ohne Primärschlüssel oder das Durchführen einer deklarativen Partitionierung. Weitere Informationen erhalten Sie unter the section called “Einschränkungen” und the section called “Fehlerbehebung bei Null-ETL-Integrationen”.
`DROP TABLE`	Amazon Redshift löscht die Tabelle.
`TRUNCATE TABLE`	Amazon Redshift kürzt die Tabelle.
`ALTER TABLE (RENAME... )`	Amazon Redshift benennt die Tabelle oder Spalte um.

DDL-Vorgang	Reaktion des Redshift-Systems
`ALTER TABLE (SET SCHEMA)`	Amazon Redshift löscht die Tabelle im ursprünglichen Schema und synchronisiert die Tabelle im neuen Schema erneut.
`ALTER TABLE (ADD PRIMARY KEY)`	Amazon Redshift fügt einen Primärschlüssel hinzu und synchronisiert die Tabelle erneut.
`ALTER TABLE (ADD COLUMN)`	Amazon Redshift fügt der Tabelle eine Spalte hinzu.
`ALTER TABLE (DROP COLUMN)`	Amazon Redshift löscht die Spalte, wenn es sich nicht um eine Primärschlüsselspalte handelt. Andernfalls wird die Tabelle erneut synchronisiert.
`ALTER TABLE (SET LOGGED/UNLOGGED )`	Wenn Sie die Tabelle auf protokolliert ändern, synchronisiert Amazon Redshift die Tabelle erneut. Wenn Sie die Tabelle auf unlogged ändern, löscht Amazon Redshift die Tabelle.

Amazon RDS Zero-ETL-Integrationen anzeigen und überwachen

Sie können die Details einer Null-ETL-Integration von Amazon RDS anzeigen, um die zugehörigen Konfigurationsinformationen und den aktuellen Status einzusehen. Sie können außerdem den Status Ihrer Integration überwachen, indem Sie bestimmte Systemansichten in Amazon Redshift abfragen. Darüber hinaus veröffentlicht Amazon Redshift bestimmte integrationsbezogene Metriken auf Amazon CloudWatch, die Sie in der Amazon Redshift Redshift-Konsole einsehen können.

Themen

- Anzeigen von Integrationen
- Überwachen von Integrationen mithilfe von Systemtabellen für Amazon Redshift
- Überwachung von Integrationen mit Amazon EventBridge für Amazon Redshift

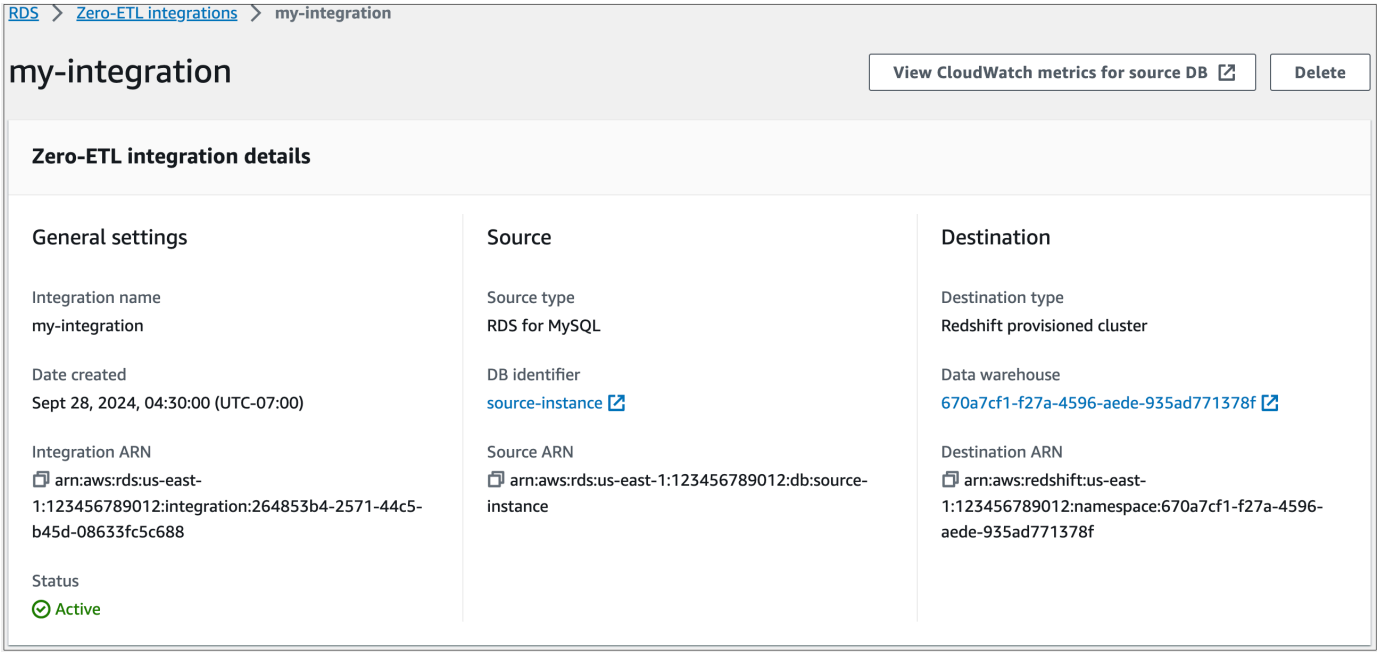
Anzeigen von Integrationen

Sie können Amazon RDS Zero-ETL-Integrationen mithilfe der AWS Management Console, der oder der AWS CLI RDS-API anzeigen.

Konsole

So zeigen Sie die Details einer Null-ETL-Integration an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Null-ETL-Integrationen aus.
3. Wählen Sie eine Integration aus, um weitere Details zu ihr anzuzeigen, z. B. ihren und ihr Ziel-Data Warehouse.



Eine Integration kann folgende Status aufweisen:

- `Creating` – Die Integration wird erstellt.
- `Active` – Die Integration sendet Transaktionsdaten an das Ziel-Data-Warehouse.
- `Syncing` – Bei der Integration ist ein behebbarer Fehler aufgetreten und die Daten werden erneut gesendet. Betroffene Tabellen können erst abgefragt werden, wenn die Neusynchronisierung abgeschlossen ist.

- `Needs attention` – Bei der Integration ist ein Ereignis oder ein Fehler aufgetreten, für dessen Behebung ein manuelles Eingreifen erforderlich ist. Befolgen Sie zur Behebung des Problems die Anweisungen in der Fehlermeldung auf der Seite mit den Integrationsdetails.
- `Failed` – Bei der Integration ist ein nicht behebbares Ereignis oder ein Fehler aufgetreten, der nicht behoben werden kann. Sie müssen die Integration löschen und erneut erstellen.
- `Deleting` – Die Integration wird gelöscht.

AWS CLI

Um alle Zero-ETL-Integrationen im aktuellen Konto mithilfe von anzuzeigen, verwenden Sie den Befehl describe-integrations AWS CLI und geben Sie die Option an. `--integration-identifier`

Example

LinuxmacOSUnixFür, oder:

```
aws rds describe-integrations \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Windows:

```
aws rds describe-integrations ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

RDS-API

Um die Null-ETL-Integration mithilfe der Amazon-RDS-API anzuzeigen, verwenden Sie die Operation `DescribeIntegrations` mit dem Parameter `IntegrationIdentifier`.

Überwachen von Integrationen mithilfe von Systemtabellen für Amazon Redshift

Amazon Redshift verfügt über Systemtabellen und Ansichten, die Informationen zur Funktionsweise des Systems enthalten. Sie können diese Systemtabellen und Ansichten genauso abfragen wie andere Datenbanktabellen. Weitere Informationen zu Systemtabellen und Ansichten in Amazon Redshift finden Sie unter Referenz zu Systemtabellen und Ansichten im Amazon Redshift Database Developer Guide.

Sie können die folgenden Systemansichten und Tabellen abfragen, um Informationen über Ihre Zero-ETL-Integrationen zu erhalten:

- SVV_INTEGRATION – Stellt Konfigurationsdetails für Ihre Integrationen bereit.
- SVV_INTEGRATION_TABLE_STATE – Beschreibt den Status jeder Tabelle innerhalb einer Integration.
- SYS_INTEGRATION_TABLE_STATE_CHANGE – Zeigt Protokolle zur Statusänderung von Tabellen für eine Integration an.
- SYS_INTEGRATION_ACTIVITY – Stellt Informationen zu abgeschlossenen Integrationsausführungen bereit.

Alle integrationsbezogenen CloudWatch Amazon-Metriken stammen von Amazon Redshift. Weitere Informationen finden Sie unter Metriken für Zero-ETL-Integrationen im Amazon Redshift Management Guide. Derzeit veröffentlicht Amazon RDS keine Integrationsmetriken für CloudWatch.

Überwachung von Integrationen mit Amazon EventBridge für Amazon Redshift

Amazon Redshift sendet integrationsbezogene Ereignisse an Amazon. EventBridge Eine Liste der Ereignisse und der entsprechenden Ereignisse finden Sie unter Benachrichtigungen über Zero-ETL-Integrationsereignisse mit Amazon EventBridge im Amazon Redshift Management Guide. IDs

Änderung der Amazon RDS Zero-ETL-Integrationen

Sie können nur den Namen, die Beschreibung und die Datenfilteroptionen für eine Zero-ETL-Integration in einem unterstützten Data Warehouse ändern. Sie können den AWS KMS Schlüssel, der zur Verschlüsselung der Integration verwendet wird, oder die Quell- oder Zieldatenbanken nicht ändern.

Wenn Sie einer vorhandenen Integration einen Datenfilter hinzufügen, bewertet Amazon RDS den Filter neu, als ob er schon immer existiert hätte. Es entfernt alle Daten, die sich derzeit im Ziel-Data Warehouse befinden und nicht den neuen Filterkriterien entsprechen. Wenn Sie einen Datenfilter aus einer Integration entfernen, repliziert er alle Daten, die zuvor nicht den Filterkriterien entsprachen (dies jetzt aber tun), in das Ziel-Data Warehouse. Weitere Informationen finden Sie unter the section called “Datenfilterung für Zero-ETL-Integrationen”.

Sie können eine Zero-ETL-Integration mithilfe der AWS Management Console, der oder der AWS CLI Amazon RDS-API ändern.

RDS-Konsole

Um eine Zero-ETL-Integration zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Zero-ETL-Integrationen und dann die Integration aus, die Sie ändern möchten.
3. Wählen Sie Ändern und nehmen Sie Änderungen an allen verfügbaren Einstellungen vor.
4. Wenn alle Änderungen Ihren Wünschen entsprechen, wählen Sie Ändern.

AWS CLI

Um eine Zero-ETL-Integration mit dem zu ändern AWS CLI, rufen Sie den Befehl modify-integration auf. Geben Sie zusammen mit dem eine der `--integration-identifier` folgenden Optionen an:

- `--integration-name`— Geben Sie einen neuen Namen für die Integration an.
- `--description`— Geben Sie eine neue Beschreibung für die Integration an.
- `--data-filter`— Geben Sie Datenfilteroptionen für die Integration an. Weitere Informationen finden Sie unter the section called “Datenfilterung für Zero-ETL-Integrationen”.

Example

Die folgende Anfrage ändert eine bestehende Integration.

Für LinuxmacOS, oderUnix:

```
aws rds modify-integration \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374 \
    --integration-name my-renamed-integration
```

Windows:

```
aws rds modify-integration ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374 ^
```

```
    --integration-name my-renamed-integration
```

RDS-API

Um eine Zero-ETL-Integration mithilfe der RDS-API zu ändern, rufen Sie den Vorgang auf ModifyIntegration. Geben Sie die Integrations-ID und die Parameter an, die Sie ändern möchten.

Löschen von Amazon RDS Zero-ETL-Integrationen

Wenn Sie eine Zero-ETL-Integration löschen, entfernt Amazon RDS Amazon sie aus dem der Quelldatenbank. Ihre Transaktionsdaten werden nicht aus Amazon RDS Amazon oder dem Analyseziel gelöscht, aber Amazon RDS sendet keine neuen Daten an Amazon Redshift oder. Amazon SageMaker

Sie können eine Integration nur löschen, wenn sie den Status`Active`,`Failed`, `Syncing` oder hat. `Needs attention`

Sie können Zero-ETL-Integrationen mithilfe der AWS Management Console, der oder der AWS CLI RDS-API löschen.

Konsole

So löschen Sie eine Null-ETL-Integration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im linken Navigationsbereich Null-ETL-Integrationen aus.
3. Wählen Sie die Null-ETL-Integration aus, die Sie löschen möchten.
4. Klicken Sie auf Aktionen, Löschen und bestätigen Sie den Löschvorgang.

AWS CLI

Verwenden Sie zum Löschen einer Null-ETL-Integration den Befehl delete-integration und geben Sie die Option `--integration-identifier` an.

Example

Für LinuxmacOS, oderUnix:

```
aws rds delete-integration \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Windows:

```
aws rds delete-integration ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

RDS-API

Verwenden Sie zum Löschen einer Null-ETL-Integration mithilfe der Amazon-RDS-API die Operation `DeleteIntegration` mit dem Parameter `IntegrationIdentifier`.

Fehlerbehebung bei Amazon RDS Zero-ETL-Integrationen

Sie können den Status einer Zero-ETL-Integration überprüfen, indem Sie die SVV_INTEGRATION-Systemtabelle im Analyseziel abfragen. Wenn die Spalte `state` den Wert `ErrorState` aufweist, bedeutet das, dass ein Fehler vorliegt. Weitere Informationen finden Sie unter the section called “Überwachen mithilfe von Systemtabellen”.

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme mit Amazon RDS Zero-ETL-Integrationen zu beheben.

> **Important**
>
> Neusynchronisierungs- und Aktualisierungsvorgänge sind für Zero-ETL-Integrationen mit einem Lakehouse nicht verfügbar. Amazon SageMaker AI Wenn es Probleme mit einer Integration gibt, müssen Sie die Integration löschen und eine neue Integration erstellen. Sie können eine bestehende Integration nicht aktualisieren oder erneut synchronisieren.

Themen

- Ich kann keine Null-ETL-Integration erstellen
- Meine Integration steckt in einem Zustand von Syncing
- Meine Tabellen werden nicht auf Amazon Redshift repliziert
- Eine oder mehrere meiner Amazon-Redshift-Tabellen erfordern eine erneute Synchronisation
- Fehler bei der Integration, Probleme bei Lakehouse Zero-ETL-Integrationen Amazon SageMaker AI

Ich kann keine Null-ETL-Integration erstellen

Wenn Sie keine Zero-ETL-Integration erstellen können, stellen Sie sicher, dass die folgenden Angaben für Ihre Quelldatenbank korrekt sind:

- In Ihrer Quelldatenbank muss eine unterstützte DB-Engine-Version ausgeführt werden. Eine Liste der unterstützten Versionen finden Sie unter the section called “Null-ETL-Integrationen ”.
- Sie haben die DB-Parameter korrekt konfiguriert. Wenn die erforderlichen Parameter falsch gesetzt oder nicht mit der Datenbank verknüpft sind, schlägt die Erstellung fehl. Siehe the section called “Schritt 1: Erstellen einer benutzerdefinierten DB--Parametergruppe”.

Stellen Sie außerdem Folgendes für Ihr Ziel-Data-Warehouse sicher:

- Die Unterscheidung zwischen Groß- und Kleinschreibung ist aktiviert. Siehe Aktivieren der Unterscheidung zwischen Groß- und Kleinschreibung für Ihr Data Warehouse.
- Sie haben den richtigen autorisierten Prinzipal und die richtige Integrationsquelle hinzugefügt. Weitere Informationen finden Sie unter Autorisierung für Ihr Amazon Redshift Data Warehouse konfigurieren.
- Das Data Warehouse ist verschlüsselt (wenn es sich um einen bereitgestellten Cluster handelt). Siehe Amazon Redshift Redshift-Datenbankverschlüsselung.

Meine Integration steckt in einem Zustand von **`Syncing`**

Ihre Integration zeigt möglicherweise durchgängig den Status an, `Syncing` wenn Sie den Wert eines der erforderlichen DB-Parameter ändern.

Um dieses Problem zu beheben, überprüfen Sie die Werte der Parameter in der Parametergruppe, die dem zugeordnet ist, und stellen Sie sicher, dass sie den erforderlichen Werten entsprechen. Weitere Informationen finden Sie unter the section called “Schritt 1: Erstellen einer benutzerdefinierten DB--Parametergruppe”.

Wenn Sie Parameter ändern, stellen Sie sicher, dass Sie den neu starten, um die Änderungen zu übernehmen.

Meine Tabellen werden nicht auf Amazon Redshift repliziert

Wenn Sie eine oder mehrere Tabellen nicht in Amazon Redshift sehen, können Sie den folgenden Befehl ausführen, um sie erneut zu synchronisieren:

```
ALTER DATABASE dbname INTEGRATION REFRESH TABLES table1, table2;
```

Weitere Informationen finden Sie unter ALTER DATABASE in der Amazon Redshift SQL-Referenz.

Ihre Daten werden möglicherweise nicht repliziert, weil eine oder mehrere Ihrer Quelltabellen keinen Primärschlüssel haben. Das Monitoring-Dashboard in Amazon Redshift zeigt den Status dieser Tabellen als an`Failed`, und der Status der gesamten Zero-ETL-Integration ändert sich auf. `Needs attention` Um dieses Problem zu lösen, können Sie einen vorhandenen Schlüssel in Ihrer Tabelle identifizieren, der zu einem Primärschlüssel werden kann, oder Sie können einen synthetischen Primärschlüssel hinzufügen. Ausführliche Lösungen finden Sie unter Behandeln von Tabellen ohne Primärschlüssel bei der Erstellung von Amazon Aurora MySQL- oder Amazon RDS for MySQL Zero-ETL-Integrationen mit Amazon Redshift.

Eine oder mehrere meiner Amazon-Redshift-Tabellen erfordern eine erneute Synchronisation

Wenn Sie bestimmte Befehle in Ihrer Quelldatenbank ausführen, müssen Ihre Tabellen möglicherweise erneut synchronisiert werden. In diesen Fällen zeigt die Systemansicht SVV_INTEGRATION_TABLE_STATE für `table_state` den Wert `ResyncRequired` an. Dies bedeutet, dass die Integration die Daten für diese spezifische Tabelle vollständig neu von MySQL in Amazon Redshift laden muss.

Wenn die Tabelle mit der erneuten Synchronisation beginnt, wechselt sie in den Status `Syncing`. Sie müssen keine manuellen Maßnahmen ergreifen, um eine Tabelle erneut zu synchronisieren. Während die Tabellendaten erneut synchronisiert werden, können Sie in Amazon Redshift nicht darauf zugreifen.

Im Folgenden finden Sie einige Beispieloperationen, mit denen eine Tabelle in den Status `ResyncRequired` versetzt werden kann, sowie mögliche Alternativen, die Sie in Betracht ziehen sollten.

Operation	Beispiel	Alternative
Hinzufügen einer Spalte an einer bestimmten Position	`ALTER TABLE table_name ADD COLUMN column_name INTEGER NOT NULL first;`	Amazon Redshift unterstützt nicht das Hinzufügen

Operation	Beispiel	Alternative
		von Spalten an bestimmte n Positione n mithilfe der Schlüssel wörter `first` oder `af` Wenn die Reihenfolge der Spalten in der Zieltabelle nicht wichtig ist, fügen Sie die Spalte mit einem einfacheren Befehl am Ende der Tabelle hinzu: `ALTER TABLE table_name ADD COLUMN column_name column_type ;`

Operation	Beispiel	Alternative

Hinzufügen einer Zeitstempelspalte mit dem Standardwert CURRENT_TIMESTAMP	`ALTER TABLE table_name`
	`  ADD COLUMN column_name  TIMESTAMP`
	`  NOT NULL DEFAULT CURRENT_TIMESTAMP;`
		Der CURRENT_TIMESTAMP Wert für bestehende Tabellenzeilen wird von RDS für MySQL MySQL berechnet und kann in Amazon Redshift ohne vollständige Resynchronisierung der Tabellendaten nicht simuliert werden.
		
		Wenn möglich, ändern Sie den Standardwert in eine Literalkonstante wie 2023-01-01 00:00:15, um Latenzen bei der

Operation	Beispiel	Alternative
		Tabellenv erfügbarkeit zu vermeiden.
Ausführen mehrerer Spaltenop erationen innerhalb eines einzigen Befehls	`ALTER TABLE table_name ADD COLUMN column_1, RENAME COLUMN column_2 TO column_3;`	Überlegen Sie, ob Sie den Befehl in zwei separate Operation en, `ADD` und `RENAME`, aufteilen sollten, die keine erneute Synchroni sation erfordern.

Fehler bei der Integration, Probleme bei Lakehouse Zero-ETL-Integrationen Amazon SageMaker AI

Wenn Sie Probleme mit einer bestehenden Zero-ETL-Integration mit einem Amazon SageMaker AI Lakehouse haben, besteht die einzige Lösung darin, die Integration zu löschen und eine neue zu erstellen. Im Gegensatz zu anderen AWS Diensten unterstützen Zero-ETL-Integrationen keine Aktualisierungs- oder Resynchronisierungsvorgänge.

So lösen Sie Integrationsprobleme:

1. Löschen Sie die problematische Zero-ETL-Integration mithilfe der Konsole, CLI oder API.
2. Stellen Sie sicher, dass die Konfigurationen der Quelldatenbank und des Ziel-Data Warehouses korrekt sind.
3. Erstellen Sie eine neue Zero-ETL-Integration mit derselben oder einer aktualisierten Konfiguration.

Dieser Prozess führt zu einer vollständigen Neuinitialisierung der Datenpipeline, was je nach Größe Ihrer Quelldatenbank einige Zeit in Anspruch nehmen kann.

Amazon RDS für Db2

Amazon RDS unterstützt DB-Instances, auf denen die folgenden Editionen von ausgeführt werdenIBM Db2:

- Db2 Standard Edition
- Db2 Advanced Edition

Amazon RDS unterstützt DB-Instances, auf denen die folgenden Versionen von Db2 ausgeführt werden:

- Db2 11.5

Weitere Informationen über den Support für Minor-Versionen finden Sie unter Versionen von Db2 auf Amazon RDS.

Bevor Sie eine DB-Instance erstellen, führen Sie die Schritte im Einrichtung Ihrer Amazon RDS-Umgebung Abschnitt dieses Benutzerhandbuchs durch. Wenn Sie eine DB-Instance mit Ihrem Master-Benutzer erstellen, erhält `DBADM` der Benutzer mit einigen Einschränkungen die erforderlichen Rechte. Verwenden Sie diesen Benutzer für administrative Aufgaben wie das Erstellen zusätzlicher Datenbankkonten. Sie können weder Berechtigungen auf `SYSMAINT` Instanzebene noch auf `SECADM` Datenbankebene verwendenSYSADM. `SYSCTRL`

Sie können das folgende erstellen:

- DB-Instances
- DB-Snapshots
- Point-in-time stellt wieder her
- Automatisierte Speicher-Backups
- Manuelle Speicher-Backups

Sie können DB-Instances verwenden, auf denen Db2 in einer Virtual Private Cloud (VPC) ausgeführt wird. Sie können Ihrer Amazon RDS for Db2-DB-Instance auch Funktionen hinzufügen, indem Sie verschiedene Optionen aktivieren. Amazon RDS unterstützt Multi-AZ-Bereitstellungen für RDS for Db2 als Failover-Lösung mit hoher Verfügbarkeit.

> **Important**
>
> Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Shell-Zugriff auf DB-Instances. Es schränkt auch den Zugriff auf bestimmte Systemprozeduren und Tabellen ein, für die erhöhte Rechte erforderlich sind. Sie können mit Standard-SQL-Clients wie IBM Db2 CLP auf Ihre Datenbank zugreifen. Sie können jedoch nicht direkt auf den Host zugreifen, indem Sie Telnet oder Secure Shell (SSH) verwenden.

Themen

- Überblick über Db2 auf Amazon RDS
- Voraussetzungen für die Erstellung einer Amazon RDS for Db2-DB-Instance
- Mehrere Datenbanken auf einer Amazon RDS for Db2-DB-Instance
- Verbindung zu Ihrer Db2-DB-Instance herstellen
- Sicherung von Amazon RDS für Db2-DB-Instance-Verbindungen
- Verwaltung Ihrer Amazon RDS for Db2-DB-Instance
- Integrieren einer Amazon RDS for Db2-DB-Instance mit Amazon S3
- Migrieren von Daten zu Amazon RDS for Db2
- Amazon RDS für den Db2-Verbund
- Arbeiten mit Replikaten für Amazon RDS for Db2
- Optionen für Amazon RDS für Db2-DB-Instances
- Externe gespeicherte Prozeduren für Amazon RDS for Db2
- Bekannte Probleme und Einschränkungen für Amazon RDS for Db2
- Referenz für gespeicherte Prozeduren in Amazon RDS für Db2
- Referenz RDS für benutzerdefinierte Funktionen von Amazon for Db2
- Fehlerbehebung für Amazon RDS for Db2

Überblick über Db2 auf Amazon RDS

Sie können die folgenden Abschnitte lesen, um sich einen Überblick über Db2 auf Amazon RDS zu verschaffen.

Themen

- Funktionen von Amazon RDS für Db2
- Versionen von Db2 auf Amazon RDS
- Lizenzierungsoptionen für Amazon RDS für Db2
- Amazon RDS für Db2-Instance-Klassen
- Standardrollen von Amazon RDS für Db2
- Amazon RDS für Db2-Parameter
- EBCDICSortierung für Db2-Datenbanken auf Amazon RDS
- Lokale Zeitzone für Amazon RDS für Db2-DB-Instances

Funktionen von Amazon RDS für Db2

Amazon RDS for Db2 unterstützt die meisten Funktionen und Fähigkeiten der IBM Db2 Datenbank. Einige Funktionen werden möglicherweise nur begrenzt unterstützt oder haben eingeschränkte Berechtigungen. Weitere Informationen zu den Db2-Datenbankfunktionen für bestimmte Db2-Versionen finden Sie in der Dokumentation. IBM Db2

Sie können neue Amazon RDS Funktionen auf der Was ist neu mit Datenbank?-Seite filtern. Wählen Sie für Produkte Amazon RDS aus. Anschließend können Sie anhand von Schlüsselwörtern wie suchen. **`Db2 2023`**

> **Note**
>
> Die folgenden Listen erheben keinen Anspruch auf Vollständigkeit.

Themen

- Unterstützte Funktionen in RDS für Db2
- Nicht unterstützte Funktionen in RDS für Db2

Unterstützte Funktionen in RDS für Db2

RDS for Db2 unterstützt Funktionen, zu denen Funktionen gehören, die in Amazon RDS integriert sind, IBM Db2 und Funktionen, die zu den Kernfunktionen von Amazon RDS gehören.

Funktionen, die systemeigenen Funktionen von IBM Db2

RDS for Db2 unterstützt die folgenden Db2-Datenbankfunktionen:

- Erstellung einer Standarddatenbank, die einen vom Kunden definierten Codesatz, die Sortierung, die Seitengröße und das Gebiet verwendet. Verwenden Sie die rdsadmin.create_database gespeicherte Amazon RDS-Prozedur.
- Hinzufügen, Löschen oder Ändern von lokalen Benutzern und Gruppen. Verwenden Sie die gespeicherten Amazon RDS-Prozeduren fürGespeicherte Prozeduren zum Erteilen und Widerrufen von Rechten für RDS for Db2.
- Erstellung von Rollen mit der rdsadmin.create_role gespeicherten Amazon RDS-Prozedur.
- Support für standardmäßige zeilenorganisierte Tabellen.
- Support der analytischen Arbeitslast für spaltenorganisierte Tabellen.
- Fähigkeit, DB2-Kompatibilitätsmerkmale wie und zu definieren. Oracle MySQL
- Support für Java basierte externe gespeicherte Prozeduren.
- Support für Datenverschlüsselung bei der Übertragung mithilfe von SSL/TLS.
- Überwachung des Status einer Datenbank (`ALIVE`,, `DOWN` `STORAGE_FULLUNKNOWN`, und`STANDBY_CONNECTABLE`).
- Wiederherstellung einer vom Kunden bereitgestellten Offline- oder Linux (LE) Online-Datenbank. Verwenden Sie gespeicherte Amazon RDS-Prozeduren fürGespeicherte Prozeduren für Datenbanken für RDS for Db2.
- Verwendung von vom Kunden bereitgestellten Db2-Archivprotokollen, um die Datenbank mit selbstverwalteten Db2-Datenbanken zu synchronisieren. Verwenden Sie gespeicherte Amazon RDS-Prozeduren fürGespeicherte Prozeduren für Datenbanken für RDS for Db2.
- Support für Prüfungen auf Db2-Instanz- und Datenbankebene.
- Support für eine homogene Föderation.
- Möglichkeit, eine Tabelle aus Datendateien in Amazon Simple Storage Service (Amazon S3) zu laden.
- Benutzern, Gruppen oder Rollen erteilte Berechtigungen, wie`CONNECT`,,`SYSMON`, `ACCESSCTRL`,`DATAACCESS`,`SQLADM`,`WLMADM`, `EXPLAIN`,`LOAD`, oder`IMPLICIT_SCHEMA`.
- Erstellung mehrerer Datenbanken.

> **Note**
>
> Eine RDS for Db2-DB-Instance kann bis zu 50 Datenbanken enthalten. Weitere Informationen finden Sie unter the section called “Mehrere Db2-Datenbanken”.

Kernfunktionen von Amazon RDS

RDS for Db2 unterstützt die folgenden Kernfunktionen von Amazon RDS:

- Benutzerdefinierte Parametergruppen, die DB-Instances zugewiesen werden sollen
- Erstellung, Änderung und Löschung von DB-Instances
- Wiederherstellung einer selbstverwalteten Db2-Offline- oder Linux (LE) Online-Datenbanksicherung

> **Note**
>
> Um Ihr Backup wiederherstellen zu können, geben Sie beim Erstellen einer DB-Instance keinen Namen für Ihre Datenbank an. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

- Support der Speichertypen gp3, io2 und io1
- Verwendung von AWS Managed Microsoft AD für die Kerberos Authentifizierung und LDAP-Gruppenautorisierung für RDS for Db2
- Änderung von Sicherheitsgruppen, Ports, Instance-Typen, Speicher, Aufbewahrungsfristen für Backups und anderen Einstellungen für bestehende Db2-Instances
- Löschschutz für DB-Instances
- Regionsübergreifende point-in-time Wiederherstellung (PITR), auch für verschlüsselte Backups
- Verwendung von AWS Key Management Service (AWS KMS) für Speicherverschlüsselung und Verschlüsselung im Ruhezustand
- Multi-AZ-DB-Instances mit einem Standby-Modus für hohe Verfügbarkeit
- Neustarts von DB-Instances
- Aktualisierungen der Master-Passwörter
- Wiederherstellung von DB-Instances zu einem bestimmten Zeitpunkt
- Backup und Wiederherstellung von DB-Instances mithilfe von Backups auf Speicherebene

- Starten und Stoppen von DB-Instances
- Wartung von DB-Instances

Nicht unterstützte Funktionen in RDS für Db2

RDS for Db2 unterstützt die folgenden Db2-Datenbankfunktionen nicht:

- `SYSADMSECADM`, und `SYSMAINT` Zugriff für den Masterbenutzer.
- In C, C++ oder Cobol geschriebene externe gespeicherte Prozeduren.
- Mehrere Db2-DB-Instances auf einem einzigen Host.
- Externe GSS-API-Plugins zur Authentifizierung.
- Externe Plugins von Drittanbietern zum Sichern oder Wiederherstellen von Db2-Datenbanken.
- Massiv-Parallelverarbeitung (MPP) mit mehreren Knoten, wie z. IBM Db2 Warehouse
- IBM Db2 pureScale.
- Manuelles Einrichten von High Availability Disaster Recovery (HADR) für RDS für Db2.

> **Note**
>
> Amazon RDS unterstützt und verwaltet HADR für RDS for Db2 über Replikate. Weitere Informationen finden Sie unter Arbeiten mit Replikaten für Amazon RDS for Db2.
> RDS for Db2 unterstützt Multi-AZ-Bereitstellungen, regionsübergreifende automatische Backups und Replikation. Weitere Informationen erhalten Sie unter Multi-AZ-DB-Instance-Bereitstellungen für Amazon RDS und Automatisierte Backups auf ein anderes replizieren AWS-Region.

- Native Datenbankverschlüsselung.
- Heterogener Verbund mit Informix, Sybase und Teradata. Weitere Informationen finden Sie unter the section called “Verbund”.
- Erstellung von Routinen ohne Fencing und Migration vorhandener Routinen ohne Fencing durch Sicherung und Wiederherstellung von Daten. Weitere Informationen finden Sie unter Routinen ohne Umzäunung.
- Erstellung neuer nichtautomatischer Speicher-Tablespaces. Weitere Informationen finden Sie unter Nichtautomatische Speicher-Tablespaces während der Migration.
- Externe Tabellen.

Versionen von Db2 auf Amazon RDS

Für Db2 haben die Versionsnummern die Form major.minor.build.revision, zum Beispiel 11.5.9.0.sb00000000.r1. Unsere Versionsimplementierung entspricht der von Db2.

Major

Die Hauptversionsnummer ist sowohl die Ganzzahl als auch der erste Bruchteil der Versionsnummer, zum Beispiel 11.5. Eine Versionsänderung gilt als schwerwiegend, wenn sich die Hauptversionsnummer ändert, z. B. wenn von Version 11.5 auf 12.1 umgestellt wird.

geringfügig

Die Nebenversionsnummer ist sowohl der dritte als auch der vierte Teil der Versionsnummer, z. B. 9.0 in 11.5.9.0. Der dritte Teil gibt das Db2-Modpack an, zum Beispiel 9 in 9.0. Der vierte Teil gibt das Db2-Fixpack an, zum Beispiel 0 in 9.0. Eine Versionsänderung wird als geringfügig angesehen, wenn sich entweder das Db2-Modpack oder das Db2-Fixpack ändert, z. B. wenn von Version 11.5.9.0 auf 11.5.9.1 oder von 11.5.9.0 auf 11.5.10.0 umgestellt wird, mit Ausnahmen bei der Bereitstellung von Katalogtabellen-Updates. (Amazon RDS kümmert sich um diese Ausnahmen.)

bauen

Die Buildnummer ist der fünfte Teil der Versionsnummer, zum Beispiel sb00000000 in 11.5.9.0.sb00000000. Eine Buildnummer, bei der die Zahl nur aus Nullen besteht, weist auf einen Standard-Build hin. Eine Buildnummer, bei der die Zahl nicht nur aus Nullen besteht, weist auf einen speziellen Build hin. Eine Buildnummer ändert sich, wenn es einen Sicherheitspatch oder einen speziellen Build einer vorhandenen Db2-Version gibt. Eine Änderung der Build-Nummer weist auch darauf hin, dass Amazon RDS automatisch eine neue Nebenversion angewendet hat.

Änderung

Die Revisionsnummer ist der sechste Teil der Versionsnummer, zum Beispiel r1 in 11.5.9.0.sb00000000.r1. Eine Revision ist eine Amazon RDS-Revision einer bestehenden Db2-Version. Eine Änderung der Revisionsnummer bedeutet, dass Amazon RDS automatisch eine neue Nebenversion angewendet hat.

Themen

- Unterstützte Db2-Nebenversionen auf Amazon RDS
- Unterstützte Db2-Hauptversionen auf Amazon RDS

Unterstützte Db2-Nebenversionen auf Amazon RDS

Die folgende Tabelle zeigt die Nebenversionen von Db2 11.5, die Amazon RDS derzeit unterstützt.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.

Version der Db2-Engine	IBMDatum der Veröffentlichung	Datum der Veröffentlichung von RDS	
11.5.9.0	15. November 2023	27. November 2023	

Sie können jede derzeit unterstützte Db2-Version angeben, wenn Sie eine neue DB-Instance erstellen. Sie können die Hauptversion (z. B. Db2 11.5) und jede unterstützte Nebenversion für die angegebene Hauptversion angeben. Wenn keine Version angegeben wird, verwendet Amazon RDS standardmäßig eine unterstützte Version - in der Regel die aktuelle Version. Wenn die Hauptversion, jedoch nicht die Unterversion, festgelegt ist, verwendet Amazon RDS standardmäßig den letzten Release der Hauptversion, die Sie festgelegt haben. Verwenden Sie den Befehl describe-db-engine-versions AWS Command Line Interface (AWS CLI), um eine Liste der unterstützten Versionen sowie die Standardeinstellungen für neu erstellte DB-Instances anzuzeigen.

Um beispielsweise die unterstützten Engine-Versionen für Amazon RDS for Db2 aufzulisten, führen Sie den folgenden AWS CLI Befehl aus. Ersetzen Sie *region* durch Ihre AWS-Region.

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-engine-versions \
    --filters Name=engine,Values=db2-ae,db2-se \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region region
```

Windows:

```
aws rds describe-db-engine-versions ^
```

```
    --filters Name=engine,Values=db2-ae,db2-se ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
    --region region
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    },
    {
    "Engine": "db2-se",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-se-11.5"
    }
]
```

Die standardmäßige Db2-Version kann je nach AWS-Region variieren. Um eine DB-Instance mit einer bestimmten Unterversion zu erstellen, geben Sie die Unterversion bei der Erstellung der DB-Instance an. Sie können die Standardversion AWS-Region für `db2-ae` und `db2-se` Datenbank-Engines ermitteln, indem Sie den `describe-db-engine-versions` Befehl ausführen. Das folgende Beispiel gibt die Standardversion für `db2-ae` in US East (Nord-Virginia) zurück.

Für LinuxmacOS, oderUnix:

```
aws rds describe-db-engine-versions \
    --default-only --engine db2-ae \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region us-east-1
```

Windows:

```
aws rds describe-db-engine-versions ^
    --default-only --engine db2-ae ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
    --region us-east-1
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    }
]
```

Mit Amazon RDS haben Sie die Kontrolle darüber, wann Sie Ihre Db2-Instance auf eine neue Hauptversion aktualisieren müssen, die von Amazon RDS unterstützt wird. Sie können die Kompatibilität mit bestimmten Db2-Versionen aufrechterhalten, neue Versionen mit Ihrer Anwendung testen, bevor Sie sie in der Produktion einsetzen, und Hauptversions-Upgrades zu Zeiten durchführen, die Ihrem Zeitplan am besten entsprechen.

Wenn das automatische Upgrade von Nebenversionen aktiviert ist, aktualisiert Amazon RDS Ihre DB-Instances automatisch auf neue Db2-Nebenversionen, da diese von Amazon RDS unterstützt werden. Dieser Patch tritt während Ihres geplanten Wartungsfensters auf. Sie können eine DB-Instance ändern, um automatische Upgrades der Nebenversion zu aktivieren oder zu deaktivieren.

Mit Ausnahme der Db2-Versionen 11.5.9.1 und 11.5.10.0 beinhalten automatische Upgrades auf neue Db2-Nebenversionen automatische Upgrades auf neue Builds und Revisionen. Führen Sie für 11.5.9.1 und 11.5.10.0 ein manuelles Upgrade der Nebenversionen durch.

Wenn Sie sich von automatisch geplanten Upgrades abmelden, können Sie ein manuelles Upgrade auf eine der unterstützten Unterversionen durchführen, indem Sie die selben Schritte befolgen, wie bei einem Update auf eine Hauptversion. Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Unterstützte Db2-Hauptversionen auf Amazon RDS

RDS für Db2-Hauptversionen sind im Rahmen der Standardunterstützung mindestens bis zum IBM Ende des Supports (Basis) für die entsprechende IBM Version verfügbar. In der folgenden Tabelle sind die Termine aufgeführt, an denen Sie Ihre Test- und Upgrade-Zyklen planen können. Falls Amazon den Support für eine RDS for Db2-Version länger als ursprünglich angegeben verlängert, planen wir, diese Tabelle zu aktualisieren, um das spätere Datum widerzuspiegeln.

Sie können die folgenden Daten verwenden, um Ihre Test- und Upgrade-Zyklen zu planen.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.
> Sie können die Hauptversionen Ihrer Db2-Datenbanken anzeigen, indem Sie den AWS CLI Befehl describe-db-major-engine-versions ausführen oder den API-Vorgang Describe DBMajor EngineVersions RDS verwenden.

Db2-Hauptversion	IBMDatum der Veröffentlichung	Datum der Veröffentlichung von RDS	IBMEnde des Supports (Standard und Advanced Edition)	IBMEnde des Supports (verlängert)	
Db2 11.5	27. Juni 2019	27. November 2023	Noch nicht angekündigt vonIBM. [1]	4 Jahre nach Ende des Supports	

[1] Die neuesten Informationen finden Sie in der IBM Db2 Dokumentation unter end-of-supportDb2-Produktdaten.

Lizenzierungsoptionen für Amazon RDS für Db2

Amazon RDS for Db2 bietet zwei Lizenzierungsoptionen: Bring Your Own License (BYOL) und Db2-Lizenz durch. AWS Marketplace

Themen

- Bringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit
- Db2-Lizenz über AWS Marketplace
- Zwischen Db2-Lizenzen wechseln

Bringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit

Im BYOL-Modell verwenden Sie Ihre vorhandenen Db2-Datenbanklizenzen, um Datenbanken auf Amazon RDS bereitzustellen. Stellen Sie sicher, dass Sie über die entsprechende Db2-Datenbanklizenz für die DB-Instance-Klasse und die Db2-Datenbank-Edition verfügen, die Sie ausführen möchten. Sie müssen auch IBM die Richtlinien für die Lizenzierung von IBM Datenbanksoftware in der Cloud-Computing-Umgebung befolgen.

> **Note**
>
> Multi-AZ-DB-Instances sind Cold-Standbys, da die Db2-Datenbank installiert ist, aber nicht läuft. Standbys sind nicht lesbar, laufen nicht und bearbeiten keine Anfragen. Weitere Informationen finden Sie unter IBM Db2Lizenzinformationen auf der IBM-Website.

Bei diesem Modell verwenden Sie weiterhin Ihr aktives IBM Support-Konto und wenden sich bei Serviceanfragen für Db2-Datenbanken IBM direkt an uns. Wenn Sie ein Support Konto beim Fallsupport haben, können Sie sich Support bei Problemen mit Amazon RDS an uns wenden. Amazon Web Services und IBM verfügen über einen herstellerübergreifenden Support-Prozess für Fälle, in denen die Unterstützung beider Organisationen erforderlich ist.

Amazon RDS unterstützt das BYOL-Modell für Db2 Standard Edition undDb2 Advanced Edition.

Themen

- IBM IDs für Bring Your Own License (BYOL) für Db2
- Hinzufügen IBM IDs zu einer Parametergruppe für RDS für Db2-DB-Instances
- Integrieren mit AWS License Manager

IBM IDs für Bring Your Own License (BYOL) für Db2

Im BYOL-Modell müssen Sie IBM Customer ID und Sie IBM Site ID RDS für Db2-DB-Instances erstellen, ändern oder wiederherstellen. Sie müssen eine benutzerdefinierte Parametergruppe mit Ihren IBM Customer ID und Ihrer erstellen, IBM Site ID bevor Sie eine RDS for Db2-DB-Instance erstellen. Weitere Informationen finden Sie unter Hinzufügen IBM IDs zu einer Parametergruppe für RDS für Db2-DB-Instances. Sie können mehrere RDS for Db2-DB-Instances mit unterschiedlichen IBM Customer IDs und IBM Site IDs in derselben AWS-Konto oder ausführen. AWS-Region

> **Important**
>
> Wenn wir Ihre Lizenz nicht anhand Ihrer IBM Customer ID und Ihrer verifizieren könnenIBM Site ID, beenden wir möglicherweise alle DB-Instances, die mit diesen nicht verifizierten Lizenzen laufen.

Wenn Sie ein neuer IBM Db2 Kunde sind, müssen Sie zunächst eine Db2-Softwarelizenz von erwerben. IBM Nach dem Kauf einer Db2-Softwarelizenz erhalten Sie einen Berechtigungsnachweis vonIBM, in dem Ihre und Ihre IBM Customer ID Rechte aufgeführt sind. IBM Site ID

Wenn Sie bereits IBM Db2 Kunde sind, finden Sie Ihr IBM Customer ID und Ihr Zertifikat IBM Site ID auf Ihrem Berechtigungsnachweis unter. IBM

Sie finden Ihr IBM Customer ID und Ihr auch IBM Site ID in Ihrem IBM Passport Advantage Online-Konto. Nachdem Sie sich angemeldet haben, können Sie beide entweder IDs auf der Hauptseite oder auf der Seite für Software-Downloads ansehen.

Hinzufügen IBM IDs zu einer Parametergruppe für RDS für Db2-DB-Instances

Da Sie Standardparametergruppen nicht ändern können, müssen Sie eine benutzerdefinierte Parametergruppe erstellen und diese dann so ändern, dass sie die Werte für Ihre IBM Customer ID und Ihre IBM Site ID enthält. Informationen zu Parametergruppen finden Sie unter DB-Parametergruppen für Amazon RDS Amazon.

> **Important**
>
> Sie müssen eine benutzerdefinierte Parametergruppe mit Ihren IBM Customer ID und Ihrer erstellen, IBM Site ID bevor Sie eine RDS for Db2-DB-Instance erstellen.

Verwenden Sie die Parametereinstellungen in der folgenden Tabelle.

Parameter	Wert
`rds.ibm_customer_id`	`<your IBM Customer ID>`
`rds.ibm_site_id`	`<your IBM Site ID>`

Parameter	Wert
`ApplyMethod`	`immediate` , `pending-reboot`

Diese Parameter sind dynamisch, was bedeutet, dass alle Änderungen an ihnen sofort wirksam werden und Sie die DB-Instance nicht neu starten müssen. Wenn Sie nicht möchten, dass die Änderungen sofort wirksam werden, können Sie festlegen, `ApplyMethod` dass `pending-reboot` diese Änderungen während eines Wartungsfensters vorgenommen werden.

Sie können eine benutzerdefinierte Parametergruppe mithilfe der AWS Management Console, der oder der Amazon RDS-API erstellen und ändern. AWS CLI

Konsole

Um Ihren IBM Customer ID und Ihren IBM Site ID zu einer Parametergruppe hinzuzufügen

1. Erstellen Sie eine neue DB-Parametergruppe. Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.
2. Ändern Sie die von Ihnen erstellte Parametergruppe. Weitere Informationen zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

AWS CLI

So fügen Sie Ihren IBM Customer ID und Ihren IBM Site ID zu einer Parametergruppe hinzu

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem create-db-parameter-groupSie den Befehl ausführen.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - `--db-parameter-group-name`— Ein Name für die Parametergruppe, die Sie erstellen.
   - `--db-parameter-group-family`— Die Db2-Engine-Edition und die Hauptversion. Zulässige Werte: `db2-se-11.5`, `db2-ae-11.5`.
   - `--description`— Eine Beschreibung für diese Parametergruppe.

Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.

2. Ändern Sie die Parameter in der benutzerdefinierten Parametergruppe, die Sie durch Ausführen des modify-db-parameter-groupBefehls erstellt haben.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - `--db-parameter-group-name`— Der Name der Parametergruppe, die Sie erstellt haben.
   - `--parameters`— Eine Reihe von Parameternamen, Werten und Anwendungsmethoden für die Parameteraktualisierung.

   Weitere Hinweise zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

RDS-API

So fügen Sie Ihren IBM Customer ID und Ihren IBM Site ID zu einer Parametergruppe hinzu

1. Erstellen Sie mithilfe des Amazon CreateDBParameterGroupRDS-API-Vorgangs eine benutzerdefinierte DB-Parametergruppe.

   Nutzen Sie die folgenden erforderlichen Parameter:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.

2. Ändern Sie die Parameter in der benutzerdefinierten Parametergruppe, die Sie mithilfe des ModifyDBParameterGroupRDS-API-Vorgangs erstellt haben.

   Nutzen Sie die folgenden erforderlichen Parameter:

   - `DBParameterGroupName`
   - `Parameters`

Weitere Informationen zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Jetzt sind Sie bereit, eine DB-Instance zu erstellen und die benutzerdefinierte Parametergruppe an die DB-Instance anzuhängen. Weitere Informationen erhalten Sie unter Erstellen einer Amazon RDS-DB-Instance und Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon.

Integrieren mit AWS License Manager

AWS License ManagerIntegriert in RDS for Db2, um die Überwachung der Lizenznutzung von RDS for Db2 im BYOL-Modell zu unterstützen. License Manager unterstützt die Nachverfolgung von RDS for Db2-Engine-Editionen auf Basis von virtual CPUs (vCPUs). Sie können License Manager auch verwenden AWS Organizations , um alle Ihre Unternehmenskonten zentral zu verwalten.

Um die Lizenznutzung Ihrer RDS for Db2-DB-Instances nachzuverfolgen, müssen Sie selbstverwaltete Lizenzen erstellen. Sie können selbstverwaltete Lizenzen mithilfe der AWS Management Console, der AWS License Manager CLI und der AWS License Manager API erstellen. Oder Sie können die Erstellung von selbstverwalteten Lizenzen mit AWS CloudFormation Vorlagen automatisieren. Terraform

RDS for Db2-Ressourcen, die dem Produktinformationsfilter entsprechen, werden automatisch der selbstverwalteten Lizenz zugeordnet. Die Erkennung von RDS für Db2-DB-Instances kann bis zu 24 Stunden dauern.

Die folgende Tabelle zeigt die verfügbaren Werte für den Engine Edition-Produktinformationsfilter für RDS for Db2.

Wert	Beschreibung
`db2-se`	Db2 Standard Edition
`db2-ae`	Db2 Advanced-Ausgabe

Themen

- Terminologie

- Eine selbstverwaltete Lizenz erstellen in AWS License Manager
- Automatisieren Sie die Erstellung von selbstverwalteten Lizenzen mithilfe von Vorlagen AWS License Manager
- Einstellungen für die Erstellung von selbstverwalteten Lizenzen

Terminologie

Auf dieser Seite wird die folgende Terminologie verwendet, wenn es um die Amazon RDS-Integration mit geht AWS License Manager.

Selbstverwaltete Lizenz

Selbstverwaltete Lizenz ist ein Begriff, der in verwendet wird. AWS License Manager Die Amazon RDS-Konsole bezeichnet die Lizenz als AWS License Manager Konfiguration. Eine selbst verwaltete Lizenz enthält Lizenzregeln, die auf den Bedingungen Ihrer Unternehmensvereinbarungen basieren. Die Regeln, die Sie erstellen, bestimmen, wie Befehle AWS verarbeitet werden, die Lizenzen verbrauchen. Arbeiten Sie bei der Erstellung einer selbst verwalteten Lizenz eng mit dem Compliance-Team Ihres Unternehmens zusammen, um Ihre Unternehmensvereinbarungen zu überprüfen. Weitere Informationen finden Sie unter Selbstverwaltete Lizenzen im License Manager.

Eine selbstverwaltete Lizenz erstellen in AWS License Manager

Sie können eine selbstverwaltete Lizenz mithilfe der AWS Management Console, der AWS License Manager CLI und der AWS License Manager API erstellen.

> **Note**
>
> Wenn Sie eine RDS for Db2-DB-Instance mithilfe von erstellen AWS Management Console, erstellen Sie eine selbstverwaltete Lizenz, indem Sie einen Namen für die Lizenz eingeben. Dann ordnet Amazon RDS die DB-Instance dieser Lizenz zu. (In der Amazon RDS-Konsole wird diese Lizenz als AWS License Manager Konfiguration bezeichnet.) Wenn Sie eine RDS for Db2-DB-Instance mithilfe der AWS License Manager CLI oder AWS License Manager API erstellen möchten, müssen Sie zunächst eine selbstverwaltete Lizenz mit den folgenden Schritten erstellen. Die gleiche Situation gilt für die Wiederherstellung einer RDS for Db2-DB-Instance zu einem bestimmten Zeitpunkt oder aus einem Snapshot.

Konsole

Um eine selbstverwaltete Lizenz zu erstellen, um die Lizenznutzung Ihrer RDS für Db2-DB-Instances nachzuverfolgen

1. Wechseln Sie zu https://console.aws.amazon.com/license-manager/.
2. Erstellen Sie eine selbstverwaltete Lizenz.

   Anweisungen finden Sie im AWS License Manager Benutzerhandbuch unter Erstellen einer selbstverwalteten Lizenz.

   Fügen Sie im Bedienfeld Produktinformationen eine Regel für einen RDS-Produktinformationsfilter hinzu.

   Weitere Informationen finden Sie unter ProductInformation in der AWS License Manager -API-Referenz.

AWS License Manager CLI

> **Note**
>
> Dieses Verfahren verwendet einen AWS License Manager CLI-Befehl.

Führen Sie den AWS License Manager create-license-configurationBefehl aus AWS CLI, um mit dem eine selbstverwaltete Lizenz zu erstellen. Verwenden Sie die `--cli-input-yaml` Optionen `--cli-input-json` oder, um die Optionen an den Befehl zu übergeben.

Weitere Informationen finden Sie unter the section called “Einstellungen für die Erstellung von selbstverwalteten Lizenzen”.

Mit dem folgenden Befehl wird eine selbstverwaltete Lizenz für Db2 Standard Edition erstellt.

```
aws license-manager create-license-configuration --cli-input-json file://rds-db2-se.json
```

Die folgende JSON-Datei ist der Inhalt der `rds-db2-se.json` Datei, die im vorherigen Befehl verwendet wurde.

```
{
```

```
    "Name": "rds-db2-se",
    "Description": "RDS Db2 Standard Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["db2-se"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Weitere Informationen zu Produktinformationen finden Sie unter Automatisiertes Erkennen des Ressourcenbestands im AWS License Manager -Benutzerhandbuch.

Weitere Informationen zum Parameter `--cli-input` finden Sie unter Generieren der AWS CLI -Skeleton- und -Eingabeparameter aus einer JSON- oder YAML-Eingabedatei im AWS CLI -Benutzerhandbuch.

AWS License Manager API

> **Note**
>
> Dieses Verfahren verwendet einen AWS License Manager API-Befehl.

Verwenden Sie den `CreateLicenseConfiguration` AWS License Manager API-Vorgang mit den folgenden erforderlichen Parametern, um eine selbstverwaltete Lizenz zu erstellen:

- `Name`
- `LicenseCountingType`
- `ProductInformationList`
- `ResourceType`
- `ProductInformationFilterList`

- `ProductInformationFilterName`
- `ProductInformationFilterValue`
- `ProductInformationFilterComparator`

Weitere Informationen zu den Parametern finden Sie unter the section called “Einstellungen für die Erstellung von selbstverwalteten Lizenzen”.

Automatisieren Sie die Erstellung von selbstverwalteten Lizenzen mithilfe von Vorlagen AWS License Manager

Sie können die Erstellung von selbstverwalteten Lizenzen mithilfe AWS CloudFormation von Vorlagen automatisieren. Terraform

Mit der folgenden AWS CloudFormation Beispielvorlage werden selbstverwaltete Lizenzen für Db2 Standard Edition auf RDS for Db2 erstellt. Aktualisieren Sie für eine Vorlage für Db2 Advanced Edition die Werte für`Name`, und. `Description` `ProductInformationFilter`

```
AWSTemplateFormatVersion: "2010-09-09"
Description: CloudFormation template to create a License Configuration for Db2 Standard
 Edition on RDS for Db2.

Resources:
  Db2LicenseConfiguration:
    Type: "AWS::LicenseManager::LicenseConfiguration"
    Properties:
      Name: "rds-db2-se"
      Description: "Db2 Standard Edition on RDS for Db2"
      LicenseCountingType: "vCPU"
      LicenseCountHardLimit: false
      ProductInformationList:
        - ResourceType: "RDS"
          ProductInformationFilterList:
            - ProductInformationFilterName: "Engine Edition"
              ProductInformationFilterValue:
                - "db2-se"
              ProductInformationFilterComparator: "EQUALS"
```

Weitere Informationen zur Verwendung AWS CloudFormation mit Amazon RDS finden Sie unterErstellen von Amazon RDS Amazon mit AWS CloudFormation.

Die folgende Terraform Beispielvorlage erstellt selbstverwaltete Lizenzen für Db2 Standard Edition auf RDS for Db2. Ersetzen Sie durch Ihre*us-east-1*. AWS-Region Aktualisieren Sie für eine Vorlage für Db2 Advanced Edition die Werte für `name``description`, und`product_information_filter`.

```
provider "aws" {
  region = "us-east-1"
}

resource "aws_licensemanager_license_configuration" "rds_db2_license_config" {
  name                     = "rds-db2-se"
  description              = "Db2 Standard Edition on RDS for Db2
  license_counting_type    = "vCPU"
  license_count_hard_limit = false

  product_information_list {
    resource_type = "RDS"

    product_information_filter {
      name       = "Engine Edition"
      comparator = "EQUALS"
      value      = ["db2-se"]
    }
  }
}
```

Weitere Informationen zur Verwendung von Terraform Amazon RDS finden Sie unter Verwendung Terraform als IaC-Tool für das AWS Cloud und Bewährte Methoden für die Verwendung des AWS Terraform-Anbieters in AWS Prescriptive Guidance.

Einstellungen für die Erstellung von selbstverwalteten Lizenzen

In der folgenden Tabelle finden Sie Details zu den Einstellungen für die Erstellung von selbstverwalteten Lizenzen mithilfe der AWS License Manager CLI, der AWS License Manager API, einer AWS CloudFormation Vorlage und einer Terraform Vorlage. Der Parametername in der folgenden Tabelle wird im Format des Namens angezeigt, der in der AWS License Manager API und der AWS CloudFormation Vorlage verwendet wird.

Parametername	Datentyp	Erforderlich	Beschreibung
Name	Zeichenfolge	Ja	Der Name der Lizenzkonfiguration.
Beschreibung	Zeichenfolge	Nein	Die Beschreibung der Lizenzkonfiguration.
LicenseCountingType	Zeichenfolge	Ja	Die Dimension, die zur Nachverfolgung des Lizenzinventars verwendet wird. Zulässiger Wert: `vCPU`.
LicenseCountHardLimit	boolesch	Nein	Gibt an, ob eine harte oder eine sanfte Lizenzerzwingung verwendet wird. Bei Überschreitung eines festen Limits wird der Start neuer Instances blockiert.
ProductInformationList	eine Reihe von Objekten	Ja	Eine Liste mit Produktinformationen für eine Lizenzkonfiguration.
ResourceType	Zeichenfolge	Ja	Der Ressourcentyp. Zulässiger Wert: `RDS`.
ProductInformationFilterList	eine Reihe von Objekten	Ja	Eine Liste von Produktinformationsfiltern für eine Lizenzkonfiguration.

Parametername	Datentyp	Erforderlich	Beschreibung
ProductInformation FilterName	Zeichenfolge	Ja	Der Name des Filtertyps, der deklariert wird. Zulässiger Wert: `Engine Edition`.
ProductInformation FilterValue	Zeichenfolgen-Array	Ja	Der Wert, nach dem gefiltert werden soll. Sie dürfen nur einen Wert angeben. Gültige Werte: `db2-se` oder `db2-ae`.
ProductInformation FilterComparator	Zeichenfolge	Ja	Der logische Operator für`ProductInformation FilterName` . Zulässiger Wert: `EQUALS`.

Db2-Lizenz über AWS Marketplace

Beim AWS Marketplace Modell mit durchgehender Db2-Lizenz zahlen Sie für das Abonnement von Db2-Lizenzen einen Stundensatz. Mit diesem Modell können Sie schnell mit RDS für Db2 loslegen, ohne Lizenzen erwerben zu müssen.

Um die Db2-Lizenz nutzen zu können AWS Marketplace, benötigen Sie ein aktives AWS Marketplace Abonnement für die jeweilige IBM Db2 Edition, die Sie verwenden möchten. Wenn Sie noch keine haben, abonnieren AWS Marketplace Sie diese IBM Db2 Edition.

Amazon RDS unterstützt Db2-Lizenzen AWS Marketplace für die IBM Db2 Standard Edition und die IBM Db2 Advanced Edition.

Themen

- Terminologie

- Zahlungen und Rechnungsstellung
- Angebote von Db2 Marketplace abonnieren und sich registrieren bei IBM
- Einholung eines privaten Angebots

Terminologie

Auf dieser Seite wird die folgende Terminologie verwendet, wenn es um die Amazon RDS-Integration mit geht AWS Marketplace.

SaaS Subscription (SaaS-Abonnement)

Bei AWS Marketplace software-as-a-service (SaaS-) Produkten wie dem pay-as-you-go Lizenzmodell wird ein nutzungsbasiertes Abonnementmodell verwendet. IBM, der Softwareanbieter für Db2, verfolgt Ihre Nutzung und Sie zahlen nur für das, was Sie tatsächlich nutzen.

Öffentliches Angebot

Öffentliche Angebote ermöglichen es Ihnen, AWS Marketplace Produkte direkt bei der zu kaufen AWS Management Console.

Privates Angebot

Private Angebote sind ein Einkaufsprogramm, das es Verkäufern und Käufern ermöglicht, individuelle Preise und Bedingungen der Endbenutzer-Lizenzvereinbarung (EULA) für Käufe in AWS Marketplace auszuhandeln.

Gebühren für den Db2 Marketplace

Gebühren für die Nutzung der Db2-Softwarelizenz von. IBM Diese Servicegebühren sind abgerechnet AWS Marketplace und erscheinen auf Ihrer AWS Rechnung im AWS Marketplace Abschnitt.

Amazon RDS-Gebühren

AWS Gebühren, die für die Dienste RDS für Db2 anfallen. Davon ausgenommen sind Lizenzen bei der Nutzung AWS Marketplace für Db2-Lizenzen. Die Gebühren werden über den verwendeten Amazon RDS-Service berechnet und erscheinen auf Ihrer AWS Rechnung.

Zahlungen und Rechnungsstellung

RDS for Db2 ist in Db2 integriert, AWS Marketplace um stündliche pay-as-you-go Lizenzen für Db2 anzubieten. Die Gebühren für den Db2 Marketplace decken die Lizenzkosten der Db2-Software ab, und die Amazon RDS-Gebühren decken die Kosten für die Nutzung Ihrer RDS für die Db2-DB-Instance ab. Preisinformationen finden Sie unter Amazon RDS for Db2 — Preise.

Um diese Gebühren zu beenden, müssen Sie alle RDS for Db2-DB-Instances löschen. Darüber hinaus können Sie Ihre Abonnements AWS Marketplace für Db2-Lizenzen entfernen. Wenn Sie Ihre Abonnements entfernen, ohne Ihre DB-Instances zu löschen, stellt Amazon RDS Ihnen weiterhin die Nutzung der DB-Instances in Rechnung. Weitere Informationen finden Sie unter the section called “Löschen einer DB-Instance”.

In der Konsole können Sie Rechnungen für Ihre RDS für Db2-DB-Instances, die eine Db2-Lizenz verwenden, einsehen und AWS Marketplace Zahlungen verwalten.AWS Billing Ihre Rechnungen beinhalten zwei Gebühren: eine für Ihre Nutzung der Db2-Lizenz AWS Marketplace und eine für Ihre Nutzung von Amazon RDS. Weitere Informationen zur Abrechnung finden Sie im AWS Fakturierung und Kostenmanagement Benutzerhandbuch unter Ihre Rechnung anzeigen.

Angebote von Db2 Marketplace abonnieren und sich registrieren bei IBM

Um die Db2-Lizenz über verwenden zu können AWS Marketplace, müssen Sie die verwenden, AWS Management Console um die folgenden beiden Aufgaben auszuführen. Sie können diese Aufgaben nicht über die AWS CLI oder die RDS-API ausführen.

> **Note**
>
> Wenn Sie Ihre DB-Instances mithilfe der AWS CLI oder der RDS-API erstellen möchten, müssen Sie diese beiden Aufgaben zuerst ausführen.

Themen

- Aufgabe 1: Abonnieren Sie Db2 in AWS Marketplace
- Aufgabe 2: Registrieren Sie Ihr Abonnement bei IBM

Aufgabe 1: Abonnieren Sie Db2 in AWS Marketplace

Um die Db2-Lizenz mit verwenden zu können AWS Marketplace, benötigen Sie ein aktives AWS Marketplace Abonnement für Db2. Da Abonnements mit einer bestimmten IBM Db2 Edition verknüpft

sind, müssen Sie Db2 AWS Marketplace für jede Edition von Db2 abonnieren, die Sie verwenden möchten: IBM Db2Advanced Edition, Standard Edition. IBM Db2 Informationen zu AWS Marketplace Abonnements finden Sie im Buyer Guide unter nutzungsbasierte SaaS-Abonnements.AWS Marketplace

Wir empfehlen, dass Sie Db2 abonnieren, AWS Marketplace bevor Sie mit der Erstellung einer DB-Instance beginnen.

Aufgabe 2: Registrieren Sie Ihr Abonnement bei IBM

Nachdem Sie Db2 in abonniert haben AWS Marketplace, schließen Sie die Registrierung Ihrer IBM-Bestellung AWS Marketplace auf der Seite für den von Ihnen ausgewählten Db2-Abonnementtyp ab. Wählen Sie auf der AWS Marketplace Seite Kaufoptionen anzeigen und anschließend Konto einrichten aus. Du kannst dich entweder mit deinem bestehenden IBM Konto registrieren oder indem du ein kostenloses IBM Konto erstellst.

Einholung eines privaten Angebots

Sie können ein AWS Marketplace privates Angebot für Db2 bei IBM anfordern. Weitere Informationen finden Sie im AWS Marketplace Buyer Guide unter Private Angebote.

> **Note**
>
> Wenn Sie ein AWS Organizations Benutzer sind und ein privates Angebot erhalten haben, das auf Ihre Zahler- und Mitgliedskonten ausgestellt wurde, gehen Sie wie folgt vor, um Db2 direkt für jedes Konto in Ihrer Organisation zu abonnieren.

Um ein privates Db2-Angebot zu erhalten

1. Nachdem ein privates Angebot erstellt wurde, melden Sie sich bei der AWS Marketplace Konsole an.
2. Öffnen Sie die E-Mail mit einem Link zu einem privaten Db2-Angebot.
3. Folgen Sie dem Link, um direkt auf das private Angebot zuzugreifen.

> **Note**
>
> Wenn Sie diesem Link folgen, bevor Sie sich mit dem richtigen Konto anmelden, wird der Fehler „Seitennotiz gefunden“ (404) angezeigt.

4. Lesen Sie die Allgemeinen Geschäftsbedingungen.
5. Wählen Sie Bedingungen akzeptieren aus.

> **Note**
>
> Wenn ein AWS Marketplace privates Angebot nicht akzeptiert wird, werden die Db2-Servicegebühren von AWS Marketplace weiterhin zum öffentlichen Stundensatz in Rechnung gestellt.

6. Um die Angebotsdetails zu überprüfen, wählen Sie in der Produktliste die Option Details anzeigen aus.

Nachdem Sie das Verfahren abgeschlossen haben, können Sie Ihre DB-Instance erstellen, indem Sie die Schritte unter befolgenthe section called “Erstellen einer DB-Instance”. Vergewissern Sie sich AWS Management Console, dass Sie unter Lizenz die Option Through auswählen AWS Marketplace.

Zwischen Db2-Lizenzen wechseln

Sie können in RDS für Db2 zwischen Db2-Lizenzen wechseln. Sie können beispielsweise mit Bring Your Own License (BYOL) beginnen und dann zur Db2-Lizenz wechseln. AWS Marketplace

> **Important**
>
> Wenn Sie zur Db2-Lizenz wechseln möchten, stellen Sie sicher AWS Marketplace, dass Sie über ein aktives AWS Marketplace Abonnement für die IBM Db2 Edition verfügen, die Sie verwenden möchten. Wenn Sie dies nicht tun, abonnieren Sie zuerst Db2 AWS Marketplace für diese Db2-Edition und schließen Sie dann den Wiederherstellungsvorgang ab.

Konsole

Um zwischen Db2-Lizenzen zu wechseln

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Automated backups (Automatisierte Backups) aus.

   Die automatisierten Backups werden auf der Registerkarte Current Region (Aktuelle Region) angezeigt.

3. Wählen Sie die DB-Instance aus, die Sie wiederherstellen möchten.
4. Wählen Sie unter Aktionen die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

   Anschließend wird das Fenster Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) angezeigt.
5. Wählen Sie Späteste Wiederherstellungszeit, um auf den spätesten möglichen Zeitpunkt wiederherzustellen oder wählen Sie Benutzerdefiniert, um eine Zeit auszuwählen.

   Wenn Sie Benutzerdefiniert wählen, geben Sie das Datum und die Uhrzeit ein, für die Sie die Instance wiederherstellen möchten.

   > **Note**
   >
   > Zeiten werden in Ihrer lokalen Zeitzone angezeigt, die durch einen Offset von Coordinated Universal Time (UTC) angezeigt wird. UTC-5 ist beispielsweise Eastern Standard Time/Central Daylight Time.
6. Wählen Sie für DB-Engine die Db2-Lizenz aus, die Sie verwenden möchten.
7. Geben Sie für DB-Instance-Kennung den Namen der wiederhergestellten DB-Ziel-Instance ein. Der Name muss eindeutig sein.
8. Wählen Sie nach Bedarf andere Optionen wie die DB-Instance-Klasse und Speicher aus, sowie ob Sie Speicher-Autoscaling verwenden möchten.

   Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.
9. Wählen Sie Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen) aus.

Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

AWS CLI

Um zwischen Db2-Lizenzen zu wechseln, führen Sie den Befehl restore-db-instance-to- point-in-time aus. Das folgende Beispiel stellt die neueste point-in-time Version wieder her, setzt die DB-Engine auf IBM Db2 Advanced Edition und setzt das Lizenzmodell auf Db2-Lizenz über. AWS Marketplace

Sie können andere Einstellungen festlegen. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Example

Für LinuxmacOS, oderUnix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my_source_db_instance \
    --target-db-instance-identifier my_target_db_instance \
    --use-latest-restorable-time \
    --engine db2-ae \
    --license-model marketplace-license
```

Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my_source_db_instance ^
    --target-db-instance-identifier my_target_db_instance ^
    --use-latest-restorable-time ^
    --engine db2-ae ^
    --license-model marketplace-license
```

Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

RDS-API

Um zwischen Db2-Lizenzen zu wechseln, rufen Sie den Amazon `RestoreDBInstanceToPointInTime`RDS-API-Vorgang mit den folgenden Parametern auf:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`
- `Engine`
- `LicenseModel`

Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Amazon RDS für Db2-Instance-Klassen

Die Rechen- und Speicherkapazität von DB-Instances wird über deren Instance-Klasse festgelegt. Die benötigte DB-Instance-Klasse richtet sich nach Ihren Rechen- und Speicheranforderungen.

Unterstützte RDS für Db2-Instance-Klassen

Die unterstützten Amazon RDS for Db2-Instance-Klassen sind eine Teilmenge der Amazon RDS-DB-Instance-Klassen. Die vollständige Liste der Amazon RDS-Instance-Klassen finden Sie unter.

Themen

- Unterstützte RDS für Db2-Instance-Klassen für Db2 Standard Edition
- Unterstützte RDS für Db2-Instanzklassen für Db2 Advanced Edition

Unterstützte RDS für Db2-Instance-Klassen für Db2 Standard Edition

In der folgenden Tabelle sind alle Instance-Klassen aufgeführt, die für die Db2 Standard Edition Db2-Datenbankversion 11.5.9.0 unterstützt werden. Diese Instanzklassen sind sowohl für Bring Your Own License (BYOL) als auch für Db2-Lizenzen verfügbar. AWS Marketplace

Typ der Instance-Klasse	Instance-Klasse
Allzweck-Instanzklassen mit Intel Xeon Scalable Prozessoren der 3. Generation, SSD-Speicher und Netzwerkoptimierung	db.m6idn.large — db.m6idn.8xlarge
Allzweck-Instance-Klassen, die auf Prozessoren der dritten Generation basieren Intel Xeon Scalable	db.m6in.large — db.m6in.8xlarge
Instanzklassen für allgemeine Zwecke	db.m7i.large — db.m7i.8xlarge db.m6i.large — db.m6i.8xlarge
Speicheroptimierte Instanzklassen mit lokaler Basis, die von Prozessoren der 3. Generation angetrieben werden NVMe SSDs Intel Xeon Scalable	db.x2iedn.xlarge
Speicheroptimierte Instanzklassen, die auf Intel Xeon Scalable Prozessoren der 3. Generation basieren	db.r6idn.large — db.r6idn.4xlarge

Typ der Instance-Klasse	Instance-Klasse
	db.r6in.large — db.r6in.4xlarge
Arbeitsspeicheroptimierte Instance-Klassen	db.r7i.large — db.r7i.8xlarge db.r6i.large–db.r6i.4xlarge
Instance-Klassen mit Spitzenleistung	db.t3.small–db.t3.2xlarge

Unterstützte RDS für Db2-Instanzklassen für Db2 Advanced Edition

In der folgenden Tabelle sind alle Instance-Klassen aufgeführt, die für die Db2 Advanced Edition Db2-Datenbankversion 11.5.9.0 unterstützt werden. Diese Instanzklassen sind sowohl für Bring Your Own License (BYOL) als auch für Db2-Lizenzen verfügbar. AWS Marketplace

Typ der Instance-Klasse	Instance-Klasse
Allzweck-Instanzklassen mit Intel Xeon Scalable Prozessoren der 3. Generation, SSD-Speicher und Netzwerkoptimierung	db.m6idn.large — db.m6idn.32xlarge
Allzweck-Instance-Klassen, die auf Prozessoren der dritten Generation basieren Intel Xeon Scalable	db.m6in.large — db.m6in.32xlarge
Instanzklassen für allgemeine Zwecke	db.m6i.large — db.m7i.48xlarge db.m7i.large — db.m7i.48xlarge
Speicheroptimierte Instanzklassen mit lokaler Basis, die von Prozessoren der 3. Generation angetrieben werden NVMe SSDs Intel Xeon Scalable	db.x2iedn.xlarge–db.x2iedn.32xlarge
Speicheroptimierte Instanzklassen, die auf Intel Xeon Scalable Prozessoren der 3. Generation basieren	db.r6idn.large — db.r6idn.32xlarge

Typ der Instance-Klasse	Instance-Klasse
	db.r6in.8xlarge — db.r6in.32xlarge
Arbeitsspeicheroptimierte Instance-Klassen	db.r6i.large — db.r7i.48xlarge db.r7i.large — db.r7i.48xlarge
Instance-Klassen mit Spitzenleistung	db.t3.small–db.t3.2xlarge

Standardrollen von Amazon RDS für Db2

RDSfor Db2 fügt die folgenden sechs Rollen hinzu und weist sie `master_user_role` mit der `ADMIN` Option zu. Wenn die Datenbank bereitgestellt wird, werden RDS für Db2 die Berechtigungen dem Masterbenutzer `master_user_role` zugewiesen. Der Masterbenutzer kann diese Rollen wiederum anderen Benutzern, Gruppen oder Rollen mit systemeigenen `GRANT` Anweisungen zuweisen, indem er eine Verbindung zur Datenbank herstellt.

- DBA— RDS für Db2 erstellt diese leere Rolle mit `DATAACCESS` Autorisierung. Der Hauptbenutzer kann dieser Rolle weitere Autorisierungen oder Rechte hinzufügen und die Rolle dann anderen Benutzern, Gruppen oder Rollen zuweisen.
- DBA_ RESTRICTED — RDS für Db2 erstellt diese leere Rolle. Der Hauptbenutzer kann dieser Rolle Rechte hinzufügen und die Rolle dann anderen Benutzern, Gruppen oder Rollen gewähren.
- DEVELOPER— RDS für Db2 erstellt diese leere Rolle mit `DATAACCESS` Autorisierung. Der Hauptbenutzer kann dieser Rolle weitere Autorisierungen oder Rechte hinzufügen und die Rolle dann anderen Benutzern, Gruppen oder Rollen zuweisen.
- ROLE_ NULLID _ PACKAGES — RDS für Db2 gewährt dieser Rolle `EXECUTE` Rechte für `ALL NULLID` Pakete, die bei `CREATE DATABASE` der Ausführung an Db2 gebunden waren.
- ROLE_ PROCEDURES — RDS für Db2 gewährt dieser Rolle `EXECUTE` Rechte für alle Prozeduren. `SYSIBM`
- ROLE_ TABLESPACES — RDS für Db2 gewährt `USAGE` Rechte für Tablespaces, die mit dem Befehl erstellt wurden. `CREATE DATABASE`

Amazon RDS für Db2-Parameter

Amazon RDS for Db2 verwendet drei Arten von Parametern: Datenbank-Manager-Konfigurationsparameter, Registrierungsvariablen und Datenbankkonfigurationsparameter. Sie können die ersten beiden Typen über Parametergruppen und den letzten Typ über die rdsadmin.update_db_param gespeicherte Prozedur verwalten.

Standardmäßig verwendet eine RDS for Db2-DB-Instance eine DB-Parametergruppe, die für eine Db2-Datenbank und eine DB-Instance spezifisch ist. Diese Parametergruppe enthält Parameter für die IBM Db2 Datenbank-Engine, insbesondere die Konfigurationsparameter und Registrierungsvariablen des Datenbankmanagers. Hinweise zum Arbeiten mit Parametergruppen finden Sie unterParametergruppen für Amazon RDS.

Die Konfigurationsparameter für RDS for Db2-Datenbanken sind auf die Standardwerte der von Ihnen ausgewählten Speicher-Engine festgelegt. Weitere Informationen zu Db2-Parametern finden Sie in den Db2-Datenbankkonfigurationsparametern in der Dokumentation. IBM Db2

Themen

- Parameter in Parametergruppen anzeigen
- Alle Parameter mit Db2-Befehlen anzeigen
- Ändern der Parameter in Parametergruppen
- Ändern der Datenbankkonfigurationsparameter mit Db2-Befehlen

Parameter in Parametergruppen anzeigen

Die Konfigurationsparameter des Datenbankmanagers und die Registrierungsvariablen werden in Parametergruppen festgelegt. Sie können die Database Manager-Konfigurationsparameter und die Registrierungsvariablen für eine bestimmte Db2-Version anzeigen, indem Sie die AWS Management Console AWS CLI, oder die RDS-API verwenden.

Konsole

So können Sie die Parameterwerte für eine DB-Parametergruppe ansehen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

Die verfügbaren DB-Parametergruppen erscheinen in einer Liste.

3. Wählen Sie den Namen der Parametergruppe, um deren Parameterliste anzuzeigen.

AWS CLI

Sie können die Konfigurationsparameter des Datenbankmanagers und die Registrierungsvariablen für eine Db2-Version anzeigen, indem Sie den describe-engine-default-parametersBefehl ausführen. Geben Sie einen der folgenden Werte für die Option `--db-parameter-group-family` an:

- `db2-ae-11.5`
- `db2-se-11.5`

Um beispielsweise die Parameter für Db2 Standard Edition 11.5 anzuzeigen, führen Sie den folgenden Befehl aus:

```
aws rds describe-engine-default-parameters --db-parameter-group-family db2-se-11.5
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "agent_stack_sz",
                "ParameterValue": "1024",
                "Description": "You can use this parameter to determine the amount of
 memory that is allocated by Db2 for each agent thread stack.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "256-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "agentpri",
                "ParameterValue": "-1",
                "Description": "This parameter controls the priority given to all
 agents and to other database manager instance processes and threads by the operating
 system scheduler. This priority determines how CPU time is allocated to the database
```

```
 manager processes, agents, and threads relative to other processes and threads running
 on the machine.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1-99",
                "IsModifiable": false
            },
            ...
        ]
    }
}
```

Führen Sie den folgenden Befehl aus, um nur die änderbaren Parameter für Db2 Standard Edition 11.5 aufzulisten:

FürLinux, odermacOS: Unix

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family db2-se-11.5 \
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

Windows:

```
aws rds describe-engine-default-parameters ^
    --db-parameter-group-family db2-se-11.5 ^
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

RDS-API

Um die Parameterwerte für eine DB-Parametergruppe anzuzeigen, verwenden Sie den `DescribeDBParameters`Vorgang mit dem folgenden erforderlichen Parameter.

- `DBParameterGroupName`

Alle Parameter mit Db2-Befehlen anzeigen

Sie können die Einstellungen für Database Manager-Konfigurationsparameter, Datenbankkonfigurationsparameter und Registrierungsvariablen mithilfe von Db2-Befehlen anzeigen.

Um die Einstellungen anzuzeigen

1. Connect zu Ihrer Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel, *database_namemaster_username*, und *master_password* durch Ihre Informationen.

```
db2 "connect to database_name user master_username using master_password"
```

2. Suchen Sie nach der unterstützten Db2-Version.

```
db2 "select service_level, fixpack_num from table(sysproc.env_get_inst_info()) as
 instanceinfo"
```

3. Zeigen Sie die Parameter für eine bestimmte Db2-Version an.

   - Zeigen Sie die Konfigurationsparameter von Database Manager an, indem Sie den folgenden Befehl ausführen:

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from sysibmadm.dbmcfg
    order by name asc with UR"
```

   - Zeigen Sie alle Ihre Datenbankkonfigurationsparameter an, indem Sie den folgenden Befehl ausführen:

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from table(db_get_cfg(null)) order by name asc, member asc with UR"
```

   - Zeigen Sie die aktuell eingestellten Registrierungsvariablen an, indem Sie den folgenden Befehl ausführen:

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
    cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
    level from table(env_get_reg_variables(null))
    order by reg_var_name,member with UR"
```

Ändern der Parameter in Parametergruppen

Sie können die Konfigurationsparameter des Database Managers und die Registrierungsvariablen in benutzerdefinierten Parametergruppen ändern, indem Sie die AWS Management Console AWS CLI, oder die RDS-API verwenden. Weitere Informationen finden Sie unter DB-Parametergruppen für Amazon RDS Amazon.

Konsole

Um die Konfigurationsparameter und Registrierungsvariablen für den Datenbankmanager zu ändern

1. Erstellen Sie eine benutzerdefinierte Parametergruppe. Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.
2. Ändern Sie die Parameter in dieser benutzerdefinierten Parametergruppe. Weitere Informationen zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

AWS CLI

So ändern Sie die Konfigurationsparameter und Registrierungsvariablen von Database Manager

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem create-db-parameter-groupSie den Befehl ausführen.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - `--db-parameter-group-name`— Ein Name für die Parametergruppe, die Sie erstellen.
   - `--db-parameter-group-family`— Die Db2-Engine-Edition und die Hauptversion. Zulässige Werte: `db2-se-11.5`, `db2-ae-11.5`.
   - `--description`— Eine Beschreibung für diese Parametergruppe.

   Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.
2. Ändern Sie die Parameter in der benutzerdefinierten Parametergruppe, die Sie durch Ausführen des modify-db-parameter-groupBefehls erstellt haben.

   Verwenden Sie den folgenden erforderlichen Parameter:

- `--db-parameter-group-name`— Der Name der Parametergruppe, die Sie erstellt haben.
- `--parameters`— Eine Reihe von Parameternamen, Werten und Anwendungsmethoden für die Parameteraktualisierung.

Weitere Hinweise zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

RDS-API

So ändern Sie die Konfigurationsparameter und Registrierungsvariablen von Database Manager

1. Erstellen Sie mithilfe des CreateDBParameterGroupVorgangs eine benutzerdefinierte DB-Parametergruppe.

   Nutzen Sie die folgenden erforderlichen Parameter:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.
2. Ändern Sie die Parameter in der benutzerdefinierten Parametergruppe, die Sie mithilfe der ModifyDBParameterGroupOperation erstellt haben.

   Nutzen Sie die folgenden erforderlichen Parameter:

   - `DBParameterGroupName`
   - `Parameters`

   Weitere Informationen zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Ändern der Datenbankkonfigurationsparameter mit Db2-Befehlen

Sie können die Datenbankkonfigurationsparameter mit Db2-Befehlen ändern.

Um die Datenbankkonfigurationsparameter zu ändern

1. Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Ändern Sie die Datenbankkonfigurationsparameter, indem Sie die `rdsadmin.update_db_param` gespeicherte Prozedur aufrufen. Weitere Informationen finden Sie unter rdsadmin.update_db_param.

   ```
   db2 "call rdsadmin.update_db_param(
       'database_name',
       'parameter_to_modify',
       'changed_value',
       'restart_database')"
   ```

EBCDICSortierung für Db2-Datenbanken auf Amazon RDS

Amazon RDS for Db2 unterstützt die EBCDIC Sortierung für Db2-Datenbanken. Sie können eine EBCDIC Sortierreihenfolge für eine Datenbank nur angeben, wenn Sie die Datenbank mithilfe der RDS the section called “rdsadmin.create_database” gespeicherten Amazon-Prozedur erstellen.

Wenn Sie eine RDS For-DB2-DB-Instance mithilfe von AWS Management Console, oder RDS API erstellen AWS CLI, können Sie einen Datenbanknamen angeben. Wenn Sie einen Datenbanknamen angeben, RDS erstellt Amazon eine Datenbank mit der Standardsortierung von`SYSTEM`. Wenn Sie eine Datenbank mit EBCDIC Sortierung erstellen müssen, geben Sie beim Erstellen einer DB-Instance keinen Datenbanknamen an.

Die Sortierung für eine Datenbank in RDS für Db2 wird zum Zeitpunkt der Erstellung festgelegt und ist unveränderlich.

Um eine Db2-Datenbank mit Kollation zu erstellen EBCDIC

1. Wenn Sie keine RDS für Db2-DB-Instance haben, erstellen Sie eine DB-Instance, geben aber keinen Datenbanknamen an. Sie können eine DB-Instance erstellen, indem Sie AWS

Management Console, AWS CLI oder verwenden. RDS API Weitere Informationen finden Sie unter Erstellen einer DB-Instance.

2. Erstellen Sie eine Db2-Datenbank und setzen Sie die Kollationsoption auf einen EBCDIC Wert, indem Sie die `rdsadmin.create_database` gespeicherte Prozedur aufrufen. Weitere Informationen finden Sie unter rdsadmin.create_database.

> **Important**
>
> Nachdem Sie eine Datenbank mithilfe der gespeicherten Prozedur erstellt haben, können Sie die Sortierreihenfolge nicht mehr ändern. Wenn Sie möchten, dass eine Datenbank eine andere Sortierreihenfolge verwendet, löschen Sie die Datenbank, indem Sie die the section called “rdsadmin.drop_database” gespeicherte Prozedur aufrufen. Erstellen Sie dann eine Datenbank mit der erforderlichen Sortierreihenfolge.

Lokale Zeitzone für Amazon RDS für Db2-DB-Instances

Die Zeitzone einer Amazon RDS-DB-Instance, auf der Db2 ausgeführt wird, ist standardmäßig festgelegt. Die Standardeinstellung ist Coordinated Universal Time (UTC). Um der Zeitzone Ihrer Anwendungen zu entsprechen, können Sie die Zeitzone Ihrer DB-Instance stattdessen auf eine lokale Zeitzone festlegen.

Sie legen die Zeitzone bei der Erstellung Ihrer DB-Instance fest. Sie können Ihre DB-Instance mithilfe der AWS Management Console, der RDS-API oder der erstellen AWS CLI. Weitere Informationen finden Sie unter Erstellen einer DB-Instance.

Wenn Ihre DB-Instance Teil einer Multi-AZ-Bereitstellung ist, bleibt ihre Zeitzone beim Failover die von Ihnen festgelegte lokale Zeitzone.

Sie können Ihre DB-Instance zu einem von Ihnen angegebenen Zeitpunkt wiederherstellen. Die Uhrzeit wird in Ihrer lokalen Zeitzone angezeigt. Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Das Einstellen der lokalen Zeitzone auf Ihrer DB-Instance hat die folgenden Einschränkungen:

- Sie können die Zeitzone einer vorhandenen Amazon RDS for Db2-DB-Instance nicht ändern.
- Sie können einen Snapshot aus einer DB-Instance in einer Zeitzone nicht in eine DB-Instance in einer anderen Zeitzone wiederherstellen.

- Es wird dringend davon abgeraten, eine Sicherungsdatei aus einer Zeitzone für eine andere Zeitzone wiederherzustellen. Wenn Sie eine Sicherungsdatei von einer Zeitzone in eine andere wiederherstellen, müssen Sie Ihre Abfragen und Anwendungen auf die Auswirkungen der Zeitzonenänderung überprüfen.

Verfügbare Zeitzonen

Sie können die folgenden Werte für die Zeitzoneneinstellung verwenden.

Bereich	Zeitzone
Afrika	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
Amerika	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York, America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asien	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damascus, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Jakarta, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantic/Azores, Atlantic/Cape_Verde
Australien	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brasilien	Brasilien/, Brasilien/Ost DeNoronha

Bereich	Zeitzone
Kanada	Canada/Newfoundland, Canada/Saskatchewan
Etc	Etc/GMT-3
Europa	Europe/Amsterdam, Europe/Athens, Europe/Berlin, Europe/Dublin, Europe/Helsinki, Europe/Kaliningrad, Europe/London, Europe/Madrid, Europe/Moscow, Europe/Paris, Europe/Prague, Europe/Rom, Europe/Sarajevo, Europe/Stockholm
Pazifik	Pacific/Apia, Pacific/Auckland, Pacific/Chatham, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Kiritimati, Pacific/Marquesas, Pacific/Samoa, Pacific/Tongatapu, Pacific/Wake
USA	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific
UTC	UTC

Voraussetzungen für die Erstellung einer Amazon RDS for Db2-DB-Instance

Die folgenden Punkte sind Voraussetzungen für die Erstellung einer DB-Instance.

Themen

- Administratorkonto
- Weitere Überlegungen

Administratorkonto

Wenn Sie eine DB-Instance erstellen, müssen Sie ein Administratorkonto für die Instance angeben. Amazon RDS erteilt diesem lokalen Datenbankadministratorkonto `DBADM` Autorität.

Das Administratorkonto weist die folgenden Merkmale, Funktionen und Einschränkungen auf:

- Ist ein lokaler Benutzer und kein AWS-Konto.
- Hat keine Berechtigungen auf Db2-Instanzebene wie`SYSADM`,, `SYSMAINT` oder. `SYSCTRL`
- Eine Db2-Instanz kann nicht gestoppt oder gestartet werden.
- Eine Db2-Datenbank kann nicht gelöscht werden, wenn Sie den Namen beim Erstellen der DB-Instance angegeben haben.
- Hat vollen Zugriff auf die Db2-Datenbank, einschließlich Katalogtabellen und Ansichten.
- Kann mithilfe von gespeicherten Amazon RDS-Prozeduren lokale Benutzer und Gruppen erstellen.
- Kann Befugnisse und Privilegien gewähren und entziehen.

Das Administratorkonto kann die folgenden Aufgaben ausführen:

- DB-Instances erstellen, ändern oder löschen.
- Erstellen Sie DB-Snapshots.
- Initiieren Sie point-in-time Wiederherstellungen.
- Erstellen Sie automatische Backups von DB-Snapshots.
- Erstellen Sie manuelle Backups von DB-Snapshots.
- Verwenden Sie andere Amazon RDS-Funktionen.

Weitere Überlegungen

Bevor Sie eine DB-Instance erstellen, sollten Sie die folgenden Punkte berücksichtigen:

- Jede Amazon RDS for Db2-DB-Instance kann bis zu 50 Db2-Datenbanken hosten. Weitere Informationen finden Sie unter Mehrere Datenbanken auf einer Amazon RDS for Db2-DB-Instance.
- Anfänglicher Datenbankname
  - Wenn Sie beim Erstellen einer DB-Instance keinen Datenbanknamen angeben, erstellt Amazon RDS keine Datenbank.
  - Geben Sie unter den folgenden Umständen keinen Datenbanknamen an:
    - Sie möchten den `db2_compatibility_vector` Parameter ändern. Weitere Informationen finden Sie unter Einstellung des Parameters db2_compatibility_vector.
- Beim Modell „Bring Your Own License“ (BYOL) müssen Sie zunächst eine benutzerdefinierte Parametergruppe erstellen, die Ihre IBM Customer ID und Ihre IBM Site ID Parametergruppe enthält. Weitere Informationen finden Sie unter Bringen Sie Ihre eigene Lizenz (BYOL) für Db2 mit.
- Beim AWS Marketplace Db2-Lizenzmodell benötigen Sie ein aktives AWS Marketplace Abonnement für die jeweilige IBM Db2 Edition, die Sie verwenden möchten. Wenn Sie noch keines haben, abonnieren Sie Db2 AWS Marketplace für die IBM Db2 Edition, die Sie verwenden möchten. Weitere Informationen finden Sie unter Db2-Lizenz über AWS Marketplace.

Mehrere Datenbanken auf einer Amazon RDS for Db2-DB-Instance

Sie können mehrere Datenbanken auf einer einzigen RDS für Db2-DB-Instance erstellen, indem Sie die gespeicherte Prozedur rdsadmin.create_database aufrufen. Eine einzelne DB-Instance RDS für Db2 ist auf 50 Datenbanken beschränkt. Diese Zahl umfasst Datenbanken sowohl im aktivierten als auch im deaktivierten Zustand.

> **Note**
>
> Wenn Sie mehrere Datenbanken auf einer RDS für Db2 DB-Instance erstellen, die vor dem 15. November 2024 erstellt wurde, müssen Sie die DB-Instance neu starten, um die Unterstützung für mehrere Datenbanken zu aktivieren.

Standardmäßig RDS aktiviert Amazon Datenbanken, wenn Sie sie erstellen. Um die Speicherressourcen zu optimieren, können Sie Datenbanken, die Sie selten verwenden, deaktivieren und sie später bei Bedarf aktivieren. Weitere Informationen erhalten Sie unter the section called “Eine Datenbank deaktivieren” und the section called “Eine Datenbank aktivieren”.

Die Anzahl der aktivierten Datenbanken auf einer DB-Instance hängt von den verfügbaren Speicherressourcen auf dem Server ab. Die Speicherressourcen unterscheiden sich je nach DB-Instance-Klasse und der für die Datenbank konfigurierten Speichermenge. Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”. Hinweise zum Aktualisieren des Speichers RDS für eine Db2-Datenbank finden Sie unterthe section called “rdsadmin.update_db_param”.

Wir empfehlen Ihnen, eine DB-Instance-Klasse mit 2 GB Arbeitsspeicher für allgemeine Datenbankaufgaben, Betriebssystemanforderungen und andere RDS Amazon-Automatisierungsaufgaben wie Backups zu wählen. Weitere Informationen zum Ändern der DB-Instance-Klasse finden Sie unterthe section called “Ändern einer DB-Instance”.

Darüber hinaus IBM empfiehlt es sich, mindestens 1 GB Arbeitsspeicher für jede aktive Datenbank zu verwenden. Weitere Informationen finden Sie in der IBM Dokumentation unter Festplatten- und Speicheranforderungen.

Sie können die maximale Anzahl aktiver Datenbanken, die eine DB-Instance haben kann, mit der folgenden Formel berechnen:

```
Active database limit = (total server memory - 2 GB) / 1 GB
```

Das folgende Beispiel zeigt die maximale Anzahl aktiver Datenbanken für eine DB-Instance mit der DB-Instance-Klasse db.m6i.xlarge:

```
Active database limit = (total server memory - 2 GB) / 1 GB
                      = (16 GB - 2 GB) / 1 GB
                      = 14 databases
```

Wenn Amazon eine Datenbank nach einem Absturz RDS wiederherstellt, aktiviert es die Datenbank, sofern sie zuvor aktiv war. In bestimmten Fällen, z. B. wenn Sie eine DB-Instance-Klasse auf eine niedrigere Speicherkonfiguration ändern, ist möglicherweise nicht genügend Speicher vorhanden, um alle Datenbanken auf der DB-Instance zu aktivieren. In diesen Fällen RDS aktiviert Amazon Datenbanken in der Reihenfolge, in der sie erstellt wurden.

> **Note**
>
> Alle Datenbanken, die Amazon aufgrund unzureichenden Speichers nicht aktivieren RDS kann, bleiben deaktiviert.

Verbindung zu Ihrer Db2-DB-Instance herstellen

Nachdem Amazon RDS Ihre Amazon RDS for Db2-DB-Instance bereitgestellt hat, können Sie eine beliebige Standard-SQL-Client-Anwendung verwenden, um eine Verbindung mit der DB-Instance herzustellen. Da Amazon RDS ein verwalteter Service ist, können Sie sich nicht als`SYSADM`, `SYSCTRLSECADM`, oder anmelden`SYSMAINT`.

Sie können eine Verbindung zu einer DB-Instance herstellen, auf der das ausgeführt wird IBM Db2 Datenbank-Engine mithilfe von IBM Db2 CLP, IBM CLPPlus, DBeaver, oder IBM Db2 Data Management Console.

> **Note**
>
> Die Verbindung zu einer Db2-Datenbank kann fehlschlagen, wenn Ihre RDS for Db2-DB-Instance nicht über ausreichend Arbeitsspeicher verfügt. Weitere Informationen finden Sie unter the section called “Fehler bei der Datenbankverbindung”.

Themen

- Den Endpunkt Ihrer Amazon RDS for Db2-DB-Instance finden
- Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 CLP
- Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM CLPPlus
- Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit DBeaver
- Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 Data Management Console
- Überlegungen zu Sicherheitsgruppen mit Amazon RDS for Db2

Den Endpunkt Ihrer Amazon RDS for Db2-DB-Instance finden

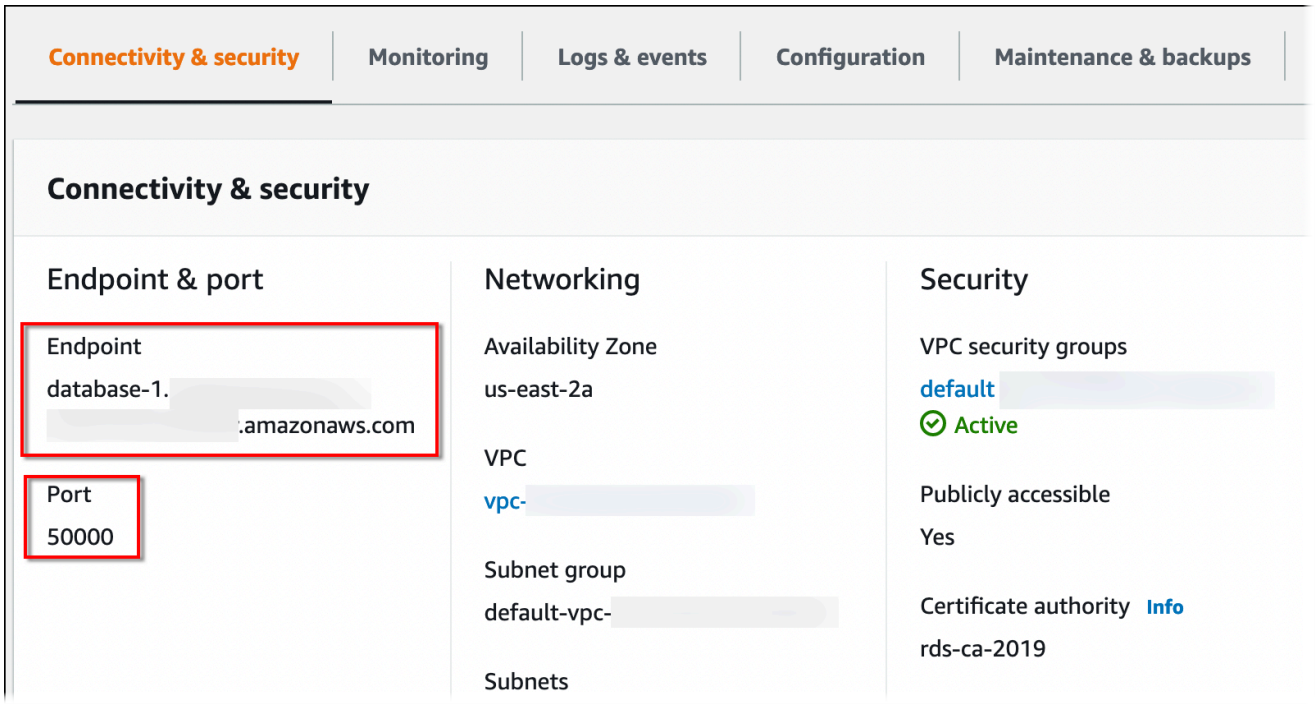
Jede Amazon RDS-DB-Instance hat einen Endpunkt und jeder Endpunkt hat einen DNS-Namen und eine Portnummer für die DB-Instance. Um mit einer SQL-Client-Anwendung eine Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance herzustellen, benötigen Sie den DNS-Namen und die Portnummer für Ihre DB-Instance.

Sie können den Endpunkt für eine DB-Instance finden, indem Sie den AWS Management Console oder den AWS CLI verwenden.

Konsole

Um den Endpunkt einer RDS for Db2-DB-Instance zu finden

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Konsole die Ihrer DB-Instance AWS-Region aus.
3. Suchen Sie den DNS-Namen und die Portnummer für Ihre RDS for Db2-DB-Instance.
   a. Wählen Sie Databases (Datenbanken) aus, um eine Liste Ihrer DB-Instances anzuzeigen.
   b. Wählen Sie den Namen der RDS for Db2-DB-Instance, um die Instance-Details anzuzeigen.
   c. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



AWS CLI

Führen Sie den describe-db-instancesBefehl aus, um den Endpunkt einer RDS for Db2-DB-Instance zu ermitteln. Ersetzen Sie im folgenden Beispiel *database-1* durch den Namen Ihrer DB-Instance.

FürLinux, odermacOS: Unix

```
aws rds describe-db-instances \
    --db-instance-identifier database-1 \
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' \
    --output json
```

Windows:

```
aws rds describe-db-instances ^
    --db-instance-identifier database-1 ^
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' ^
    --output json
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt. Die Linie `Address` in der Ausgabe enthält den DNS-Namen.

```
[
    {
        "DBInstanceIdentifier": "database-1",
        "DBName": "DB2DB",
        "Endpoint": {
            "Address": "database-1.123456789012.us-east-2.amazonaws.com",
            "Port": 50000,
            "HostedZoneId": "Z2OC4A7DETW6VH"
        }
    }
]
```

Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 CLP

Sie können ein Befehlszeilenprogramm verwenden, IBM Db2 CLP um beispielsweise eine Verbindung zu Amazon RDS für Db2-DB-Instances herzustellen. Dieses Hilfsprogramm ist Teil von. IBM Data Server Runtime Client Informationen zum Herunterladen des Clients finden Sie unter IBMData Server Client Packages Version 11.5 Mod 8 Fix Pack 0 im IBM Support. IBM Fix Central

Themen

- Terminologie

- Den Client installieren
- Herstellen einer Verbindung mit einer DB-Instance
- Problembehandlung bei Verbindungen zu Ihrer RDS for Db2-DB-Instance

Terminologie

Die folgenden Begriffe erläutern die Befehle, die beim Herstellen einer Verbindung mit Ihrer RDS for Db2-DB-Instance verwendet werden.

TCP/IP-Knoten für den Katalog

Dieser Befehl registriert einen entfernten Datenbankknoten bei einem lokalen Db2-Client, wodurch der Knoten für die Client-Anwendung zugänglich ist. Um einen Knoten zu katalogisieren, geben Sie Informationen wie den Hostnamen, die Portnummer und das Kommunikationsprotokoll des Servers an. Der katalogisierte Knoten stellt dann einen Zielserver dar, auf dem sich eine oder mehrere entfernte Datenbanken befinden. Weitere Informationen finden Sie unter CATALOG TCPIP/TCPIP4/TCPIP6 NODEBefehl in der IBM Db2 Dokumentation.

Katalog-Datenbank

Dieser Befehl registriert eine entfernte Datenbank bei einem lokalen Db2-Client, wodurch die Client-Anwendung auf die Datenbank zugreifen kann. Um eine Datenbank zu katalogisieren, geben Sie Informationen wie den Alias der Datenbank, den Knoten, auf dem sie sich befindet, und den Authentifizierungstyp an, der für die Verbindung mit der Datenbank erforderlich ist. Weitere Informationen finden Sie in der IBM Db2 Dokumentation unter CATALOG DATABASEBefehl.

Den Client installieren

Installieren Sie downloading the package for Linux anschließend den Client mit Root- oder Administratorrechten.

> **Note**
>
> Um den Client auf AIX oder zu installierenWindows, gehen Sie genauso vor, ändern Sie jedoch die Befehle für Ihr Betriebssystem.

Um den Client zu installieren Linux

1. Führen Sie den **./db2_install -f sysreq**Befehl aus und wählen **yes**Sie, ob Sie die Lizenz akzeptieren möchten.
2. Wählen Sie den Ort, an dem der Client installiert werden soll.
3. Führen Sie **clientInstallDir/instance/db2icrt -s client*instance_name*** aus. Ersetzen Sie *instance_name* durch einen gültigen Betriebssystembenutzer auf. Linux In Linux ist der Name der Db2-DB-Instance an den Betriebssystem-Benutzernamen gebunden.

   Mit diesem Befehl wird ein **sqllib**Verzeichnis im Home-Verzeichnis des angegebenen Benutzers erstellt. Linux

Herstellen einer Verbindung mit einer DB-Instance

Um eine Verbindung zu Ihrer RDS for Db2-DB-Instance herzustellen, benötigen Sie deren DNS-Namen und Portnummer. Informationen darüber, wie Sie sie finden, finden Sie unterErmitteln des Endpunkts. Sie müssen auch den Datenbanknamen, den Master-Benutzernamen und das Master-Passwort kennen, die Sie bei der Erstellung Ihrer RDS for Db2-DB-Instance definiert haben. Weitere Informationen darüber, wie Sie sie finden, finden Sie unterErstellen einer DB-Instance.

Um eine Verbindung zu einer RDS for Db2-DB-Instance herzustellen, verwenden Sie IBM Db2 CLP

1. Melden Sie sich mit dem Benutzernamen an, den Sie bei der IBM Db2 CLP Client-Installation angegeben haben.
2. Katalogisieren Sie Ihre RDS for Db2-DB-Instance. Ersetzen Sie im folgenden Beispiel *node_name*, *dns_name* und *port* durch einen Namen für den Knoten im lokalen Katalog, den DNS-Namen für Ihre DB-Instance und die Portnummer.

   ```
   db2 catalog TCPIP node node_name remote dns_name server port
   ```

   Beispiel

   ```
   db2 catalog TCPIP node remnode remote database-1.123456789012.us-
   east-1.amazonaws.com server 50000
   ```
3. Katalogisieren Sie die `rdsadmin` Datenbank und Ihre Datenbank. Auf diese Weise können Sie eine Verbindung zur `rdsadmin` Datenbank herstellen, um einige Verwaltungsaufgaben mithilfe

von gespeicherten Amazon RDS-Prozeduren auszuführen. Weitere Informationen finden Sie unter Verwaltung Ihrer RDS für Db2-DB-Instance.

Ersetzen Sie im folgenden Beispiel *database_alias*, *node_name und database_name* durch Aliase für diese Datenbank, den Namen des im vorherigen Schritt definierten Knotens und den Namen Ihrer Datenbank. `server_encrypt`verschlüsselt Ihren Benutzernamen und Ihr Passwort über das Netzwerk.

```
db2 catalog database rdsadmin [ as database_alias ] at node node_name
 authentication server_encrypt

db2 catalog database database_name [ as database_alias ] at node node_name
 authentication server_encrypt
```

Beispiel

```
db2 catalog database rdsadmin at node remnode authentication server_encrypt

db2 catalog database testdb as rdsdb2 at node remnode authentication server_encrypt
```

4. Connect zu Ihrer RDS for Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_alias, master_username und master_password durch den Namen Ihrer Datenbank, den Master-Benutzernamen und das Master-Passwort* Ihrer RDS for Db2-DB-Instance.

```
db2 connect to rds_database_alias user master_username using master_password
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
Database Connection Information

    Database server        = DB2/LINUXX8664 11.5.9.0
    SQL authorization ID   = ADMIN
    Local database alias   = TESTDB
```

5. Abfragen ausführen und Ergebnisse anzeigen. Das folgende Beispiel zeigt eine SQL-Anweisung, die die von Ihnen erstellte Datenbank auswählt.

```
db2 "select current server from sysibm.dual"
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
1
    ------------------
    TESTDB

    1 record(s) selected.
```

Problembehandlung bei Verbindungen zu Ihrer RDS for Db2-DB-Instance

Wenn Sie die folgende `NULLID` Fehlermeldung erhalten, deutet dies in der Regel darauf hin, dass Ihre Client- und RDS for Db2-Serverversionen nicht übereinstimmen. Informationen zu den unterstützten Db2-Client-Versionen finden Sie in der Dokumentation unter Unterstützte Kombinationen von Clients, Treibern und Serverstufen. IBM Db2

```
db2 "select * from syscat.tables"
SQL0805N Package "NULLID.SQLC2O29 0X4141414141454A69" was not found.
SQLSTATE=51002
```

Nachdem Sie diesen Fehler erhalten haben, müssen Sie Pakete von Ihrem älteren Db2-Client an eine von RDS für Db2 unterstützte Db2-Serverversion binden.

Um Pakete von einem älteren Db2-Client an einen neueren Db2-Server zu binden

1. Suchen Sie die Bindungsdateien auf dem Client-Computer. In der Regel befinden sich diese Dateien im Verzeichnis bnd des Installationspfads des Db2-Clients und haben die Erweiterung .bnd.
2. Connect zum Db2-Server her. Ersetzen Sie im folgenden Beispiel *database_name durch den Namen Ihres Db2-Servers*. *Ersetzen Sie master_username und master_password durch Ihre Informationen.* Dieser Benutzer hat Autorität. `DBADM`

   ```
   db2 connect to database_name user master_username using master_password
   ```
3. Führen Sie den `bind` Befehl aus, um die Pakete zu binden.

   a. Navigieren Sie zu dem Verzeichnis, in dem sich die Bind-Dateien auf dem Client-Computer befinden.

   b. Führen Sie den `bind` Befehl für jede Datei aus.

Die folgenden Optionen sind erforderlich:

- `blocking all`— Bindet alle Pakete in der Bind-Datei in einer einzigen Datenbankanfrage.
- `grant public`— Erteilt die Erlaubnis`public`, das Paket auszuführen.
- `sqlerror continue`— Gibt an, dass der `bind` Prozess auch dann fortgesetzt wird, wenn Fehler auftreten.

Weitere Informationen zum `bind` Befehl finden Sie in der IBM Db2 Dokumentation zum BINDBefehl.

4. Stellen Sie sicher, dass die Bindung erfolgreich war, indem Sie entweder die `syscat.package` Katalogansicht abfragen oder die nach dem `bind` Befehl zurückgegebene Nachricht überprüfen.

Weitere Informationen finden Sie unter Bindungsdatei und Paketnamenliste für DB2 v11.5 im Support. IBM

Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM CLPPlus

Sie können ein Hilfsprogramm verwenden wie IBM CLPPlus um eine Verbindung zu einer Amazon RDS for Db2-DB-Instance herzustellen. Dieses Hilfsprogramm ist Teil von IBM Data Server Runtime Client. Um den Client herunterzuladen von IBM Fix Central, siehe IBM Datenserver-Client-Pakete Version 11.5 Mod 8 Fix Pack 0 in IBM Support.

> **Important**
>
> Wir empfehlen dir zu laufen IBM CLPPlus auf einem Betriebssystem, das grafische Benutzeroberflächen unterstützt, wie macOS, Windows, oder Linux mit Desktop. Wenn kopflos läuft Linux, benutze den Schalter -nw mit CLPPlus Befehle.

Themen

- Den Client installieren
- Herstellen einer Verbindung mit einer DB-Instance

Den Client installieren

Nach dem Herunterladen des Pakets für Linux, installieren Sie den Client.

> **Note**
>
> Um den Client zu installieren AIX or Windows, gehen Sie genauso vor, ändern Sie jedoch die Befehle für Ihr Betriebssystem.

Um den Client zu installieren Linux

1. Führen Sie **`./db2_install`**.
2. Führen Sie Folgendes aus:**`clientInstallDir/instance/db2icrt -s client`** ***`instance_name`***. Ersetzen *`instance_name`* mit einem gültigen Betriebssystembenutzer auf Linux. In Linux, der Name der Db2-DB-Instance ist an den Benutzernamen des Betriebssystems gebunden.

   Dieser Befehl erstellt ein **`sqllib`**Verzeichnis im Home-Verzeichnis des angegebenen Benutzers unter Linux.

Herstellen einer Verbindung mit einer DB-Instance

Um eine Verbindung RDS zu Ihrer DB2-DB-Instance herzustellen, benötigen Sie deren DNS Namen und Portnummer. Informationen darüber, wie Sie sie finden, finden Sie unterErmitteln des Endpunkts. Sie müssen auch den Datenbanknamen, den Master-Benutzernamen und das Master-Passwort kennen, die Sie bei der Erstellung Ihrer RDS for Db2-DB-Instance definiert haben. Weitere Informationen darüber, wie Sie sie finden, finden Sie unterErstellen einer DB-Instance.

Um eine Verbindung zu einer RDS for Db2-DB-Instance herzustellen, verwenden Sie IBM CLPPlus

1. Überprüfen Sie die Befehlssyntax. Ersetzen Sie im folgenden Beispiel *`clientDir`* durch den Ort, an dem der Client installiert ist.

   ```
   cd clientDir/bin
       ./clpplus -h
   ```

2. Konfigurieren Sie Ihren Db2-Server. Ersetzen Sie im folgenden Beispiel *`dsn_name`*, *`database_name`*, *`endpoint`*, und *`port`* durch den DSN Namen, den Datenbanknamen, den

Endpunkt und den Port RDS für Ihre DB2-DB-Instance. Weitere Informationen finden Sie unter Den Endpunkt Ihrer Amazon RDS for Db2-DB-Instance finden.

```
db2cli writecfg add -dsn dsn_name -database database_name -host endpoint -port port
 -parameter "Authentication=SERVER_ENCRYPT"
```

3. Connect zu Ihrer RDS DB2-DB-Instance her. Ersetzen Sie im folgenden Beispiel *master_username* and *dsn_name* durch den Hauptbenutzernamen und DSN -namen.

```
./clpplus -nw master_username@dsn_name
```

4. A Java Shell Fenster öffnet sich. Geben Sie das Master-Passwort RDS für Ihre DB2-DB-Instance ein.

> **Note**
>
> Wenn ein Java Shell Das Fenster wird nicht geöffnet. Führen Sie **`./clpplus -nw`** es aus, um dasselbe Befehlszeilenfenster zu verwenden.

```
Enter password: *********
```

Es wird eine Verbindung hergestellt und eine Ausgabe erzeugt, die dem folgenden Beispiel ähnelt:

```

Database Connection Information :
---------------------------------
Hostname = database-1.abcdefghij.us-east-1.rds.amazonaws.com
Database server = DB2/LINUXX8664  SQL110590
SQL authorization ID = admin
Local database alias = DB2DB
Port = 50000
```

5. Abfragen ausführen und Ergebnisse anzeigen. Das folgende Beispiel zeigt eine SQL Anweisung, die die von Ihnen erstellte Datenbank auswählt.

```
SQL > select current server from sysibm.dual;
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
1
    --------------------
    DB2DB
    SQL>
```

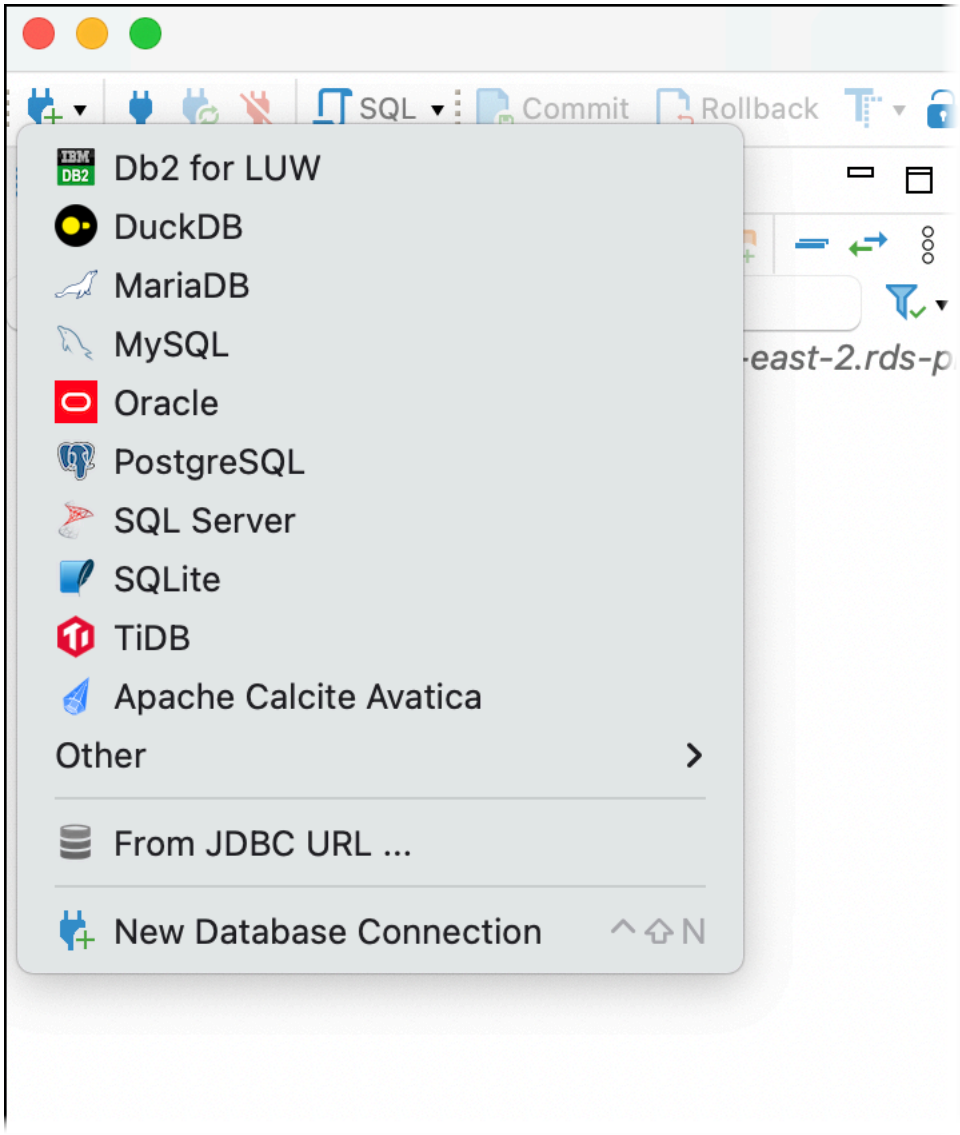
Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit DBeaver

Sie können Tools von Drittanbietern verwenden, DBeaver um beispielsweise eine Verbindung zu Amazon RDS für Db2-DB-Instances herzustellen. Informationen zum Herunterladen dieses Dienstprogramms finden Sie unter DBeaverCommunity.

Um eine Verbindung zu Ihrer RDS for Db2-DB-Instance herzustellen, benötigen Sie deren DNS-Namen und Portnummer. Informationen darüber, wie Sie sie finden, finden Sie unterErmitteln des Endpunkts. Sie müssen auch den Datenbanknamen, den Master-Benutzernamen und das Master-Passwort kennen, die Sie bei der Erstellung Ihrer RDS for Db2-DB-Instance definiert haben. Weitere Informationen darüber, wie Sie sie finden, finden Sie unterErstellen einer DB-Instance.

Um eine Verbindung zu einer RDS for Db2-DB-Instance herzustellen, verwenden Sie DBeaver

1. Starten DBeaver.
2. Wählen Sie in der Werkzeugleiste das Symbol „Neue Verbindung“ und dann „Db2 for LUW“.



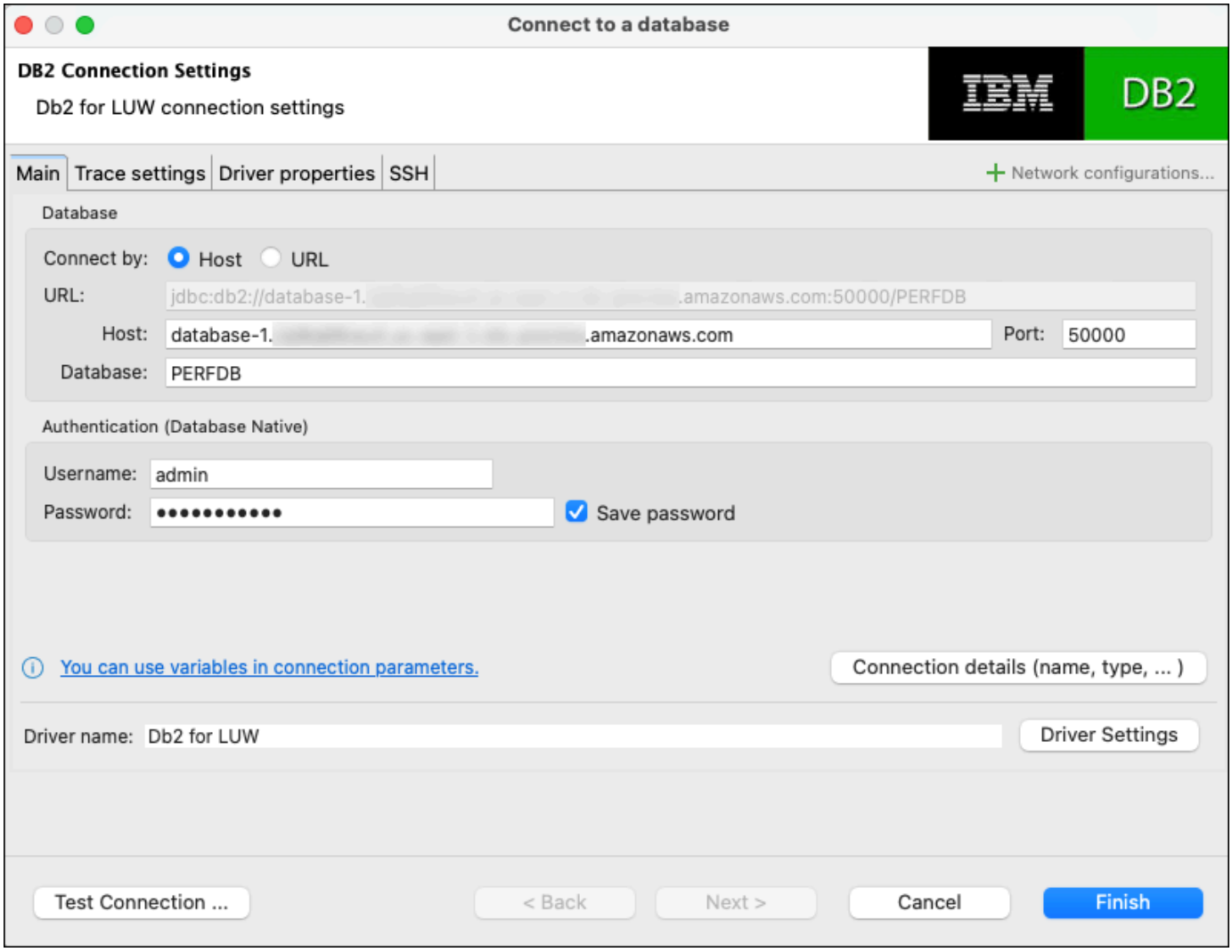
3. Geben Sie im Fenster Mit einer Datenbank verbinden Informationen für Ihre RDS for Db2-DB-Instance ein.

   a. Geben Sie die folgenden Informationen ein:

   - Geben Sie für Host den DNS-Namen der DB-Instance ein.
   - Geben Sie unter Port die Portnummer für die DB-Instance ein.
   - Geben Sie für Datenbank den Namen der Datenbank ein.
   - Geben Sie in das Feld Username (Benutzername) den Namen des Datenbankadministrators für Ihre DB-Instance ein.
   - Geben Sie unter Passwort das Passwort des Datenbankadministrators für die DB-Instance ein.

b. Wählen Sie Passwort speichern aus.

c. Wählen Sie Treibereinstellungen.



4. Geben Sie im Fenster „Treiber bearbeiten“ zusätzliche Sicherheitseigenschaften an.

a. Wählen Sie die Registerkarte Treibereigenschaften.

b. Fügen Sie zwei Benutzereigenschaften hinzu.

i. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Neue Eigenschaft hinzufügen.

ii. Fügen Sie als Eigenschaftsname EncryptionAlgorithm hinzu, und wählen Sie dann OK.

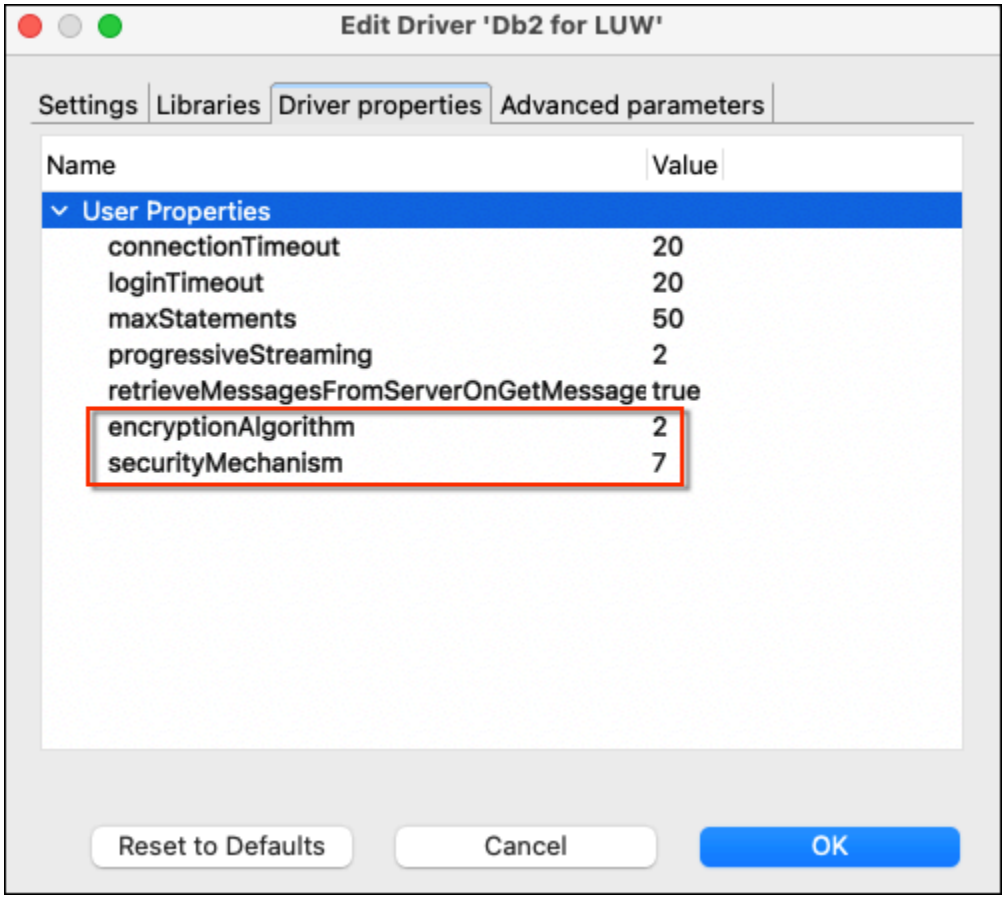
iii. Wählen Sie die Zeile EncryptionAlgorithm aus, wählen Sie die Spalte Wert aus und fügen Sie 2 hinzu.

iv. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Neue Eigenschaft hinzufügen.

v. Fügen Sie als Eigenschaftsname SecurityMechanism hinzu, und wählen Sie dann OK.

vi. Wählen Sie die Zeile SecurityMechanism aus, wählen Sie die Spalte Wert aus und fügen Sie 7 hinzu.

c. Wählen Sie OK aus.

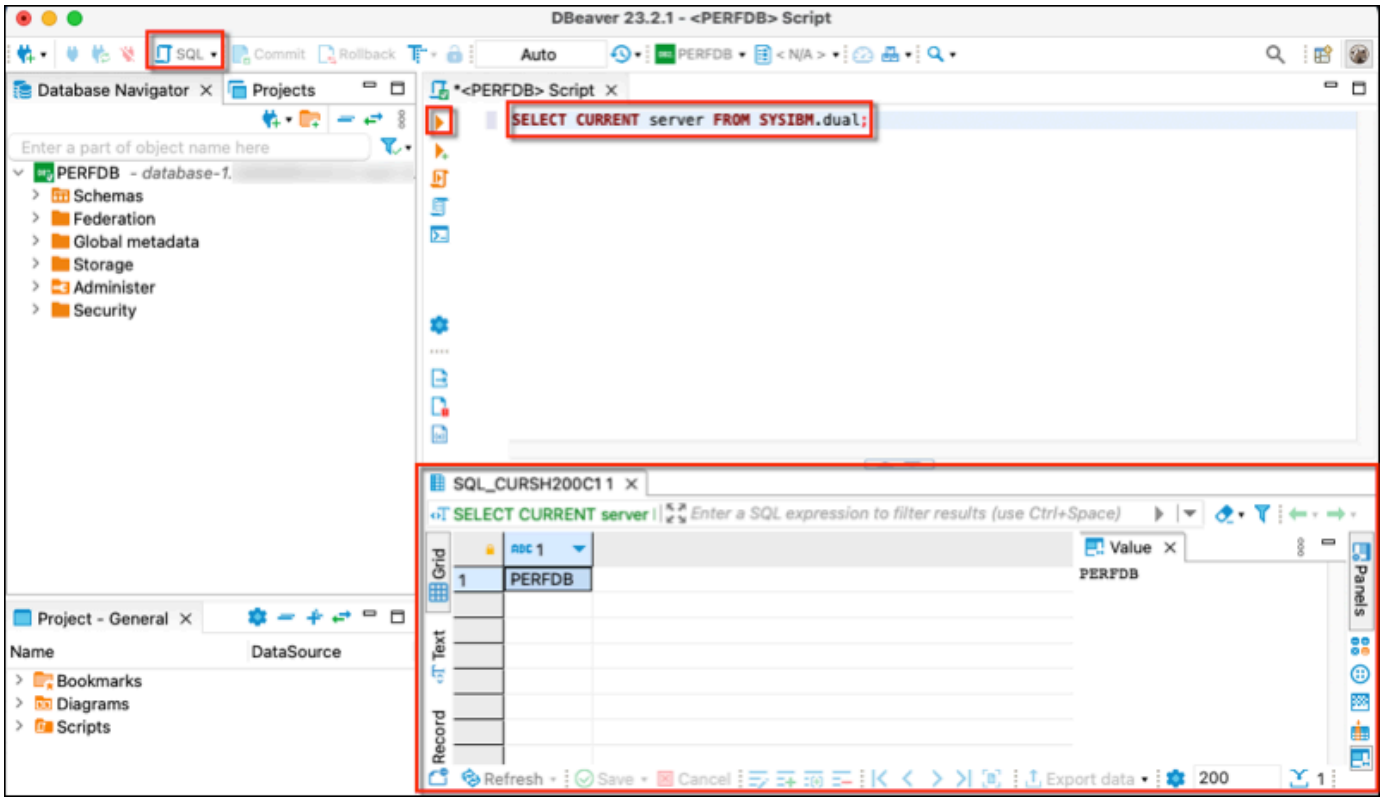


5. Wählen Sie im Fenster Mit einer Datenbank verbinden die Option Verbindung testen aus. Wenn auf Ihrem Computer kein DB2 JBDC-Treiber installiert ist, wird der Treiber automatisch heruntergeladen.
6. Wählen Sie OK aus.
7. Wählen Sie Finish (Abschließen).
8. Wählen Sie auf der Registerkarte Datenbanknavigation den Namen der Datenbank aus. Sie können jetzt Objekte erkunden.

Sie sind jetzt bereit, SQL-Befehle auszuführen.

Um SQL-Befehle auszuführen und die Ergebnisse anzusehen

1. Wählen Sie im oberen Menü SQL. Dadurch wird ein SQL-Skriptfenster geöffnet.
2. Geben Sie im Skriptfenster einen SQL-Befehl ein.
3. Um den Befehl auszuführen, wählen Sie die Schaltfläche „SQL-Abfrage ausführen“.
4. Sehen Sie sich im Bereich „SQL-Ergebnisse“ die Ergebnisse Ihrer SQL-Abfragen an.



Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 Data Management Console

Sie können eine Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance herstellen mit IBM Db2 Data Management Console. IBM Db2 Data Management Console kann mehrere RDS für Db2-DB-Instances verwalten und überwachen.

> **Note**
>
> Du musst einen Amazon haben EC2 Linux oder Windows-Computer, der sich in derselben VPC Sicherheitsgruppe wie Ihre RDS DB2-DB-Instance befindet. Die Sicherheitsgruppe VPC und steuert die Verbindung zu Ihrer DB-Instance über das interne Netzwerk.

IBM Db2 Data Management Console benötigt eine Repository-Db2-Datenbank zum Speichern von Metadaten und Leistungsmetriken, kann aber nicht automatisch eine Repository-Datenbank RDS für Db2 erstellen. Stattdessen müssen Sie zunächst eine Repository-Datenbank erstellen, um eine oder mehrere RDS Db2-DB-Instances zu überwachen. Dann können Sie installieren IBM Db2 Data Management Console und stellen Sie eine Verbindung zu Ihrer RDS Db2-DB-Instance her mit IBM Db2 Data Management Console.

Themen

- Schritt 1: Erstellen einer Repository-Datenbank zur Überwachung von DB-Instances
- Schritt 2: Installation und Einrichtung IBM Db2 Data Management Console
- Schritt 3: Konfiguration der Repository-Datenbank und Verbindung zu RDS vier Db2-DB-Instances
- Die Verwendung von IBM Db2 Data Management Console

Schritt 1: Erstellen einer Repository-Datenbank zur Überwachung von DB-Instances

Sie können eine vorhandene DB-Instance mit der richtigen Größe RDS für Db2 als Repository verwenden für IBM Db2 Data Management Console um andere RDS für Db2-DB-Instances zu überwachen. Da der Admin-Benutzer jedoch nicht `SYSCTRL` befugt ist, Pufferpools und Tablespaces zu erstellen, verwendet er IBM Db2 Data Management Console Die Erstellung eines Repositorys zur Erstellung einer Repository-Datenbank schlägt fehl. Stattdessen müssen Sie eine Repository-Datenbank erstellen. Diese Repository-Datenbank überwacht Ihre RDS vier DB2-DB-Instances.

Sie können eine Repository-Datenbank auf zwei verschiedene Arten erstellen. Sie können eine RDS For-Db2-Datenbank und dann manuell einen Pufferpool, einen Benutzer-Tablespace und einen temporären System-Tablespace erstellen. Oder Sie können eine separate EC2 Amazon-Instance erstellen, um eine zu hosten IBM Db2 Data Management Console Repository-Datenbank.

Themen

- Manuelles Erstellen eines Pufferpools, eines Benutzer-Tablespaces und eines temporären System-Tablespaces

- Erstellen einer EC2 Amazon-Instance zum Hosten einer IBM Db2 Data Management Console Repository

Manuelles Erstellen eines Pufferpools, eines Benutzer-Tablespaces und eines temporären System-Tablespaces

Um einen Pufferpool, einen Benutzer-Tablespace und einen temporären System-Tablespace zu erstellen

1. Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Erstellen Sie einen Pufferpool für IBM Db2 Data Management Console. Ersetzen Sie im folgenden Beispiel *`database_name`* durch den Namen des Repositorys, für das Sie erstellt haben IBM Db2 Data Management Console um Ihre RDS vier Db2-DB-Instances zu überwachen.

   ```
   db2 "call rdsadmin.create_bufferpool('database_name',
        'BP4CONSOLE', 1000, 'Y', 'Y', 32768)"
   ```

3. Erstellen Sie einen Benutzer-Tablespace für IBM Db2 Data Management Console. Ersetzen Sie im folgenden Beispiel *`database_name`* durch den Namen des Repositorys, für das Sie erstellt haben IBM Db2 Data Management Console um Ihre RDS vier Db2-DB-Instances zu überwachen.

   ```
   db2 "call rdsadmin.create_tablespace('database_name',
        'TS4CONSOLE', 'BP4CONSOLE', 32768)"
   ```

4. Erstellen Sie einen temporären System-Tablespace für IBM Db2 Data Management Console. Ersetzen Sie im folgenden Beispiel *`database_name`* durch den Namen des Repositorys, für das Sie erstellt haben IBM Db2 Data Management Console um Ihre RDS vier Db2-DB-Instances zu überwachen.

   ```
   db2 "call rdsadmin.create_tablespace('database_name',
       'TS4CONSOLE_TEMP', 'BP4CONSOLE', 32768, 0, 0, 'S')"
   ```

Sie sind jetzt bereit für die Installation IBM Db2 Data Management Console. Weitere Informationen zur Installation und Einrichtung finden Sie unterSchritt 2: Installation und Einrichtung IBM Db2 Data Management Console.

Erstellen einer EC2 Amazon-Instance zum Hosten einer IBM Db2 Data Management Console Repository

Sie können eine separate Amazon Elastic Compute Cloud (AmazonEC2) -Instance zum Hosten einer IBM Db2 Data Management Console Repository. Informationen zum Erstellen einer EC2 Amazon-Instance finden Sie unter Tutorial: Erste Schritte mit Amazon EC2 Linux Instanzen im EC2Amazon-Benutzerhandbuch.

Schritt 2: Installation und Einrichtung IBM Db2 Data Management Console

Nachdem Sie einen Pufferpool, einen Benutzer-Tablespace und einen temporären System-Tablespace erstellt haben, sind Sie bereit für die Installation und Einrichtung IBM Db2 Data Management Console.

> **Important**
>
> Du musst einen Amazon haben EC2 Linux oder Windows-Computer, der sich in derselben VPC Sicherheitsgruppe wie Ihre RDS DB2-DB-Instance befindet. Die Sicherheitsgruppe VPC und steuert die Verbindung zu Ihrer DB-Instance über das interne Netzwerk. Außerdem müssen Sie bereits eine Repository-Datenbank für erstellt haben IBM Db2 Data Management Console.

Zur Installation und Einrichtung IBM Db2 Data Management Console

1. Herunterladen IBM Db2 Data Management Console von IBM Db2 Data Management Console Version 3.1x wird auf der IBM Support-Website veröffentlicht.
2. Installieren IBM Db2 Data Management Console.
3. Öffnen IBM Db2 Data Management Console und verwenden Sie die IP-Adresse Ihres EC2 Amazon-Computers und die Portnummer, die Sie für die HTTP HTTPS Oder-Verbindung zu Ihrer EC2 Amazon-Instance verwendet haben. Verwenden Sie beispielsweise `http://xx.xx.xx.xx:11080` oder`https://xx.xx.xx.xx.11081`. *xx.xx.xx.xx*Ersetzen Sie es durch die IP-Adresse Ihres EC2 Amazon-Computers. `11080`und `11081` sind die Standardports für HTTP und HTTPS Verbindungen.

4. (Optional) Wenn Sie Port 80 oder 443 auf Ihrer EC2 Amazon-Instance verwenden möchten, können Sie entweder Apache httpd oder einen HTTP Nginx-Server als Proxy für die IBM Db2 Data Management Console Port entweder auf Port 80 oder 443. Weitere Informationen finden Sie unter Apache HTTP Server Project und auf der Nginx-Website.

   Um eine Verbindung zu zuzulassen IBM Db2 Data Management Console, müssen Sie die Regeln für eingehende Nachrichten in Ihrer Sicherheitsgruppe bearbeiten. Wenn Sie einen Proxy verwenden, ändern Sie den TCP /IP-Port 80 oder 443, um auf den IBM Db2 Data Management Console Anschlüsse. Wenn Sie keinen Proxy verwenden, ändern Sie den TCP /IP-Port 80 oder 443 auf die Standardports 11080 (HTTP) oder 11081 (). HTTPS

Sie sind jetzt bereit, sich anzumelden bei IBM Db2 Data Management Console um die Repository-Datenbank zu konfigurieren und eine Verbindung zu Ihren vier RDS Db2-DB-Instances herzustellen. Weitere Informationen finden Sie unter Konfiguration der Repository-Datenbank und Verbindung zu DB-Instances.

Schritt 3: Konfiguration der Repository-Datenbank und Verbindung zu RDS vier Db2-DB-Instances

Wenn Sie zum ersten Mal eine Verbindung zur Repository-Datenbank herstellen, IBM Db2 Data Management Console konfiguriert das Repository automatisch. Nachdem die Repository-Datenbank konfiguriert wurde, können Sie Datenbankverbindungen zu hinzufügen IBM Db2 Data Management Console.

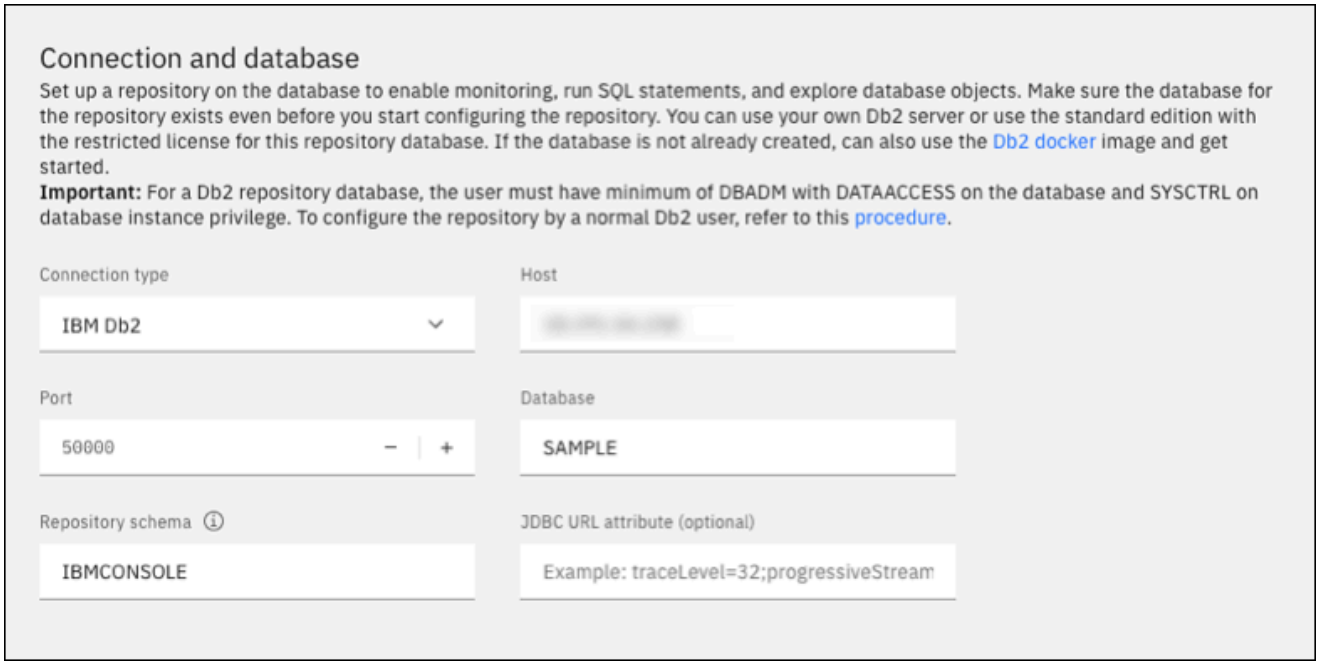
Um eine Verbindung RDS zu Ihrer DB2-DB-Instance herzustellen, benötigen Sie deren DNS Namen und Portnummer. Informationen darüber, wie Sie sie finden, finden Sie unterErmitteln des Endpunkts. Sie müssen auch den Datenbanknamen, den Master-Benutzernamen und das Master-Passwort kennen, die Sie bei der Erstellung Ihrer RDS for Db2-DB-Instance definiert haben. Weitere Informationen darüber, wie Sie sie finden, finden Sie unterErstellen einer DB-Instance. Wenn Sie eine Verbindung über das Internet herstellen, lassen Sie den Datenverkehr zum Datenbankport zu. Weitere Informationen finden Sie unter Erstellen einer DB-Instance.

Um eine Verbindung zu vier RDS Db2-DB-Instances herzustellen mit IBM Db2 Data Management Console

1. Loggen Sie sich ein bei IBM Db2 Data Management Console mit den Anmeldeinformationen, die Sie bei der Installation festgelegt haben.
2. Konfigurieren Sie das Repository.

a. Geben Sie im Abschnitt Verbindung und Datenbank die folgenden Informationen RDS für Ihre DB2-DB-Instance ein:

- Geben Sie für Host den DNS Namen der DB-Instance ein.
- Geben Sie unter Port die Portnummer für die DB-Instance ein.
- Geben Sie für Datenbank den Namen der Datenbank ein.



b. Geben Sie im Abschnitt Sicherheit und Anmeldeinformationen die folgenden Informationen RDS für Ihre DB2-DB-Instance ein:

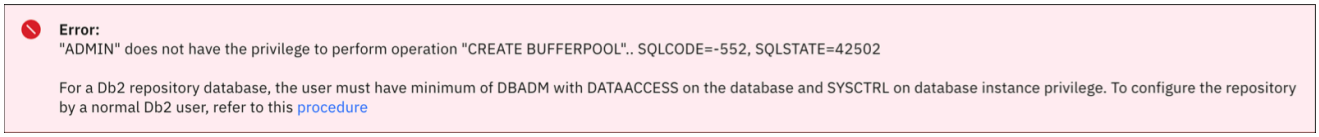
- Wählen Sie als Sicherheitstyp die Option Verschlüsselter Benutzer und Passwort aus.
- Geben Sie in das Feld Username (Benutzername) den Namen des Datenbankadministrators für Ihre DB-Instance ein.
- Geben Sie unter Passwort das Passwort des Datenbankadministrators für die DB-Instance ein.

c. Wählen Sie Test connection (Verbindung testen) aus.

> **Note**
>
> Wenn die Verbindung nicht erfolgreich ist, überprüfen Sie anhand der Regeln für eingehenden Datenverkehr in Ihrer Sicherheitsgruppe, dass der

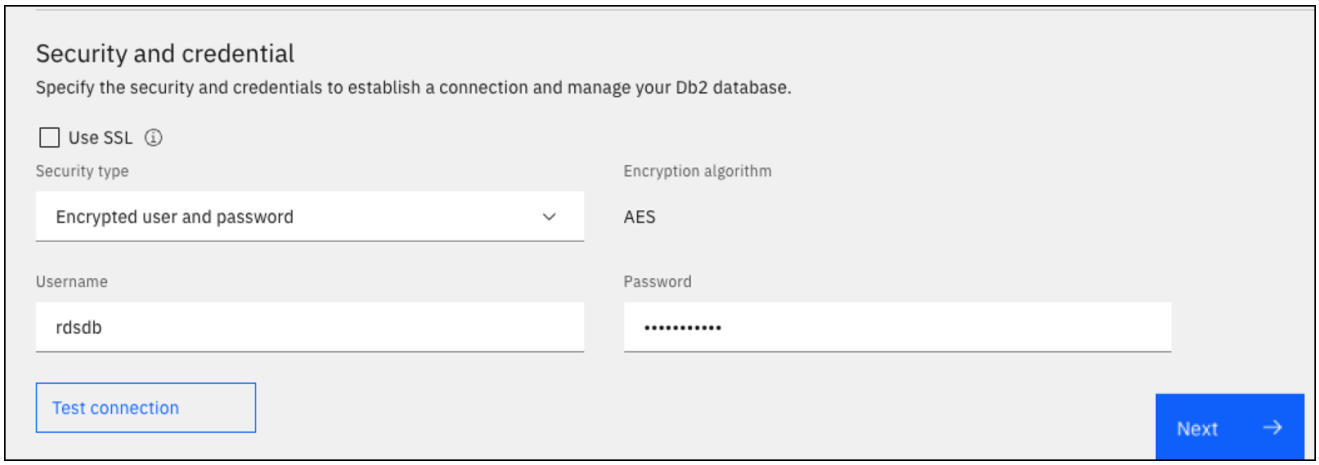
Datenbankport geöffnet ist. Weitere Informationen finden Sie unter Überlegungen zu Sicherheitsgruppen mit Amazon RDS for Db2.

Wenn Sie in Db2 nicht manuell einen Pufferpool, einen Benutzer-Tablespace und einen temporären System-Tablespace erstellt haben, wird möglicherweise die folgende Fehlermeldung angezeigt: RDS



Stellen Sie sicher, dass Sie eine Puffertabelle, einen Tablespace und Objekte für ein erstellt haben IBM Db2 Data Management Console Repository zur Überwachung Ihrer RDS Db2-DB-Instance. Oder Sie können eine Amazon EC2 Db2-DB-Instance verwenden, um eine zu hosten IBM Db2 Data Management Console Repository zur Überwachung Ihrer RDS eigenen Db2-DB-Instance. Weitere Informationen finden Sie unter Schritt 1: Erstellen einer Repository-Datenbank zur Überwachung von DB-Instances.

d. Nachdem Sie Ihre Verbindung erfolgreich getestet haben, wählen Sie Weiter.



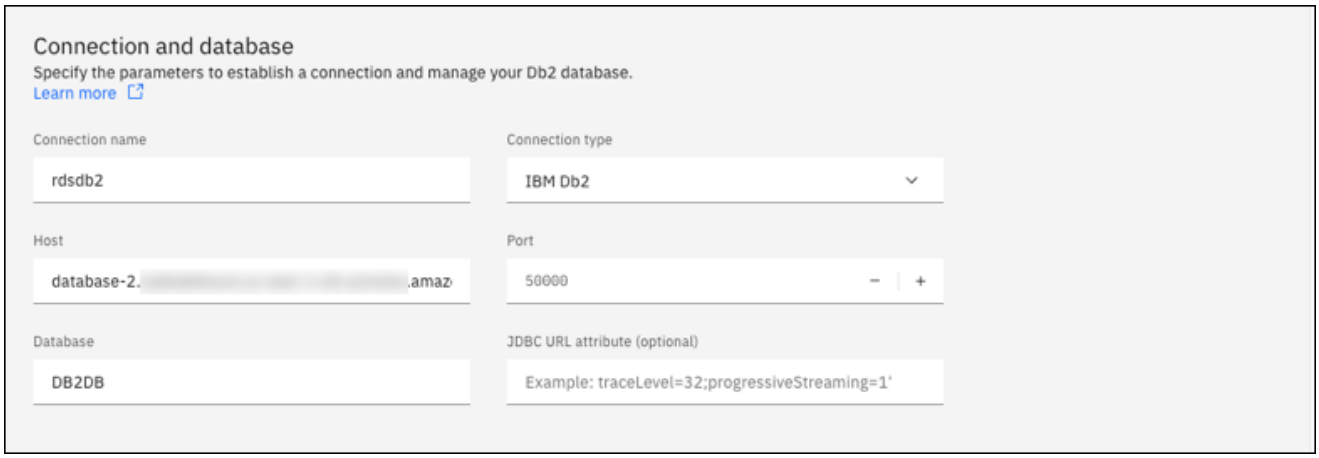
Wenn IBM Db2 Data Management Console findet dann den Pufferpool, den Benutzer-Tablespace und den temporären System-Tablespace in der RDS for Db2-DB-Instance IBM Db2 Data Management Console konfiguriert automatisch die Repository-Datenbank. Wenn Sie Ihre Db2-Instance auf Ihrer EC2 Amazon-Instance als Repository-Datenbank verwenden, IBM Db2 Data Management Console erstellt automatisch den Pufferpool und andere Objekte.

3. Wählen Sie im Anmeldefenster Set Statistics Event Monitor die Option Next aus.

4. (Optional) Neue Verbindung hinzufügen. Wenn Sie für die Verwaltung und Überwachung eine andere DB-Instance RDS für DB2 verwenden möchten, fügen Sie eine Verbindung zu einer DB-Instance hinzu, die kein Repository RDS für Db2 ist.

   a. Geben Sie im Abschnitt Verbindung und Datenbank die folgenden Informationen für die DB2-DB-Instance ein, die RDS für die Verwaltung und Überwachung verwendet werden soll:

      - Geben Sie als Verbindungsname den Db2-Datenbank-Identifier ein.
      - Geben Sie für Host den DNS Namen der DB-Instance ein.
      - Geben Sie unter Port die Portnummer für die DB-Instance ein.
      - Geben Sie für Datenbank den Namen der Datenbank ein.



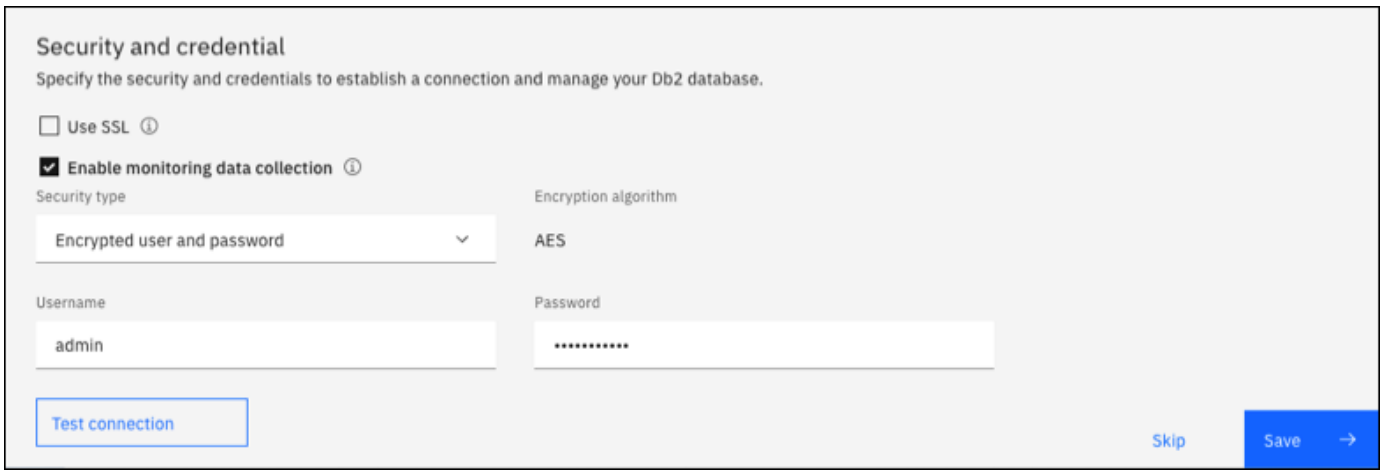
   b. Wählen Sie im Abschnitt Sicherheit und Anmeldeinformationen die Option Erfassung von Überwachungsdaten aktivieren aus.

   c. Geben Sie die folgenden Informationen RDS für Ihre DB2-DB-Instance ein:

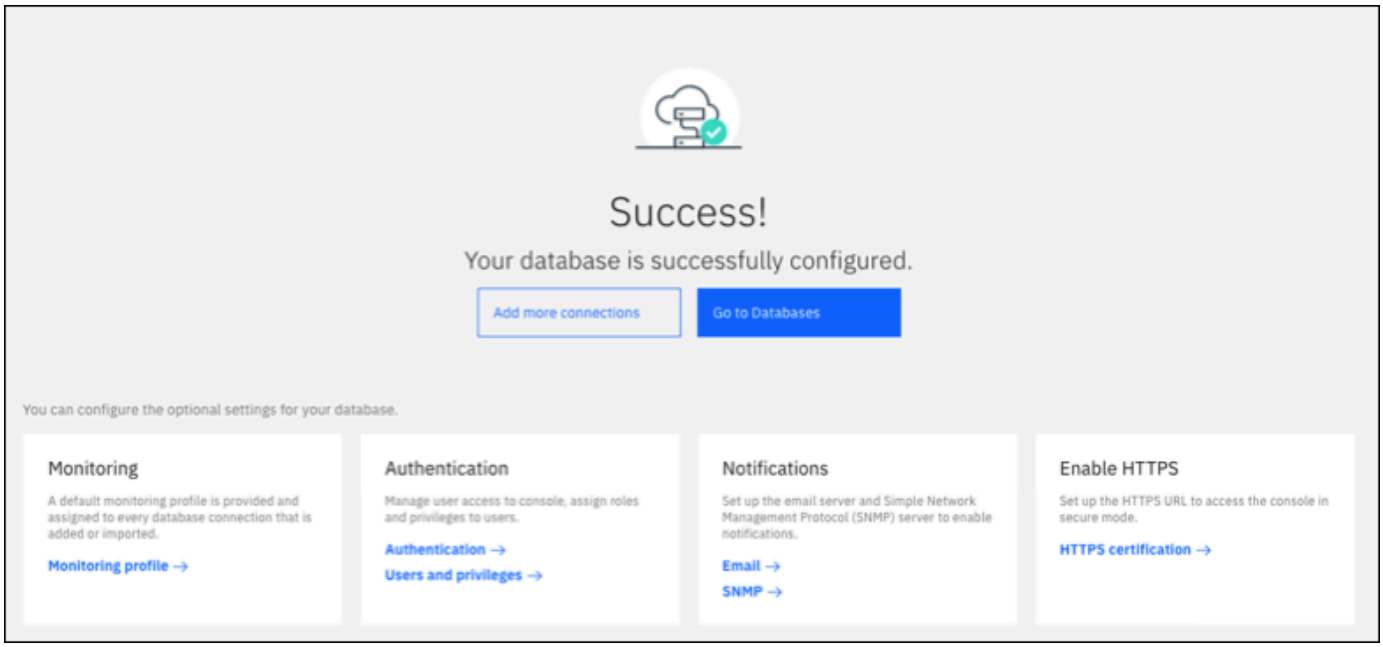
      - Geben Sie in das Feld Username (Benutzername) den Namen des Datenbankadministrators für Ihre DB-Instance ein.
      - Geben Sie unter Passwort das Passwort des Datenbankadministrators für die DB-Instance ein.

   d. Wählen Sie Test connection (Verbindung testen) aus.

   e. Nachdem Sie Ihre Verbindung erfolgreich getestet haben, wählen Sie Speichern.



Nachdem die Verbindung hinzugefügt wurde, erscheint ein Fenster, das dem folgenden ähnelt. Dieses Fenster zeigt an, dass Ihre Datenbank erfolgreich konfiguriert wurde.



5. Wählen Sie Gehe zu Datenbanken. Ein Datenbankfenster, das dem folgenden ähnelt, wird angezeigt. Dieses Fenster ist ein Dashboard, das Metriken, Status und Verbindungen anzeigt.



Sie können jetzt mit der Verwendung beginnen IBM Db2 Data Management Console.

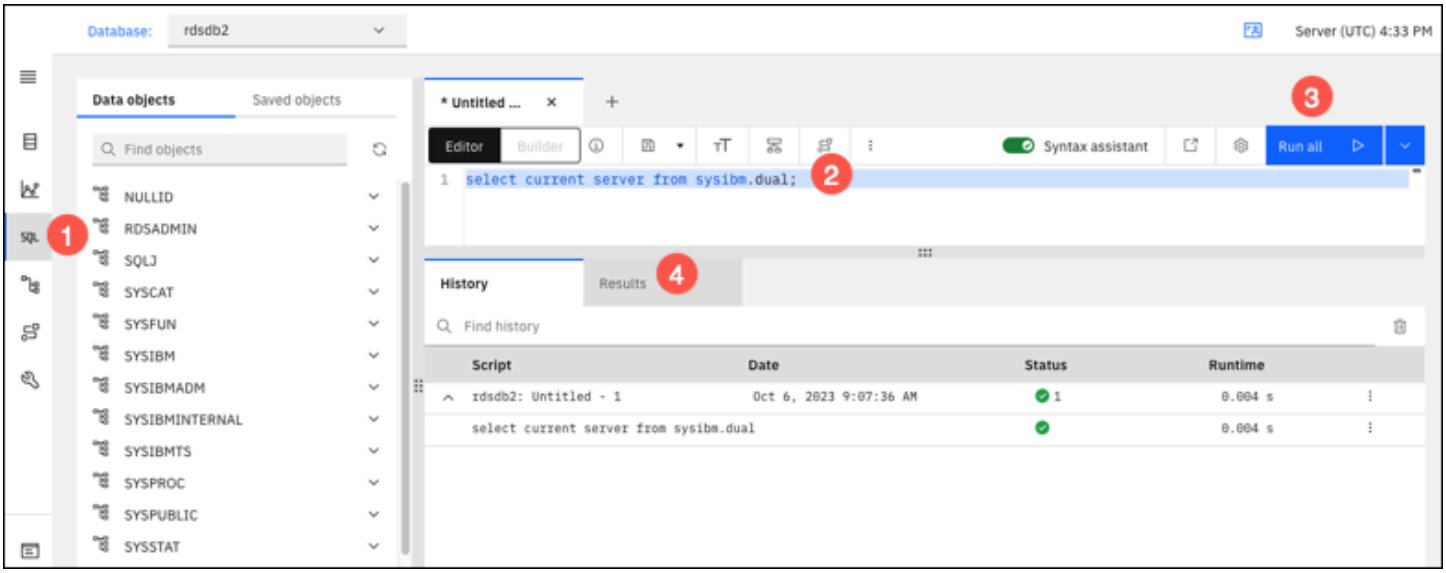
Die Verwendung von IBM Db2 Data Management Console

Sie können Folgendes verwenden … IBM Db2 Data Management Console um die folgenden Arten von Aufgaben zu erledigen:

- Verwalten Sie mehrere RDS DB2-DB-Instances.
- SQLBefehle ausführen.
- Erkunden, erstellen oder ändern Sie Daten und Datenbankobjekte.
- Erstellen Sie `EXPLAIN PLAN` Anweisungen inSQL.
- Optimieren Sie Abfragen.

Um SQL Befehle auszuführen und die Ergebnisse anzuzeigen

1. Wählen Sie in der linken Navigationsleiste SQL.
2. Geben Sie einen SQL Befehl ein.
3. Wählen Sie Alle ausführen.
4. Um die Ergebnisse anzuzeigen, wählen Sie die Registerkarte Ergebnisse.



Überlegungen zu Sicherheitsgruppen mit Amazon RDS for Db2

Damit Sie eine Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance herstellen können, muss diese einer Sicherheitsgruppe zugeordnet sein, die die erforderlichen IP-Adressen und die Netzwerkkonfiguration enthält. Ihre RDS for Db2-DB-Instance verwendet möglicherweise die Standardsicherheitsgruppe. Wenn Sie bei der Erstellung der RDS for Db2-DB-Instance eine nicht konfigurierte Standardsicherheitsgruppe zugewiesen haben, verhindert die Firewall Internetverbindungen. Informationen zum Erstellen einer neuen Sicherheitsgruppe finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Nachdem Sie die neue Sicherheitsgruppe erstellt haben, ändern Sie Ihre DB-Instance, um ihr die Sicherheitsgruppe zuzuordnen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sie können die Sicherheitsstufe mithilfe von SSL erhöhen, um Verbindungen zu Ihrer DB-Instance zu verschlüsseln. Weitere Informationen finden Sie unter Verwendung SSL/TLS mit einer Amazon RDS for Db2-DB-Instance.

Sicherung von Amazon RDS für Db2-DB-Instance-Verbindungen

Amazon RDS for Db2 unterstützt Möglichkeiten zur Verbesserung der Sicherheit Ihrer RDS for Db2-DB-Instance.

Themen

- Verwendung SSL/TLS mit einer Amazon RDS for Db2-DB-Instance
- KerberosAuthentifizierung für Amazon RDS for Db2 verwenden

Verwendung SSL/TLS mit einer Amazon RDS for Db2-DB-Instance

SSL ist ein Industriestandardprotokoll zur Sicherung von Netzwerkverbindungen zwischen Client und Server. Nach der SSL-Version 3.0 wurde der Name in TLS geändert, aber wir bezeichnen das Protokoll immer noch oft als SSL. Amazon RDS unterstützt SSL-Verschlüsselung für Amazon RDS für Db2-DB-Instances. Die Nutzung der SSL/TLS, you can encrypt a connection between your application client and your RDS for Db2 DB instance. SSL/TLS Unterstützung ist AWS-Regionen für RDS for Db2 in allen Versionen verfügbar.

Um die SSL/TLS Verschlüsselung für eine RDS for Db2-DB-Instance zu aktivieren, fügen Sie die Db2-SSL-Option zu der Parametergruppe hinzu, die der DB-Instance zugeordnet ist. Amazon RDS verwendet einen zweiten Port, wie von Db2 gefordert, für SSL/TLS Verbindungen. Auf diese Weise können sowohl Klartext- als auch SSL-verschlüsselte Kommunikation zwischen einer DB-Instance und einem Db2-Client gleichzeitig erfolgen. Sie können z. B. den Port mit Klartext-Kommunikation verwenden, um mit anderen Ressourcen innerhalb einer VPC zu kommunizieren, und den Port mit SSL-verschlüsselter Kommunikation, um mit Ressourcen außerhalb der VPC zu kommunizieren.

Themen

- Eine Verbindung herstellen SSL/TLS
- Connect zu Ihrem Db2-Datenbankserver her

Eine Verbindung herstellen SSL/TLS

Um eine SSL/TLS Verbindung herzustellen, wählen Sie eine Zertifizierungsstelle (CA) aus, laden Sie ein Zertifikatspaket für alle AWS-Regionen herunter und fügen Sie Parameter zu einer benutzerdefinierten Parametergruppe hinzu.

Schritt 1: Wählen Sie eine Zertifizierungsstelle und laden Sie ein Zertifikat herunter

Wählen Sie eine Zertifizierungsstelle (CA) und laden Sie ein Zertifikatspaket für alle herunter AWS-Regionen. Weitere Informationen finden Sie unter .

Schritt 2: Aktualisieren Sie die Parameter in einer benutzerdefinierten Parametergruppe

> **Important**
>
> Wenn Sie das Modell Bring Your Own License (BYOL) für RDS for Db2 verwenden, ändern Sie die benutzerdefinierte Parametergruppe, die Sie für Ihre IBM Customer ID und Ihre erstellt haben. IBM Site ID Wenn Sie ein anderes Lizenzmodell für RDS for Db2 verwenden, gehen Sie wie folgt vor, um einer benutzerdefinierten Parametergruppe Parameter hinzuzufügen. Weitere Informationen finden Sie unter Lizenzierungsoptionen für Amazon RDS für Db2.

Sie können Standardparametergruppen für RDS for Db2-DB-Instances nicht ändern. Daher müssen Sie eine benutzerdefinierte Parametergruppe erstellen, sie ändern und sie dann an Ihre RDS for Db2-DB-Instances anhängen. Informationen zu Parametergruppen finden Sie unter DB-Parametergruppen für Amazon RDS Amazon.

Verwenden Sie die Parametereinstellungen in der folgenden Tabelle.

Parameter	Wert
`DB2COMM`	`TCPIP,SSL` oder `SSL`
`SSL_SVCENAME`	`<any port number except the number used for the non-SSL port>`

Um Parameter in einer benutzerdefinierten Parametergruppe zu aktualisieren

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem create-db-parameter-groupSie den Befehl ausführen.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - `--db-parameter-group-name`— Ein Name für die Parametergruppe, die Sie erstellen.

- `--db-parameter-group-family`— Die Db2-Engine-Edition und die Hauptversion. Zulässige Werte: `db2-se-11-5`, `db2-ae-11.5`.
- `--description`— Eine Beschreibung für diese Parametergruppe.

Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.

2. Ändern Sie die Parameter in der benutzerdefinierten Parametergruppe, die Sie durch Ausführen des modify-db-parameter-groupBefehls erstellt haben.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - `--db-parameter-group-name`— Der Name der Parametergruppe, die Sie erstellt haben.
   - `--parameters`— Eine Reihe von Parameternamen, Werten und Anwendungsmethoden für die Parameteraktualisierung.

   Weitere Hinweise zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

3. Ordnen Sie die Parametergruppe Ihrer RDS for Db2-DB-Instance zu. Weitere Informationen finden Sie unter Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon.

Connect zu Ihrem Db2-Datenbankserver her

Die Anweisungen für die Verbindung mit Ihrem Db2-Datenbankserver sind sprachspezifisch.

Java

Um eine Verbindung zu Ihrem Db2-Datenbankserver herzustellen, verwenden Sie Java

1. Laden Sie den JDBC-Treiber herunter. Weitere Informationen finden Sie unter DB2 JDBC-Treiberversionen und Downloads in der IBM Support-Dokumentation.
2. Erstellen Sie eine Shell-Skriptdatei mit dem folgenden Inhalt. Dieses Skript fügt alle Zertifikate aus dem Paket zu einem hinzuJava KeyStore.

> **Important**
>
> Stellen Sie sicher, dass der Pfad im Skript `keytool` vorhanden ist, damit das Skript ihn finden kann. Wenn Sie einen Db2-Client verwenden, finden Sie den `keytool` unter`~sqlib/java/jdk64/jre/bin`.

```
#!/bin/bash
PEM_FILE=$1
PASSWORD=$2
KEYSTORE=$3
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)
for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cat $PEM_FILE |
    awk "n==$N { print }; /END CERTIFICATE/ { n++ }" |
    keytool -noprompt -import -trustcacerts -alias $ALIAS -keystore $KEYSTORE -
storepass $PASSWORD
done
```

3. Führen Sie den folgenden Befehl aus, um das Shell-Skript auszuführen und die PEM Datei mit dem Zertifikatspaket in a Java KeyStore zu importieren. *`shell_file_name.sh`*Ersetzen Sie es durch den Namen Ihrer Shell-Skriptdatei und *`password`* durch das Passwort für IhreJava KeyStore.

```
./shell_file_name.sh global-bundle.pem password truststore.jks
```

4. Führen Sie den folgenden Befehl aus, um eine Verbindung zu Ihrem Db2-Server herzustellen. Ersetzen Sie die folgenden Platzhalter im Beispiel durch Ihre RDS for Db2-DB-Instance-Informationen.

   - *`ip_address`*— Die IP-Adresse für Ihren DB-Instance-Endpunkt.
   - *`port`*— Die Portnummer für die SSL-Verbindung. Dies kann eine beliebige Portnummer sein, mit Ausnahme der Nummer, die für den Nicht-SSL-Port verwendet wird.
   - *`database_name`*— Der Name Ihrer Datenbank in Ihrer DB-Instance.
   - *`master_username`*— Der Master-Benutzername für Ihre DB-Instance.
   - *`master_password`*— Das Master-Passwort für Ihre DB-Instance.

```
export trustStorePassword=MyPassword
java -cp ~/dsdriver/jdbc_sqlj_driver/linuxamd64/db2jcc4.jar \
com.ibm.db2.jcc.DB2Jcc -url \
"jdbc:db2://ip_address:port/database_name:\
sslConnection=true;sslTrustStoreLocation=\
~/truststore.jks;\
sslTrustStorePassword=${trustStorePassword};\
sslVersion=TLSv1.2;\
encryptionAlgorithm=2;\
securityMechanism=7;" \
-user master_username -password master_password
```

Node.js

Um eine Verbindung zu Ihrem Db2-Datenbankserver herzustellen, verwenden Sie Node.js

1. Installieren Sie den node-ibm_dbTreiber. Weitere Informationen finden Sie in der Dokumentation unter Installation des node-ibm_db-Treibers auf Linux und UNIX -Systemen. IBM Db2
2. Erstellen Sie eine JavaScript Datei, die auf dem folgenden Inhalt basiert. Ersetzen Sie die folgenden Platzhalter im Beispiel durch Ihre RDS for Db2-DB-Instance-Informationen.
   - *ip_address*— Die IP-Adresse für Ihren DB-Instance-Endpunkt.
   - *master_username*— Der Master-Benutzername für Ihre DB-Instance.
   - *master_password*— Das Master-Passwort für Ihre DB-Instance.
   - *database_name*— Der Name Ihrer Datenbank in Ihrer DB-Instance.
   - *port*— Die Portnummer für die SSL-Verbindung. Dies kann eine beliebige Portnummer sein, mit Ausnahme der Nummer, die für den Nicht-SSL-Port verwendet wird.

```
var ibmdb = require("ibm_db");
const hostname = "ip_address";
const username = "master_username";
const password = "master_password";
const database = "database_name";
const port = "port";
const certPath = "/root/qa-bundle.pem";
```

```
ibmdb.open("DRIVER={DB2};DATABASE=" + database + ";HOSTNAME=" +
 hostname + ";UID=" + username + ";PWD=" + password + ";PORT=" + port +
 ";PROTOCOL=TCPIP;SECURITY=SSL;SSLServerCertificate=" + certPath + ";", function
 (err, conn){
 if (err) return console.log(err);
 conn.close(function () {
 console.log('done');
 });
});
```

3. Führen Sie den folgenden Befehl aus, um die JavaScript Datei auszuführen.

```
node ssl-test.js
```

Python

Um eine Verbindung zu Ihrem Db2-Datenbankserver herzustellen, verwenden Sie Python

1. Erstellen Sie eine Python Datei mit dem folgenden Inhalt. Ersetzen Sie die folgenden Platzhalter im Beispiel durch Ihre RDS for Db2-DB-Instance-Informationen.

   - *port*— Die Portnummer für die SSL-Verbindung. Dies kann eine beliebige Portnummer sein, mit Ausnahme der Nummer, die für den Nicht-SSL-Port verwendet wird.
   - *master_username*— Der Master-Benutzername für Ihre DB-Instance.
   - *master_password*— Das Master-Passwort für Ihre DB-Instance.
   - *database_name*— Der Name Ihrer Datenbank in Ihrer DB-Instance.
   - *ip_address*— Die IP-Adresse für Ihren DB-Instance-Endpunkt.

```
import click
import ibm_db
import sys

port = port;
master_user_id = "master_username" # Master id used to create your DB instance
master_password = "master_password" # Master password used to create your DB
 instance
db_name = "database_name" # If not given "db-name'
vpc_customer_private_ip = "ip_address" # Hosts end points - Customer private IP
 Addressicert_path = "/root/ssl/global-bundle.pem" # cert path
```

```
@click.command()
@click.option("--path", help="certificate path")
def db2_connect(path):

    try:
        conn =
 ibm_db.connect(f"DATABASE={db_name};HOSTNAME={vpc_customer_private_ip};PORT={port};

 PROTOCOL=TCPIP;UID={master_user_id};PWD={master_password};SECURITY=ssl;SSLServerCertifi
 "", "")
        try:
            ibm_db.exec_immediate(conn, 'create table tablename (a int);')
            print("Query executed successfully")
        except Exception as e:
            print(e)
        finally:
            ibm_db.close(conn)
            sys.exit(1)
    except Exception as ex:
        print("Trying to connect...")

if __name__ == "__main__":
    db2_connect()
```

2. Erstellen Sie das folgende Shell-Skript, das die von Ihnen erstellte Python Datei ausführt. *python_file_name.py*Ersetzen Sie es durch den Namen Ihrer Python Skriptdatei.

```
#!/bin/bash
PEM_FILE=$1
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)

for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cert=`cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }"`
    cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }" >
 $ALIAS.pem
    python3 <python_file_name.py> --path $ALIAS.pem
    output=`echo $?`
    if [ $output == 1 ]; then
        break
    fi
```

```
done
```

3. Führen Sie den folgenden Befehl aus, um die PEM Datei mit dem Zertifikatspaket zu importieren und das Shell-Skript auszuführen. *shell_file_name.sh*Ersetzen Sie es durch den Namen Ihrer Shell-Skriptdatei.

```
./shell_file_name.sh global-bundle.pem
```

Db2 CLP

Um eine Verbindung zu Ihrem Db2-Datenbankserver herzustellen, verwenden Sie Db2 CLP

1. Um eine Verbindung zu Ihrer Db2-Instance herzustellenDb2 CLP, benötigen Sie GSKit, das Sie bei IBM Fix Central herunterladen können. Zur Verwendung Db2 CLP benötigen Sie außerdem den IBM Db2 Client, den Sie im IBM Support unter Client und Treiber der ersten Version 11.5 herunterladen können.
2. Erstellen Sie einen Keystore.

```
gsk8capicmd_64 -keydb -create -db "directory/keystore-filename" -pw
 "changeThisPassword" -type pkcs12 -stash
```

3. Importieren Sie die Zertifikatspakete in den Keystore.

```
gsk8capicmd_64 -cert -import -file global-bundle.pem -target directory/keystore-
filename> -target_stashed
```

4. Aktualisieren Sie die Db2-Instanzkonfiguration.

```
db2 update dbm cfg using SSL_CLNT_KEYDB keystore-filename
 SSL_CLNT_STASH keystore stash file immediate
```

5. Katalogisieren Sie den Knoten und die Datenbank.

```
db2 catalog tcpip node ssluse1 REMOTE endpoint SERVER ssl_svcename security ssl

db2 catalog database testdb as ssltest at node ssluse1
```

6. Verbinden Sie mit der Datenbank.

```
db2 connect to sltest user username using password
```

KerberosAuthentifizierung für Amazon RDS for Db2 verwenden

Sie können die Kerberos Authentifizierung verwenden, um Benutzer zu authentifizieren, wenn sie sich mit Ihrer Amazon RDS for Db2-DB-Instance verbinden. In dieser Konfiguration arbeitet Ihre DB-Instance mit AWS Directory Service for Microsoft Active Directory, auch genannt. AWS Managed Microsoft AD Sie fügen die Domain und andere Informationen Ihres AWS Managed Microsoft AD Verzeichnisses zu Ihrer RDS for Db2-DB-Instance hinzu. Wenn sich Benutzer mit einer RDS for Db2-DB-Instance authentifizieren, die der vertrauenswürdigen Domäne hinzugefügt wurde, werden Authentifizierungsanforderungen an das AWS Managed Microsoft AD Verzeichnis weitergeleitet, mit dem Sie sie erstellt haben. AWS Directory Service

Wenn Sie alle Ihre Anmeldeinformationen im selben Verzeichnis aufbewahren, können Sie Zeit und Mühe sparen. Mit diesem Ansatz haben Sie einen zentralen Ort für die Speicherung und Verwaltung von Anmeldedaten für mehrere DB-Instances. Die Verwendung eines Verzeichnisses kann auch Ihr allgemeines Sicherheitsprofil verbessern.

Außerdem können Sie von Ihrem eigenen On-Premises Microsoft Active Directory auf Anmeldeinformationen zugreifen. Dazu erstellen Sie eine vertrauensvolle Domain-Beziehung, damit das AWS Managed Microsoft AD -Verzeichnis Ihrem On-Premises Microsoft Active Directory vertraut. Auf diese Weise können Ihre Benutzer mit derselben Windows-SSO-Erfahrung (Single Sign-On) auf Ihre RDS for Db2-DB-Instances zugreifen wie beim Zugriff auf Workloads in Ihrem lokalen Netzwerk.

Weitere Informationen finden Sie unter Was ist? AWS Directory Service im AWS Directory Service Administratorhandbuch.

Informationen zur Kerberos Authentifizierung finden Sie in den folgenden Themen:

Themen

- KerberosAuthentifizierung für Amazon RDS für Db2-DB-Instances einrichten
- Herstellen einer Verbindung zu Amazon RDS for Db2 mit Kerberos Authentifizierung

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Verfügbarkeit von RDS für Db2 mit Kerberos Authentifizierung in Version und Region finden Sie unterUnterstützte Regionen und DB-Engines für die Kerberos-Authentifizierung in Amazon RDS.

> **Note**
>
> KerberosDie Authentifizierung wird nicht für DB-Instance-Klassen unterstützt, die für RDS for Db2-DB-Instances veraltet sind. Weitere Informationen finden Sie unter Amazon RDS für Db2-Instance-Klassen.

Überblick über die Kerberos Authentifizierung für RDS für Db2-DB-Instances

Um die Kerberos Authentifizierung für eine RDS for Db2-DB-Instance einzurichten, führen Sie die folgenden allgemeinen Schritte aus, die später ausführlicher beschrieben werden:

1. Wird verwendet AWS Managed Microsoft AD , um ein AWS Managed Microsoft AD Verzeichnis zu erstellen. Sie können das AWS Management Console, das AWS Command Line Interface (AWS CLI) oder verwenden, AWS Directory Service um das Verzeichnis zu erstellen. Weitere Informationen finden Sie unter AWS Managed Microsoft AD Verzeichnis erstellen im AWS Directory Service Administratorhandbuch.
2. Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle, die die verwaltete IAM-Richtlinie verwendet. `AmazonRDSDirectoryServiceAccess` Die IAM-Rolle ermöglicht Amazon RDS, Ihr Verzeichnis aufzurufen.

   Damit die IAM-Rolle den Zugriff ermöglicht, muss der Endpunkt AWS Security Token Service (AWS STS) in der AWS-Region für Sie richtigen Weise aktiviert sein. AWS-Konto AWS STS Endpoints sind standardmäßig in allen aktiv AWS-Regionen, und Sie können sie ohne weitere Aktionen verwenden. Weitere Informationen finden Sie unter Aktivierung und Deaktivierung AWS STSAWS-Region im IAM-Benutzerhandbuch.
3. Erstellen oder ändern Sie eine RDS for Db2-DB-Instance mithilfe der AWS Management Console AWS CLI, der oder der RDS-API mit einer der folgenden Methoden:
   - Erstellen Sie mithilfe der Konsole, des create-db-instanceBefehls oder der Operation Create DBInstance API eine neue RDS for Db2-DB-Instance. Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

- Ändern Sie eine vorhandene RDS for Db2-DB-Instance mithilfe der Konsole, des modify-db-instanceBefehls oder der ModifyDBInstanceAPI-Operation. Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie mithilfe der Konsole, des restore-db-instance-from-db-snapshotBefehls oder der RestoreDBInstanceFromDBSnapshotAPI-Operation eine RDS for Db2-DB-Instance aus einem DB-Snapshot wieder her. Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.
- Stellen Sie eine RDS for Db2-DB-Instance point-in-time mithilfe der Konsole, des restore-db-instance-to-point-in-timeBefehls oder der RestoreDBInstanceToPointInTimeAPI-Operation wieder her. Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Sie können die DB-Instance in derselben Amazon Virtual Private Cloud (VPC) wie das Verzeichnis oder in einer anderen AWS-Konto oder VPC lokalisieren. Wenn Sie die RDS for Db2-DB-Instance erstellen oder ändern, führen Sie die folgenden Aufgaben aus:

- Geben Sie den Domänenbezeichner (`d-*`-Bezeichner) an, der beim Erstellen Ihres Verzeichnisses generiert wurde.
- Geben Sie außerdem den Namen der IAM-Rolle an, die Sie erstellt haben.
- Stellen Sie sicher, dass die Sicherheitsgruppe der DB-Instance eingehenden Datenverkehr von der Verzeichnissicherheitsgruppe empfangen kann.

4. Konfigurieren Sie Ihren Db2-Client und stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und AWS Directory Service für die folgenden Ports fließen kann:
   - TCP/UDP-Port 53 — DNS
   - TCP 88 — Authentifizierung Kerberos
   - TCP 389 — LDAP
   - TCP 464 — Authentifizierung Kerberos

Verwalten einer DB-Instance in einer Domäne

Sie können die AWS Management Console, oder die RDS-API verwenden AWS CLI, um Ihre DB-Instance und ihre Beziehung zu Ihrer zu verwaltenMicrosoft Active Directory. Sie können beispielsweise eine zuordnen, um die Kerberos Authentifizierung Active Directory zu aktivieren. Sie können auch die Zuordnung für eine entfernenActive Directory, um die Kerberos Authentifizierung zu deaktivieren. Sie können eine DB-Instance auch so verschieben, dass sie von einer Instanz extern authentifiziert werden Microsoft Active Directory soll.

Wenn Sie beispielsweise den modify-db-instanceCLI-Befehl ausführen, können Sie die folgenden Aktionen ausführen:

- Versuchen Sie erneut, die Kerberos Authentifizierung für eine fehlgeschlagene Mitgliedschaft zu aktivieren, indem Sie die Verzeichnis-ID der aktuellen Mitgliedschaft für die `--domain` Option angeben.
- Deaktivieren Sie die Kerberos Authentifizierung auf einer DB-Instance, indem Sie dies `none` für die `--domain` Option angeben.
- Verschieben Sie eine DB-Instance von einer Domain in eine andere, indem Sie die Domain-ID der neuen Domain für die `--domain` Option angeben.

Grundlegendes zur Domänenmitgliedschaft

Nachdem Sie Ihre DB-Instance erstellt oder geändert haben, wird sie Mitglied der Domäne. Sie können den Status der Domänenmitgliedschaft in der Konsole oder durch Ausführen des describe-db-instancesBefehls anzeigen. Der Status der DB-Instance kann einer der folgenden sein:

- `kerberos-enabled`— Für die DB-Instance ist die Kerberos Authentifizierung aktiviert.
- `enabling-kerberos`— AWS ist dabei, die Kerberos Authentifizierung für diese DB-Instance zu aktivieren.
- `pending-enable-kerberos`— Die Aktivierung der Kerberos Authentifizierung für diese DB-Instance steht noch aus.
- `pending-maintenance-enable-kerberos`— AWS wird versuchen, die Kerberos Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu aktivieren.
- `pending-disable-kerberos`— Die Deaktivierung der Kerberos Authentifizierung für diese DB-Instance steht noch aus.
- `pending-maintenance-disable-kerberos`— AWS wird versuchen, die Kerberos Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu deaktivieren.
- `enable-kerberos-failed`— Aufgrund eines Konfigurationsproblems konnte die AWS Kerberos Authentifizierung auf der DB-Instance nicht aktiviert werden. Korrigieren Sie das Konfigurationsproblem, bevor Sie den Befehl zum Ändern der DB-Instance erneut ausgeben.
- `disabling-kerberos`— AWS ist dabei, die Kerberos Authentifizierung für diese DB-Instance zu deaktivieren.

Eine Anfrage zur Aktivierung der Kerberos Authentifizierung kann aufgrund eines Problems mit der Netzwerkkonnektivität oder einer falschen IAM-Rolle fehlschlagen. In einigen Fällen schlägt der Versuch, die Kerberos Authentifizierung zu aktivieren, möglicherweise fehl, wenn Sie eine DB-Instance erstellen oder ändern. Stellen Sie in diesem Fall sicher, dass Sie die richtige IAM-Rolle verwenden, und ändern Sie dann die DB-Instance so, dass sie der Domäne beitritt.

KerberosAuthentifizierung für Amazon RDS für Db2-DB-Instances einrichten

Sie verwenden AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD), um die Kerberos Authentifizierung für eine RDS for Db2-DB-Instance einzurichten. Gehen Sie folgendermaßen vor, um die Kerberos Authentifizierung einzurichten:

Themen

- Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD
- Schritt 2: Erstellen einer Vertrauensstellung
- Schritt 3: Erstellen Sie eine IAM-Rolle, auf die Amazon RDS zugreifen kann AWS Directory Service
- Schritt 4: Anlegen und Konfigurieren von Benutzern
- Schritt 5: Erstellen Sie eine RDS for Db2-Administratorgruppe in AWS Managed Microsoft AD
- Schritt 6: DB-Parameter ändern
- Schritt 7: Erstellen oder ändern Sie eine RDS for Db2-DB-Instance
- Schritt 8: Rufen Sie die Active Directory-Gruppen-SID ab in PowerShell
- Schritt 9: Fügen Sie SID zu GroupName Zuordnungen zu Ihrer RDS for Db2-DB-Instance hinzu
- Schritt 10: Konfigurieren Sie einen Db2-Client

Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD

AWS Directory Service erstellt ein vollständig verwaltetes Active Directory im AWS Cloud. Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service erstellt zwei Domänencontroller und DNS-Server für Sie. Die Verzeichnisserver werden in verschiedenen Subnetzen in einer VPC erstellt. Diese Redundanz hilft sicherzustellen, dass Ihr Verzeichnis verfügbar bleibt, auch wenn ein Fehler auftritt.

Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service führt in Ihrem Namen die folgenden Aufgaben aus:

- Richtet eine Active Directory innerhalb Ihrer VPC ein.

- Erstellt ein Konto für den Verzeichnisadministrator mit dem Benutzernamen `Admin` und dem angegebenen Passwort. Mit diesem Konto verwalten Sie das Verzeichnis.

  > **Important**
  >
  > Stellen Sie sicher, dass Sie dieses Passwort speichern. AWS Directory Service speichert dieses Passwort nicht und es kann nicht abgerufen oder zurückgesetzt werden.

- Erstellt eine Sicherheitsgruppe für die Verzeichniscontroller. Die Sicherheitsgruppe muss die Kommunikation mit der RDS for Db2-DB-Instance zulassen.

AWS Erstellt beim Start AWS Directory Service for Microsoft Active Directory eine Organisationseinheit (OU), die alle Objekte Ihres Verzeichnisses enthält. Diese OU erhält den NetBIOS-Namen, den Sie beim Erstellen des Verzeichnisses eingegeben haben, und befindet sich im Domänenstamm. Der Domänenstamm gehört und wird von verwaltet AWS.

Das `Admin` Konto, das mit Ihrem AWS Managed Microsoft AD Verzeichnis erstellt wurde, verfügt über Berechtigungen für die gängigsten Verwaltungsaktivitäten Ihrer Organisationseinheit:

- Benutzer erstellen, aktualisieren oder löschen.
- Fügen Sie Ihrer Domain Ressourcen wie Datei- oder Druckserver hinzu, und weisen Sie dann Benutzern in Ihrer Organisationseinheit Berechtigungen für diese Ressourcen zu.
- Erstellen Sie zusätzliche OUs Container.
- Delegieren von Befugnissen.
- Stellen Sie gelöschte Objekte aus dem Active Directory Papierkorb wieder her.
- Ausführen Active Directory und DNS-Module (Domain Name Service) für Windows PowerShell auf dem AWS Directory Service.

Das `Admin`-Konto hat auch die Berechtigung, die folgenden domänenweiten Aktivitäten durchzuführen:

- Verwalten von DNS-Konfigurationen (Hinzufügen, Entfernen oder Aktualisieren von Datensätzen, Zonen und Weiterleitungen).
- Aufrufen von DNS-Ereignisprotokollen.
- Anzeigen von Sicherheitsereignisprotokollen.

Um ein Verzeichnis zu erstellen mit AWS Managed Microsoft AD

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die AWS Directory Service Konsole unter https://console.aws.amazon.com/directoryservicev2/.
2. Wählen Sie Verzeichnis einrichten.
3. Wählen Sie AWS Managed Microsoft AD. AWS Managed Microsoft AD ist die einzige Option, die derzeit für die Verwendung mit Amazon RDS unterstützt wird.
4. Wählen Sie Weiter aus.
5. Geben Sie auf der Seite Enter directory information (Verzeichnisinformationen eingeben) die folgenden Informationen ein:

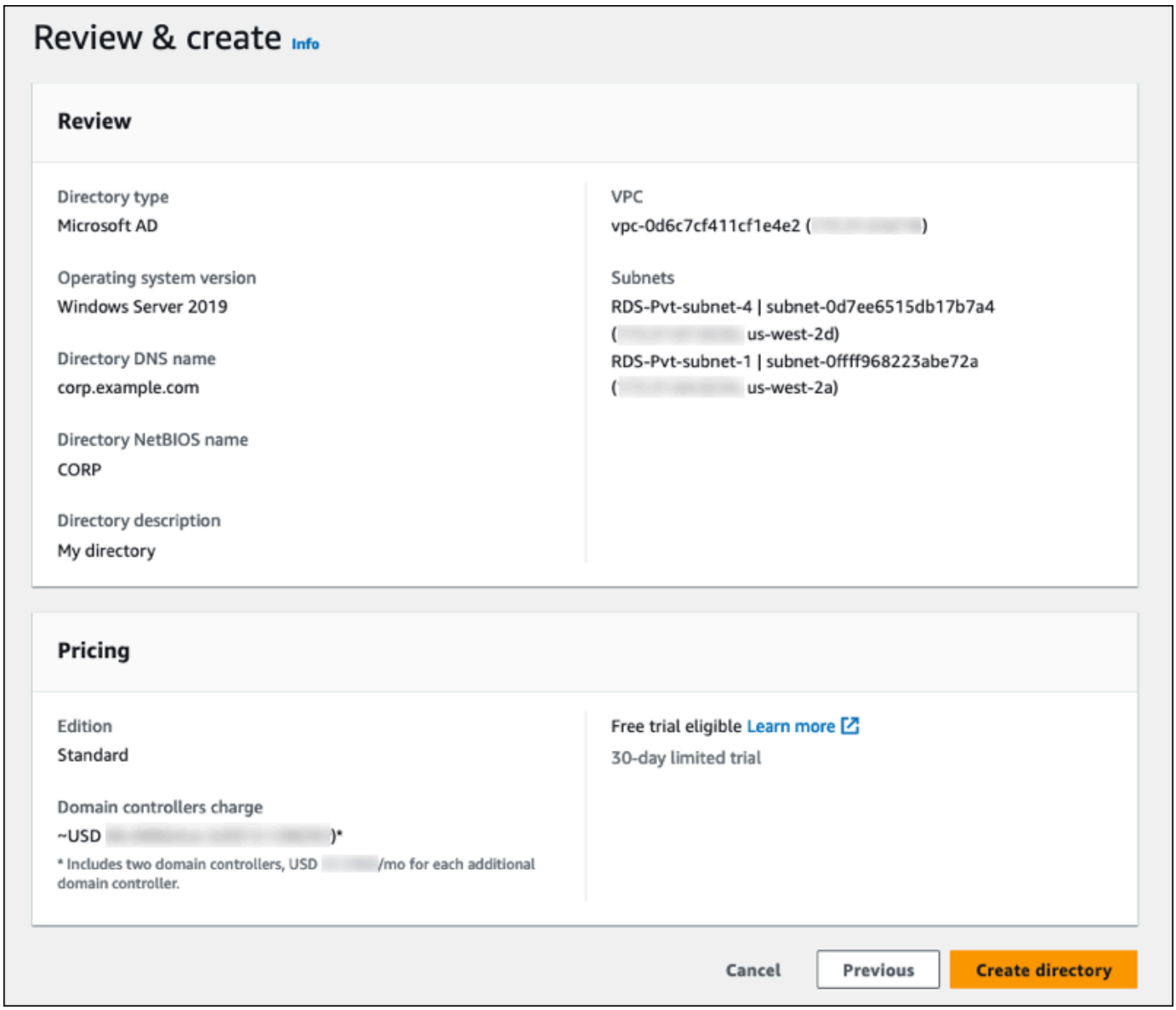
   - Edition — Wählen Sie die Edition, die Ihren Anforderungen entspricht.
   - DNS-Name des Verzeichnisses — Der vollständig qualifizierte Name für das Verzeichnis, z. `corp.example.com` B.
   - NetBIOS-Name des Verzeichnisses — Ein optionaler Kurzname für das Verzeichnis, z. B. `CORP`
   - Verzeichnisbeschreibung — Eine optionale Beschreibung für das Verzeichnis.
   - Admin-Passwort — Das Passwort für den Verzeichnisadministrator. Beim Erstellen des Verzeichnisses wird ein Administratorkonto mit dem Benutzernamen `Admin` und diesem Passwort erstellt.

     Das Passwort für den Verzeichnisadministrator darf nicht das Wort "admin" enthalten. Beachten Sie beim Passwort die Groß- und Kleinschreibung und es muss 8 bis 64 Zeichen lang sein. Zudem muss es mindestens ein Zeichen aus dreien der vier folgenden Kategorien enthalten:

     - Kleinbuchstaben (a–z)
     - Großbuchstaben (A–Z)
     - Zahlen (0–9)
     - Nicht-alphanumerische Zeichen (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)
     - Passwort bestätigen — Geben Sie das Administratorkennwort erneut ein.

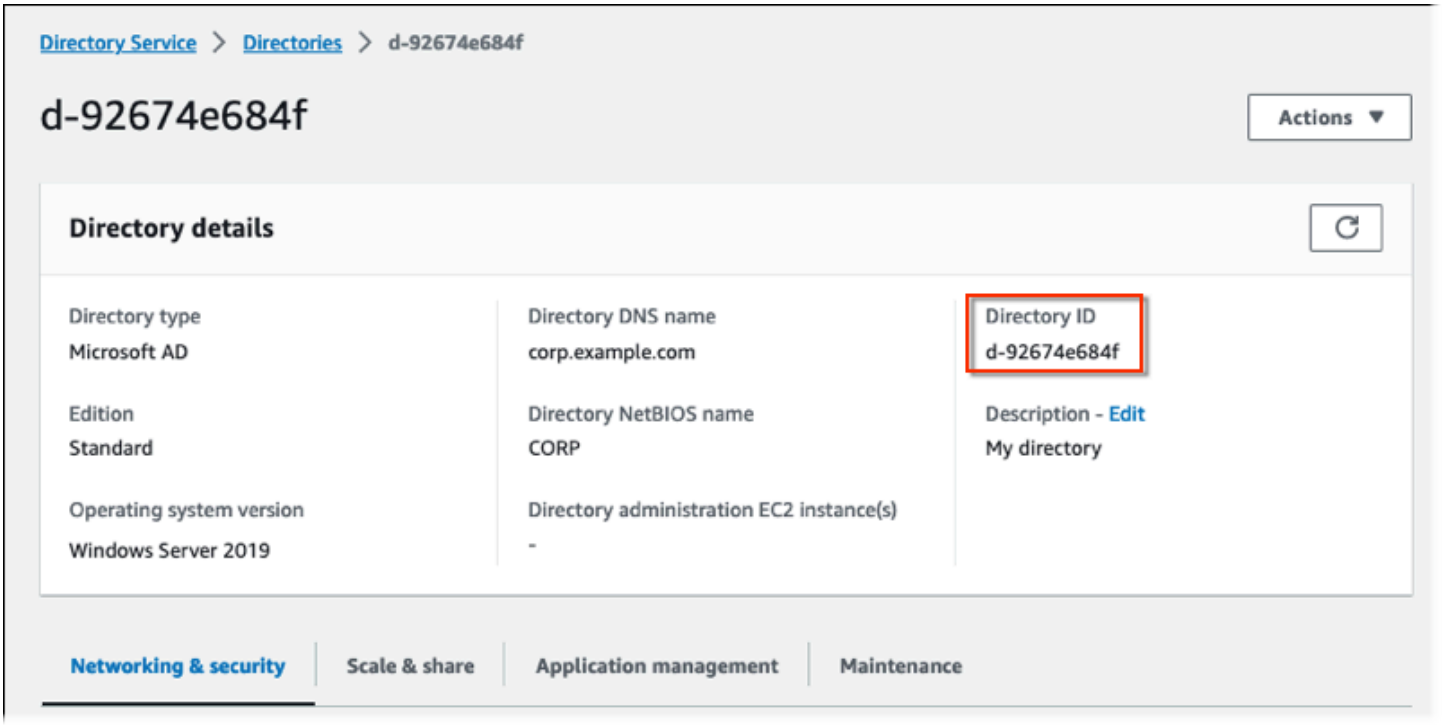
> **Important**
>
> Stellen Sie sicher, dass Sie dieses Passwort speichern. AWS Directory Service speichert dieses Passwort nicht und es kann nicht abgerufen oder zurückgesetzt werden.

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Choose VPC and subnets (VPC und Subnetze wählen) die folgenden Informationen an.
   - VPC — Wählen Sie die VPC für das Verzeichnis aus. Sie können die DB-Instance RDS for Db2 in derselben VPC oder in einer anderen VPC erstellen.
   - Subnetze — Wählen Sie die Subnetze für die Verzeichnisserver aus. Die beiden Subnetze müssen zu verschiedenen Availability-Zonen gehören.
8. Wählen Sie Weiter aus.
9. Überprüfen Sie die Verzeichnisinformationen. Wenn Änderungen erforderlich sind, klicken Sie auf Previous (Zurück) und nehmen Sie die Änderungen vor. Wenn die Informationen richtig sind, wählen Sie Create directory (Verzeichnis erstellen).



Es dauert einige Minuten, bis das Verzeichnis erstellt wurde. Wenn es erfolgreich erstellt wurde, ändert sich der Wert Status in Active (Aktiv).

Um Informationen zu Ihrem Verzeichnis zu sehen, wählen Sie die Verzeichnis-ID unter Verzeichnis-ID aus. Notieren Sie sich den Wert Directory ID. Sie benötigen diesen Wert, wenn Sie Ihre RDS for Db2-DB-Instance erstellen oder ändern.



Schritt 2: Erstellen einer Vertrauensstellung

Wenn Sie nur AWS Managed Microsoft AD verwenden möchten, fahren Sie mit fortSchritt 3: Erstellen Sie eine IAM-Rolle, auf die Amazon RDS zugreifen kann AWS Directory Service.

Um die Kerberos-Authentifizierung mithilfe Ihres selbstverwalteten Active Directory zu aktivieren, müssen Sie eine Gesamtstrukturvertrauensstellung zwischen Ihrem selbstverwalteten Active Directory und dem einrichten. Eine Gesamtvertrauensstellung ist eine Vertrauensstellung zwischen einem Microsoft AD und einem selbstverwalteten Active Directory, das im vorherigen Schritt AWS Managed Microsoft AD erstellt wurde. Die Vertrauensstellung kann auch bidirektional erfolgen, wobei beide Active Directories einander vertrauen. Weitere Informationen zum Einrichten von Gesamtvertrauensstellungen mithilfe AWS Directory Service von finden Sie unter Wann sollte eine Vertrauensstellung erstellt werden? im AWS Directory Service Administration Guide.

Schritt 3: Erstellen Sie eine IAM-Rolle, auf die Amazon RDS zugreifen kann AWS Directory Service

Damit Amazon RDS AWS Directory Service für Sie anrufen kann, AWS-Konto benötigen Sie eine IAM-Rolle, die die verwaltete IAM-Richtlinie verwendet. `AmazonRDSDirectoryServiceAccess` Diese Rolle ermöglicht es Amazon RDS, Anrufe an zu tätigen AWS Directory Service.

Wenn Sie eine DB-Instance mit dem erstellen AWS Management Console und Ihr Konsolen-Benutzerkonto über die `iam:CreateRole` entsprechende Berechtigung verfügt, erstellt die

Konsole automatisch die benötigte IAM-Rolle. In diesem Fall lautet der Rollenname `rds-directoryservice-kerberos-access-role`. Andernfalls müssen Sie die IAM-Rolle manuell erstellen. Wenn Sie diese IAM-Rolle erstellen`Directory Service`, wählen Sie die AWS verwaltete Richtlinie aus und hängen Sie `AmazonRDSDirectoryServiceAccess` sie an.

Weitere Informationen zum Erstellen von IAM-Rollen für einen Dienst finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

> **Note**
>
> Die für die Windows Authentifizierung für RDS verwendete IAM-Rolle Microsoft SQL Server kann nicht für RDS for Db2 verwendet werden.

Alternativ können Sie Richtlinien mit den erforderlichen Berechtigungen erstellen, anstatt die verwaltete Richtlinie `AmazonRDSDirectoryServiceAccess` zu verwenden. In diesem Fall muss die IAM-Rolle über die folgende IAM-Vertrauensrichtlinie verfügen:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Die Rolle muss außerdem die folgende IAM-Rollenrichtlinie haben:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Schritt 4: Anlegen und Konfigurieren von Benutzern

Sie können Benutzer mithilfe des Active Directory Users and Computers Tools erstellen. Dies ist eines der Active Directory Lightweight Directory Services Tools Active Directory Domain Services und. Weitere Informationen finden Sie in der Microsoft Dokumentation unter Benutzer und Computer zur Active Directory Domäne hinzufügen. In diesem Fall handelt es sich bei Benutzern um Einzelpersonen oder andere Entitäten, z. B. deren Computer, die Teil der Domäne sind und deren Identitäten im Verzeichnis verwaltet werden.

Um Benutzer in einem AWS Directory Service Verzeichnis zu erstellen, müssen Sie mit einer Windows basierten EC2 Amazon-Instance verbunden sein, die Mitglied des AWS Directory Service Verzeichnisses ist. Gleichzeitig müssen Sie als Benutzer angemeldet sein, der über die Rechte zum Erstellen von Benutzern verfügt. Weitere Informationen finden Sie unter Erstellen eines Benutzers im AWS Directory Service Administration Guide.

Schritt 5: Erstellen Sie eine RDS for Db2-Administratorgruppe in AWS Managed Microsoft AD

RDS for Db2 unterstützt keine Kerberos Authentifizierung für den Master-Benutzer oder die beiden reservierten Amazon RDS-Benutzer `rdsdb` und`rdsadmin`. Stattdessen müssen Sie eine neue Gruppe mit dem Namen „`masterdba`In AWS Managed Microsoft AD“ erstellen. Weitere Informationen finden Sie Active Directoryin der Microsoft Dokumentation unter Erstellen

eines Gruppenkontos. Alle Benutzer, die Sie zu dieser Gruppe hinzufügen, verfügen über Masterbenutzerrechte.

Nachdem Sie die Kerberos Authentifizierung aktiviert haben, verliert der Masterbenutzer die `masterdba` Rolle. Daher kann der Masterbenutzer nur dann auf die Mitgliedschaft in der lokalen Benutzergruppe der Instanz zugreifen, wenn Sie die Kerberos Authentifizierung deaktivieren. Um den Masterbenutzer mit Kennwortanmeldung weiterhin zu verwenden, erstellen Sie einen Benutzer AWS Managed Microsoft AD mit demselben Namen wie der Masterbenutzer. Fügen Sie diesen Benutzer dann der Gruppe hinzu`masterdba`.

Schritt 6: DB-Parameter ändern

Wenn Sie AWS Managed Microsoft AD nur verwenden möchten, fahren Sie mit fortSchritt 7: Erstellen oder ändern Sie eine RDS for Db2-DB-Instance.

Um die Kerberos-Authentifizierung mit Ihrem selbstverwalteten Active Directory zu aktivieren, müssen Sie den Parameter `AWS_MANAGED_AD_WITH_TRUST` in Ihrer `rds.active_directory_configuration` Parametergruppe auf setzen. Standardmäßig ist dieser Parameter auf Nur `AWS_MANAGED_AD` für die Verwendung von AWS Managed Microsoft AD festgelegt.

Informationen zum Ändern von DB-Parametern finden Sie unterÄndern der Parameter in Parametergruppen.

Schritt 7: Erstellen oder ändern Sie eine RDS for Db2-DB-Instance

Erstellen oder ändern Sie eine RDS for Db2-DB-Instance zur Verwendung mit Ihrem Verzeichnis. Sie können die AWS Management Console, oder die RDS-API verwenden AWS CLI, um eine DB-Instance einem Verzeichnis zuzuordnen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Erstellen Sie mithilfe der Konsole, des create-db-instanceBefehls oder der CreateDBInstanceAPI-Operation eine neue RDS for Db2-DB-Instance. Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine vorhandene RDS for Db2-DB-Instance mithilfe der Konsole, des modify-db-instanceBefehls oder der Operation „DBInstanceAPI modifizieren“. Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie mithilfe der Konsole, des restore-db-instance-from-db-snapshotBefehls oder der RestoreDBInstanceFromDBSnapshotAPI-Operation eine RDS for Db2-DB-Instance aus einem DB-

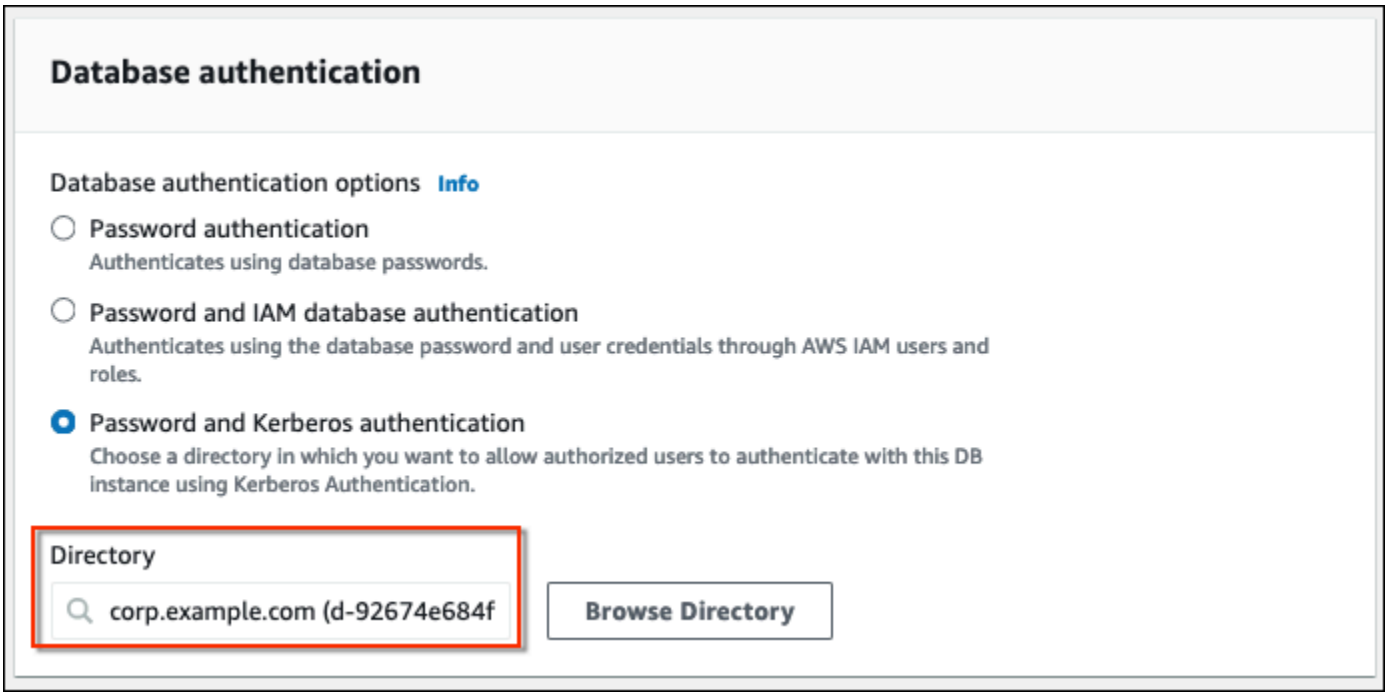
Snapshot wieder her. Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.

- Stellen Sie eine RDS for Db2-DB-Instance point-in-time mithilfe der Konsole, des restore-db-instance-to-point-in-timeBefehls oder der RestoreDBInstanceToPointInTimeAPI-Operation wieder her. Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

KerberosDie Authentifizierung wird nur für RDS für Db2-DB-Instances in einer VPC unterstützt. Die DB-Instance kann sich in derselben VPC wie das Verzeichnis oder in einer anderen VPC befinden. Die DB-Instance muss eine Sicherheitsgruppe verwenden, die Ein- und Ausgänge innerhalb der VPC des Verzeichnisses zulässt, damit die DB-Instance mit dem Verzeichnis kommunizieren kann.

Konsole

Wenn Sie die Konsole verwenden, um eine DB-Instance zu erstellen, zu ändern oder wiederherzustellen, wählen Sie im Abschnitt Datenbankauthentifizierung die Option Passwort und Kerberos Authentifizierung aus. Dann wählen Sie Verzeichnis durchsuchen. Wählen Sie das Verzeichnis aus oder wählen Sie Verzeichnis erstellen, um den Directory Service zu verwenden.



AWS CLI

Wenn Sie den verwenden AWS CLI, sind die folgenden Parameter erforderlich, damit die DB-Instance das von Ihnen erstellte Verzeichnis verwenden kann:

- Verwenden Sie für den `--domain` Parameter die Domänen-ID (“ `d-*` "Identifier), die bei der Erstellung des Verzeichnisses generiert wurde.
- Verwenden Sie für den `--domain-iam-role-name`-Parameter die von Ihnen erstellte Rolle, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet.

Im folgenden Beispiel wird eine DB-Instance so geändert, dass sie ein Verzeichnis verwendet. Ersetzen Sie die folgenden Platzhalter im Beispiel durch Ihre eigenen Werte:

- *`db_instance_name`*— Der Name Ihrer RDS for Db2-DB-Instance.
- *`directory_id`*— Die ID des AWS Directory Service for Microsoft Active Directory Verzeichnisses, das Sie erstellt haben.
- *`role_name`*— Der Name der IAM-Rolle, die Sie erstellt haben.

```
aws rds modify-db-instance --db-instance-identifier db_instance_name --domain
 d-directory_id --domain-iam-role-name role_name
```

> **Important**
>
> Wenn Sie eine DB-Instance ändern, um die Kerberos Authentifizierung zu aktivieren, starten Sie die DB-Instance neu, nachdem Sie die Änderung vorgenommen haben.

Schritt 8: Rufen Sie die Active Directory-Gruppen-SID ab in PowerShell

Eine Sicherheits-ID (SID) identifiziert einen Sicherheitsprinzipal oder eine Sicherheitsgruppe eindeutig. Wenn eine Sicherheitsgruppe oder ein Konto in Active Directory erstellt wird, weist Active Directory der Gruppe eine SID zu. Um die SID der AD-Sicherheitsgruppe aus Active Directory abzurufen, verwenden Sie das `Get-ADGroup` Cmdlet auf einem Windows-Client-Computer, der Teil der Active Directory-Domäne ist. Der `Identity` Parameter gibt den Namen der Active Directory-Gruppe an, für die Sie die SID benötigen.

Das folgende Beispiel gibt die SID der Active Directory-Gruppe zurück`adgroup1`.

```
C:\Users\Admin> Get-ADGroup -Identity adgroup1 | select SID

             SID
-----------------------------------------------
```

```
S-1-5-21-3168537779-1985441202-1799118680-1612
```

Sie müssen diese Zuordnung für alle Gruppen generieren, die für die Datenbank relevant sind.

Schritt 9: Fügen Sie SID zu GroupName Zuordnungen zu Ihrer RDS for Db2-DB-Instance hinzu

Sie müssen die SID zu den im vorherigen Schritt erstellten GroupName Zuordnungen zu Ihrer RDS for Db2-DB-Instance hinzufügen. Rufen Sie für jede Zuordnung die folgende gespeicherte Prozedur auf. Ersetzen Sie das *SID* und *group_name* durch Ihre eigenen Informationen.

```
db2 connect to rdsadmin
db2 "call rdsadmin.set_sid_group_mapping(?, 'SID','group_name')"
```

Weitere Informationen finden Sie unter rdsadmin.set_sid_group_mapping.

Informationen zur Überprüfung des Aufgabenstatus finden Sie unterrdsadmin.get_task_status.

Schritt 10: Konfigurieren Sie einen Db2-Client

Um einen Db2-Client zu konfigurieren

1. Erstellen Sie eine /etc/krb5.conf-Datei (oder eine gleichwertige Datei), die auf die Domäne verweist.

   > **Note**
   >
   > Erstellen Sie für Windows-Betriebssysteme eine Datei C:\windows\krb5.ini.

2. Stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und fließen kann AWS Directory Service. Verwenden Sie ein Netzwerkdienstprogramm, z. B. Netcat für die folgenden Aufgaben:

   a. Überprüfen Sie den Datenverkehr über DNS für Port 53.

   b. Überprüfen Sie den Datenverkehr TCP/UDP für Port 53 und fürKerberos, einschließlich der Ports 88 und 464 für AWS Directory Service.

3. Stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und der DB-Instance über den Datenbank-Port fließen kann. Sie können den Befehl verwendendb2, um eine Verbindung zur Datenbank herzustellen und darauf zuzugreifen.

Das folgende Beispiel ist der Inhalt der Datei /etc/krb5.conf für: AWS Managed Microsoft AD

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
kdc = example.com
admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Herstellen einer Verbindung zu Amazon RDS for Db2 mit Kerberos Authentifizierung

Gehen Sie wie folgt vor, um eine Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance herzustellen mit Kerberos Authentifizierung.

Um eine Verbindung zu RDS für Db2 herzustellen mit Kerberos Authentifizierung

1. Führen Sie an einer Eingabeaufforderung den folgenden -Befehl aus. Ersetzen Sie im folgenden Beispiel *username* mit deinem Microsoft Active Directory Nutzername.

   ```
   kinit username
   ```

2. Wenn die RDS for Db2-DB-Instance eine öffentlich zugängliche verwendetVPC, fügen Sie die IP-Adresse für Ihren DB-Instance-Endpunkt zu Ihrer `/etc/hosts` Datei auf dem EC2 Amazon-Client hinzu. Das folgende Beispiel ruft die IP-Adresse ab und fügt sie dann der `/etc/hosts` Datei hinzu.

   ```
   % dig +short Db2-endpoint.AWS-Region.rds.amazonaws.com
   ;; Truncated, retrying in TCP mode.
   ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
   34.210.197.118

   % echo "34.210.197.118  Db2-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/hosts
   ```

3. Verwenden Sie den folgenden Befehl, um sich bei einer RDS for Db2-DB-Instance anzumelden, die verknüpft ist mit Active Directory. Ersetze *database_name* durch den Namen Ihrer RDS für Db2-Datenbank.

   ```
   db2 connect to database_name
   ```

Verwaltung Ihrer Amazon RDS for Db2-DB-Instance

Dieses Thema behandelt die allgemeinen Verwaltungsaufgaben, die Sie mit einer Amazon RDS for Db2-DB-Instance ausführen. Einige Aufgaben sind für alle Amazon RDS-DB-Instances identisch. Andere Aufgaben sind spezifisch für RDS for Db2.

Die folgenden Aufgaben sind allen RDS-Datenbanken gemeinsam. Es gibt auch spezielle Aufgaben für RDS for Db2, wie z. B. die Verbindung zu einer RDS for Db2-Datenbank mit einem Standard-SQL-Client.

Aufgabenbereich	Relevante Dokumentation
Instance-Klassen, Speicher und PIOPS Wenn Sie eine Produktionsinstance erstellen, erfahren Sie, wie Instance-Klassen, Speichertypen und bereitgestellte IOPS in Amazon RDS funktionieren.	Amazon RDS-Speichertypen
Multi-AZ-Bereitstellungen Bei einer DB-Instance für die Produktion sollten Multi-AZ-Bereitstellungen eingesetzt werden. Multi-AZ-Bereitstellungen bieten eine erhöhte Verfügbarkeit, eine längere Lebensdauer von Daten sowie eine höhere Fehlertoleranz für DB-Instances.	Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS
Amazon VPC Wenn Sie AWS-Konto über eine standardmäßige Virtual Private Cloud (VPC) verfügen, wird Ihre DB-Instance automatisch in der Standard-VPC erstellt. Wenn Ihr Konto über keine Standard-VPC verfügt und Sie die DB-Instance in einer VPC erstellen möchten, müssen Sie zunächst die VPC und Subnetz-Gruppen erstellen.	Arbeiten mit einer DB-Instance in einer VPC
Sicherheitsgruppen Standardmäßig verwenden DB-Instances eine Firewall, die den Zugriff verhindert. Stellen Sie sicher, dass Sie eine Sicherhei	Zugriffskontrolle mit Sicherheitsgruppen

Aufgabenbereich	Relevante Dokumentation
tsgruppe mit den korrekten IP-Adressen und Netzwerkkonfigurationen erstellen, um auf die DB-Instance zugreifen zu können.	
Parametergruppen Da Ihre RDS for Db2-DB-Instance erfordert, dass Sie die `rds.ibm_site_id` Parameter `rds.ibm_customer_id` und hinzufügen, erstellen Sie eine Parametergruppe, bevor Sie die DB-Instance erstellen. Wenn Ihre DB-Instance weitere spezifische Datenbankparameter benötigt, fügen Sie diese ebenfalls zu dieser Parametergruppe hinzu, bevor Sie die DB-Instance erstellen.	Hinzufügen IBM IDs zu einer Parametergruppe für RDS für Db2-DB-Instances Parametergruppen für Amazon RDS
Optionsgruppen Wenn Ihre DB-Instance bestimmte Datenbankoptionen erfordert, erstellen Sie eine Optionsgruppe, bevor Sie die DB-Instance erstellen.	Optionen für Amazon RDS für Db2-DB-Instances
Herstellen einer Verbindung mit einer DB-Instance Nachdem Sie eine Sicherheitsgruppe erstellt und sie einer DB-Instance zugeordnet haben, können Sie mit jeder Standard-SQL-Clientanwendung eine Verbindung zur DB-Instance herstellen, z. IBM Db2 CLP	Verbindung zu Ihrer Db2-DB-Instance herstellen
Backup und Wiederherstellung Sie können Ihre DB-Instance so konfigurieren, dass automatische Speicher-Backups oder manuelle Speicher-Snapshots erstellt und dann Instances aus den Backups oder Snapshots wiederhergestellt werden.	Daten sichern, wiederherstellen und exportieren

Aufgabenbereich	Relevante Dokumentation
Überwachung Sie können eine RDS for Db2-DB-Instance mit überwachen. IBM Db2 Data Management Console Sie können eine RDS for Db2-DB-Instance auch mithilfe von CloudWatch Amazon RDS-Metriken, Ereignissen und erweiterter Überwachung überwachen.	Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 Data Management Console Metriken in der RDS Amazon-Konsole anzeigen RDSAmazon-Ereignisse anzeigen Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung)
Protokolldateien Sie können auf die Protokolldateien für Ihre RDS for Db2-DB-Instance zugreifen.	Überwachung von Amazon RDS Amazon

Themen

- Ausführen allgemeiner Systemaufgaben für Amazon RDS for Db2-DB-Instances
- Durchführung allgemeiner Datenbankaufgaben für Amazon RDS for Db2-DB-Instances

Ausführen allgemeiner Systemaufgaben für Amazon RDS for Db2-DB-Instances

Sie können bestimmte allgemeine Datenbankadministratoraufgaben im Zusammenhang mit dem System auf Ihren RDS Amazon-DB-Instances ausführen, auf denen Db2 ausgeführt wird. Um ein Managed Service-Erlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances und schränkt den Zugriff auf bestimmte Systemprozeduren und Tabellen ein, für die erweiterte Rechte erforderlich sind.

Informationen zum Erteilen und Widerrufen von Rechten und zum Anhängen an die entfernte Datenbank RDS für Db2 finden Sie in den folgenden Themen.

Themen

- Erteilen und Widerrufen von Rechten für RDS for Db2
- Verbindung zur Remote-Datenbankinstanz RDS für DB2 herstellen

Einen benutzerdefinierten Datenbank-Endpunkt erstellen

Wenn Sie zu Amazon RDS for Db2 migrieren, können Sie einen benutzerdefinierten Datenbank-Endpunkt verwenden, URLs um Änderungen an Ihrer Anwendung zu minimieren. Wenn Sie beispielsweise `db2.example.com` als Ihren aktuellen DNS Datensatz verwenden, können Sie ihn zu Amazon Route 53 hinzufügen. In Route 53 können Sie private gehostete Zonen verwenden, um Ihren aktuellen DNS Datenbank-Endpunkt einem RDS for Db2-Datenbank-Endpunkt zuzuordnen. Informationen zum Hinzufügen eines benutzerdefinierten `CNAME` Datensatzes `A` oder Datensatzes für einen RDS Amazon-Datenbank-Endpunkt finden Sie unter Registrierung und Verwaltung von Domains mithilfe von Amazon Route 53 im Amazon Route 53-Entwicklerhandbuch.

> **Note**
>
> Wenn Sie Ihre Domain nicht auf Route 53 übertragen können, können Sie Ihren DNS Anbieter verwenden, um einen `CNAME` Datensatz für den RDS for Db2-Datenbank-Endpunkt URL zu erstellen. Schlagen Sie in der Dokumentation Ihres DNS Anbieters nach.

Erteilen und Widerrufen von Rechten für RDS for Db2

Benutzer erhalten Zugriff auf Datenbanken durch Mitgliedschaft in Gruppen, die Datenbanken zugeordnet sind.

Gehen Sie wie folgt vor, um Rechte zur Steuerung des Zugriffs auf Ihre Datenbank zu gewähren oder zu entziehen.

Bei diesen Verfahren wird die IBM Db2 CLP Ausführung auf einem lokalen Computer verwendet, um eine Verbindung zu einer RDS for Db2-DB-Instance herzustellen. Achten Sie darauf, den TCPIP-Knoten und die Datenbank zu katalogisieren, um eine Verbindung zu Ihrer RDS for Db2-DB-Instance herzustellen, die auf Ihrem lokalen Computer ausgeführt wird. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 CLP.

Themen

- Gewähren Sie einem Benutzer Zugriff auf Ihre Datenbank
- Das Passwort eines Benutzers ändern
- Hinzufügen von Gruppen zu einem Benutzer
- Gruppen von einem Benutzer entfernen
- Einen Benutzer entfernen
- Benutzer auflisten
- Erstellen einer Rolle
- Eine Rolle gewähren
- Eine Rolle widerrufen
- Eine Rolle löschen
- Datenbankautorisierung gewähren
- Widerrufen der Datenbankautorisierung

Gewähren Sie einem Benutzer Zugriff auf Ihre Datenbank

Um einem Benutzer Zugriff auf Ihre Datenbank zu gewähren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

   Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

   ```
   Database Connection Information

   Database server        = DB2/LINUXX8664 11.5.8.0
   SQL authorization ID   = ADMIN
   Local database alias   = RDSADMIN
   ```

2. Fügen Sie Ihrer Autorisierungsliste einen Benutzer hinzu, indem Sie anrufen`rdsadmin.add_user`. Weitere Informationen finden Sie unter rdsadmin.add_user.

   ```
   db2 "call rdsadmin.add_user(
   ```

```
    'username',
    'password',
    'group_name,group_name')"
```

3. (Optional) Fügen Sie dem Benutzer weitere Gruppen hinzu, indem Sie anrufen`rdsadmin.add_groups`. Weitere Informationen finden Sie unter rdsadmin.add_groups.

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

4. Bestätigen Sie die Berechtigungen, die dem Benutzer zur Verfügung stehen. Ersetzen Sie im folgenden Beispiel *rds_database_aliasmaster_user*, und *master_password* durch Ihre eigenen Informationen. Ersetzen Sie es auch *username* durch den Benutzernamen des Benutzers.

```
db2 terminate
db2 connect to rds_database_alias user master_user using master_password
db2 "SELECT SUBSTR(AUTHORITY,1,20) AUTHORITY, D_USER, D_GROUP, D_PUBLIC
       FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('username', 'U') ) AS
 T
       ORDER BY AUTHORITY"
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
AUTHORITY            D_USER D_GROUP D_PUBLIC
-------------------- ------ ------- --------
ACCESSCTRL           N      N       N
BINDADD              N      N       N
CONNECT              N      N       N
CREATETAB            N      N       N
CREATE_EXTERNAL_ROUT N      N       N
CREATE_NOT_FENCED_RO N      N       N
CREATE_SECURE_OBJECT N      N       N
DATAACCESS           N      N       N
DBADM                N      N       N
EXPLAIN              N      N       N
IMPLICIT_SCHEMA      N      N       N
LOAD                 N      N       N
QUIESCE_CONNECT      N      N       N
SECADM               N      N       N
SQLADM               N      N       N
```

```
SYSADM                  *       N       *
SYSCTRL                 *       N       *
SYSMAINT                *       N       *
SYSMON                  *       N       *
WLMADM                  N       N       N
```

5. Erteilen Sie der GruppeROLE_NULLID_PACKAGES, ROLE_PROCEDURES zu der Sie den Benutzer hinzugefügt habenROLE_TABLESPACES, die Rollen RDS für Db2 und. Weitere Informationen finden Sie unter Standardrollen von Amazon RDS für Db2.

> **Note**
>
> Wir erstellen RDS für Db2-DB-Instances im RESTRICTIVE Modus. Daher übernimmt der RDS für Db2 die Rollen ROLE_NULLID_PACKAGESROLE_TABLESPACES, und ROLE_PROCEDURES gewährt Ausführungsberechtigungen für NULLID Pakete für IBM Db2 CLP und. Dynamic SQL Diese Rollen gewähren auch Benutzerrechte für Tablespaces.

   a. Connect zu Ihrer Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *database_namemaster_user*, und *master_password* durch Ihre eigenen Informationen.

```
db2 connect to database_name user master_user using master_password
```

   b. Weisen Sie die Rolle einer Gruppe ROLE_NULLED_PACKAGES zu. Im folgenden Beispiel *group_name* ersetzen Sie sie durch den Namen der Gruppe, zu der Sie die Rolle hinzufügen möchten.

```
db2 "grant role ROLE_NULLID_PACKAGES to group group_name"
```

   c. Weisen Sie derselben Gruppe die Rolle ROLE_TABLESPACES zu. Im folgenden Beispiel *group_name* ersetzen Sie es durch den Namen der Gruppe, zu der Sie die Rolle hinzufügen möchten.

```
db2 "grant role ROLE_TABLESPACES to group group_name"
```

d. Weisen Sie derselben Gruppe die Rolle `ROLE_PROCEDURES` zu. Im folgenden Beispiel *`group_name`* ersetzen Sie es durch den Namen der Gruppe, zu der Sie die Rolle hinzufügen möchten.

```
db2 "grant role ROLE_PROCEDURES to group group_name"
```

6. Gewähren Sie der Gruppe `connect bindaddcreatetab`, `IMPLICIT_SCHEMA` zu der Sie den Benutzer hinzugefügt haben, Berechtigungen, und Berechtigungen. Ersetzen Sie im folgenden Beispiel *`group_name`* durch den Namen der zweiten Gruppe, zu der Sie den Benutzer hinzugefügt haben.

```
db2 "grant usage on workload SYSDEFAULTUSERWORKLOAD to public"
db2 "grant connect, bindadd, createtab, implicit_schema on database to
 group group_name"
```

7. Wiederholen Sie die Schritte 4 bis 6 für jede weitere Gruppe, zu der Sie den Benutzer hinzugefügt haben.
8. Testen Sie den Zugriff des Benutzers, indem Sie eine Verbindung als Benutzer herstellen, eine Tabelle erstellen, Werte in die Tabelle einfügen und Daten aus der Tabelle zurückgeben. Im folgenden Beispiel ersetzen Sie *`rds_database_aliasusername`*, und *`password`* durch den Namen der Datenbank sowie den Benutzernamen und das Passwort des Benutzers.

```
db2 connect to rds_database_alias user username using password
db2 "create table t1(c1 int not null)"
db2 "insert into t1 values (1),(2),(3),(4)"
db2 "select * from t1"
```

Das Passwort eines Benutzers ändern

So ändern Sie das Passwort eines Benutzers

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Ändern Sie das Passwort, indem Sie anrufen`rdsadmin.change_password`. Weitere Informationen finden Sie unter rdsadmin.change_password.

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

Hinzufügen von Gruppen zu einem Benutzer

Um einem Benutzer Gruppen hinzuzufügen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Fügen Sie einem Benutzer Gruppen hinzu, indem Sie anrufen`rdsadmin.add_groups`. Weitere Informationen finden Sie unter rdsadmin.add_groups.

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Gruppen von einem Benutzer entfernen

Um Gruppen von einem Benutzer zu entfernen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Entfernen Sie Gruppen, indem Sie anrufen`rdsadmin.remove_groups`. Weitere Informationen finden Sie unter rdsadmin.remove_groups.

```
db2 "call rdsadmin.remove_groups(
```

```
    'username',
    'group_name,group_name')"
```

Einen Benutzer entfernen

Um einen Benutzer aus der Autorisierungsliste zu entfernen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Entfernen Sie einen Benutzer aus Ihrer Autorisierungsliste, indem Sie anrufen`rdsadmin.remove_user`. Weitere Informationen finden Sie unter rdsadmin.remove_user.

   ```
   db2 "call rdsadmin.remove_user('username')"
   ```

Benutzer auflisten

Rufen Sie die `rdsadmin.list_users` gespeicherte Prozedur auf, um Benutzer auf einer Autorisierungsliste aufzulisten. Weitere Informationen finden Sie unter rdsadmin.list_users.

```
db2 "call rdsadmin.list_users()"
```

Erstellen einer Rolle

Sie können die rdsadmin.create_role gespeicherte Prozedur verwenden, um eine Rolle zu erstellen.

So erstellen Sie eine Rolle

1. Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Stellen Sie Db2 für die Ausgabe von Inhalten ein.

```
db2 set serveroutput on
```

3. Erstellen Sie eine Rolle. Weitere Informationen finden Sie unter the section called “rdsadmin.create_role”.

```
db2 "call rdsadmin.create_role(
    'database_name',
    'role_name')"
```

4. Stellen Sie Db2 so ein, dass kein Inhalt ausgegeben wird.

```
db2 set serveroutput off
```

Eine Rolle gewähren

Sie können die rdsadmin.grant_role gespeicherte Prozedur verwenden, um einer Rolle, einem Benutzer oder einer Gruppe eine Rolle zuzuweisen.

Um eine Rolle zuzuweisen

1. Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre Informationen.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Stellen Sie Db2 für die Ausgabe von Inhalten ein.

```
db2 set serveroutput on
```

3. Weisen Sie eine Rolle zu. Weitere Informationen finden Sie unter the section called “rdsadmin.grant_role”.

```
db2 "call rdsadmin.grant_role(
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

4. Stellen Sie Db2 so ein, dass kein Inhalt ausgegeben wird.

```
db2 set serveroutput off
```

Eine Rolle widerrufen

Sie können die rdsadmin.revoke_role gespeicherte Prozedur verwenden, um einer Rolle, einem Benutzer oder einer Gruppe eine Rolle zu entziehen.

Um eine Rolle zu widerrufen

1. Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Widerrufen Sie eine Rolle. Weitere Informationen finden Sie unter the section called “rdsadmin.revoke_role”.

   ```
   db2 "call rdsadmin.revoke_role(
       ?,
       'database_name',
       'role_name',
       'grantee')"
   ```

Eine Rolle löschen

Sie können die rdsadmin.drop_role gespeicherte Prozedur verwenden, um eine Rolle zu löschen.

Um eine Rolle zu löschen

1. Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Löschen Sie eine Rolle. Weitere Informationen finden Sie unter the section called “rdsadmin.drop_role”.

   ```
   db2 "call rdsadmin.drop_role(
   ```

```
    ?,
    'database_name',
    'role_name')"
```

Datenbankautorisierung gewähren

Der Masterbenutzer, der über eine DBADM Autorisierung verfügt, kann einer Rolle DBADMACCESSCTRL, einem Benutzer oder einer Gruppe eine DATAACCESS Autorisierung erteilen oder eine Autorisierung erteilen.

Um eine Datenbankautorisierung zu erteilen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur rdsadmin Datenbank her. Ersetzen Sie im folgenden Beispiel *master_username* und *master_password* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Gewähren Sie einem Benutzer Zugriff, indem Sie anrufenrdsadmin.dbadm_grant. Weitere Informationen finden Sie unter rdsadmin.dbadm_grant.

   ```
   db2 "call rdsadmin.dbadm_grant(
       ?,
       'database_name,
       'authorization',
       'grantee')"
   ```

Beispiel für einen Anwendungsfall

Das folgende Verfahren führt Sie durch das Erstellen einer Rolle, das Erteilen der DBADM Autorisierung für die Rolle, das Zuweisen der Rolle an einen Benutzer und das Erteilen der Rolle an eine Gruppe.

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur rdsadmin Datenbank her. Ersetzen Sie im folgenden Beispiel *master_username* und *master_password* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Erstellen Sie eine Rolle namens PROD_ROLE für eine Datenbank namensTESTDB. Weitere Informationen finden Sie unter rdsadmin.create_role.

```
db2 "call rdsadmin.create_role(
    'TESTDB',
    'PROD_ROLE')"
```

3. Weisen Sie die Rolle einem Benutzer mit dem Namen zuPROD_USER. Er PROD_USER erhält die Administratorberechtigung zur Zuweisung von Rollen. Weitere Informationen finden Sie unter rdsadmin.grant_role.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'PROD_ROLE',
    'USER PROD_USER',
    'Y')"
```

4. (Optional) Geben Sie zusätzliche Autorisierungen oder Rechte ein. Im folgenden Beispiel wird eine DBADM Autorisierung PROD_ROLE für eine Rolle erteilt, die nach einer Datenbank namens benannt istFUNDPROD. Weitere Informationen finden Sie unter rdsadmin.dbadm_grant.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'FUNDPROD',
    'DBADM',
    'ROLE PROD_ROLE')"
```

5. Beenden Sie Ihre Sitzung.

```
db2 terminate
```

6. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur TESTDB Datenbank her. Ersetzen Sie im folgenden Beispiel *master_username* und *master_password* durch Ihre eigenen Informationen.

```
db2 connect to TESTDB user master_username using master_password
```

7. Fügen Sie der Rolle weitere Autorisierungen hinzu.

```
db2 "grant connect, implicit_schema on database to role PROD_ROLE"
```

8. Erteilen Sie PROD_ROLE die Rolle einer Gruppe.

```
db2 "grant role PROD_ROLE to group PRODGRP"
```

Benutzer, die der Gruppe angehören, PRODGRP können jetzt Aktionen wie das Herstellen einer Verbindung zur TESTDB Datenbank, das Erstellen von Tabellen oder das Erstellen von Schemas ausführen.

Widerrufen der Datenbankautorisierung

Der Hauptbenutzer, der über eine DBADM Autorisierung verfügt DBADMACCESSCTRL, kann die DATAACCESS Autorisierung einer Rolle, einem Benutzer oder einer Gruppe entziehen.

Um die Datenbankautorisierung zu widerrufen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Widerrufen Sie den Benutzerzugriff, indem Sie anrufen`rdsadmin.dbadm_revoke`. Weitere Informationen finden Sie unter rdsadmin.dbadm_revoke.

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Verbindung zur Remote-Datenbankinstanz RDS für DB2 herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu Ihrer Remote RDS for Db2-DB-Instance herzustellen und Operationen auszuführen`get snapshot`.

Um eine Verbindung RDS zur Remote-DB2-DB-Instance herzustellen

1. Führen Sie eine clientseitige Ausführung aus IBM Db2 CLP Sitzung. Hinweise zur Katalogisierung Ihrer RDS DB2-DB-Instance und -Datenbank finden Sie unter. Herstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 CLP Notieren Sie sich den Master-Benutzernamen und das Master-Passwort RDS für Ihre DB2-DB-Instance.
2. Stellen Sie eine Verbindung zur RDS für die DB2-DB-Instance her. Ersetzen Sie im folgenden Beispiel *node_name*, *master_username*, und *master_password* durch den TCPIP Knotennamen, den Sie katalogisiert haben, sowie den Master-Benutzernamen und das Master-Passwort RDS für Ihre DB2-DB-Instance.

```
db2 attach to node_name user master_username using master_password
```

Nachdem Sie eine Verbindung zur Remote-Datenbankinstanz RDS für DB2 hergestellt haben, können Sie die folgenden Befehle und andere Befehle ausführen. `get snapshot` Weitere Informationen finden Sie unter GET SNAPSHOT Befehl in der IBM Db2 -Dokumentation.

```
db2 list applications
db2 get snapshot for all databases
db2 get snapshot for database manager
db2 get snapshot for all applications
```

Durchführung allgemeiner Datenbankaufgaben für Amazon RDS for Db2-DB-Instances

Sie können bestimmte allgemeine DBA Aufgaben im Zusammenhang mit Datenbanken auf Ihren Amazon RDS for Db2-DB-Instances ausführen. Um ein Managed-Serviceerlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances. Außerdem kann der Masterbenutzer keine Befehle oder Dienstprogramme ausführen `SYSADMSYSMAINT`, für die Berechtigungen erforderlich `SYSCTRL` sind.

Informationen zu allgemeinen Aufgaben für Pufferpools, Datenbanken und Tablespaces finden Sie in den folgenden Themen.

Themen

- Allgemeine Aufgaben für Pufferpools

- Allgemeine Aufgaben für Datenbanken
- Allgemeine Aufgaben für Tablespaces

Allgemeine Aufgaben für Pufferpools

Sie können Pufferpools für eine RDS for Db2-Datenbank erstellen, ändern oder löschen. Das Erstellen, Ändern oder Löschen von Pufferpools erfordert eine höhere SYSCTRL AutoritätsebeneSYSADM, die dem Masterbenutzer nicht zur Verfügung steht. Verwenden Sie stattdessen gespeicherte Amazon RDS-Prozeduren.

Sie können auch Pufferpools leeren.

Themen

- Einen Pufferpool erstellen
- Einen Pufferpool ändern
- Einen Pufferpool löschen
- Die Pufferpools leeren

Einen Pufferpool erstellen

Rufen Sie die `rdsadmin.create_bufferpool` gespeicherte Prozedur auf, um einen Pufferpool für Ihre RDS for Db2-Datenbank zu erstellen. Weitere Informationen finden Sie in der CREATE BUFFERPOOLErklärung in der IBM Db2 Dokumentation.

Um einen Pufferpool zu erstellen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Erstellen Sie einen Pufferpool, indem Sie aufrufen`rdsadmin.create_bufferpool`. Weitere Informationen finden Sie unter rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool(
       'database_name',
       'buffer_pool_name',
   ```

```
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

Einen Pufferpool ändern

Um einen Pufferpool für Ihre RDS for Db2-Datenbank zu ändern, rufen Sie die `rdsadmin.alter_bufferpool` gespeicherte Prozedur auf. Weitere Informationen finden Sie in der ALTER BUFFERPOOLErklärung in der IBM Db2 Dokumentation.

Um einen Pufferpool zu ändern

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Ändern Sie einen Pufferpool, indem Sie aufrufen`rdsadmin.alter_bufferpool`. Weitere Informationen finden Sie unter rdsadmin.alter_bufferpool.

   ```
   db2 "call rdsadmin.alter_bufferpool(
       'database_name',
       'buffer_pool_name',
       buffer_pool_size,
       'immediate',
       'automatic',
       change_number_blocks,
       number_block_pages,
       block_size)"
   ```

Einen Pufferpool löschen

Um einen Pufferpool für Ihre RDS for Db2-Datenbank zu löschen, rufen Sie die `rdsadmin.drop_bufferpool` gespeicherte Prozedur auf. Weitere Informationen finden Sie in der IBM Db2 Dokumentation unter Löschen von Pufferpools.

> **Important**
>
> Stellen Sie sicher, dass dem Pufferpool, den Sie löschen möchten, keine Tablespaces zugewiesen sind.

Um einen Pufferpool zu löschen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Löschen Sie einen Pufferpool, indem Sie aufrufen`rdsadmin.drop_bufferpool`. Weitere Informationen finden Sie unter rdsadmin.drop_bufferpool.

   ```
   db2 "call rdsadmin.drop_bufferpool(
       'database_name',
       'buffer_pool_name')"
   ```

Die Pufferpools leeren

Sie können die Pufferpools leeren, um einen Checkpoint zu erzwingen, sodass RDS for Db2 Seiten aus dem Speicher in den Speicher schreibt.

> **Note**
>
> Sie müssen die Pufferpools nicht leeren. Db2 schreibt Protokolle synchron, bevor es Transaktionen festschreibt. Die Dirty Pages befinden sich möglicherweise immer noch in einem Pufferpool, aber Db2 schreibt sie asynchron in den Speicher. Selbst wenn das System unerwartet heruntergefahren wird, führt Db2 beim Neustart der Datenbank automatisch eine Wiederherstellung nach einem Absturz durch. Während der Wiederherstellung nach einem Absturz schreibt Db2 festgeschriebene Änderungen in die Datenbank oder macht Änderungen für nicht festgeschriebene Transaktionen rückgängig.

Um die Pufferpools zu leeren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zu Ihrer Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_aliasmaster_username*, und durch Ihre *master_password* eigenen Informationen.

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Leeren Sie die Pufferpools.

```
db2 flush bufferpools all
```

Allgemeine Aufgaben für Datenbanken

Sie können Datenbanken auf Ihrer RDS for Db2-DB-Instance erstellen, löschen oder wiederherstellen. Für das Erstellen, Löschen oder Wiederherstellen von Datenbanken sind höhere Berechtigungen erforderlich, SYSADM die dem Masterbenutzer nicht zur Verfügung stehen. Verwenden Sie stattdessen gespeicherte Amazon RDS-Prozeduren.

Sie können auch allgemeine Verwaltungsaufgaben wie Überwachung, Wartung und Erfassung von Informationen über Ihre Datenbanken ausführen.

Themen

- Erstellen einer Datenbank
- Einstellungen für eine Datenbank konfigurieren
- Datenbankparameter ändern
- Konfiguration der Protokollaufbewahrung
- Protokollinformationen auflisten
- Eine Datenbank deaktivieren
- Eine Datenbank aktivieren
- Eine Datenbank reaktivieren
- Eine Datenbank löschen
- Wiederherstellen einer Datenbank
- Datenbanken auflisten

- Sammeln von Informationen über Datenbanken
- Anwendungen aus Datenbanken abzwingen
- Generieren von Leistungsberichten

Erstellen einer Datenbank

Rufen Sie die `rdsadmin.create_database` gespeicherte Prozedur auf, um eine Datenbank auf Ihrer RDS for Db2-DB-Instance zu erstellen. Weitere Informationen finden Sie unter CREATE DATABASEBefehl in der IBM Db2 Dokumentation.

> **Note**
>
> Wenn Sie den `db2_compatibility_vector` Parameter ändern möchten, ändern Sie den Parameter, bevor Sie eine Datenbank erstellen. Weitere Informationen finden Sie unter Einstellung des Parameters db2_compatibility_vector.

Eine Datenbank erstellen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Erstellen Sie eine Datenbank, indem Sie aufrufen`rdsadmin.create_database`. Weitere Informationen finden Sie unter rdsadmin.create_database.

   ```
   db2 "call rdsadmin.create_database('database_name')"
   ```

3. (Optional) Erstellen Sie zusätzliche Datenbanken, indem Sie `rdsadmin.create_database` jede Datenbank aufrufen, die Sie erstellen möchten. Jede Db2-DB-Instance kann bis zu 50 Datenbanken enthalten. Weitere Informationen finden Sie unter rdsadmin.create_database.

   ```
   db2 "call rdsadmin.create_database('database_name')"
   ```

4. (Optional) Vergewissern Sie sich, dass Ihre Datenbank mit einer der folgenden Methoden erstellt wurde:

- Rufen Sie die folgende Seite auf `rdsadmin.list_databases`. Weitere Informationen finden Sie unter rdsadmin.list_databases.
- Führen Sie den folgenden SQL-Befehl aus:

```
db2 "select varchar(r.task_type,25) as task_type, r.database_name,
    varchar(r.lifecycle,15) as lifecycle, r.created_at, r.database_name,
    varchar(bson_to_json(task_input_params),256) as input_params,
    varchar(r.task_output,1024) as task_output
    from table(rdsadmin.get_task_status(null,null,'create_database'))
    as r order by created_at desc"
```

Einstellungen für eine Datenbank konfigurieren

Rufen Sie die `rdsadmin.set_configuration` gespeicherte Prozedur auf, um die Einstellungen für eine Datenbank auf Ihrer RDS for Db2-DB-Instance zu konfigurieren. Sie könnten beispielsweise die Anzahl der Puffer oder Puffermanipulatoren konfigurieren, die während eines Wiederherstellungsvorgangs erstellt werden sollen.

Um Einstellungen für eine Datenbank zu konfigurieren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. (Optional) Überprüfen Sie Ihre aktuellen Konfigurationseinstellungen, indem Sie anrufen`rdsadmin.show_configuration`. Weitere Informationen finden Sie unter the section called “rdsadmin.show_configuration”.

```
db2 "call rdsadmin.show_configuration('name')"
```

3. Konfigurieren Sie die Einstellungen für die Datenbank, indem Sie anrufen`rdsadmin.set_configuration`. Weitere Informationen finden Sie unter the section called “rdsadmin.set_configuration”.

```
db2 "call rdsadmin.set_configuration(
    'name',
```

```
    'value')"
```

Datenbankparameter ändern

Amazon RDS for Db2 verwendet drei Arten von Parametern: Datenbank-Manager-Konfigurationsparameter, Registrierungsvariablen und Datenbankkonfigurationsparameter. Sie können die ersten beiden Typen über Parametergruppen und den letzten Typ über die rdsadmin.update_db_param gespeicherte Prozedur aktualisieren.

> **Note**
>
> Sie können nur die Werte vorhandener Parameter ändern. Sie können keine neuen Parameter hinzufügen, die RDS for Db2 nicht unterstützt.

Weitere Informationen zu diesen Parametern und zur Änderung ihrer Werte finden Sie unterthe section called “Db2-Parameter”.

Konfiguration der Protokollaufbewahrung

Rufen Sie die `rdsadmin.set_archive_log_retention` gespeicherte Prozedur auf, um zu konfigurieren, wie lange Amazon RDS Protokolldateien für Ihre RDS for Db2-Datenbank aufbewahrt.

Um die Protokollaufbewahrung für eine Datenbank zu konfigurieren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. (Optional) Überprüfen Sie Ihre aktuelle Konfiguration für die Aufbewahrung von Protokollen, indem Sie anrufen`rdsadmin.show_archive_log_retention`. Weitere Informationen finden Sie unter the section called “rdsadmin.show_archive_log_retention”.

   ```
   db2 "call rdsadmin.show_archive_log_retention(
       ?,
       'database_name')"
   ```

3. Konfigurieren Sie die Protokollspeicherung für die Datenbank, indem Sie aufrufen`rdsadmin.set_archive_log_retention`. Weitere Informationen finden Sie unter the section called “rdsadmin.set_archive_log_retention”.

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'database_name',
    'archive_log_retention_hours')"
```

Protokollinformationen auflisten

Rufen Sie die `rdsadmin.list_archive_log_information` gespeicherte Prozedur auf, um Details zu Archivprotokolldateien aufzulisten, einschließlich Details wie der insgesamt verwendeten Speichergröße.

Um Protokollinformationen für eine Datenbank aufzulisten

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Geben Sie eine Liste mit Protokolldateiinformationen zurück, indem Sie aufrufen`rdsadmin.list_archive_log_information`. Weitere Informationen finden Sie unter the section called “rdsadmin.list_archive_log_information”.

```
db2 "call rdsadmin.list_archive_log_information(
    ?,
    'database_name')"
```

Eine Datenbank deaktivieren

Um eine Datenbank auf Ihrer RDS for Db2-DB-Instance zu deaktivieren, rufen Sie die `rdsadmin.deactivate_database` gespeicherte Prozedur auf.

Standardmäßig aktiviert Amazon RDS eine Datenbank, wenn Sie eine Datenbank auf Ihrer RDS for Db2-DB-Instance erstellen. Sie können selten verwendete Datenbanken deaktivieren, um Speicherressourcen zu schonen.

Um eine Datenbank zu deaktivieren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Deaktivieren Sie eine Datenbank, indem Sie sie aufrufen`rdsadmin.deactivate_database`. Weitere Informationen finden Sie unter rdsadmin.deactivate_database.

   ```
   db2 "call rdsadmin.deactivate_database(
       ?,
       'database_name')"
   ```

Eine Datenbank aktivieren

Um eine Datenbank auf einer eigenständigen RDS for Db2-DB-Instance zu aktivieren, rufen Sie die `rdsadmin.activate_database` gespeicherte Prozedur auf.

Standardmäßig aktiviert Amazon RDS eine Datenbank, wenn Sie eine Datenbank auf Ihrer RDS for Db2-DB-Instance erstellen. Sie können selten verwendete Datenbanken deaktivieren, um Speicherressourcen zu schonen, und später eine deaktivierte Datenbank aktivieren.

Um eine Datenbank zu aktivieren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Aktivieren Sie eine Datenbank, indem Sie aufrufen`rdsadmin.activate_database`. Weitere Informationen finden Sie unter rdsadmin.activate_database.

   ```
   db2 "call rdsadmin.activate_database(
       ?,
       'database_name')"
   ```

Eine Datenbank reaktivieren

Um eine Datenbank zu reaktivieren, rufen Sie die `rdsadmin.reactivate_database` gespeicherte Prozedur auf. Nachdem Sie Änderungen an den Datenbankkonfigurationen vorgenommen haben, müssen Sie möglicherweise eine Datenbank auf einer RDS for Db2-DB-Instance reaktivieren. Um festzustellen, ob Sie eine Datenbank reaktivieren müssen, stellen Sie eine Verbindung mit der Datenbank her und führen Sie den Befehl aus. `db2 get db cfg show detail`

Sie können diese gespeicherte Prozedur auch aufrufen, um eine Datenbank auf einer eigenständigen RDS for Db2-DB-Instance zu reaktivieren, nachdem Sie Änderungen an den Datenbankkonfigurationen vorgenommen haben. Oder Sie könnten eine Datenbank auf einer eigenständigen RDS for Db2-DB-Instance reaktivieren, indem Sie zuerst die `rdsadmin.deactivate_database` gespeicherte Prozedur und dann die gespeicherte Prozedur aufrufen. `rdsadmin.activate_database` Weitere Informationen erhalten Sie unter Eine Datenbank deaktivieren und Eine Datenbank aktivieren.

Um eine Datenbank zu reaktivieren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Reaktivieren Sie eine Datenbank, indem Sie aufrufen`rdsadmin.reactivate_database`. Weitere Informationen finden Sie unter rdsadmin.reactivate_database.

   ```
   db2 "call rdsadmin.reactivate_database(
       ?,
       'database_name')"
   ```

Eine Datenbank löschen

Um eine Datenbank aus Ihrer RDS for Db2-DB-Instance zu löschen, rufen Sie die `rdsadmin.drop_database` gespeicherte Prozedur auf. Weitere Informationen finden Sie in der IBM Db2 Dokumentation unter Löschen von Datenbanken.

> **Note**
>
> Sie können eine Datenbank nur löschen, indem Sie die gespeicherte Prozedur aufrufen, wenn bestimmte Bedingungen erfüllt sind. Weitere Informationen finden Sie unter the section called “Nutzungshinweise” für`rdsadmin.drop_database`.

Um eine Datenbank zu löschen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Löschen Sie eine Datenbank, indem Sie aufrufen`rdsadmin.drop_database`. Weitere Informationen finden Sie unter rdsadmin.drop_database.

   ```
   db2 "call rdsadmin.drop_database('database_name')"
   ```

Wiederherstellen einer Datenbank

Um eine Datenbank von einem Amazon S3 S3-Bucket in Ihre RDS for Db2-DB-Instance zu verschieben, rufen Sie die `rdsadmin.restore_database` gespeicherte Prozedur auf. Weitere Informationen finden Sie unter RESTORE DATABASEBefehl in der IBM Db2 Dokumentation.

So stellen Sie eine -Datenbank wieder her

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. (Optional) Überprüfen Sie Ihre aktuellen Konfigurationseinstellungen, um den Wiederherstellungsvorgang zu optimieren, indem Sie anrufen`rdsadmin.show_configuration`. Weitere Informationen finden Sie unter the section called “rdsadmin.show_configuration”.

```
db2 "call rdsadmin.show_configuration('name')"
```

3. Konfigurieren Sie die Einstellungen, um den Wiederherstellungsvorgang zu optimieren, indem Sie anrufen`rdsadmin.set_configuration`. Das explizite Festlegen dieser Werte kann die Leistung bei der Wiederherstellung von Datenbanken mit großen Datenmengen verbessern. Weitere Informationen finden Sie unter the section called “rdsadmin.set_configuration”.

```
db2 "call rdsadmin.set_configuration(
    'name',
    'value')"
```

4. Stellen Sie die Datenbank wieder her, indem Sie aufrufen`rdsadmin.restore_database`. Weitere Informationen finden Sie unter the section called “rdsadmin.restore_database”.

```
db2 "call rdsadmin.restore_database(
    ?,
    'database_name',
    's3_bucket_name',
    's3_prefix',
    restore_timestamp,
    'backup_type')"
```

5. (Optional) Bestätigen Sie, dass Ihre Datenbank wiederhergestellt wurde, indem Sie anrufen `rdsadmin.list_databases` und überprüfen, ob die wiederhergestellte Datenbank aufgeführt ist. Weitere Informationen finden Sie unter rdsadmin.list_databases.
6. Schalten Sie die Datenbank wieder online und wenden Sie zusätzliche Transaktionsprotokolle an, indem Sie anrufen`rdsadmin.rollforward_database`. Weitere Informationen finden Sie unter the section called “rdsadmin.rollforward_database”.

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'database_name',
    's3_bucket_name',
    s3_prefix,
    'rollforward_to_option',
    'complete_rollforward')"
```

7. (Optional) Überprüfen Sie den Status der `rdsadmin.rollforward_database` gespeicherten Prozedur, indem Sie die the section called “rdsadmin.rollforward_status” gespeicherte Prozedur aufrufen.

8. Wenn Sie FALSE im vorherigen Schritt `complete_rollforward` auf gesetzt haben, müssen Sie die Datenbank durch einen Anruf wieder online stellen`rdsadmin.complete_rollforward`. Weitere Informationen finden Sie unter the section called “rdsadmin.complete_rollforward”.

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'database_name')"
```

9. (Optional) Überprüfen Sie den Status der `rdsadmin.complete_rollforward` gespeicherten Prozedur, indem Sie die the section called “rdsadmin.rollforward_status” gespeicherte Prozedur aufrufen.

Datenbanken auflisten

Sie können alle Ihre Datenbanken auflisten, die auf Amazon RDS for Db2 laufen, indem Sie die `rdsadmin.list_databases` benutzerdefinierte Funktion aufrufen.

Um Ihre Datenbanken aufzulisten

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Listen Sie Ihre Datenbanken auf, indem Sie anrufen`rdsadmin.list_databases`. Weitere Informationen finden Sie unter rdsadmin.list_databases.

```
db2 "select * from table(rdsadmin.list_databases())"
```

Sammeln von Informationen über Datenbanken

Rufen Sie die `rdsadmin.db2pd_command` gespeicherte Prozedur auf, um Informationen über eine Datenbank auf einer RDS for Db2-DB-Instance zu sammeln. Diese Informationen können Ihnen bei der Überwachung Ihrer Datenbanken oder bei der Behebung von Problemen helfen.

Um Informationen über eine Datenbank zu sammeln

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Sammeln Sie Informationen über die Datenbank, indem Sie anrufen`rdsadmin.db2pd_command`. Weitere Informationen finden Sie unter rdsadmin.db2pd_command.

   ```
   db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
   ```

Anwendungen aus Datenbanken abzwingen

Um zu erzwingen, dass Anwendungen aus einer Datenbank auf Ihrer RDS for Db2-DB-Instance entfernt werden, rufen Sie die `rdsadmin.force_application` gespeicherte Prozedur auf. Bevor Sie Wartungsarbeiten an Ihren Datenbanken durchführen, zwingen Sie Anwendungen dazu, Ihre Datenbanken zu deaktivieren.

Um Anwendungen aus einer Datenbank zu entfernen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Erzwingen Sie, dass Anwendungen aus einer Datenbank entfernt werden, indem Sie sie aufrufen`rdsadmin.force_application`. Weitere Informationen finden Sie unter rdsadmin.force_application.

   ```
   db2 "call rdsadmin.force_application(
       ?,
       'applications')"
   ```

Generieren von Leistungsberichten

Sie können Leistungsberichte mit einer Prozedur oder einem Skript generieren. Informationen zur Verwendung eines Verfahrens finden Sie in der IBM Db2 Dokumentation unter DBSUMMARYVerfahren - Generieren eines zusammenfassenden Berichts über System- und Anwendungsleistungskennzahlen.

Db2 enthält eine `db2mon.sh` Datei in seinem `~sqllib/sample/perf` Verzeichnis. Durch die Ausführung des Skripts wird ein kostengünstiger, umfangreicher SQL-Metrikbericht erstellt. Informationen zum Herunterladen der `db2mon.sh` Datei und der zugehörigen Skriptdateien finden Sie im perfVerzeichnis im IBM db2-samples RepositoryGitHub.

Um Leistungsberichte mit dem Skript zu generieren

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Connect zu Ihrer Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Erstellen Sie einen Pufferpool `db2monbp` mit einem Namen und einer Seitengröße von 4096, indem Sie aufrufen`rdsadmin.create_bufferpool`. Weitere Informationen finden Sie unter rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool('database_name','db2monbp',4096)"
   ```

3. Erstellen Sie einen temporären Tablespace mit dem Namen`db2montmptbsp`, der den db2monbp Pufferpool verwendet, indem Sie aufrufen. `rdsadmin.create_tablespace` Weitere Informationen finden Sie unter rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace('database_name',\
       'db2montmptbsp','db2monbp',4096,1000,100,'T')"
   ```

4. Öffnen Sie das `db2mon.sh` Skript und ändern Sie die Zeile zur Verbindung mit einer Datenbank.

   a. Entfernen Sie die folgende Zeile.

      ```
      db2 -v connect to $dbName
      ```

b. Ersetzen Sie die Zeile im vorherigen Schritt durch die folgende Zeile. Ersetzen Sie im folgenden Beispiel *master_username* und *master_password* durch den Master-Benutzernamen und das Master-Passwort für Ihre RDS for Db2-DB-Instance.

```
db2 -v connect to $dbName user master_username using master_password
```

c. Entfernen Sie die folgenden Zeilen.

```
db2 -v create bufferpool db2monbp

db2 -v create user temporary tablespace db2montmptbsp bufferpool db2monbp

db2 -v drop tablespace db2montmptbsp

db2 -v drop bufferpool db2monbp
```

5. Führen Sie das `db2mon.sh` Skript aus, um in bestimmten Intervallen einen Bericht auszugeben. *absolute_path*Ersetzen Sie im folgenden Beispiel durch den vollständigen Pfad zur Skriptdatei, *rds_database_alias* durch den Namen Ihrer Datenbank und *seconds* durch die Anzahl der Sekunden (0 bis 3600) zwischen der Berichtsgenerierung.

```
absolute_path/db2mon.sh rds_database_alias seconds | tee -a db2mon.out
```

Beispiele

Das folgende Beispiel zeigt, dass sich die Skriptdatei im `perf` Verzeichnis unter dem home Verzeichnis befindet.

```
/home/db2inst1/sqllib/samples/perf/db2mon.sh rds_database_alias seconds | tee -a
 db2mon.out
```

6. Löschen Sie den Pufferpool und den Tablespace, die für die `db2mon.sh` Datei erstellt wurden. Ersetzen Sie im folgenden Beispiel und durch den Master-Benutzernamen *master_username* und *master_password* das Master-Passwort für Ihre RDS for Db2-DB-Instance. *database_name*Ersetzen Sie durch den Namen Ihrer Datenbank. Weitere Informationen erhalten Sie unter rdsadmin.drop_tablespace und rdsadmin.drop_bufferpool.

```
db2 connect to rdsadmin user master_username using master_password

db2 "call rdsadmin.drop_tablespace('database_name','db2montmptbsp')"
```

```
db2 "call rdsadmin.drop_bufferpool('database_name','db2monbp')"
```

Verwalten des Speichers

DB2 verwendet automatischen Speicher, um den physischen Speicher für Datenbankobjekte wie Tabellen, Indizes und temporäre Dateien zu verwalten. Anstatt Speicherplatz manuell zuzuweisen und zu verfolgen, welche Speicherpfade verwendet werden, ermöglicht die automatische Speicherung dem Db2-System, Speicherpfade nach Bedarf zu erstellen und zu verwalten. Dies kann die Verwaltung von Db2-Datenbanken vereinfachen und die Wahrscheinlichkeit von Fehlern aufgrund menschlicher Fehler verringern. Weitere Informationen finden Sie unter Automatische Speicherung im IBM Db2 -Dokumentation.

Mit RDS for Db2 können Sie die Speichergröße dynamisch erhöhen, indem Sie die logischen Volumes und das Dateisystem automatisch erweitern. Weitere Informationen finden Sie unter Arbeiten mit Speicher für Amazon RDS DB-Instances.

Allgemeine Aufgaben für Tablespaces

Sie können Tablespaces für eine for Db2-Datenbank erstellen, ändern, umbenennen oder löschen. RDS Für das Erstellen, Ändern, Umbenennen oder Löschen von Tablespaces sind Berechtigungen auf höherer Ebene erforderlich, die dem Masterbenutzer nicht zur `SYSADM` Verfügung stehen. Verwenden Sie stattdessen RDS gespeicherte Amazon-Prozeduren.

Themen

- Einen Tablespace erstellen
- Einen Tablespace ändern
- Einen Tablespace umbenennen
- Einen Tablespace löschen
- Den Status eines Tablespaces überprüfen
- Rückgabe detaillierter Informationen über Tablespaces
- Status und Speichergruppe für einen Tablespace auflisten
- Auflisten der Tablespaces einer Tabelle
- Tablespace-Container auflisten

Einen Tablespace erstellen

Um einen Tablespace für Ihre RDS for Db2-Datenbank zu erstellen, rufen Sie die `rdsadmin.create_tablespace` gespeicherte Prozedur auf. Weitere Informationen finden Sie unter CREATE TABLESPACE Anweisung in der IBM Db2 -Dokumentation.

> **Important**
>
> Um einen Tablespace zu erstellen, benötigen Sie einen Pufferpool mit derselben Seitengröße, den Sie dem Tablespace zuordnen können. Weitere Informationen finden Sie unter Allgemeine Aufgaben für Pufferpools.

Um einen Tablespace zu erstellen

1. Stellen Sie mit dem Master-Benutzernamen und dem Master-Passwort für Ihre DB2-DB-Instance eine Connect RDS zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Erstellen Sie einen Tablespace, indem Sie aufrufen`rdsadmin.create_tablespace`. Weitere Informationen finden Sie unter rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace(
       'database_name',
       'tablespace_name',
       'buffer_pool_name',
       tablespace_initial_size,
       tablespace_increase_size,
       'tablespace_type')"
   ```

Einen Tablespace ändern

Um einen Tablespace für Ihre RDS for Db2-Datenbank zu ändern, rufen Sie die gespeicherte Prozedur auf. `rdsadmin.alter_tablespace` Sie können diese gespeicherte Prozedur verwenden, um den Pufferpool eines Tablespaces zu ändern, den Höchstwert zu senken oder einen Tablespace online zu schalten. Weitere Informationen finden Sie unter ALTER TABLESPACE Aussage in der IBM Db2 -Dokumentation.

Um einen Tablespace zu ändern

1. Stellen Sie mit dem Master-Benutzernamen und dem Master-Passwort für Ihre DB2-DB-Instance eine Connect RDS zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Ändern Sie einen Tablespace, indem Sie aufrufen`rdsadmin.alter_tablespace`. Weitere Informationen finden Sie unter rdsadmin.alter_tablespace.

```
db2 "call rdsadmin.alter_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    buffer_pool_size,
    tablespace_increase_size,
    'max_size', 'reduce_max',
    'reduce_stop',
    'reduce_value',
    'lower_high_water',
    'lower_high_water_stop',
    'switch_online')"
```

Einen Tablespace umbenennen

Um den Namen eines Tablespaces für Ihre Datenbank RDS für Db2 zu ändern, rufen Sie die gespeicherte Prozedur auf. `rdsadmin.rename_tablespace` Weitere Informationen finden Sie unter RENAME TABLESPACE Anweisung in der IBM Db2 -Dokumentation.

Um einen Tablespace umzubenennen

1. Stellen Sie mit dem Master-Benutzernamen und dem Master-Passwort für Ihre DB2-DB-Instance eine Connect RDS zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Benennen Sie einen Tablespace um, indem Sie aufrufen`rdsadmin.rename_tablespace`. Weitere Informationen, einschließlich der Einschränkungen, wie Sie einen Tablespace benennen können, finden Sie unter. rdsadmin.rename_tablespace

```
db2 "call rdsadmin.rename_tablespace(
    'database_name',
    'source_tablespace_name',
    'target_tablespace_name')"
```

Einen Tablespace löschen

Um einen Tablespace für Ihre RDS for Db2-Datenbank zu löschen, rufen Sie die `rdsadmin.drop_tablespace` gespeicherte Prozedur auf. Bevor Sie einen Tablespace löschen, löschen Sie zunächst alle Objekte im Tablespace, z. B. Tabellen, Indizes oder große Objekte (). LOBs Weitere Informationen finden Sie unter Löschen von Tabellenräumen im IBM Db2 - Dokumentation.

So löschen Sie einen Tablespace

1. Stellen Sie mit dem Master-Benutzernamen und dem Master-Passwort für Ihre DB2-DB-Instance eine Connect RDS zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Löschen Sie einen Tablespace, indem Sie aufrufen`rdsadmin.drop_tablespace`. Weitere Informationen finden Sie unter rdsadmin.drop_tablespace.

```
db2 "call rdsadmin.drop_tablespace(
    'database_name',
    'tablespace_name')"
```

Den Status eines Tablespaces überprüfen

Sie können den Status eines Tablespaces mithilfe der `cast` Funktion überprüfen.

Um den Status eines Tablespaces zu überprüfen

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre DB2-DB-Instance eine Connect zu Ihrer RDS Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_alias*, *master_username*, und *master_password* durch Ihre eigenen Informationen.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Gibt eine zusammenfassende Ausgabe zurück.

   Für eine zusammenfassende Ausgabe:

   ```
   db2 "select cast(tbsp_id as smallint) as tbsp_id,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   cast(tbsp_type as varchar(3)) as tbsp_type,
   cast(tbsp_state as varchar(10)) as state,
   cast(tbsp_content_type as varchar(8)) as contents from
    table(mon_get_tablespace(null,-1)) order by tbsp_id"
   ```

Rückgabe detaillierter Informationen über Tablespaces

Mit der Funktion können Sie Informationen über einen Tablespace für ein Mitglied oder für alle Mitglieder zurückgeben. `cast`

Um detaillierte Informationen über Tablespaces zurückzugeben

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre DB2-DB-Instance eine Connect zu Ihrer RDS Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_alias*, *master_username*, und *master_password* durch Ihre eigenen Informationen.

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Gibt Details zu allen Tablespaces in der Datenbank für ein Mitglied oder für alle Mitglieder zurück.

   Für ein Mitglied:

   ```
   db2 "select cast(member as smallint) as member,
   ```

```
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-1)) order by member, tbsp_id "
```

Für alle Mitglieder:

```
db2 "select cast(member as smallint) as member
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-2)) order by member, tbsp_id "
```

Status und Speichergruppe für einen Tablespace auflisten

Sie können den Status und die Speichergruppe für einen Tablespace auflisten, indem Sie eine SQL Anweisung ausführen.

Führen Sie die folgende SQL Anweisung aus, um den Status und die Speichergruppe für einen Tablespace aufzulisten:

```
db2 "SELECT varchar(tbsp_name, 30) as tbsp_name,
                  varchar(TBSP_STATE, 30) state,
                  tbsp_type,
                  varchar(storage_group_name,30) storage_group
FROM TABLE(MON_GET_TABLESPACE('',-2)) AS t"
```

Auflisten der Tablespaces einer Tabelle

Sie können die Tablespaces für eine Tabelle auflisten, indem Sie eine Anweisung ausführen. SQL

Führen Sie die folgende Anweisung aus, um die Tablespaces einer Tabelle aufzulisten. SQL Ersetzen Sie im folgenden Beispiel *SCHEMA_NAME* and *TABLE_NAME* durch die Namen Ihres Schemas und Ihrer Tabelle:

```
db2 "SELECT
   VARCHAR(SD.TBSPACE,30) AS DATA_SPACE,
   VARCHAR(SL.TBSPACE,30) AS LONG_SPACE,
   VARCHAR(SI.TBSPACE,30) AS INDEX_SPACE
 FROM
   SYSCAT.DATAPARTITIONS P
   JOIN SYSCAT.TABLESPACES SD ON SD.TBSPACEID = P.TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SL ON SL.TBSPACEID = P.LONG_TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SI ON SI.TBSPACEID = P.INDEX_TBSPACEID
 WHERE
    TABSCHEMA = 'SCHEMA_NAME'
 AND TABNAME   = 'TABLE_NAME'"
```

Tablespace-Container auflisten

Sie können alle Tablespace-Container oder bestimmte Tablespace-Container auflisten, indem Sie den Befehl verwenden. `cast`

Um die Tablespace-Container für einen Tablespace aufzulisten

1. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre DB2-DB-Instance eine Connect zu Ihrer RDS Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_alias*, *master_username*, und *master_password* durch Ihre eigenen Informationen:

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Gibt eine Liste aller Tablespace-Container in der Datenbank oder bestimmter Tablespace-Container zurück.

   Für alle Tablespace-Container:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(container_id as smallint) as id,
cast(container_name as varchar(60)) as container_path, container_type as type from
 table(mon_get_container(null,-2)) order by member,tbsp_id,container_id"
```

   Für bestimmte Tablespace-Container:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(container_id as smallint) as id,
cast(container_name as varchar(60)) as container_path, container_type as type from
 table(mon_get_container('TBSP_1',-2)) order by member, tbsp_id,container_id"
```

Integrieren einer Amazon RDS for Db2-DB-Instance mit Amazon S3

Sie können Dateien zwischen Ihrer Amazon RDS for Db2-DB-Instance und einem Amazon Simple Storage Service (Amazon S3) -Bucket mit RDS gespeicherten Amazon-Prozeduren übertragen. Weitere Informationen finden Sie unter Referenz für gespeicherte Prozeduren in Amazon RDS für Db2.

> **Note**
>
> Die DB-Instance und der Amazon-S3-Bucket müssen sich in der gleichen AWS-Region befinden.

RDSDamit Db2 in Amazon S3 integriert werden kann, muss Ihre DB-Instance Zugriff auf einen Amazon S3 S3-Bucket haben, in dem sich Ihr RDS for Db2 befindet. Wenn Sie derzeit keinen S3-Bucket haben, erstellen Sie einen Bucket.

Themen

- Schritt 1: Erstellen einer IAM-Richtlinie
- Schritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu
- Schritt 3: Fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu

Schritt 1: Erstellen einer IAM-Richtlinie

In diesem Schritt erstellen Sie eine AWS Identity and Access Management (IAM) -Richtlinie mit den erforderlichen Berechtigungen, um Dateien von Ihrem Amazon S3 S3-Bucket auf Ihre RDS DB-Instance zu übertragen. In diesem Schritt wird davon ausgegangen, dass Sie bereits einen S3-Bucket erstellt haben. Weitere Informationen finden Sie unter Bucket erstellen im Amazon S3 S3-Benutzerhandbuch.

Notieren Sie sich vor dem Erstellen der Richtlinie die folgenden Informationen:

- Der Amazon-Ressourcenname (ARN) für Ihren Bucket
- Der Schlüssel ARN for your AWS Key Management Service (AWS KMS), wenn Ihr Bucket verwendet SSE-KMS or SSE-S3 Verschlüsselung.

Erstellen Sie eine IAM Richtlinie, die die folgenden Berechtigungen umfasst:

```
"kms:GenerateDataKey",
"kms:Decrypt",
"s3:PutObject",
"s3:GetObject",
"s3:AbortMultipartUpload",
"s3:ListBucket",
"s3:DeleteObject",
"s3:GetObjectVersion",
"s3:ListMultipartUploadParts"
```

Sie können eine IAM Richtlinie erstellen, indem Sie das AWS Management Console oder das AWS Command Line Interface (AWS CLI) verwenden.

Konsole

Um eine IAM Richtlinie zu erstellen, die Amazon den Zugriff RDS auf Ihren Amazon S3 S3-Bucket ermöglicht

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen und dann JSON.
4. Fügen Sie Aktionen nach Dienst hinzu. Um Dateien von einem Amazon S3 S3-Bucket nach Amazon zu übertragenRDS, müssen Sie Bucket-Berechtigungen und Objektberechtigungen auswählen.
5. Erweitern Sie Resources (Ressourcen). Sie müssen Ihren Bucket und Ihre Objektressourcen angeben.
6. Wählen Sie Weiter.
7. Geben Sie unter Richtlinienname einen Namen für diese Richtlinie ein.
8. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für diese Richtlinie ein.
9. Wählen Sie Create Policy (Richtlinie erstellen) aus.

AWS CLI

Um eine IAM Richtlinie zu erstellen, die Amazon den Zugriff RDS auf Ihren Amazon S3 S3-Bucket ermöglicht

1. Ausführen des create-policyBefehl. Ersetzen Sie im folgenden Beispiel *iam_policy_name* und *s3_bucket_name* durch einen Namen für Ihre IAM Richtlinie und den Namen des Amazon S3 S3-Buckets, in dem sich Ihre RDS for Db2-Datenbank befindet.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt",
            "s3:PutObject",
            "s3:GetObject",
            "s3:AbortMultipartUpload",
            "s3:ListBucket",
            "s3:DeleteObject",
            "s3:GetObjectVersion",
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
        }
      ]
    }'
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
```

```
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
          "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:AbortMultipartUpload",
            "s3:ListBucket",
            "s3:DeleteObject",
            "s3:GetObjectVersion",
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
      }
    ]
  }'
```

2. Notieren Sie sich nach der Erstellung der Richtlinie den ARN Inhalt der Richtlinie. Sie benötigen das ARN fürSchritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu.

Informationen zum Erstellen einer IAM Richtlinie finden Sie im IAM Benutzerhandbuch unter IAMRichtlinien erstellen.

Schritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu

Bei diesem Schritt wird davon ausgegangen, dass Sie die IAM Richtlinie in erstellt habenSchritt 1: Erstellen einer IAM-Richtlinie. In diesem Schritt erstellen Sie eine IAM Rolle RDS für Ihre DB2-DB-Instance und fügen dann Ihre IAM Richtlinie der Rolle hinzu.

Sie können eine IAM Rolle für Ihre DB-Instance erstellen, indem Sie das AWS Management Console oder das AWS CLI verwenden.

Konsole

Um eine IAM Rolle zu erstellen und ihr Ihre IAM Richtlinie anzuhängen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS-Service.
5. Wählen RDSSie für Dienst oder Anwendungsfall die Option und dann RDS— Rolle zur Datenbank hinzufügen aus.
6. Wählen Sie Weiter.
7. Suchen Sie unter Berechtigungsrichtlinien nach dem Namen der IAM Richtlinie, die Sie erstellt haben, und wählen Sie ihn aus.
8. Wählen Sie Weiter.
9. Geben Sie für Role name (Rollenname) einen Rollennamen ein.
10. (Optional) Geben Sie unter Role description (Rollenbeschreibung) eine Beschreibung für die neue Rolle ein.
11. Wählen Sie Rolle erstellen.

AWS CLI

Um eine IAM Rolle zu erstellen und ihr Ihre IAM Richtlinie anzuhängen

1. Ausführen des screate-roleBefehl. Ersetzen Sie es im folgenden Beispiel *iam_role_name* durch einen Namen für Ihre IAM Rolle.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
```

```
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

2. Notieren Sie sich nach der Erstellung der Rolle den ARN Namen der Rolle. Sie benötigen das ARN fürSchritt 3: Fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu.

3. Ausführen des sattach-role-policyBefehl. Im folgenden Beispiel *iam_policy_arn* ersetzen Sie ihn durch den Wert ARN der IAM Richtlinie, die Sie in erstellt habenSchritt 1: Erstellen einer IAM-Richtlinie. *iam_role_name*Ersetzen Sie ihn durch den Namen der IAM Rolle, die Sie gerade erstellt haben.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws iam attach-role-policy \
   --policy-arn iam_policy_arn \
   --role-name iam_role_name
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws iam attach-role-policy ^
   --policy-arn iam_policy_arn ^
```

```
    --role-name iam_role_name
```

Weitere Informationen finden Sie im Benutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM IAM Benutzer.

Schritt 3: Fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu

In diesem Schritt fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu. Beachten Sie die folgenden Voraussetzungen:

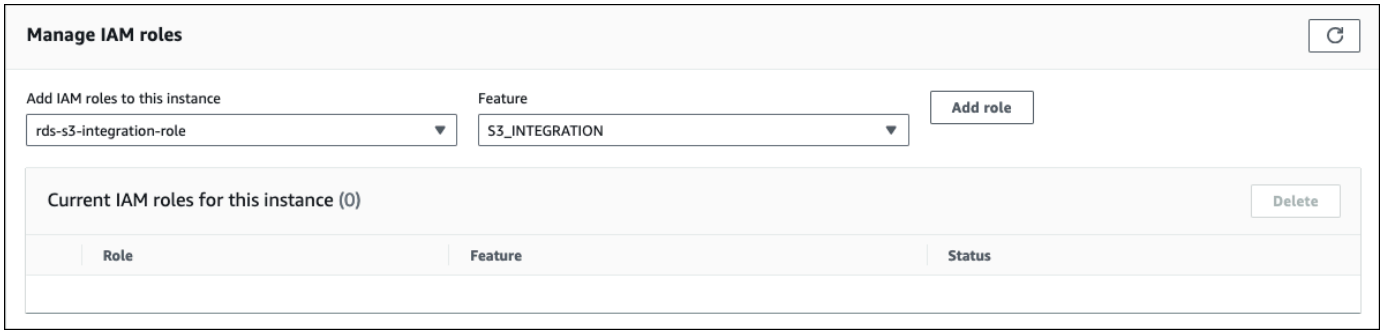
- Sie müssen Zugriff auf eine IAM Rolle haben, der die erforderlichen Amazon S3 S3-Berechtigungsrichtlinien zugeordnet sind.
- Sie können Ihrer RDS Db2-DB-Instance jeweils nur eine IAM Rolle zuordnen.
- Ihre RDS for Db2-DB-Instance muss sich im Status Verfügbar befinden.

Sie können Ihrer DB-Instance eine IAM Rolle hinzufügen, indem Sie den AWS Management Console oder den AWS CLI verwenden.

Konsole

Um Ihrer RDS Db2-DB-Instance eine IAM Rolle hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie Ihren Namen RDS für die DB2-DB-Instance.
4. Scrollen Sie auf der Registerkarte Konnektivität und Sicherheit nach unten zum Abschnitt IAMRollen verwalten unten auf der Seite.
5. Wählen Sie unter IAM Rollen zu dieser Instanz hinzufügen die Rolle aus, in der Sie sie erstellt habenSchritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu.
6. Wählen Sie für Feature die Option S3_ INTEGRATION aus.
7. Wählen Sie Rolle hinzufügen aus.



AWS CLI

Um Ihrer RDS Db2-DB-Instance eine IAM Rolle hinzuzufügen, führen Sie den add-role-to-db-instanceBefehl. Ersetzen Sie im folgenden Beispiel *db_instance_name* und *iam_role_arn* durch den Namen Ihrer DB-Instance und den Namen ARN der IAM Rolle, in der Sie sie erstellt habenSchritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier db_instance_name \
    --feature-name S3_INTEGRATION \
    --role-arn iam_role_arn \
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier db_instance_name ^
    --feature-name S3_INTEGRATION ^
    --role-arn iam_role_arn ^
```

Um zu bestätigen, dass die Rolle erfolgreich zu Ihrer RDS for Db2-DB-Instance hinzugefügt wurde, führen Sie den describe-db-instancesBefehl. Ersetzen Sie im folgenden Beispiel *db_instance_name* durch den Namen Ihrer DB-Instance.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-instances \
    --filters "Name=db-instance-id,Values=db_instance_name" \
    --query 'DBInstances[].AssociatedRoles'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-instances ^
    --filters "Name=db-instance-id,Values=db_instance_name" ^
    --query 'DBInstances[].AssociatedRoles'
```

Dieser Befehl erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

```
[
    [
        {
            "RoleArn": "arn:aws:iam::0123456789012:role/rds-db2-s3-role",
            "FeatureName": "S3_INTEGRATION",
            "Status": "ACTIVE"
        }
    ]
]
```

Migrieren von Daten zu Amazon RDS for Db2

Sie können selbstverwaltete Db2-Datenbanken zu Amazon RDS for Db2 migrieren, indem Sie entweder native Db2-Tools AWS oder native Db2-Tools verwenden.

Informationen zur Migration von Ihrer Db2-Datenbank zu Amazon RDS for Db2 mithilfe von AWS Services finden Sie unter. Nutzung von AWS Services zur Migration von Daten von Db2 zu Amazon RDS for Db2

Informationen zur Migration von Ihrer Db2-Datenbank zu Amazon RDS for Db2 mithilfe nativer Db2-Tools finden Sie unter. Verwendung nativer Db2-Tools zur Migration von Daten von Db2 nach Amazon RDS for Db2

Nutzung von AWS Services zur Migration von Daten von Db2 zu Amazon RDS for Db2

In Amazon gibt es mehrere MöglichkeitenRDS, Daten von einer Db2-Datenbank zu Amazon RDS for Db2 zu migrieren. Sie können eine einmalige Migration Ihrer Db2-Datenbank von durchführen Linux, AIX, oder Windows Umgebungen für Amazon RDS für Db2. Um Ausfallzeiten zu minimieren, können Sie eine Migration ohne Ausfallzeiten durchführen. Sie können Ihre Daten migrieren, indem Sie sie in Amazon S3 speichern und sie tabellenweise in Ihre Db2-Datenbank laden. Sie können auch eine synchrone Migration durch Replikation oder Verwendung durchführen. AWS Database Migration Service

Für einmalige Migrationen für LinuxAmazon basiert auf Db2-Datenbanken und unterstützt RDS nur Offline- und Online-Backups. Amazon unterstützt RDS keine inkrementellen und Delta Backups. Für Migrationen nahezu ohne Ausfallzeiten für LinuxAmazon basiert auf Db2-Datenbanken und RDS benötigt Online-Backups. Wir empfehlen Ihnen, Online-Backups für Migrationen ohne Ausfallzeiten und Offline-Backups für Migrationen zu verwenden, bei denen Ausfallzeiten verkraftet werden können.

Themen

- Migration von Linux to Linux für Amazon RDS für Db2
- Migration von Linux to Linux mit nahezu null Ausfallzeiten für Amazon for Db2 RDS
- Synchrone Migration von Linux to Linux für Amazon RDS für Db2
- Migration von AIX or Windows to Linux für Amazon RDS für Db2
- Migration von Db2-Daten über Amazon S3 zu Amazon RDS for Db2

- Migration zu Amazon RDS für Db2 mit AWS Database Migration Service ()AWS DMS

Migration von Linux to Linux für Amazon RDS für Db2

Mit diesem Migrationsansatz sichern Sie Ihre selbstverwaltete Db2-Datenbank in einem Amazon S3 S3-Bucket. Anschließend verwenden Sie RDS gespeicherte Amazon-Prozeduren, um Ihre Db2-Datenbank auf einer Amazon RDS for Db2-DB-Instance wiederherzustellen. Weitere Informationen zur Verwendung von Amazon S3 finden Sie unterIntegrieren einer Amazon RDS for Db2-DB-Instance mit Amazon S3.

Backup und Wiederherstellung RDS für Db2 folgen dem IBM Db2 unterstützte Upgrade-Pfade und Einschränkungen. Weitere Informationen finden Sie unter Unterstützte Upgrade-Pfade für Db2-Server und Upgradeeinschränkungen für Db2-Server in der IBM Db2 -Dokumentation.

Themen

- Einschränkungen und Empfehlungen für die Verwendung der systemeigenen Wiederherstellung
- Ihre Datenbank auf Amazon S3 sichern
- Eine automatische Standardspeichergruppe erstellen
- Ihre Db2-Datenbank wiederherstellen

Einschränkungen und Empfehlungen für die Verwendung der systemeigenen Wiederherstellung

Die folgenden Einschränkungen und Empfehlungen gelten für die Verwendung von Native Restore:

- Amazon unterstützt RDS nur die Migration von lokalen Versionen von Db2, die den unterstützten RDS Db2-Versionen entsprechen. Weitere Informationen zu den unterstützten Versionen finden Sie unter. Unterstützte Db2-Nebenversionen auf Amazon RDS
- Amazon unterstützt RDS nur Offline- und Online-Backups für die native Wiederherstellung. Amazon unterstützt RDS keine inkrementellen oder Delta Backups.
- Sie können keine Wiederherstellung aus einem Amazon S3 S3-Bucket in einer anderen Region als der Region durchführen AWS-Region , in der sich Ihre RDS for Db2-DB-Instance befindet.
- Amazon S3 begrenzt die Größe von Dateien, die in einen Amazon S3 S3-Bucket hochgeladen werden, auf 5 TB. Wenn Ihre Datenbank-Backup-Datei 5 TB überschreitet, teilen Sie die Sicherungsdatei in kleinere Dateien auf.
- Amazon unterstützt RDS keine externen Routinen, inkrementelle Wiederherstellungen oder Delta Wiederherstellungen.

- Sie können nicht aus einer verschlüsselten Quelldatenbank wiederherstellen, aber Sie können eine Wiederherstellung in einer verschlüsselten RDS Amazon-DB-Instance durchführen.

Wenn Sie Ihre Datenbank wiederherstellen, wird das Backup kopiert und dann auf Ihre RDS Db2-DB-Instance extrahiert. Wir empfehlen Ihnen, Speicherplatz RDS für Ihre Db2-DB-Instance bereitzustellen, der mindestens der Summe aus Backup-Größe und Größe der ursprünglichen Datenbank auf der Festplatte entspricht.

Die maximale Größe der wiederhergestellten Datenbank ist die maximale Datenbankgröße, die unterstützt wird, abzüglich der Größe der Sicherung. Wenn beispielsweise die maximale Datenbankgröße, die unterstützt wird, 64 TiB und die Größe der Sicherung 30 TiB beträgt, dann beträgt die maximale Größe der wiederhergestellten Datenbank 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Ihre Datenbank auf Amazon S3 sichern

Um Ihre Datenbank auf Amazon S3 zu sichern, benötigen Sie die folgenden AWS Komponenten:

- Ein Amazon S3 S3-Bucket zum Speichern Ihrer Backup-Dateien: Laden Sie alle Backup-Dateien hoch, die Sie zu Amazon migrieren möchtenRDS. Wir empfehlen Ihnen, Offline-Backups für Migrationen zu verwenden, die Ausfallzeiten verkraften können. Wenn Sie bereits einen S3-Bucket haben, können Sie diesen Bucket verwenden. Wenn Sie keinen S3-Bucket haben, finden Sie weitere Informationen unter Bucket erstellen im Amazon S3 S3-Benutzerhandbuch.

  > **Note**
  >
  > Wenn Ihre Datenbank groß ist und die Übertragung in einen S3-Bucket viel Zeit in Anspruch nehmen würde, können Sie ein AWS Snow Family Gerät bestellen und AWS die Sicherung durchführen lassen. Nachdem Sie Ihre Dateien auf das Gerät kopiert und an das Snow Family-Team zurückgegeben haben, überträgt das Team Ihre gesicherten Bilder in Ihren S3-Bucket. Weitere Informationen finden Sie in der AWS Snow Family - Dokumentation.

- Eine IAM Rolle für den Zugriff auf den S3-Bucket: Wenn Sie bereits eine IAM Rolle haben, können Sie diese Rolle verwenden. Wenn Sie noch keine Rolle haben, finden Sie weitere Informationen unterSchritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu.

- Eine IAM Richtlinie mit Vertrauensbeziehungen und Berechtigungen, die mit Ihrer IAM Rolle verknüpft sind: Weitere Informationen finden Sie unterSchritt 1: Erstellen einer IAM-Richtlinie.
- Die IAM Rolle, die Ihrer RDS DB2-DB-Instance hinzugefügt wurde: Weitere Informationen finden Sie unterSchritt 3: Fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu.

Eine automatische Standardspeichergruppe erstellen

Ihre Quelldatenbank muss über eine automatische Standardspeichergruppe verfügen. Wenn Ihre Datenbank keine automatische Standardspeichergruppe hat, müssen Sie eine erstellen.

Um eine automatische Standardspeichergruppe zu erstellen

1. Stellen Sie eine Verbindung zur Quelldatenbank her. Im folgenden Beispiel *source_database* ersetzen Sie es durch den Namen Ihrer Datenbank.

```
db2 connect to source_database
```

2. Erstellen Sie eine automatische Speichergruppe und legen Sie sie als Standard fest. Im folgenden Beispiel *storage_path* ersetzen Sie es durch den absoluten Pfad zu dem Ort, an dem sich die Speichergruppe befindet.

```
db2 "create stogroup IBMSTOGROUP ON storage_path set as default"
```

3. Beenden Sie Backend-Prozesse.

```
db2 terminate
```

4. Deaktivieren Sie die Datenbank und beenden Sie alle Datenbankdienste. Im folgenden Beispiel *source_database* ersetzen Sie es durch den Namen der Datenbank, für die Sie die Speichergruppe erstellt haben.

```
db2 deactivate db source_database
```

5. Erstellen Sie eine Sicherungskopie der Datenbank. Im folgenden Beispiel *source_database* ersetzen Sie es durch den Namen der Datenbank, für die Sie die Speichergruppe erstellt haben. *file_system_path*Ersetzen Sie es durch den absoluten Pfad zu dem Ort, an dem Sie die Datenbank sichern möchten.

```
db2 backup database source_database to file_system_path
```

Ihre Db2-Datenbank wiederherstellen

Nachdem Sie Ihre Datenbank auf Amazon S3 gesichert und eine automatische Speichergruppe erstellt haben, können Sie Ihre Db2-Datenbank auf Ihrer RDS Db2-DB-Instance wiederherstellen.

Um Ihre Db2-Datenbank auf Ihrer RDS für Db2 DB-Instance wiederherzustellen

1. Connect zu Ihrer RDS DB2-DB-Instance her. Weitere Informationen finden Sie unter Verbindung zu Ihrer Db2-DB-Instance herstellen.
2. (Optional) Um sicherzustellen, dass Ihre Datenbank mit den optimalen Einstellungen für den Wiederherstellungsvorgang konfiguriert ist, können Sie aufrufen, the section called “rdsadmin.show_configuration” um die Werte für `RESTORE_DATABASE_PARALLELISM` und `RESTORE_DATABASE_NUM_BUFFERS` zu überprüfen. Rufen Sie the section called “rdsadmin.set_configuration” auf, um diese Werte nach Bedarf zu ändern. Das explizite Festlegen dieser Werte kann die Leistung bei der Wiederherstellung von Datenbanken mit großen Datenmengen verbessern.
3. Stellen Sie Ihre Datenbank wieder her, indem Sie anrufen`rdsadmin.restore_database`. Weitere Informationen finden Sie unter rdsadmin.restore_database.

Migration von Linux to Linux mit nahezu null Ausfallzeiten für Amazon for Db2 RDS

Mit diesem Migrationsansatz migrieren Sie ein Linuxbasierte Db2-Datenbank von einer selbstverwalteten Db2-Datenbank (Quelle) zu Amazon RDS für Db2. Dieser Ansatz führt zu minimalen bis gar keinen Ausfällen oder Ausfallzeiten für die Anwendung oder die Benutzer. Dieser Ansatz sichert Ihre Datenbank und stellt sie mit Protokollwiedergabe wieder her, wodurch Unterbrechungen des laufenden Betriebs vermieden werden und eine hohe Verfügbarkeit Ihrer Datenbank gewährleistet wird.

Um eine Migration ohne Ausfallzeiten zu erreichen, implementiert RDS for Db2 eine Wiederherstellung mit Protokollwiedergabe. Bei diesem Ansatz wird eine Sicherungskopie Ihrer selbst verwalteten Daten verwendet Linuxbasierte Db2-Datenbank und stellt sie auf dem RDS for Db2-Server wieder her. Mit RDS gespeicherten Prozeduren von Amazon wenden Sie dann nachfolgende Transaktionsprotokolle an, um die Datenbank auf den neuesten Stand zu bringen.

Themen

- Einschränkungen und Empfehlungen für eine Migration ohne Ausfallzeiten
- Ihre Datenbank auf Amazon S3 sichern

- Eine automatische Standardspeichergruppe erstellen
- Migrieren Sie Ihre Db2-Datenbank

Einschränkungen und Empfehlungen für eine Migration ohne Ausfallzeiten

Die folgenden Einschränkungen und Empfehlungen gelten für die Verwendung einer Migration ohne Ausfallzeiten:

- Amazon RDS benötigt ein Online-Backup für eine Migration ohne Ausfallzeiten. Dies liegt daran, RDS dass Amazon Ihre Datenbank beim Hochladen Ihrer archivierten Transaktionsprotokolle im Status Rollforward ausstehend hält. Weitere Informationen finden Sie unter the section called “Migrieren Sie Ihre Db2-Datenbank”.
- Sie können keine Wiederherstellung aus einem Amazon S3 S3-Bucket in einer anderen Region als der Region durchführen AWS-Region , in der sich Ihre RDS for Db2-DB-Instance befindet.
- Amazon S3 begrenzt die Größe von Dateien, die in einen S3-Bucket hochgeladen werden, auf 5 TB. Wenn Ihre Datenbank-Backup-Datei 5 TB überschreitet, teilen Sie die Sicherungsdatei in kleinere Dateien auf.
- Amazon unterstützt RDS keine externen Routinen, inkrementelle Wiederherstellungen oder Delta Wiederherstellungen.
- Sie können nicht aus einer verschlüsselten Quelldatenbank wiederherstellen, aber Sie können eine Wiederherstellung in einer verschlüsselten RDS Amazon-DB-Instance durchführen.

Wenn Sie Ihre Datenbank wiederherstellen, RDS kopiert Amazon Ihr Backup und extrahiert es dann auf Ihre RDS for Db2-DB-Instance. Wir empfehlen Ihnen, Speicherplatz RDS für Ihre Db2-DB-Instance bereitzustellen, der mindestens der Summe aus Backup-Größe und Größe der ursprünglichen Datenbank auf der Festplatte entspricht.

Die maximale Größe der wiederhergestellten Datenbank ist die maximale Datenbankgröße, die unterstützt wird, abzüglich der Größe der Sicherung. Wenn beispielsweise die maximale Datenbankgröße, die unterstützt wird, 64 TiB und die Größe der Sicherung 30 TiB beträgt, dann beträgt die maximale Größe der wiederhergestellten Datenbank 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Ihre Datenbank auf Amazon S3 sichern

Um Ihre Datenbank auf Amazon S3 zu sichern, benötigen Sie die folgenden AWS Komponenten:

- Ein Amazon S3 S3-Bucket zum Speichern Ihrer Backup-Dateien: Laden Sie alle Backup-Dateien hoch, die Sie zu Amazon migrieren möchtenRDS. Amazon RDS benötigt ein Online-Backup für eine Migration ohne Ausfallzeiten. Wenn Sie bereits einen S3-Bucket haben, können Sie diesen Bucket verwenden. Wenn Sie keinen S3-Bucket haben, finden Sie weitere Informationen unter Bucket erstellen im Amazon S3 S3-Benutzerhandbuch.

  > **Note**
  >
  > Wenn Ihre Datenbank groß ist und die Übertragung in einen S3-Bucket viel Zeit in Anspruch nehmen würde, können Sie ein AWS Snow Family Gerät bestellen und AWS die Sicherung durchführen lassen. Nachdem Sie Ihre Dateien auf das Gerät kopiert und an das Snow Family-Team zurückgegeben haben, überträgt das Team Ihre gesicherten Bilder in Ihren S3-Bucket. Weitere Informationen finden Sie in der AWS Snow Family - Dokumentation.

- Eine IAM Rolle für den Zugriff auf den S3-Bucket: Wenn Sie bereits eine AWS Identity and Access Management (IAM) -Rolle haben, können Sie diese Rolle verwenden. Wenn Sie keine Rolle haben, finden Sie weitere Informationen unterSchritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu.
- Eine IAM Richtlinie mit Vertrauensbeziehungen und Berechtigungen, die mit Ihrer IAM Rolle verknüpft sind: Weitere Informationen finden Sie unterSchritt 1: Erstellen einer IAM-Richtlinie.
- Die IAM Rolle, die Ihrer RDS DB2-DB-Instance hinzugefügt wurde: Weitere Informationen finden Sie unterSchritt 3: Fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu.

Eine automatische Standardspeichergruppe erstellen

Ihre Quelldatenbank muss über eine automatische Standardspeichergruppe verfügen. Wenn Ihre Datenbank keine automatische Standardspeichergruppe hat, müssen Sie eine erstellen.

Um eine automatische Standardspeichergruppe zu erstellen

1. Stellen Sie eine Verbindung zur Quelldatenbank her. Im folgenden Beispiel *source_database* ersetzen Sie es durch den Namen Ihrer Datenbank.

   ```
   db2 connect to source_database
   ```

2. Erstellen Sie eine automatische Speichergruppe und legen Sie sie als Standard fest. Im folgenden Beispiel *storage_path* ersetzen Sie es durch den absoluten Pfad zu dem Ort, an dem sich die Speichergruppe befindet.

```
db2 "create stogroup IBMSTOGROUP ON storage_path set as default"
```

3. Beenden Sie Backend-Prozesse.

```
db2 terminate
```

Migrieren Sie Ihre Db2-Datenbank

Nachdem Sie Ihre Datenbank auf Amazon S3 gesichert und eine automatische Speichergruppe erstellt haben, können Sie Ihre Db2-Datenbank auf Ihre RDS for Db2-DB-Instance migrieren.

Um eine Migration ohne Ausfallzeiten durchzuführen

1. Führen Sie eine Online-Sicherung Ihrer Quelldatenbank durch. Weitere Informationen finden Sie unter BACKUP DATABASE Befehl in der IBM Db2 -Dokumentation.
2. Kopieren Sie das Backup Ihrer Datenbank in einen Amazon S3 S3-Bucket. Informationen zur Verwendung von Amazon S3 finden Sie im Amazon Simple Storage Service-Benutzerhandbuch.
3. Stellen Sie mit dem *master_username* und *master_password* für Ihre DB2-DB-Instance RDS eine Connect zum `rdsadmin` Server her.

```
db2 connect to rdsadmin user master_username using master_password
```

4. (Optional) Um sicherzustellen, dass Ihre Datenbank mit den optimalen Einstellungen für den Wiederherstellungsvorgang konfiguriert ist, können Sie aufrufen, the section called “rdsadmin.show_configuration” um die Werte für `RESTORE_DATABASE_PARALLELISM` und `RESTORE_DATABASE_NUM_BUFFERS` zu überprüfen. Rufen Sie the section called “rdsadmin.set_configuration” auf, um diese Werte nach Bedarf zu ändern. Das explizite Festlegen dieser Werte kann die Leistung bei der Wiederherstellung von Datenbanken mit großen Datenmengen verbessern.
5. Stellen Sie die Sicherung auf dem RDS for Db2-Server wieder her, indem Sie aufrufen`rdsadmin.restore_database`. Setzen Sie `backup_type` auf `ONLINE`. Weitere Informationen finden Sie unter rdsadmin.restore_database.

6. Kopieren Sie Ihre Archivprotokolle von Ihrem Quellserver in Ihren S3-Bucket. Weitere Informationen finden Sie unter Archiv-Protokollierung im IBM Db2 -Dokumentation.
7. Wenden Sie Archivprotokolle so oft wie nötig an, indem Sie anrufen`rdsadmin.rollforward_database`. Stellen Sie `complete_rollforward` auf ein`FALSE`, um die Datenbank in einem `ROLL-FORWARD PENDING` Zustand zu halten. Weitere Informationen finden Sie unter rdsadmin.rollforward_database.
8. Nachdem Sie alle Archivprotokolle übernommen haben, schalten Sie die Datenbank online, indem Sie anrufen`rdsadmin.complete_rollforward`. Weitere Informationen finden Sie unter rdsadmin.complete_rollforward.
9. Wechseln Sie zwischen den Anwendungsverbindungen RDS zum Db2-Server, indem Sie entweder Ihre Anwendungsendpunkte für die Datenbank aktualisieren oder indem Sie die DNS Endpunkte so aktualisieren, dass sie den Datenverkehr zum Db2-Server umleiten. RDS Sie können auch die automatische Db2-Client-Umleitungsfunktion für Ihre selbstverwaltete Db2-Datenbank mit dem Datenbankendpunkt for Db2 verwenden. RDS Weitere Informationen finden Sie unter Beschreibung und Einrichtung der automatischen Client-Umleitung im IBM Db2 -Dokumentation.
10. (Optional) Fahren Sie Ihre Quelldatenbank herunter.

Synchrone Migration von Linux to Linux für Amazon RDS für Db2

Mit diesem Migrationsansatz richten Sie die Replikation zwischen Ihrer selbst verwalteten Db2-Datenbank und Ihrer Amazon RDS for Db2-DB-Instance ein. An der selbstverwalteten Datenbank vorgenommene Änderungen werden nahezu in Echtzeit auf die RDS for Db2-DB-Instance repliziert. Dieser Ansatz kann für kontinuierliche Verfügbarkeit sorgen und Ausfallzeiten während des Migrationsprozesses minimieren.

Migration von AIX or Windows to Linux für Amazon RDS für Db2

Bei diesem Migrationsansatz verwenden Sie native Db2-Tools, um Ihre selbstverwaltete Db2-Datenbank in einem Amazon S3 S3-Bucket zu sichern. Zu den systemeigenen Db2-Tools gehören das `export` Hilfsprogramm, der `db2move` Systembefehl oder der Systembefehl. `db2look` Ihre Db2-Datenbank kann entweder selbst verwaltet werden oder sich in Amazon Elastic Compute Cloud (AmazonEC2) befinden. Sie können Daten von Ihrem verschieben AIX or Windows System zu Ihrem Amazon S3 S3-Bucket. Verwenden Sie dann einen Db2-Client, um Daten direkt aus dem S3-Bucket in Ihre Amazon RDS for Db2-Datenbank zu laden. Die Ausfallzeit hängt von der Größe Ihrer

Datenbank ab. Weitere Informationen zur Verwendung von Amazon S3 finden Sie unterIntegrieren einer Amazon RDS for Db2-DB-Instance mit Amazon S3.

So migrieren Sie Ihre Db2-Datenbank RDS zu Db2

1. Bereiten Sie die Sicherung Ihrer Datenbank vor. Konfigurieren Sie ausreichend Speicherplatz für das Backup auf Ihrem selbstverwalteten Db2-System.
2. Erstellen Sie eine Sicherungskopie Ihrer Datenbank.
   a. Ausführen des sdb2look Systembefehl zum Extrahieren der Data Definition Language (DDL) -Datei für alle Objekte.
   b. Führen Sie entweder das Db2-Exportdienstprogramm oder das db2move Systembefehl oder ein CREATE EXTERNAL TABLE Anweisung zum Entladen der Db2-Tabellendaten in den Speicher Ihres Db2-Systems.
3. Verschieben Sie Ihr Backup in einen Amazon S3 S3-Bucket. Weitere Informationen finden Sie unter Integrieren einer Amazon RDS for Db2-DB-Instance mit Amazon S3.

   > **Note**
   >
   > Wenn Ihre Datenbank groß ist und die Übertragung in einen S3-Bucket lange dauern würde, können Sie ein AWS Snow Family Gerät bestellen und AWS die Sicherung durchführen lassen. Nachdem Sie Ihre Dateien auf das Gerät kopiert und an das Snow Family-Team zurückgegeben haben, überträgt das Team Ihre gesicherten Bilder in Ihren S3-Bucket. Weitere Informationen finden Sie in der AWS Snow Family -Dokumentation.

4. Verwenden Sie einen Db2-Client, um Daten direkt aus Ihrem S3-Bucket in Ihre RDS for Db2-Datenbank zu laden. Weitere Informationen finden Sie unter Migration mit Amazon S3.

Migration von Db2-Daten über Amazon S3 zu Amazon RDS for Db2

Bei diesem Migrationsansatz speichern Sie zunächst Daten aus einer einzelnen Tabelle in einer Datendatei, die Sie in einem Amazon S3 S3-Bucket platzieren. Anschließend verwenden Sie den LOADBefehl, um die Daten aus dieser Datendatei in eine Tabelle in Ihrer Amazon RDS for Db2-Datenbank zu laden. Weitere Informationen zur Verwendung von Amazon S3 finden Sie unterIntegrieren einer Amazon RDS for Db2-DB-Instance mit Amazon S3.

Themen

- Speichern Ihrer Daten auf Amazon S3
- Laden Sie Ihre Daten in RDS Db2-Tabellen

Speichern Ihrer Daten auf Amazon S3

Um Daten aus einer einzelnen Tabelle in Amazon S3 zu speichern, verwenden Sie ein Datenbank-Hilfsprogramm, um die Daten aus Ihrem Datenbankmanagementsystem (DBMS) in eine CSV Datei zu extrahieren. Laden Sie dann die Datendatei auf Amazon S3 hoch.

Zum Speichern von Datendateien auf Amazon S3 benötigen Sie die folgenden AWS Komponenten:

- Ein Amazon S3 S3-Bucket zum Speichern Ihrer Backup-Dateien: Wenn Sie bereits einen S3-Bucket haben, können Sie diesen Bucket verwenden. Wenn Sie keinen S3-Bucket haben, finden Sie weitere Informationen unter Bucket erstellen im Amazon S3 S3-Benutzerhandbuch.
- Eine IAM Rolle für den Zugriff auf den S3-Bucket: Wenn Sie bereits eine IAM Rolle haben, können Sie diese Rolle verwenden. Wenn Sie keine Rolle haben, finden Sie weitere Informationen unterSchritt 2: Erstellen Sie eine IAM Rolle und fügen Sie Ihre IAM Richtlinie hinzu.
- Eine IAM Richtlinie mit Vertrauensbeziehungen und Berechtigungen, die mit Ihrer IAM Rolle verknüpft sind: Weitere Informationen finden Sie unterSchritt 1: Erstellen einer IAM-Richtlinie.
- Die IAM Rolle, die Ihrer RDS DB2-DB-Instance hinzugefügt wurde: Weitere Informationen finden Sie unterSchritt 3: Fügen Sie Ihre IAM Rolle zu Ihrer RDS for Db2-DB-Instance hinzu.

Laden Sie Ihre Daten in RDS Db2-Tabellen

Nachdem Sie Ihre Datendateien in Amazon S3 gespeichert haben, können Sie die Daten aus diesen Dateien in einzelne Tabellen auf Ihrer RDS for Db2-DB-Instance laden.

Um Ihre Db2-Tabellendaten in Ihre RDS für Db2 DB-Datenbanktabelle zu laden

1. Stellen Sie mit dem Master-Benutzernamen und dem Master-Passwort für Ihre DB2-DB-Instance eine Connect RDS zur `rdsadmin` Datenbank her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Katalogisieren Sie einen Alias für den Speicherzugriff, der auf den Amazon S3 S3-Bucket verweist, in dem Ihre gespeicherten Dateien gespeichert sind. Notieren Sie sich den Namen dieses Alias zur Verwendung im nächsten Schritt. Sie müssen diesen Schritt nur einmal

ausführen, wenn Sie mehrere Tabellen aus Datendateien laden möchten, die im selben Amazon S3 S3-Bucket gespeichert sind.

Das folgende Beispiel katalogisiert einen Alias mit dem Namen *my_s3_alias* das gewährt einem Benutzer mit dem Namen *jorge_souza* Zugriff auf einen Bucket mit dem Namen *amzn-s3-demo-bucket*.

```
db2 "call rdsadmin.catalog_storage_access(?, 'my_s3_alias', 'amzn-s3-demo-bucket',
 'USER', 'jorge_souza')"
```

Weitere Hinweise zu dieser gespeicherten Prozedur finden Sie unterthe section called “rdsadmin.catalog_storage_access”.

3. Führen Sie den LOAD Befehl mit dem Speicherzugriffsalias aus, der auf Ihren Amazon S3 S3-Bucket verweist.

> **Note**
>
> Wenn der LOAD Befehl einen Fehler zurückgibt, müssen Sie möglicherweise einen VPC Gateway-Endpunkt für Amazon S3 erstellen und ausgehende Regeln zur Sicherheitsgruppe hinzufügen. Weitere Informationen finden Sie unter the section called “Ein Fehler in der Datei I/O ”.

Im folgenden Beispiel werden Daten aus einer Datendatei mit dem Namen geladen *my_s3_datafile.csv* in eine Tabelle mit dem Namen *my_db2_table*. Das Beispiel geht davon aus, dass sich die Datendatei in dem Amazon S3 S3-Bucket befindet, den der Alias benannt hat *my_s3_alias* zeigt auf.

```
db2 "load from db2remote://my_s3_alias//my_s3_datafile.csv of DEL insert
 into my_db2_table";
```

Das folgende Beispiel wird LOBs aus einer Datendatei mit dem Namen geladen *my_table1_export.ixf* in eine Tabelle mit dem Namen *my_db2_table*. Das Beispiel geht davon aus, dass sich die Datendatei in dem Amazon S3 S3-Bucket befindet, den der Alias benannt hat *my_s3_alias* zeigt auf.

```
db2 "call sysproc.admin_cmd('load from
 "db2remote://my_s3_alias//my_table1_export.ixf" of ixf
```

```
        lobs from "db2remote://my_s3_alias//" xml from "db2remote://my_s3_alias//"
        modified by lobsinfile implicitlyhiddeninclude identityoverride
generatedoverride periodoverride transactionidoverride
        messages on server
        replace into "my_schema"."my_db2_table"
                              nonrecoverable
        indexing mode incremental allow no access')"
```

Wiederholen Sie diesen Schritt für jede Datendatei im Amazon S3 S3-Bucket, die Sie in eine Tabelle in Ihrer RDS for Db2-DB-Instance laden möchten.

Weitere Informationen zu dem `LOAD` Befehl finden Sie unter LOADBefehl.

Migration zu Amazon RDS für Db2 mit AWS Database Migration Service ()AWS DMS

Sie können es AWS DMS für einmalige Migrationen verwenden und dann von Db2 unter Linux, Unix (wie AIX) und Windows zu Amazon RDS für Db2 synchronisieren. Weitere Informationen finden Sie unter Was ist AWS Database Migration Service?.

Verwendung nativer Db2-Tools zur Migration von Daten von Db2 nach Amazon RDS for Db2

Sie können mehrere native Db2-Tools, -Dienstprogramme und -Befehle verwenden, um Daten direkt von einer Db2-Datenbank in eine Amazon RDS for Db2-Datenbank zu verschieben. Um diese systemeigenen Db2-Tools verwenden zu können, müssen Sie in der Lage sein, Ihren Client-Computer mit einer RDS Db2-DB-Instance zu verbinden. Weitere Informationen finden Sie unter Einen Client-Computer mit einer Amazon RDS for Db2-DB-Instance verbinden.

> **Note**
>
> Eine andere Möglichkeit, Ihre Daten zu verschieben, besteht darin, sie zuerst in einem Amazon S3 S3-Bucket zu speichern und dann den `LOAD` Befehl zu verwenden, um diese Daten in eine Tabelle in Ihrer RDS for Db2-Datenbank zu übertragen. Diese Methode bietet aufgrund der guten Netzwerkkonnektivität zwischen RDS Db2 und S3 die beste Leistung bei der Migration großer Datenmengen. Weitere Informationen finden Sie unter the section called “Migration mit Amazon S3”.

Name des Tools	Anwendungsfall	Einschränkungen
db2look	Kopieren von Metadaten aus einer selbstverwalteten Db2-Datenbank in eine RDS für Db2-Datenbank.	• Sie müssen die Syntax zum Erstellen von Pufferpoo ls, zum Erstellen von Tablespaces und zum Erstellen von Rollen so ändern, dass sie der Syntax von entspricht. Gespeiche rte RDS-Prozeduren für Db2
IMPORT Befehl	Migration kleiner Tabellen und Tabellen mit großen Objekten (LOBs) von einem Client-Co mputer zur RDS for Db2-DB-In stance.	• Langsamer als das `LOAD` Hilfsprogramm aufgrund von `INSERT` `DELETE` Protokoll ierungsvorgängen. • Schlechte Leistung bei begrenzter Netzwerkb andbreite.
INGEST Hilfsprogramm	Kontinuierliches Streamen von Daten aus Dateien und Pipes ohne große Objekte (LOBs) auf dem Client-Computer zur RDS for Db2-DB-Instance. Unterstützungen `INSERT` und `MERGE` Operationen.	• Datendateien, die enthalten , können nicht gestreamt LOBs werden. Verwenden Sie stattdessen den `IMPORT` Befehl. • Konnektivität zwischen der selbstverwalteten Db2-Datenbank und RDS der Db2-Datenbank erforderlich.
INSERT Befehl	Kopieren von Daten in kleinen Tabellen aus einer selbstver walteten Db2-Datenbank in eine RDS For-Db2-Datenbank.	• Konnektivität zwischen der selbstverwalteten Db2-Datenbank und der Db2-Datenbank erforderlich. RDS

Name des Tools	Anwendungsfall	Einschränkungen
		• Schlechte Leistung bei begrenzter Netzwerkb andbreite.
LOAD CLIENT Befehl	Migration kleiner Tabellen ohne große Objekte (LOBs) von einem Client-Computer zur RDS for Db2-DB-Instance.	• Datendateien, die enthalten , können nicht migriert werden. LOBs Verwenden Sie stattdessen den `IMPORT` Befehl. • Schlechte Leistung bei begrenzter Netzwerkb andbreite.

Einen Client-Computer mit einer Amazon RDS for Db2-DB-Instance verbinden

Um eines der nativen Db2-Tools zum Verschieben von Daten aus einer Db2-Datenbank in eine Amazon RDS for Db2-Datenbank zu verwenden, müssen Sie zuerst Ihren Client-Computer mit einer RDS for Db2-DB-Instance verbinden.

Bei dem Client-Computer kann es sich um einen der folgenden Computer handeln:

- Eine Amazon Elastic Compute Cloud (AmazonEC2) -Instanz auf Linux, Windows, oder macOS. Diese Instanz sollte sich in derselben Virtual Private Cloud (VPC) befinden wie Ihre RDS Db2-DB-Instance, AWS Cloud9, oder AWS CloudShell.
- Eine selbstverwaltete Db2-Instance in einer EC2 Amazon-Instance. Die Instanzen sollten sich in derselben befinden. VPC
- Eine selbstverwaltete Db2-Instance in einer EC2 Amazon-Instance. Die Instances können unterschiedlich sein, VPCs wenn Sie Peering aktiviert VPC haben. Weitere Informationen finden Sie unter Erstellen einer VPC Peering-Verbindung im Amazon Virtual Private Cloud VPC Peering Guide.
- Ein lokaler Computer läuft Linux, Windows, oder macOS in einer selbstverwalteten Umgebung. Sie müssen entweder über öffentliche Konnektivität RDS für Db2 verfügen oder VPN Konnektivität zwischen selbstverwalteten Db2-Instanzen aktivieren und. AWS

Um Ihren Client-Computer mit Ihrer RDS for Db2-DB-Instance zu verbinden, melden Sie sich bei Ihrem Client-Computer mit IBM Db2 Data Management Console. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance undIBM Db2 Data Management Console.

Sie können AWS Database Migration Service (AWS DMS) verwenden, um Abfragen in der Datenbank auszuführen, einen SQL Ausführungsplan auszuführen und die Datenbank zu überwachen. Weitere Informationen finden Sie unter Was ist der AWS Database Migration Service? im AWS Database Migration Service Benutzerhandbuch.

Nachdem Sie Ihren Client-Computer erfolgreich mit Ihrer RDS for Db2-DB-Instance verbunden haben, können Sie jedes native Db2-Tool zum Kopieren von Daten verwenden. Weitere Informationen finden Sie unter Verwendung nativer Db2-Tools zur Migration von Daten von Db2 nach Amazon RDS for Db2.

Kopieren von Datenbank-Metadaten von Db2 nach Amazon RDS for Db2 mit db2look

`db2look`ist ein systemeigenes Db2-Tool, das Dateien, Objekte, Autorisierungen, Konfigurationen und Datenbanklayouts in der Datendefinitionssprache (DDL) extrahiert. WLM Sie können es verwenden`db2look`, um Datenbankmetadaten aus einer selbstverwalteten Db2-Datenbank in eine Amazon RDS for Db2-Datenbank zu kopieren. Weitere Informationen finden Sie unter Datenbanken mit db2look nachahmen im IBM Db2 -Dokumentation.

Um die Datenbank-Metadaten zu kopieren

1. Führen Sie das `db2look` Tool auf Ihrem selbstverwalteten Db2-System aus, um die DDL Datei zu extrahieren. Ersetzen Sie im folgenden Beispiel *`database_name`* durch den Namen Ihrer Db2-Datenbank.

   ```
   db2look -d database_name -e -l -a -f -wlm -cor -createdb -printdbcfg -o db2look.sql
   ```

2. Wenn Ihr Client-Computer Zugriff auf die Quelldatenbank (selbstverwaltete Db2) und die RDS für Db2-DB-Instance hat, können Sie die `db2look.sql` Datei auf dem Client-Computer erstellen, indem Sie sie direkt an die Remote-Instance anhängen. Katalogisieren Sie anschließend die selbstverwaltete Db2-Remote-Instanz.

   a. Katalogisieren Sie den Knoten. Ersetzen Sie im folgenden Beispiel *`dns_ip_address`* and *`port`* durch den DNS Namen oder die IP-Adresse und die Portnummer der selbstverwalteten Db2-Datenbank.

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

b. Katalogisieren Sie die Datenbank. Ersetzen Sie im folgenden Beispiel *source_database_name* and *source_database_alias* durch den Namen der selbstverwalteten Db2-Datenbank und den Alias, den Sie für diese Datenbank verwenden möchten.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

c. An die Quelldatenbank anhängen. Ersetzen Sie im folgenden Beispiel *source_database_alias*, *user_id*, und *user_password* durch den Alias, den Sie im vorherigen Schritt erstellt haben, sowie durch die Benutzer-ID und das Kennwort für die selbstverwaltete Db2-Datenbank.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

3. Wenn Sie vom Client-Computer aus nicht auf die selbstverwaltete Db2-Remote-Datenbank zugreifen können, kopieren Sie die `db2look.sql` Datei auf den Client-Computer. Katalogisieren Sie dann die RDS für die DB2-DB-Instance.

a. Katalogisieren Sie den Knoten. Ersetzen Sie im folgenden Beispiel *dns_ip_address* and *port* durch den DNS Namen oder die IP-Adresse und die Portnummer der RDS for Db2-DB-Instance.

```
db2 catalog tcpip node remnode REMOTE dns_ip_address server port
```

b. Katalogisieren Sie die Datenbank. Ersetzen Sie im folgenden Beispiel *rds_database_name* and *rds_database_alias* durch den Namen der RDS for Db2-Datenbank und den Alias, den Sie für diese Datenbank verwenden möchten.

```
db2 catalog database rds_database_name as rds_database_alias at node remnode \
    authentication server_encrypt
```

c. Katalogisieren Sie die Admin-Datenbank, die RDS für Db2 verwaltet wird. Sie können diese Datenbank nicht zum Speichern von Daten verwenden.

```
db2 catalog database rdsadmin as rdsadmin at node remnode authentication
 server_encrypt
```

4. Erstellen Sie Pufferpools und Tablespaces. Der Administrator hat keine Rechte zum Erstellen von Pufferpools oder Tablespaces. Sie können jedoch RDS gespeicherte Amazon-Prozeduren verwenden, um sie zu erstellen.

   a. Suchen Sie in der Datei nach den Namen und Definitionen der Pufferpools und Tablespaces. `db2look.sql`

   b. Stellen Sie RDS mithilfe des Master-Benutzernamens und des Master-Passworts RDS für Ihre DB2-DB-Instance eine Connect zu Amazon her. Ersetzen Sie im folgenden Beispiel *`master_username`* and *`master_password`* durch Ihre eigenen Informationen.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

   c. Erstellen Sie einen Pufferpool durch Aufrufen`rdsadmin.create_bufferpool`. Weitere Informationen finden Sie unter rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool(
       'database_name',
       'buffer_pool_name',
       buffer_pool_size,
       'immediate',
       'automatic',
       page_size,
       number_block_pages,
       block_size)"
   ```

   d. Erstellen Sie einen Tablespace durch Aufrufen`rdsadmin.create_tablespace`. Weitere Informationen finden Sie unter rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace(
       'database_name',
       'tablespace_name',
       'buffer_pool_name',
       tablespace_initial_size,
       tablespace_increase_size,
       'tablespace_type')"
   ```

e. Wiederholen Sie die Schritte c oder d für jeden zusätzlichen Pufferpool oder Tablespace, den Sie hinzufügen möchten.

f. Beenden Sie Ihre Verbindung.

```
db2 terminate
```

5. Erstellen Sie Tabellen und Objekte.

a. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts RDS für Ihre for Db2-DB-Instance eine Connect zu Ihrer RDS for Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_name*, *master_username*, und *master_password* durch Ihre eigenen Informationen.

```
db2 connect to rds_database_name user master_username using master_password
```

b. Führen Sie die Datei `db2look.sql` aus.

```
db2 -tvf db2look.sql
```

c. Beenden Sie Ihre Verbindung.

```
db2 terminate
```

Importieren von Daten von einem Client-Computer nach Amazon RDS for Db2 mit dem IMPORT command

Sie können den `IMPORT` Befehl von einem Client-Computer aus verwenden, um Ihre Daten in den Amazon RDS for Db2-Server zu importieren.

> **Important**
>
> Die `IMPORT` Befehlsmethode ist nützlich für die Migration kleiner Tabellen und Tabellen mit großen Objekten ()LOBs. Der `IMPORT` Befehl ist aufgrund der `DELETE` Protokollierungsvorgänge langsamer als das `LOAD INSERT` Hilfsprogramm. Wenn Ihre Netzwerkbandbreite zwischen dem Client-Computer und RDS für Db2 begrenzt ist, empfehlen wir Ihnen, einen anderen Migrationsansatz zu verwenden. Weitere Informationen

finden Sie unter Verwendung nativer Db2-Tools zur Migration von Daten von Db2 nach Amazon RDS for Db2.

Um Daten in den RDS for Db2-Server zu importieren

1. Melden Sie sich bei Ihrem Client-Computer an mit IBM Db2 Data Management Console. Weitere Informationen finden Sie unterHerstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 Data Management Console.
2. Katalogisieren Sie die RDS for Db2-Datenbank auf dem Client-Computer.

   a. Katalogisieren Sie den Knoten. Ersetzen Sie im folgenden Beispiel *dns_ip_address* and *port* durch den DNS Namen oder die IP-Adresse und die Portnummer der selbstverwalteten Db2-Datenbank.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Katalogisieren Sie die Datenbank. Ersetzen Sie im folgenden Beispiel *source_database_name* and *source_database_alias* durch den Namen der selbstverwalteten Db2-Datenbank und den Alias, den Sie für diese Datenbank verwenden möchten.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. An die Quelldatenbank anhängen. Ersetzen Sie im folgenden Beispiel *source_database_alias*, *user_id*, und *user_password* durch den Alias, den Sie im vorherigen Schritt erstellt haben, sowie durch die Benutzer-ID und das Kennwort für die selbstverwaltete Db2-Datenbank.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Generieren Sie die Datendatei mithilfe des `EXPORT` Befehls auf Ihrem selbstverwalteten Db2-System. Ersetzen Sie im folgenden Beispiel *directory* durch das Verzeichnis auf Ihrem Client-Computer, in dem Ihre Datendatei existiert. Ersetzen *file_name* and *table_name* mit dem Namen der Datendatei und dem Namen der Tabelle.

```
db2 "export to /directory/file_name.txt of del lobs to /directory/lobs/ \
    modified by coldel\| select * from table_name"
```

5. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts RDS für Ihre for Db2-DB-Instance eine Connect zu Ihrer RDS for Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_alias*, *master_username,* and *master_password* durch Ihre eigenen Informationen.

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Verwenden Sie den `IMPORT` Befehl, um Daten aus einer Datei auf dem Client-Computer in die Remote-Datenbank RDS für Db2 zu importieren. Weitere Informationen finden Sie unter IMPORT Befehl in der IBM Db2 -Dokumentation. Ersetzen Sie im folgenden Beispiel *directory* and *file_name* durch das Verzeichnis auf Ihrem Client-Computer, in dem Ihre Datendatei existiert, und den Namen der Datendatei. Ersetzen *SCHEMA_NAME* and *TABLE_NAME* mit dem Namen Ihres Schemas und Ihrer Tabelle.

```
db2 "IMPORT from /directory/file_name.tbl OF DEL LOBS FROM /directory/lobs/ \
    modified by coldel\| replace into SCHEMA_NAME.TABLE_NAME"
```

7. Beenden Sie Ihre Verbindung.

```
db2 terminate
```

Importieren von Daten von einem Client-Computer nach Amazon RDS for Db2 mit dem LOAD command

Sie können den `LOAD CLIENT` Befehl verwenden, um Daten aus einer Datei auf einem Client-Computer auf den RDS for Db2-Server zu laden. Da keine SSH Konnektivität zum RDS for Db2-Server besteht, können Sie den `LOAD CLIENT` Befehl entweder auf Ihrem selbst verwalteten Db2-Server oder auf Ihrem Db2-Client-Computer verwenden.

> **Important**
>
> Die `LOAD CLIENT` Befehlsmethode ist nützlich für die Migration kleiner Tabellen. Wenn Ihre Netzwerkbandbreite zwischen dem Client und RDS für Db2 begrenzt ist, empfehlen wir Ihnen,

einen anderen Migrationsansatz zu verwenden. Weitere Informationen hierzu finden Sie unter Verwendung nativer Db2-Tools zur Migration von Daten von Db2 nach Amazon RDS for Db2. Wenn Ihre Datendatei Verweise auf große Objektdateinamen enthält, funktioniert der `LOAD` Befehl nicht, da sich große Objekte (LOBs) auf dem Db2-Server befinden müssen. Wenn Sie versuchen, LOBs vom Client-Computer auf den RDS Db2-Server zu laden, erhalten Sie eine SQL3025N Fehler. Verwenden der IMPORT Befehl stattdessen.

Um Daten auf den RDS für Db2 Server zu laden

1. Melden Sie sich bei Ihrem Client-Computer an mit IBM Db2 Data Management Console. Weitere Informationen finden Sie unterHerstellen einer Verbindung zu Ihrer Amazon RDS for Db2-DB-Instance mit IBM Db2 Data Management Console.
2. Katalogisieren Sie die RDS for Db2-Datenbank auf dem Client-Computer.

   a. Katalogisieren Sie den Knoten. Ersetzen Sie im folgenden Beispiel *dns_ip_address* and *port* durch den DNS Namen oder die IP-Adresse und die Portnummer der selbstverwalteten Db2-Datenbank.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Katalogisieren Sie die Datenbank. Ersetzen Sie im folgenden Beispiel *source_database_name* and *source_database_alias* durch den Namen der selbstverwalteten Db2-Datenbank und den Alias, den Sie für diese Datenbank verwenden möchten.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. An die Quelldatenbank anhängen. Ersetzen Sie im folgenden Beispiel *source_database_alias*, *user_id*, und *user_password* durch den Alias, den Sie im vorherigen Schritt erstellt haben, sowie durch die Benutzer-ID und das Kennwort für die selbstverwaltete Db2-Datenbank.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Generieren Sie die Datendatei mithilfe des EXPORT Befehls auf Ihrem selbstverwalteten Db2-System. Ersetzen Sie im folgenden Beispiel *directory* durch das Verzeichnis auf Ihrem Client-Computer, in dem sich Ihre Datendatei befindet. Ersetzen *file_name* and *TABLE_NAME* mit dem Namen der Datendatei und dem Namen der Tabelle.

```
db2 "export to /directory/file_name.txt of del modified by coldel\| \
    select * from TPCH.TABLE_NAME"
```

5. Stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts RDS für Ihre for Db2-DB-Instance eine Connect zu Ihrer RDS for Db2-Datenbank her. Ersetzen Sie im folgenden Beispiel *rds_database_alias*, *master_username*, und *master_password* durch Ihre eigenen Informationen.

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Verwenden Sie den LOAD Befehl, um Daten aus einer Datei auf dem Client-Computer in die Remote-Datenbank RDS für Db2 zu laden. Weitere Informationen finden Sie unter LOAD Befehl in der IBM Db2 -Dokumentation. Ersetzen Sie im folgenden Beispiel *directory* durch das Verzeichnis auf Ihrem Client-Computer, in dem sich Ihre Datendatei befindet. Ersetzen *file_name* and *TABLE_NAME* mit dem Namen der Datendatei und dem Namen der Tabelle.

```
db2 "LOAD CLIENT from /directory/file_name.txt \
    modified by coldel\| replace into TPCH.TABLE_NAME \
    nonrecoverable without prompting"
```

7. Beenden Sie Ihre Verbindung.

```
db2 terminate
```

Daten von Db2 nach Amazon RDS for Db2 importieren mit dem INSERT command

Sie können den INSERT Befehl von einem selbstverwalteten Db2-Server aus verwenden, um Ihre Daten in eine Amazon RDS for Db2-Datenbank einzufügen. Bei diesem Migrationsansatz verwenden Sie einen Spitznamen für die RDS Remote-For-DB2-DB-Instance. Ihre selbstverwaltete Db2-Datenbank (Quelle) muss in der Lage sein, eine Verbindung zur RDS for Db2-Datenbank (Ziel) herzustellen.

> **Important**
>
> Die `INSERT` Befehlsmethode ist nützlich für die Migration kleiner Tabellen. Wenn Ihre Netzwerkbandbreite zwischen Ihrer selbst verwalteten Db2-Datenbank und RDS der Db2-Datenbank begrenzt ist, empfehlen wir Ihnen, einen anderen Migrationsansatz zu verwenden. Weitere Informationen finden Sie unter Verwendung nativer Db2-Tools zur Migration von Daten von Db2 nach Amazon RDS for Db2.

Um Daten aus einer selbstverwalteten Db2-Datenbank in eine für Db2-Datenbank zu kopieren RDS

1. Katalogisieren Sie die RDS for Db2-DB-Instance auf der selbstverwalteten Db2-Instance.

   a. Katalogisieren Sie den Knoten. Ersetzen Sie im folgenden Beispiel *`dns_ip_address`* and *`port`* durch den DNS Namen oder die IP-Adresse und die Portnummer der selbstverwalteten Db2-Datenbank.

   ```
   db2 catalog tcpip node remnode REMOTE dns_ip_address SERVER port
   ```

   b. Katalogisieren Sie die Datenbank. Ersetzen Sie im folgenden Beispiel *`rds_database_name`* durch den Namen der Datenbank auf Ihrer RDS DB2-DB-Instance.

   ```
   db2 catalog database rds_database_name as remdb at node remnode \
       authentication server_encrypt
   ```

2. Aktivieren Sie den Verbund auf der selbstverwalteten Db2-Instance. Ersetzen Sie im folgenden Beispiel *`source_database_name`* durch den Namen Ihrer Datenbank auf der selbstverwalteten Db2-Instanz.

   ```
   db2 update dbm cfg using FEDERATED YES source_database_name
   ```

3. Erstellen Sie Tabellen auf der RDS for Db2-DB-Instance.

   a. Katalogisieren Sie den Knoten. Ersetzen Sie im folgenden Beispiel *`dns_ip_address`* and *`port`* durch den DNS Namen oder die IP-Adresse und die Portnummer der selbstverwalteten Db2-Datenbank.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

b. Katalogisieren Sie die Datenbank. Ersetzen Sie im folgenden Beispiel *source_database_name* and *source_database_alias* durch den Namen der selbstverwalteten Db2-Datenbank und den Alias, den Sie für diese Datenbank verwenden möchten.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

4. An die Quelldatenbank anhängen. Ersetzen Sie im folgenden Beispiel *source_database_alias*, *user_id*, und *user_password* durch den Alias, den Sie im vorherigen Schritt erstellt haben, sowie durch die Benutzer-ID und das Kennwort für die selbstverwaltete Db2-Datenbank.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

5. Richten Sie den Verbund ein und erstellen Sie einen Spitznamen für die Datenbanktabelle RDS for Db2 auf der selbstverwalteten Db2-Instanz.

a. Connect zu Ihrer lokalen Datenbank her. Ersetzen Sie im folgenden Beispiel *source_database_name* durch den Namen der Datenbank auf Ihrer selbstverwalteten Db2-Instanz.

```
db2 connect to source_database_name
```

b. Erstellen Sie einen Wrapper für den Zugriff auf Db2-Datenquellen.

```
db2 create wrapper drda
```

c. Definieren Sie eine Datenquelle in einer Verbunddatenbank. Ersetzen Sie im folgenden Beispiel *admin* and *admin_password* mit Ihren Anmeldeinformationen für Ihre selbstverwaltete Db2-Instanz. Ersetzen *rds_database_name* mit dem Namen der Datenbank auf Ihrer RDS Db2-DB-Instance.

```
db2 "create server rdsdb2 type DB2/LUW version '11.5.9.0' \
    wrapper drda authorization "admin" password "admin_password" \
    options( dbname 'rds_database_name', node 'remnode')"
```

d. Ordnen Sie die Benutzer in den beiden Datenbanken zu. Ersetzen Sie im folgenden Beispiel *master_username* and *master_password* mit Ihren Anmeldeinformationen RDS für Ihre DB2-DB-Instance.

```
db2 "create user mapping for user server rdsdb2 \
    options (REMOTE_AUTHID 'master_username', REMOTE_PASSWORD
 'master_password')"
```

e. Überprüfen Sie die Verbindung zum RDS for Db2-Server.

```
db2 set passthru rdsdb2
```

f. Erstellen Sie einen Spitznamen für die Tabelle in der Remote-Datenbank RDS für Db2. Ersetzen Sie im folgenden Beispiel *NICKNAME* and *TABLE_NAME* durch einen Spitznamen für die Tabelle und den Namen der Tabelle.

```
db2 create nickname REMOTE.NICKNAME for RDSDB2.TABLE_NAME.NICKNAME
```

6. Fügt Daten in die Tabelle in der entfernten Datenbank RDS für Db2 ein. Verwenden Sie den Spitznamen in einer `select` Anweisung für die lokale Tabelle in der selbstverwalteten Db2-Instanz. Ersetzen Sie im folgenden Beispiel *NICKNAME* and *TABLE_NAME* durch einen Spitznamen für die Tabelle und den Namen der Tabelle.

```
db2 "INSERT into REMOTE.NICKNAME select * from RDS2DB2.TABLE_NAME.NICKNAME"
```

Importieren von Daten von Db2 nach Amazon RDS for Db2 mit dem INGEST Nützlichkeit

Sie können das `INGEST` Hilfsprogramm verwenden, um kontinuierlich Daten aus Dateien und Pipes auf einem Client-Computer zu einer Amazon RDS for Db2-DB-Ziel-Instance zu streamen. Das `INGEST` Hilfsprogramm unterstützt `INSERT` und funktioniert`MERGE`. Weitere Informationen finden Sie unter Ingest Utility im IBM Db2 -Dokumentation.

Da das `INGEST` Hilfsprogramm Spitznamen unterstützt, können Sie es verwenden, um Daten aus Ihrer selbstverwalteten Db2-Datenbank in eine RDS For-Db2-Datenbank zu übertragen. Dieser Ansatz funktioniert, solange Netzwerkkonnektivität zwischen den beiden Datenbanken besteht.

> **Important**
>
> Das `INGEST` Hilfsprogramm unterstützt keine großen Objekte (LOBs). Verwenden Sie stattdessen den `IMPORT`Befehl.

Um die `RESTARTABLE` Funktion des `INGEST` Dienstprogramms zu verwenden, führen Sie den folgenden Befehl in der RDS for Db2-Datenbank aus.

```
db2 "call sysproc.sysinstallobjects('INGEST','C',NULL,NULL)"
```

Amazon RDS für den Db2-Verbund

Sie können Ihre Amazon RDS for Db2-Datenbank als Verbunddatenbank verwenden. Nachdem Sie den Verbund RDS für Db2 eingerichtet haben, können Sie von Ihrer RDS for Db2-Datenbank aus auf Daten aus mehreren Datenbanken zugreifen und diese abfragen. Der Verbund erspart Ihnen die Migration von Daten in Ihre RDS for Db2-Datenbank oder die Konsolidierung von Daten in einer einzigen Datenbank.

Wenn Sie Ihre RDS for Db2-Datenbank als föderierte Datenbank verwenden, können Sie weiterhin auf alle Funktionen von RDS for Db2 zugreifen und verschiedene Vorteile nutzen AWS-Services, und das alles, während Ihre Daten in verschiedenen Datenbanken gespeichert werden. Sie können sowohl einen homogenen Verband einrichten, der verschiedene Datenbanken desselben Typs verbindet, als auch einen heterogenen Verband, der verschiedene Datenbanken unterschiedlichen Typs verbindet.

Sie verbinden zuerst Ihre Db2-Datenbank in RDS for Db2 mit entfernten Datenbanken. Dann können Sie Abfragen für alle Ihre verbundenen Datenbanken ausführen. Sie können beispielsweise eine SQL JOIN Anweisung ausführen, die Tabellen in Ihrer RDS for Db2-Datenbank mit Tabellen in einer entfernten Db2 on z/OS-Datenbank verknüpft.

Themen

- Homogener Verbund
- Heterogener Verband

Homogener Verbund

Sie können einen homogenen Verbund zwischen Ihrer RDS for Db2-Datenbank und der folgenden Db2-Produktfamilie einrichten:

- Db2 für LinuxUNIX, Windows () LUW
- Db2 iSeries
- Db2 für z/OS

RDSfür Db2 unterstützt der homogene Verbund die folgenden Aktionen nicht:

- `CATALOG`Befehle ausführen, um ein Knotenverzeichnis und eine entfernte Datenbank in einer Hostdatenbank RDS für DB2 einzurichten

- Einrichtung des Workload Balancing (WLB) beim Verbund mit Db2 unter z/OS
- Konfiguration der IBM Datenserver-Treiber-Konfigurationsdatei () `db2dsdriver.cfg`

RDSFür den homogenen DB2-Verbund gelten die folgenden Anforderungen:

- Sie müssen den DRDA Wrapper im `UNFENCED` Modus erstellen. Wenn Sie dies nicht tun, funktioniert der Verbund in RDS Db2 nicht.
- Sie müssen eingehenden und ausgehenden Datenverkehr von Ihrer RDS für DB2 Hostdatenbank zu Ihren Remote-Hostdatenbanken zulassen. Weitere Informationen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

Themen

- Schritt 1: Erstellen Sie einen DRDA Wrapper und einen Verbundserver
- Schritt 2: Erstellen Sie eine Benutzerzuordnung
- Schritt 3: Überprüfen Sie die Verbindung

Schritt 1: Erstellen Sie einen DRDA Wrapper und einen Verbundserver

Für einen homogenen Verbund erstellen Sie einen DRDA Wrapper und einen Verbundserver. Die Verbindung zum Remote-Host verwendet`HOST`,`PORT`, und. `DBNAME`

Wählen Sie je nach Typ Ihrer entfernten Db2-Datenbank eine der folgenden Methoden:

- Db2 für Linux und Windows (LUX) -Datenbank — Führen Sie die folgenden SQL Befehle aus. UNIX Ersetzen Sie im folgenden Beispiel *`server_name`* durch den Namen des Servers, den Sie für den Verbund verwenden werden. Ersetzen *`db2_version`* mit der Version Ihrer entfernten Db2-Datenbank. Ersetzen *`username`* and *`password`* mit Ihren Anmeldeinformationen für die entfernte Db2-Datenbank, zu der Sie eine Verbindung herstellen möchten. Ersetzen *`db_name`*, *`dns_name`*, und *`port`* mit den entsprechenden Werten für die entfernte Db2-Datenbank, zu der Sie eine Verbindung herstellen möchten.

```
create wrapper drda options(DB2_FENCED 'N');
create server server_name type DB2/LUW wrapper drda version 'db2_version'
 authorization "master_username" password "master_password" options (add DBNAME
 'db_name',add HOST 'dns_name',add PORT 'port');
```

Beispiel

```
create wrapper drda options(DB2_FENCED 'N');
create server SERVER1 type DB2/LUW wrapper drda version '11.5' authorization
 "sysuser" password "******" options (add DBNAME 'TESTDB2',add HOST
 'ip-123-45-67-899.us-west-1.compute.internal',add PORT '25010');
```

- Db2 iSeries — Führen Sie die folgenden SQL Befehle aus. Ersetzen Sie im folgenden Beispiel *wrapper_name* and *library_name* durch einen Namen für Ihren DRDA Wrapper und die Wrapper-Bibliotheksdatei. Ersetzen *server_name* mit dem Namen des Servers, den Sie für den Verbund verwenden werden. Ersetzen *db2_version* mit der Version Ihrer entfernten Db2-Datenbank. Ersetzen *username* and *password* mit Ihren Anmeldeinformationen für die entfernte Db2-Datenbank, zu der Sie eine Verbindung herstellen möchten. Ersetzen *dns_name*, *port*, und *db_name* mit den entsprechenden Werten für die entfernte Db2-Datenbank, zu der Sie eine Verbindung herstellen möchten.

```
create wrapper wrapper_name library 'library name' options(DB2_FENCED 'N');
create server server_name type db2/mvs version db2_version wrapper wrapper_name
 authorization "sername" password "password" options (HOST 'dns_name', PORT 'port',
 DBNAME 'db_name');
```

Beispiel

```
create wrapper WRAPPER1 library 'libdb2drda.so' options(DB2_FENCED 'N');
create server SERVER1 type db2/mvs version 11 wrapper WRAPPER1 authorization
 "sysuser" password "******" options (HOST 'test1.123.com', PORT '446', DBNAME
 'STLEC1');
```

- Db2 for z/OS — Führen Sie die folgenden Befehle aus. SQL Ersetzen Sie im folgenden Beispiel *wrapper_name* and *library_name* durch einen Namen für Ihren DRDA Wrapper und die Wrapper-Bibliotheksdatei. Ersetzen *server_name* mit dem Namen des Servers, den Sie für den Verbund verwenden werden. Ersetzen *db2_version* mit der Version Ihrer entfernten Db2-Datenbank. Ersetzen *username* and *password* mit Ihren Anmeldeinformationen für die entfernte Db2-Datenbank, zu der Sie eine Verbindung herstellen möchten. Ersetzen *dns_name*, *port*, und *db_name* mit den entsprechenden Werten für die entfernte Db2-Datenbank, zu der Sie eine Verbindung herstellen möchten.

```
create wrapper wrapper_name library 'library_name' options(DB2_FENCED 'N');
```

```
create server server_name type db2/mvs version db2_version wrapper wrapper_name
 authorization "username" password "password" options (HOST 'dns_name', PORT 'port',
 DBNAME 'db_name');
```

Beispiel

```
create wrapper WRAPPER1 library 'libdb2drda.so' OPTIONS(DB2_FENCED 'N');
create server SERVER1 type db2/mvs version 11 wrapper WRAPPER1 authorization
 "sysuser" password "******" options (HOST 'test1.123.com', PORT '446', DBNAME
 'STLEC1');
```

Schritt 2: Erstellen Sie eine Benutzerzuordnung

Erstellen Sie eine Benutzerzuordnung, um Ihren Verbundserver mit Ihrem Datenquellenserver zu verknüpfen, indem Sie den folgenden SQL Befehl ausführen. Ersetzen Sie im folgenden Beispiel *server_name* durch den Namen des Remoteservers, auf dem Sie Operationen ausführen möchten. Dies ist der Server, den Sie in Schritt 1 erstellt haben. Ersetzen *username* and *password* mit Ihren Anmeldeinformationen für diesen Remoteserver.

```
create user mapping for user server server_name options (REMOTE_AUTHID 'username',
 REMOTE_PASSWORD 'password');
```

Weitere Informationen finden Sie unter Benutzerzuordnungen im IBM Db2 -Dokumentation.

Schritt 3: Überprüfen Sie die Verbindung

Vergewissern Sie sich, dass die Einrichtung Ihres Verbunds erfolgreich war, indem Sie die Verbindung überprüfen. Öffnen Sie eine Sitzung, um mithilfe des SET PASSTHRU Befehls systemeigene SQL Befehle an Ihre Remote-Datenquelle zu senden, und erstellen Sie dann eine Tabelle auf dem Remote-Datenserver.

1. Öffnen und schließen Sie eine Sitzung, um SQL sie an eine Datenquelle zu senden. Ersetzen Sie im folgenden Beispiel *server_name* durch den Namen des Servers, den Sie in Schritt 1 für den Verbund erstellt haben.

   ```
   set passthru server_name;
   ```

2. Erstellen Sie eine neue Tabelle. Ersetzen Sie im folgenden Beispiel *column_name*, *data_type*, und *value* durch die entsprechenden Elemente für Ihre Tabelle.

```
create table table_name
 ( column_name data_type(value), column_name data_type(value);
```

Weitere Informationen finden Sie in der CREATETABLEErklärung IBM Db2 -Dokumentation.

3. Erstellen Sie einen Index, fügen Sie Werte für Zeilen in die Tabelle ein und setzen Sie die Verbindung zurück. Beim Zurücksetzen der Verbindung wird die Verbindung unterbrochen, die Back-End-Prozesse werden jedoch beibehalten. Ersetzen Sie im folgenden Beispiel *index_name*, *table_name*, *column_name*, und *columnx_value* mit Ihren Informationen.

```
create index index_name on table_name(column_name);
insert into table_name values(column1_value,column2_value,column3_value);
insert into table_name values(column1_value,column2_value,column3_value);
set passthru reset;

connect reset;
```

4. Stellen Sie eine Connect zu Ihrer Remote-Db2-Datenbank her, erstellen Sie einen Spitznamen für Ihren Remoteserver und führen Sie Operationen aus. Wenn Sie mit dem Zugriff auf Daten in der Remote-Db2-Datenbank fertig sind, setzen Sie die Verbindung zurück und beenden Sie sie dann. Ersetzen Sie im folgenden Beispiel *database_name* durch den Namen Ihrer entfernten Db2-Datenbank. Ersetzen *nickname* mit einem Namen. Ersetzen *server_name* and *table_name* mit dem Namen des Remoteservers und der Tabelle auf dem Server, auf dem Sie Operationen ausführen möchten. Ersetzen *username* mit den Informationen für Ihren Remoteserver. Ersetzen *sql_command* mit der Operation, die auf dem Remoteserver ausgeführt werden soll.

```
connect to database_name;
create nickname nickname for server_name."username"."table_name";
select sql_command from nickname;
connect reset;
terminate;
```

Beispiel

Im folgenden Beispiel wird eine Passthrough-Sitzung erstellt, um Operationen auf dem Verbundserver zu ermöglichen. `testdb10`

Als Nächstes wird die Tabelle `t1` mit drei Spalten mit unterschiedlichen Datentypen erstellt.

Anschließend erstellt das Beispiel den Index `i1_t1` für drei Spalten in der Tabelle`t1`. Anschließend fügt es zwei Zeilen mit Werten für diese drei Spalten ein und trennt dann die Verbindung.

Zuletzt stellt das Beispiel eine Verbindung mit der entfernten Db2-Datenbank her `testdb2` und erstellt einen Spitznamen für die Tabelle auf `t1` dem Verbundserver. `testdb10` Es erstellt den Spitznamen mit dem Benutzernamen `TESTUSER` für diese Datenquelle. Ein SQL Befehl gibt alle Daten aus der Tabelle aus`t1`. Das Beispiel trennt die Verbindung und beendet die Sitzung.

```
set passthru testdbl0;

create table t1 ( c1 decimal(13,0), c2 char(200), c3 int);

create index i1_t1 on t1(c3);
insert into t1 values(1,'Test',1);
insert into t1 values(2,'Test 2',2);
connect reset;

connect to testdb2;
create nickname remote_t1 for testdbl0."TESTUSER"."T1";
select * from remote_t1;
connect reset;
terminate;
```

Heterogener Verband

Sie können einen heterogenen Verbund zwischen Ihrer RDS for Db2-Datenbank und anderen Datenquellen wie Oracle und Microsoft SQL Server einrichten. Eine vollständige Liste der Datenquellen, die Db2 LUW unterstützt, finden Sie auf der Support-Website unter Data Source Support Matrix of Federled in Db2 LUW V11.5. IBM

RDSfür Db2 unterstützt der heterogene Verbund die folgenden Elemente nicht:

- Systemeigene Wrapper für die anderen Datenquellen
- JDBCWrapper für die anderen Datenquellen
- Verbindung zu Sybase-, Informix- und Teradata-Datenquellen, da für diese Datenquellen die Installation der Clientsoftware für Db2 erforderlich ist RDS

RDSfür Db2 gelten für einen heterogenen Verbund die folgenden Anforderungen:

- RDSfür Db2 unterstützt nur die Wrapper-Methode. ODBC

- Wenn Sie eine explizite Definition eines Wrappers erstellen, müssen Sie die Option auf setzen. `DB2_FENCED 'N'` Eine Liste der gültigen Wrapper-Optionen für finden Sie ODBC unter ODBCOptionen im IBM Db2 -Dokumentation.
- Sie müssen eingehenden und ausgehenden Datenverkehr von Ihrer RDS für DB2 Host-Datenbank zu Ihrer Remote-Host-Datenbank zulassen. Weitere Informationen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

Informationen zum Verbund mit Oracle finden Sie unter How to query Oracle by using Db2 Federation and the driver? ODBC auf der IBM Support-Website.

Weitere Informationen zu Datenquellen, die den Verbund unterstützen, finden Sie auf der Support-Website unter Data Source Support Matrix of Federled in Db2 LUW V11.5. IBM

Themen

- Schritt 1: Erstellen Sie einen Wrapper ODBC
- Schritt 2: Erstellen Sie einen Verbundserver
- Schritt 3: Erstellen Sie eine Benutzerzuordnung
- Schritt 4: Überprüfen Sie die Verbindung

Schritt 1: Erstellen Sie einen Wrapper ODBC

Erstellen Sie einen Wrapper, indem Sie den folgenden Befehl ausführen:

```
db2 "create wrapper odbc options( module '/home/rdsdb/sqllib/federation/odbc/lib/
libodbc.so')"
```

Schritt 2: Erstellen Sie einen Verbundserver

Erstellen Sie einen Verbundserver, indem Sie den folgenden Befehl ausführen. Ersetzen Sie im folgenden Beispiel *`server_name`* durch den Namen des Servers, den Sie für den Verbund verwenden werden. Ersetzen *`wrapper_type`* mit dem entsprechenden Wrapper. Ersetzen *`db_version`* mit der Version Ihrer entfernten Datenbank. Ersetzen *`dns_name`*, *`port`*, und *`service_name`* mit den entsprechenden Werten für die entfernte Datenbank, zu der Sie eine Verbindung herstellen möchten.

```
db2 "create server server_name type wrapper_type version db_version options (HOST
 'dns_name', PORT 'port', SERVICE_NAME 'service_name')“
```

Informationen zu Wrappertypen finden Sie auf der Support-Website unter Data Source Support Matrix of Federation Bundled in Db2 LUW V11.5. IBM

Beispiel

Im folgenden Beispiel wird ein Verbundserver für eine Oracle-Remote-Datenbank erstellt.

```
db2 "create server server1 type oracle_odbc version 12.1 options (HOST
 'test1.amazon.com', PORT '1521', SERVICE_NAME 'pdborcl.amazon.com')“
```

Schritt 3: Erstellen Sie eine Benutzerzuordnung

Erstellen Sie eine Benutzerzuordnung, um Ihren Verbundserver mit Ihrem Datenquellenserver zu verknüpfen, indem Sie den folgenden SQL Befehl ausführen. Ersetzen Sie im folgenden Beispiel *server_name* durch den Namen des Remoteservers, auf dem Sie Operationen ausführen möchten. Dies ist der Server, den Sie in Schritt 2 erstellt haben. Ersetzen *username* and *password* mit Ihren Anmeldeinformationen für diesen Remoteserver.

```
create user mapping for user server server_name options (REMOTE_AUTHID 'username',
 REMOTE_PASSWORD 'password');
```

Weitere Informationen finden Sie unter Benutzerzuordnungen im IBM Db2 -Dokumentation.

Schritt 4: Überprüfen Sie die Verbindung

Vergewissern Sie sich, dass die Einrichtung Ihres Verbunds erfolgreich war, indem Sie die Verbindung überprüfen. Öffnen Sie eine Sitzung, um mithilfe des SET PASSTHRU Befehls systemeigene SQL Befehle an Ihre Remote-Datenquelle zu senden, und erstellen Sie dann eine Tabelle auf dem Remote-Datenserver.

1. Öffnen und schließen Sie eine Sitzung, um SQL sie an eine Datenquelle zu senden. Ersetzen Sie im folgenden Beispiel *server_name* durch den Namen des Servers, den Sie in Schritt 2 für den Verbund erstellt haben.

   ```
   set passthru server_name;
   ```

2. Erstellen Sie eine neue Tabelle. Ersetzen Sie im folgenden Beispiel *column_name*, *data_type*, und *value* durch die entsprechenden Elemente für Ihre Tabelle.

```
create table table_name
 ( column_name data_type(value), column_name data_type(value);
```

Weitere Informationen finden Sie in der CREATETABLEErklärung IBM Db2 -Dokumentation.

3. Erstellen Sie einen Index, fügen Sie Werte für Zeilen in die Tabelle ein und setzen Sie die Verbindung zurück. Beim Zurücksetzen der Verbindung wird die Verbindung unterbrochen, die Back-End-Prozesse werden jedoch beibehalten. Ersetzen Sie im folgenden Beispiel *index_name*, *table_name*, *column_name*, und *columnx_value* mit Ihren Informationen.

```
create index index_name on table_name(column_name);
insert into table_name values(column1_value,column2_value,column3_value);
insert into table_name values(column1_value,column2_value,column3_value);
set passthru reset;

connect reset;
```

4. Stellen Sie eine Connect zu Ihrer Remote-Db2-Datenbank her, erstellen Sie einen Spitznamen für Ihren Remoteserver und führen Sie Operationen aus. Wenn Sie mit dem Zugriff auf Daten in der Remote-Db2-Datenbank fertig sind, setzen Sie die Verbindung zurück und beenden Sie sie dann. Ersetzen Sie im folgenden Beispiel *database_name* durch den Namen Ihrer entfernten Db2-Datenbank. Ersetzen *nickname* mit einem Namen. Ersetzen *server_name* and *table_name* mit dem Namen des Remoteservers und der Tabelle auf dem Server, auf dem Sie Operationen ausführen möchten. Ersetzen *username* mit den Informationen für Ihren Remoteserver. Ersetzen *sql_command* mit der Operation, die auf dem Remoteserver ausgeführt werden soll.

```
connect to database_name;
create nickname nickname for server_name."username"."table_name";
select sql_command from nickname;
connect reset;
terminate;
```

Beispiel

Im folgenden Beispiel wird eine Passthrough-Sitzung erstellt, um Operationen auf dem Verbundserver zu ermöglichen. `testdb10`

Als Nächstes wird die Tabelle `t1` mit drei Spalten mit unterschiedlichen Datentypen erstellt.

Anschließend erstellt das Beispiel den Index `i1_t1` für drei Spalten in der Tabelle`t1`. Anschließend fügt es zwei Zeilen mit Werten für diese drei Spalten ein und trennt dann die Verbindung.

Zuletzt stellt das Beispiel eine Verbindung mit der entfernten Db2-Datenbank her `testdb2` und erstellt einen Spitznamen für die Tabelle auf `t1` dem Verbundserver. `testdb10` Es erstellt den Spitznamen mit dem Benutzernamen `TESTUSER` für diese Datenquelle. Ein SQL Befehl gibt alle Daten aus der Tabelle aus`t1`. Das Beispiel trennt die Verbindung und beendet die Sitzung.

```
set passthru testdbl0;

create table t1 ( c1 decimal(13,0), c2 char(200), c3 int);

create index i1_t1 on t1(c3);
insert into t1 values(1,'Test',1);
insert into t1 values(2,'Test 2',2);
connect reset;

connect to testdb2;
create nickname remote_t1 for testdbl0."TESTUSER"."T1";
select * from remote_t1;
connect reset;
terminate;
```

Arbeiten mit Replikaten für Amazon RDS for Db2

Um die Replikation zwischen RDS für Db2-DB-Instances zu konfigurieren, können Sie Replikatdatenbanken erstellen. Eine Zusammenfassung der Unterschiede zwischen RDS for Db2-Replikaten und anderen DB-Engines finden Sie unter. Unterschiede zwischen Read Replicas für DB-Engines

Eine Db2-Replikatdatenbank ist eine physische Kopie Ihrer Primärdatenbank. Ein Db2-Replikat im Standby-Modus wird als Standby-Replikat bezeichnet. Db2 erlaubt keine Lese- oder Schreibvorgänge in einem Replikat, aber Sie können ein Replikat heraufstufen, um es schreibbar zu machen. Das heraufgestufte Replikat verfügt über die replizierten Daten bis zu dem Punkt, an dem die Anforderung gestellt wurde, es heraufzustufen. RDS for Db2 verwendet Db2 High Availability Disaster Recover (HADR) für die Replikation. Weitere Informationen finden Sie in der Dokumentation unter High Availability Disaster Recovery (HADR). IBM Db2

Standby-Replikate

Wenn Sie ein Db2-Replikat erstellen oder ändern, können Sie es in den Standby-Modus versetzen. Für Standby-Replikate `DB2_HADR_ROS` ist die Umgebungsvariable Db2 auf `OFF` so gesetzt, dass die Replikatdatenbank keine Benutzerverbindungen akzeptiert. Die primäre Verwendung von Standby-Replikaten ist die regionsübergreifende Notfallwiederherstellung.

Ein Standby-Replikat kann keine schreibgeschützte Arbeitslast bewältigen. Das Standby-Replikat hat keine Archivprotokolle.

Sie können bis zu drei Replikate aus einer Quell-DB-Instance erstellen.

Datenbank-Aktivierungen

Db2 HADR ist auf Datenbankebene konfiguriert. Nachdem Sie Replikate erstellt haben, wird HADR für alle Db2-Datenbanken festgelegt, einschließlich `rdsadmin` der Datenbanken, die RDS vollständig verwaltet. Bevor Sie Db2-Replikate erstellen, müssen Sie alle Datenbanken explizit aktivieren. Andernfalls schlägt die Erstellung von Replikaten fehl und Amazon RDS gibt ein Ereignis aus. Sobald eine DB-Instance über ein oder mehrere Replikate verfügt, können Sie mit den oder gespeicherten Prozeduren keine Datenbanken auf der DB-Instance aktivieren `rdsadmin.activate_database` oder `rdsadmin.deactivate_database` deaktivieren. Sie können jedoch die `rdsadmin.reactivate_database` gespeicherte Prozedur aufrufen, um Datenbanken zu reaktivieren. Weitere Informationen finden Sie unter Gespeicherte Prozeduren für Datenbanken für RDS for Db2.

HARD-Konfigurationen

Sie können alle HADR-Konfigurationen für eine Datenbank anzeigen, indem Sie eine Verbindung mit der Datenbank herstellen und dann ausführen. `db2 get db cfg`

Aufbewahrung von Archivprotokollen

Amazon RDS löscht Protokolle aus einer primären DB-Instance, wenn die folgenden Bedingungen erfüllt sind:

- Die Protokolle sind mindestens zwei Stunden alt.
- Die Einstellung für die Aufbewahrungszeit von Archivprotokollen ist abgelaufen.
- Die Archivprotokolle wurden erfolgreich auf alle Replikat-DB-Instances repliziert. Diese Bedingung gilt sowohl für DB-Instances in derselben AWS-Region als auch für regionsübergreifende DB-Instances.

Weitere Informationen zum Festlegen von Aufbewahrungszeiten für Archivprotokolle finden Sie unter rdsadmin.set_archive_log_retention.

Amazon RDS überprüft und bereinigt jede Datenbank einzeln. Wenn eine Datenbank die HADR-Verbindung verliert oder wenn Informationen über die Verbindung nicht verfügbar sind, überspringt Amazon RDS die Datenbank und löscht die Archivprotokolle nicht.

Ausfälle während der Db2-Replikation

Wenn Sie ein Replikat erstellen, erstellt Amazon RDS einen DB-Snapshot Ihrer Quell-DB-Instance und beginnt mit der Replikation. Wenn der DB-Snapshot-Vorgang beginnt, wird die Quell-DB-Instance kurzzeitig unterbrochen I/O . Die I/O Unterbrechung dauert in der Regel etwa eine Sekunde. Wenn es sich bei der Quell-DB-Instance jedoch um eine Multi-AZ-Bereitstellung handelt, wird die Quell-DB-Instance nicht I/O gesperrt. Dies liegt daran, dass bei Multi-AZ-Bereitstellungen der Snapshot von der sekundären DB-Instance erstellt wird.

Der DB-Snapshot wird zum Db2-Replikat. Amazon RDS legt die erforderlichen Parameter und Berechtigungen für die Quelldatenbank und das Replikat ohne Serviceunterbrechung fest. Wenn Sie ein Replikat löschen, tritt ebenfalls kein Ausfall auf.

Anforderungen und Überlegungen für RDS for Db2-Replikate

Machen Sie sich vor dem Erstellen eines Db2-Replikats mit den folgenden Anforderungen und Überlegungen vertraut.

Versions- und Lizenzanforderungen für RDS für Db2-Replikate

Bevor Sie ein RDS für Db2-Replikat erstellen, sollten Sie die folgenden Informationen überprüfen:

- Sowohl Db2 Advanced Edition (AE) als auch Standard Edition (SE) können im Standby-Modus Replikate sowohl für das Bring Your Own License (BYOL) -Modell als auch für das Db2-Lizenz-Durchgangsmodell erstellen. AWS Marketplace
- Sie müssen eine benutzerdefinierte Parametergruppe für die Read Replica angeben. Bei Replikaten, die das BYOL-Modell verwenden, muss diese benutzerdefinierte Parametergruppe Ihr und enthalten. IBM Site ID IBM Customer ID Sie können diese benutzerdefinierte Parametergruppe für das Read Replica angeben, indem Sie die AWS Management Console AWS CLI , oder die RDS-API verwenden. Weitere Informationen finden Sie unter IBM IDs für Bring Your Own License (BYOL) für Db2.
- Bei Standby-Replikaten beträgt die Anzahl der vCPUs unabhängig von der Größe der DB-Instance zwei. In den folgenden Abschnitten werden die Unterschiede zwischen den Lizenztypen RDS für Db2 beschrieben:
  - Für das BYOL-Modell zeigen AWS License Manager Konfigurationen, dass RDS für Db2-DB-Instances zwei v verwendet. CPUs
  - Beim AWS Marketplace Modell mit durchgehender Db2-Lizenz sind in den Rechnungen die Lizenzkosten für zwei V angegeben. CPUs
- Alle Versionen von Db2 11.5 unterstützen Replikat-DB-Instances. Db2-Replikate unterstützen kleinere Versions-Upgrades.

Überlegungen zur Backup und Wiederherstellung von RDS for Db2-Replikaten

Bevor Sie ein RDS für Db2-Replikat erstellen, sollten Sie die folgenden Punkte berücksichtigen:

- Um Snapshots von RDS für Db2-Replikate zu erstellen oder automatische Backups zu aktivieren, stellen Sie sicher, dass Sie den Aufbewahrungszeitraum für Backups manuell festlegen. Automatische Backups sind standardmäßig nicht aktiviert.
- Wenn Sie ein Replikat-Backup wiederherstellen, erfolgt die Wiederherstellung auf die Datenbankzeit, nicht auf die Uhrzeit, zu der das Backup erstellt wurde. Die Datenbank-Zeit bezieht

sich auf die letzte angewendete Transaktionszeit der Daten im Backup. Der Unterschied ist erheblich, da ein Replikat Minuten oder Stunden hinter der Primärdatenbank zurückbleiben kann. Wenn mehrere Datenbanken vorhanden sind, verwendet RDS for Db2 die früheste Datenbankzeit.

Um den Unterschied herauszufinden, führen Sie den AWS CLI describe-db-snapshotsBefehl aus oder rufen Sie den DBSnapshots RDS-API-Vorgang Describe auf. Vergleichen Sie den `SnapshotDatabaseTime` Wert mit dem `OriginalSnapshotCreateTime` Wert. Der `SnapshotDatabaseTime` Wert ist die Datenbankzeit der Replikatsicherung. Der `OriginalSnapshotCreateTime` Wert ist die letzte angewendete Transaktion in der Primärdatenbank.

Überlegungen zur Replikation von RDS für Db2-Replikate

Bevor Sie ein RDS für Db2-Replikat erstellen, sollten Sie die folgenden Punkte berücksichtigen:

- Bei der Replikation wird Db2 HADR für alle Datenbanken auf der RDS for Db2-DB-Instance verwendet.
- Die Replikation unterstützt den Befehl nicht. `LOAD` Wenn Sie den `LOAD` Befehl von der Quell-DB-Instance aus ausführen, erhalten Sie inkonsistente Daten.
- RDS for Db2 repliziert die folgenden Elemente nicht:
  - Speicherzugriff. Achten Sie auf Daten, wie z. B. externe Tabellen, die auf Speicherzugriff angewiesen sind.
  - Nicht inline LOBs.
  - Binärdateien externer gespeicherter Prozeduren (in C oder Java).
- Für Standby-Replikate repliziert RDS for Db2 die folgenden Elemente:
  - Lokale Benutzer, außer Masterbenutzern
  - Konfigurationsparameter für die Datenbank

Verschiedene Überlegungen zu RDS für Db2-Replikate

Bevor Sie ein RDS für Db2-Replikat erstellen, sollten Sie die folgenden Punkte berücksichtigen:

- RDS for Db2 repliziert Datenbankkonfigurationen auf die Replikate. Wenn RDS for Db2 ein Replikat hochstuft, deaktiviert und aktiviert es jede Datenbank.

- RDS for Db2 repliziert die lokalen Benutzer, aber nicht den Masterbenutzer, auf die Replikate. Sie können den Masterbenutzer auf dem Replikat ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Alle Datenbanken müssen sich in einem aktiven Zustand befinden. Hinweise zur Aktivierung von Datenbanken finden Sie unterGespeicherte Prozeduren für Datenbanken für RDS for Db2.
- Alle gespeicherten Prozeduren zum Erstellen, Löschen, Wiederherstellen oder Rollforward von Datenbanken müssen abgeschlossen sein, bevor ein Replikat erstellt werden kann. Hinweise zu diesen gespeicherten Prozeduren finden Sie unterGespeicherte Prozeduren für Datenbanken für RDS for Db2.
- Wenn das Replikat erstellt wird, setzt Amazon RDS den Parameter auf Datenbankebene `blocknonlogged` für alle Datenbanken auf der Quell-DB-Instance auf. `YES` Wenn das Quellreplikat wieder zu einer eigenständigen Instance wird, setzt Amazon RDS den Wert wieder auf`N0`. Weitere Informationen finden Sie in der Dokumentation unter dem Konfigurationsparameter blocknonlogged — Erstellung von Tabellen blockieren, die nicht protokollierte Aktivitäten zulassen. IBM Db2
- Wenn das Replikat erstellt wird, setzt Amazon RDS den Parameter auf Datenbankebene `logindexbuild` für alle Datenbanken auf der Quell-DB-Instance auf. `YES` Wenn das Quellreplikat wieder zu einer eigenständigen Instance wird, setzt Amazon RDS den Wert wieder auf`N0`. Weitere Informationen finden Sie unter dem Konfigurationsparameter logindexbuild — Die erstellten Indexseiten protokollieren in der Dokumentation. IBM Db2

Vorbereitung der Erstellung eines RDS für Db2-Replikats

Bevor Sie Ihr Replikat verwenden können, führen Sie die folgenden Aufgaben aus:

Aufgaben

- Aktivieren automatischer Sicherungen.
- Planen Sie Rechen- und Speicherressourcen
- Bereiten Sie Datenbanken vor

Aktivieren automatischer Sicherungen.

Bevor eine DB-Instance als eine Quell-DB-Instance eingesetzt werden kann, müssen Sie automatische Sicherungen auf der Quell-DB-Instance aktivieren. Weitere Informationen zum Ausführen dieses Verfahrens finden Sie unter Aktivieren von automatisierten Backups.

Planen Sie Rechen- und Speicherressourcen

Stellen Sie sicher, dass die Quell-DB-Instance und ihre Replikate in Bezug auf die Datenverarbeitungs- und Speicherkapazität die für ihre Betriebslast angemessene Größe aufweisen. Wenn ein Replikat die Kapazität von Rechen-, Netzwerk- und Speicherressourcen erreicht hat, stellt das Replikat den Empfang und die Anwendung von Änderungen aus seiner Quelle ein. RDS for Db2 greift nicht ein, um die hohe Replikatverzögerung zwischen einer Quell-DB-Instance und ihren Replikaten zu verringern. Sie können die Speicher- und CPU-Ressourcen eines Replikats unabhängig von seiner Quelle und anderen Replikaten ändern.

Bereiten Sie Datenbanken vor

Stellen Sie vor der Replikaterstellung sicher, dass die DB-Instance alle Datenbanken enthält, die in der DB-Instance vorhanden sein sollen. Nach der Replikaterstellung können Sie keine Datenbank auf der DB-Instance erstellen, löschen oder nativ wiederherstellen. Alle Aufrufe der rdsadmin.restore_database gespeicherten Prozeduren rdsadmin.create_databaserdsadmin.drop_database, oder schlagen fehl.

Stellen Sie außerdem sicher, dass sich alle Datenbanken auf der DB-Instance in einem aktiven Zustand befinden. Andernfalls schlägt die Replikaterstellung fehl. Hinweise zur Aktivierung von Datenbanken finden Sie unterGespeicherte Prozeduren für Datenbanken für RDS for Db2.

Erstellen eines RDS for Db2-Replikats im Standby-Modus

Sie können ein Replikat im Standby-Modus erstellen, indem Sie die AWS Management Console AWS CLI, oder die RDS-API verwenden.

Konsole

Um ein Standby-Replikat aus einer RDS-Quell-DB-Instance für DB2 zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die RDS for Db2-DB-Instance aus, die Sie als Quelle für ein Standby-Replikat verwenden möchten.
4. Wählen Sie unter Actions (Aktionen) Create replica (Replikat erstellen) aus.
5. Wählen Sie für den Replikatmodus Standby aus.

6. Wählen Sie die Einstellungen, die Sie verwenden möchten. Geben Sie unter DB instance identifier (DB-Instance-Kennung) einen Namen für das Lesereplikat ein. Passen Sie weitere Einstellungen nach Ihrem Bedarf an.
7. Wählen Sie für Regionen den AWS-Region Ort aus, an dem das Standby-Replikat gestartet werden soll.
8. Wählen Sie die Instance-Größe und den Speichertyp aus. Wir empfehlen Ihnen, dieselbe DB-Instance-Klasse und denselben Speichertyp wie bei der Quell-DB-Instance für das Lesereplikat zu verwenden.
9. Wählen Sie für eine Multi-AZ-Bereitstellung die Option Standby-Instance erstellen aus, um eine Standby-Instanz Ihres Replikats in einer anderen Availability Zone für die Failover-Unterstützung für das Standby-Replikat zu erstellen.
10. Wählen Sie weitere Einstellungen aus, die Sie verwenden möchten.
11. Wählen Sie Create replica (Replikat erstellen).

Auf der Seite Datenbanken hat das Standby-Replikat die Rolle Replikat.

AWS CLI

Um ein Db2-Replikat im Standby-Modus zu erstellen, legen Sie im Befehl -replica `--replica-mode` den Wert auf `mounted` fest. AWS CLI create-db-instance-read

Example

FürLinux, oder: macOS Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier my_read_replica \
    --source-db-instance-identifier my_db_instance \
    --replica-mode mounted
```

Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier my_read_replica ^
    --source-db-instance-identifier my_db_instance ^
    --replica-mode mounted
```

RDS-API

Um ein Db2-Replikat im Standby-Modus zu erstellen, geben Sie `ReplicaMode=mounted` im RDS-API-Vorgang Create an. DBInstance ReadReplica

Arbeiten mit RDS für DB2-Replikat-Backups

Sie können Backups eines RDS for Db2-Replikats erstellen und wiederherstellen. RDS für Db2 unterstützt sowohl automatische Backups als auch manuelle Snapshots. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren. In den folgenden Abschnitten werden die wichtigsten Unterschiede zwischen der Verwaltung von Backups einer Primärdatenbank und einer RDS-for-Db2-Replik beschrieben.

Themen

- RDS für Db2-Replikat-Backups aktivieren
- Wiederherstellung einer RDS-für-Db2-Replikatsicherung

RDS für Db2-Replikat-Backups aktivieren

Bei einem RDS für Db2-Replikat sind automatische Backups standardmäßig nicht aktiviert. Aktivieren Sie automatische Backups, indem Sie die Aufbewahrungsdauer für Backups auf einen positiven Wert ungleich Null setzen.

Konsole

Um automatische Backups sofort zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance, die Sie ändern möchten.
3. Wählen Sie Modify aus.
4. Wählen Sie für die Aufbewahrungsdauer von Backup einen positiven Wert ungleich Null aus, z. B. drei Tage.
5. Klicken Sie auf Continue.
6. Wählen Sie Apply immediately (Sofort anwenden) aus.

7. Wählen Sie Modify DB-Instance, um Ihre Änderungen zu speichern und automatische Backups zu aktivieren.

AWS CLI

Verwenden Sie den AWS CLI modify-db-instanceBefehl, um automatische Backups zu aktivieren.

Verwenden Sie die folgenden Parameter:

- `--db-instance-identifier`
- `--backup-retention-period`
- `--apply-immediately` oder `--no-apply-immediately`

Im folgenden Beispiel werden automatische Backups aktiviert, indem die Aufbewahrungsfrist für Backups auf drei Tage festgelegt wird. Die Änderungen werden sofort übernommen.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my_db_instance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my_db_instance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

RDS-API

Um automatische Backups zu aktivieren, verwenden Sie den DBInstance Vorgang Modify der RDS-API mit den folgenden erforderlichen Parametern:

- `DBInstanceIdentifier`
- `BackupRetentionPeriod`

Wiederherstellung einer RDS-für-Db2-Replikatsicherung

Sie können eine RDS für Db2-Replikatsicherung auf die gleiche Weise wiederherstellen, wie Sie eine Sicherung der Primärdatenbank wiederherstellen können. Weitere Informationen finden Sie unter den folgenden Themen:

- Wiederherstellung auf einer DB-Instance
- Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS

Die wichtigste Überlegung beim Wiederherstellen eines Replikat-Backups ist die Bestimmung des Zeitpunkts, zu dem Sie die Wiederherstellung durchführen. Die Datenbank-Zeit bezieht sich auf die letzte angewendete Transaktionszeit der Daten im Backup. Wenn Sie ein Replikat-Backup wiederherstellen, erfolgt die Wiederherstellung auf die Datenbankzeit, nicht auf den Zeitpunkt, zu dem das Backup abgeschlossen wurde. Der Unterschied ist erheblich, da ein Replikat der Primärdatenbank um Minuten oder Stunden hinterherhinken kann. Daher liegen die Datenbankzeit einer Replikatsicherung und der Zeitpunkt, zu dem Sie sie wiederherstellen, möglicherweise viel vor der Erstellung der Sicherung.

Um den Unterschied zwischen Datenbankzeit und Erstellungszeit zu ermitteln, führen Sie den AWS CLI describe-db-snapshotsBefehl aus oder rufen Sie den DBSnapshots RDS-API-Vorgang Describe auf. Vergleichen Sie den `SnapshotDatabaseTime` Wert und den `OriginalSnapshotCreateTime` Wert. Der `SnapshotDatabaseTime` Wert ist der früheste Datenbankzeitpunkt unter allen Datenbanken der Replikatsicherung. Der `OriginalSnapshotCreateTime` Wert ist die letzte angewendete Transaktion in der Primärdatenbank. Beachten Sie, dass die Replikationsverzögerungen für mehrere Datenbanken unterschiedlich sein können und dass die Datenbankzeit zwischen diesen beiden Zeiten liegen kann.

Das folgende AWS CLI Beispiel zeigt den Unterschied zwischen den beiden Zeiten:

```
aws rds describe-db-snapshots \
    --db-instance-identifier my_db2_replica
    --db-snapshot-identifier my_replica_snapshot

{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "my_replica_snapshot",
            "DBInstanceIdentifier": "my_db2_replica",
            "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
```

```
            ...
            "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
        }
    ]
}
```

Behebung von Problemen mit der RDS for Db2-Replikation

In diesem Inhalt werden mögliche Probleme und Lösungen bei der Replikation von RDS für Db2 beschrieben.

Themen

- Überwachung der Verzögerung bei der Db2-Replikation
- Fehler bei der DB2-Replikation

Überwachung der Verzögerung bei der Db2-Replikation

Um die Replikationsverzögerung in Amazon zu überwachen CloudWatch, sehen Sie sich die Amazon `ReplicaLag` RDS-Metrik an. Weitere Informationen zur zeitlichen Verzögerung bei der Replikation finden Sie unter Überwachen der Lesereplikation und CloudWatch Amazon-Metriken für Amazon RDS.

Wenn bei einem Standby-Replikat die Verzögerungszeit zu lang ist, fragen Sie die `MON_GET_HADR` Tabelle nach dem Status der Quell-DB-Instance ab. Fragen Sie die Replikat-DB-Instance nicht ab, da Replikat-DB-Instances keine Benutzerverbindungen akzeptieren.

Fehler bei der DB2-Replikation

Die DB2-Replikation kann aus einer Reihe von Gründen einen Fehlerstatus aufweisen. Überwachen Sie Ereignisse und den Status der DB-Instance, um sicherzustellen, dass die DB-Instance repliziert. Um zu bestätigen, dass die DB-Instance repliziert, überprüfen Sie die Diagnoseprotokolle für das Replikat in der Amazon RDS-Konsole. Suchen Sie in den Protokollen nach Fehlern in HADR-Meldungen. Vergleichen Sie die Protokollsequenznummer mit der Primärsequenznummer. Weitere Informationen finden Sie unter Amazon RDS for Db2-Datenbank-Protokolldateien.

Optionen für Amazon RDS für Db2-DB-Instances

Im Folgenden werden die Optionen oder zusätzlichen Funktionen aufgeführt, die für RDS Amazon-Instances verfügbar sind, auf denen die Db2-DB-Engine ausgeführt wird. Damit diese Optionen aktiviert werden, können Sie diese einer benutzerdefinierten Optionsgruppe hinzufügen und anschließend der Optionsgruppe für Ihre DB-Instance zuordnen. Weitere Informationen über das Arbeiten mit Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Amazon RDS unterstützt die folgenden Optionen für Db2:

Option	Options-ID
Db2-Auditprotokollierung	`DB2_AUDIT`

Db2-Auditprotokollierung

Mit der Db2-Audit-Protokollierung zeichnet Amazon RDS Datenbankaktivitäten auf, darunter Benutzer, die sich an der Datenbank anmelden, und Abfragen, die in der Datenbank ausgeführt werden. RDS lädt die abgeschlossenen Audit-Logs mithilfe der von Ihnen angegebenen AWS Identity and Access Management (IAM) -Rolle in Ihren Amazon S3 S3-Bucket hoch.

Themen

- Einrichtung der Db2-Audit-Protokollierung
- Verwaltung der Db2-Auditprotokollierung
- Anzeigen von Audit-Protokollen
- Fehlerbehebung bei der Db2-Audit-Protokollierung

Einrichtung der Db2-Audit-Protokollierung

Um die Audit-Protokollierung für eine Amazon RDS for Db2-Datenbank zu aktivieren, aktivieren Sie die `DB2_AUDIT` Option auf der RDS for Db2-DB-Instance. Konfigurieren Sie anschließend eine Überwachungsrichtlinie, um die Funktion für die jeweilige Datenbank zu aktivieren. Um die Option auf der DB-Instance RDS for Db2 zu aktivieren, konfigurieren Sie die Optionseinstellungen für die `DB2_AUDIT` Option. Dazu geben Sie die Amazon-Ressourcennamen (ARNs) für Ihren Amazon S3-Bucket und die IAM-Rolle mit Zugriffsberechtigungen für Ihren Bucket an.

Gehen Sie wie folgt vor, um die Db2-Auditprotokollierung für eine RDS for Db2-Datenbank einzurichten.

Themen

- Schritt 1: Einen Amazon-S3-Bucket erstellen
- Schritt 2: Erstellen Sie eine IAM-Richtlinie
- Schritt 3: Erstellen Sie eine IAM-Rolle und fügen Sie Ihre IAM-Richtlinie hinzu
- Schritt 4: Konfigurieren Sie eine Optionsgruppe für die Db2-Auditprotokollierung
- Schritt 5: Konfigurieren Sie die Überwachungsrichtlinie
- Schritt 6: Überprüfen Sie die Audit-Konfiguration

Schritt 1: Einen Amazon-S3-Bucket erstellen

Falls Sie dies noch nicht getan haben, erstellen Sie einen Amazon S3 S3-Bucket, in den Amazon RDS die Audit-Protokolldateien Ihrer RDS for Db2-Datenbank hochladen kann. Für den S3-Bucket, den Sie als Ziel für die Überwachungsdateien verwenden, gelten folgende Einschränkungen:

- Er muss sich in derselben Datei AWS-Region wie Ihre RDS for Db2-DB-Instance befinden.
- Er darf nicht öffentlich zugänglich sein.
- Der Bucket-Eigentümer muss auch der Eigentümer der IAM-Rolle sein.

Informationen zum Erstellen eines Amazon S3 S3-Buckets finden Sie unter Erstellen eines Buckets im Amazon S3 S3-Benutzerhandbuch.

Nachdem Sie die Audit-Protokollierung aktiviert haben, sendet Amazon RDS die Protokolle automatisch von Ihrer DB-Instance an die folgenden Speicherorte:

- Protokolle auf DB-Instance-Ebene — *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`
- Protokolle auf Datenbankebene — *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`*db_name*`/`

Notieren Sie sich den Amazon-Ressourcennamen (ARN) für Ihren Bucket. Diese Informationen werden benötigt, um die nachfolgenden Schritte abzuschließen.

Schritt 2: Erstellen Sie eine IAM-Richtlinie

Erstellen Sie eine IAM-Richtlinie mit den erforderlichen Berechtigungen, um Audit-Protokolldateien von Ihrer DB-Instance in Ihren Amazon S3 S3-Bucket zu übertragen. Bei diesem Schritt wird davon ausgegangen, dass Sie über einen S3-Bucket verfügen.

Bevor Sie die Richtlinie erstellen, sollten Sie die folgenden Informationen sammeln:

- Der ARN für deinen Bucket.
- Der ARN für Ihren AWS Key Management Service (AWS KMS) -Schlüssel, wenn Ihr Bucket SSE-KMS Verschlüsselung verwendet.

Erstellen Sie eine IAM-Richtlinie, die die folgenden Berechtigungen umfasst:

```
"s3:ListBucket",
 "s3:GetBucketAcl",
 "s3:GetBucketLocation",
 "s3:PutObject",
 "s3:ListMultipartUploadParts",
 "s3:AbortMultipartUpload",
 "s3:ListAllMyBuckets"
```

> **Note**
>
> Amazon RDS benötigt die `s3:ListAllMyBuckets` Aktion intern, um zu überprüfen, ob dieselbe Person sowohl den S3-Bucket als auch die RDS for Db2-DB-Instance AWS-Konto besitzt.

Wenn Ihr Bucket SSE-KMS Verschlüsselung verwendet, geben Sie auch die folgenden Berechtigungen für Ihre IAM-Rolle und AWS KMS Ihren IAM-Schlüssel an.

Nehmen Sie die folgenden Berechtigungen in die Richtlinie für Ihre IAM-Rolle auf.

```
"kms:GenerateDataKey",
 "kms:Decrypt"
```

Nehmen Sie die folgenden Berechtigungen in die Schlüsselrichtlinie für Ihren AWS KMS Schlüssel auf. *111122223333*Ersetzen Sie es durch Ihre Kontonummer und *AR0A123456789EXAMPLE* durch Ihren IAM-Rollennamen.

```
{
  "Sid": "Allow RDS role to use the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:sts::111122223333:assumed-role/AR0A123456789EXAMPLE/RDS-Db2Audit",
      "arn:aws:iam::111122223333:role/AR0A123456789EXAMPLE"
    ]
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
```

```
  "Resource": "*"
}
```

Sie können eine IAM-Richtlinie erstellen, indem Sie das AWS Management Console oder das AWS Command Line Interface ()AWS CLI verwenden.

Konsole

So erstellen Sie eine IAM-Richtlinie, die Amazon RDS Zugriff auf Ihren Amazon-S3-Bucket gewährt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen und anschließend JSON aus.
4. Filtern Sie unter Aktionen hinzufügen nach S3. Zugriff hinzufügen ListBucketGetBucketAcl, und GetBucketLocation.
5. Wählen Sie unter Ressource hinzufügen die Option Hinzufügen aus. Wählen Sie unter Ressourcentyp die Option Bucket aus und geben Sie den Namen Ihres Buckets ein. Wählen Sie dann Ressource hinzufügen aus.
6. Wählen Sie Neuen Kontoauszug hinzufügen aus.
7. Filtern Sie unter Aktionen hinzufügen nach S3. Zugriff hinzufügen PutObjectListMultipartUploadParts, und AbortMultipartUpload.
8. Wählen Sie unter Ressource hinzufügen die Option Hinzufügen aus. Wählen Sie für Ressourcentyp die Option Objekt und geben Sie ein*your bucket name/**. Wählen Sie dann Ressource hinzufügen aus.
9. Wählen Sie Neuen Kontoauszug hinzufügen aus.
10. Filtern Sie unter Aktionen hinzufügen nach S3. Zugriff hinzufügen ListAllMyBuckets.
11. Wählen Sie unter Ressource hinzufügen die Option Hinzufügen aus. Wählen Sie als Ressourcentyp die Option Alle Ressourcen aus. Wählen Sie dann Ressource hinzufügen aus.
12. Wenn Sie Ihre eigenen KMS-Schlüssel zum Verschlüsseln der Daten verwenden:

    1. Wählen Sie Neue Aussage hinzufügen.
    2. Filtern Sie unter Aktionen hinzufügen nach KMS. Fügen Sie Zugriff hinzu GenerateDataKeyund Entschlüsseln.
    3. Wählen Sie unter Ressource hinzufügen die Option Hinzufügen aus. Wählen Sie als Ressourcentyp die Option Alle Ressourcen aus. Wählen Sie dann Ressource hinzufügen aus.

13. Wählen Sie Weiter aus.
14. Geben Sie unter Richtlinienname einen Namen für diese Richtlinie ein.
15. (Optional) Geben Sie im Feld Description (Beschreibung) eine Beschreibung für diese Richtlinie ein.
16. Wählen Sie Richtlinie erstellen aus.

AWS CLI

So erstellen Sie eine IAM-Richtlinie, die Amazon RDS Zugriff auf Ihren Amazon-S3-Bucket gewährt

1. Führen Sie den Befehl create-policy aus. Ersetzen Sie im folgenden Beispiel *iam_policy_name* und *amzn-s3-demo-bucket* durch einen Namen für Ihre IAM-Richtlinie und den Namen Ihres Amazon S3 S3-Ziel-Buckets.

   Für LinuxmacOS, oderUnix:

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
```

```
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
  }'
```

Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
```

```
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::amzn-s3-demo-bucket/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
}'
```

2. Notieren Sie sich nach der Erstellung der Richtlinie den ARN der Richtlinie. Sie benötigen den ARN fürSchritt 3: Erstellen Sie eine IAM-Rolle und fügen Sie Ihre IAM-Richtlinie hinzu.

Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Schritt 3: Erstellen Sie eine IAM-Rolle und fügen Sie Ihre IAM-Richtlinie hinzu

Bei diesem Schritt wird davon ausgegangen, dass Sie die IAM-Richtlinie in erstellt haben. Schritt 2: Erstellen Sie eine IAM-Richtlinie In diesem Schritt erstellen Sie eine IAM-Rolle für Ihre RDS for Db2-DB-Instance und fügen dann Ihre IAM-Richtlinie der Rolle hinzu.

Sie können eine IAM-Rolle für Ihre DB-Instance mithilfe der Konsole oder der erstellen. AWS CLI

Konsole

Um eine IAM-Rolle zu erstellen und ihr Ihre IAM-Richtlinie anzuhängen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS-Service.
5. Wählen Sie für Service oder Anwendungsfall RDS und dann RDS — Rolle zur Datenbank hinzufügen aus.
6. Wählen Sie Weiter aus.
7. Suchen Sie unter Berechtigungsrichtlinien nach dem Namen der IAM-Richtlinie, die Sie erstellt haben, und wählen Sie ihn aus.
8. Wählen Sie Weiter aus.
9. Geben Sie für Role name (Rollenname) einen Rollennamen ein.
10. (Optional) Geben Sie unter Role description (Rollenbeschreibung) eine Beschreibung für die neue Rolle ein.
11. Wählen Sie Rolle erstellen aus.

AWS CLI

Um eine IAM-Rolle zu erstellen und ihr Ihre IAM-Richtlinie anzuhängen

1. Führen Sie den Befehl create-role aus. Ersetzen Sie es im folgenden Beispiel *iam_role_name* durch einen Namen für Ihre IAM-Rolle.

Für LinuxmacOS, oderUnix:

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

2. Notieren Sie sich nach der Erstellung der Rolle den ARN dieser Rolle. Sie benötigen diesen ARN für den nächsten Schritt,Schritt 4: Konfigurieren Sie eine Optionsgruppe für die Db2-Auditprotokollierung.
3. Führen Sie den Befehl attach-role-policy aus. Ersetzen Sie im folgenden Beispiel durch den *iam_policy_arn* ARN der IAM-Richtlinie, die Sie in Schritt 2: Erstellen Sie eine IAM-Richtlinie

erstellt haben. *iam_role_name*Ersetzen Sie ihn durch den Namen der IAM-Rolle, die Sie gerade erstellt haben.

Für LinuxmacOS, oderUnix:

```
aws iam attach-role-policy \
   --policy-arn iam_policy_arn \
   --role-name iam_role_name
```

Windows:

```
aws iam attach-role-policy ^
   --policy-arn iam_policy_arn ^
   --role-name iam_role_name
```

Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.

Schritt 4: Konfigurieren Sie eine Optionsgruppe für die Db2-Auditprotokollierung

Das Hinzufügen der Db2-Audit-Logging-Option zu einer RDS for Db2-DB-Instance sieht wie folgt aus:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie alle erforderlichen Optionen hinzu und konfigurieren Sie diese.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie die Db2-Audit-Logging-Option hinzugefügt haben, müssen Sie Ihre DB-Instance nicht neu starten. Sobald die Optionsgruppe aktiv ist, können Sie Überwachungen erstellen und Audit-Protokolle in Ihrem S3-Bucket speichern.

Um die Db2-Auditprotokollierung in der Optionsgruppe einer DB-Instance hinzuzufügen und zu konfigurieren

1. Wählen Sie eine der folgenden Optionen aus:

   - Verwenden einer vorhandenen Optionsgruppe.

- Erstellen Sie eine benutzerdefinierte DB-Optionsgruppe und verwenden Sie diese Optionsgruppe. Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.

2. Fügen Sie der Optionsgruppe die Option DB2_AUDIT hinzu und konfigurieren Sie die Optionseinstellungen. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

   - Geben Sie für IAM_ROLE_ARN den ARN der IAM-Rolle ein, in der Sie erstellt haben. the section called “Erstellen Sie eine IAM-Rolle und fügen Sie Ihre IAM-Richtlinie hinzu”
   - Geben Sie für S3_BUCKET_ARN den ARN des S3-Buckets ein, der für Ihre Db2-Audit-Logs verwendet werden soll. Der Bucket muss sich in derselben Region befinden wie Ihre RDS for Db2-DB-Instance. Die Richtlinie, die mit der von Ihnen eingegebenen IAM-Rolle verknüpft ist, muss die erforderlichen Operationen auf dieser Ressource zulassen.

3. Wenden Sie die Optionsgruppe auf eine neue oder vorhandene DB-Instance an. Wählen Sie eine der folgenden Optionen aus:

   - Wenn Sie eine neue DB-Instance erstellen, weisen Sie die Optionsgruppe beim Start der Instance zu.
   - Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Schritt 5: Konfigurieren Sie die Überwachungsrichtlinie

Um die Überwachungsrichtlinie für Ihre RDS for Db2-Datenbank zu konfigurieren, stellen Sie mithilfe des Master-Benutzernamens und des Master-Passworts für Ihre RDS for Db2-DB-Instance eine Verbindung zur `rdsadmin` Datenbank her. Rufen Sie dann die `rdsadmin.configure_db_audit` gespeicherte Prozedur mit dem DB-Namen Ihrer Datenbank und den entsprechenden Parameterwerten auf.

Im folgenden Beispiel wird eine Verbindung mit der Datenbank hergestellt und eine Überwachungsrichtlinie für die Kategorien AUDIT, `testdb` CHECKING, OBJMAINT, SECMAINT, SYSADMIN und VALIDATE konfiguriert. Der Statuswert `BOTH` protokolliert Erfolge und Fehlschläge und ist standardmäßig. `ERROR  TYPE NORMAL` Weitere Hinweise zur Verwendung dieser gespeicherten Prozedur finden Sie unterthe section called “rdsadmin.configure_db_audit”.

```
db2 "connect to rdsadmin user master_user using master_password"
db2 "call rdsadmin.configure_db_audit('testdb', 'ALL', 'BOTH', ?)"
```

Schritt 6: Überprüfen Sie die Audit-Konfiguration

Um sicherzustellen, dass Ihre Audit-Richtlinie korrekt eingerichtet ist, überprüfen Sie den Status Ihrer Audit-Konfiguration.

Um die Konfiguration zu überprüfen, stellen Sie mit dem Master-Benutzernamen und dem Master-Passwort für Ihre RDS for Db2-DB-Instance eine Verbindung zur `rdsadmin` Datenbank her. Führen Sie dann die folgende SQL-Anweisung mit dem DB-Namen Ihrer Datenbank aus. Im folgenden Beispiel lautet der DB-Name*testdb*.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,'testdb','CONFIGURE_DB_AUDIT'))"

Sample Output

TASK_ID              TASK_TYPE            DATABASE_NAME       LIFECYCLE
-------------------- -------------------- -------------- -------------
                   2  CONFIGURE_DB_AUDIT            DB2DB       SUCCESS

... continued ...
TASK_PARAMS
------------------------------------------------------------
{ "AUDIT_CATEGORY" : "ALL", "CATEGORY_SETTING" : "BOTH" }

... continued ...
                                          TASK_OUTPUT
-------------------------------------------------
2023-12-22T20:27:03.029Z Task execution has started.

2023-12-22T20:27:04.285Z Task execution has completed successfully.
```

Verwaltung der Db2-Auditprotokollierung

Nachdem Sie die Db2-Auditprotokollierung eingerichtet haben, können Sie die Überwachungsrichtlinie für eine bestimmte Datenbank ändern oder die Auditprotokollierung auf Datenbankebene oder für die gesamte DB-Instance deaktivieren. Sie können auch den Amazon S3 S3-Bucket ändern, in den Ihre Protokolldateien hochgeladen werden.

Themen

- Änderung einer Db2-Audit-Richtlinie

- Ändern Sie den Speicherort Ihrer Protokolldateien
- Deaktivieren der Db2-Auditprotokollierung

Änderung einer Db2-Audit-Richtlinie

Um die Überwachungsrichtlinie für eine bestimmte RDS for Db2-Datenbank zu ändern, führen Sie die `rdsadmin.configure_db_audit` gespeicherte Prozedur aus. Mit dieser gespeicherten Prozedur können Sie die Kategorien, Kategorieeinstellungen und die Fehlertypkonfiguration der Überwachungsrichtlinie ändern. Weitere Informationen finden Sie unter the section called “rdsadmin.configure_db_audit”.

Ändern Sie den Speicherort Ihrer Protokolldateien

Gehen Sie wie folgt vor, um den Amazon S3 S3-Bucket zu ändern, in den Ihre Protokolldateien hochgeladen werden:

- Ändern Sie die aktuelle Optionsgruppe, die an Ihre RDS for Db2-DB-Instance angehängt ist — Aktualisieren Sie die `S3_BUCKET_ARN` Einstellung für die `DB2_AUDIT` Option, sodass sie auf den neuen Bucket verweist. Stellen Sie außerdem sicher, dass Sie die IAM-Richtlinie aktualisieren, die der IAM-Rolle zugeordnet ist, die in der `IAM_ROLE_ARN` Einstellung in der angehängten Optionsgruppe angegeben ist. Diese IAM-Richtlinie muss Ihrem neuen Bucket die erforderlichen Zugriffsberechtigungen gewähren. Informationen zu den in der IAM-Richtlinie erforderlichen Berechtigungen finden Sie unter. Eine IAM-Richtlinie erstellen
- Hängen Sie Ihre RDS for Db2-DB-Instance an eine andere Optionsgruppe an — Ändern Sie Ihre DB-Instance, um die ihr zugeordnete Optionsgruppe zu ändern. Stellen Sie sicher, dass die neue Optionsgruppe mit den richtigen `S3_BUCKET_ARN` `IAM_ROLE_ARN` AND-Einstellungen konfiguriert ist. Informationen zur Konfiguration dieser Einstellungen für die `DB2_AUDIT` Option finden Sie unterKonfigurieren Sie eine Optionsgruppe.

Wenn Sie die Optionsgruppe ändern, stellen Sie sicher, dass Sie die Änderungen sofort übernehmen. Weitere Informationen finden Sie unter the section called “Ändern einer DB-Instance”.

Deaktivieren der Db2-Auditprotokollierung

Gehen Sie wie folgt vor, um die Db2-Auditprotokollierung zu deaktivieren:

- Deaktivieren Sie die Audit-Protokollierung für die RDS for Db2-DB-Instance — Ändern Sie Ihre DB-Instance und entfernen Sie die Optionsgruppe mit der `DB2_AUDIT` Option daraus. Weitere Informationen finden Sie unter the section called “Ändern einer DB-Instance”.
- Deaktivieren Sie die Audit-Protokollierung für eine bestimmte Datenbank — Beenden Sie die Audit-Protokollierung und entfernen Sie die Audit-Richtlinie, indem Sie `rdsadmin.disable_db_audit` mit dem DB-Namen Ihrer Datenbank aufrufen. Weitere Informationen finden Sie unter the section called “rdsadmin.disable_db_audit”.

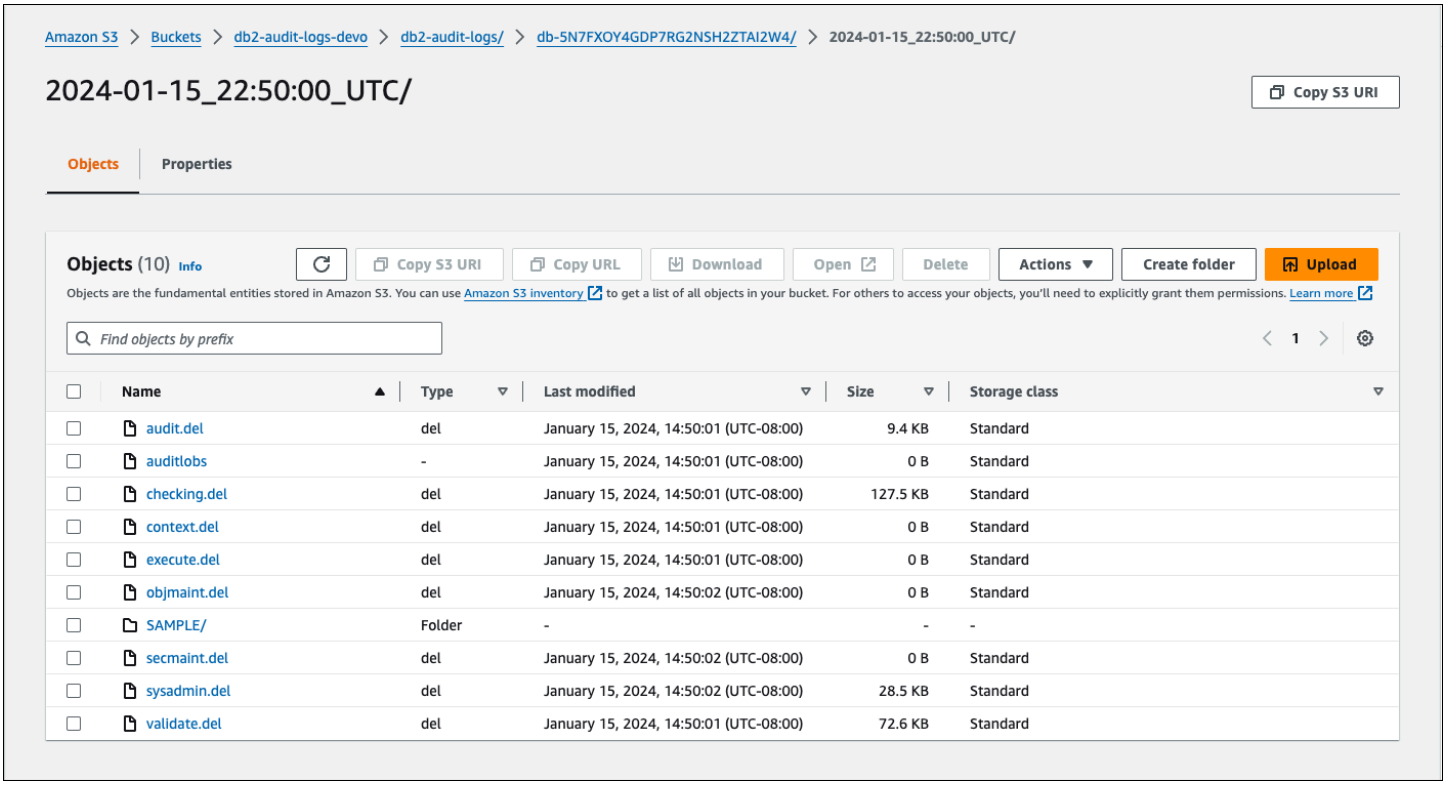
```
db2 "call rdsadmin.disable_db_audit(
    'db_name',
    ?)"
```

Anzeigen von Audit-Protokollen

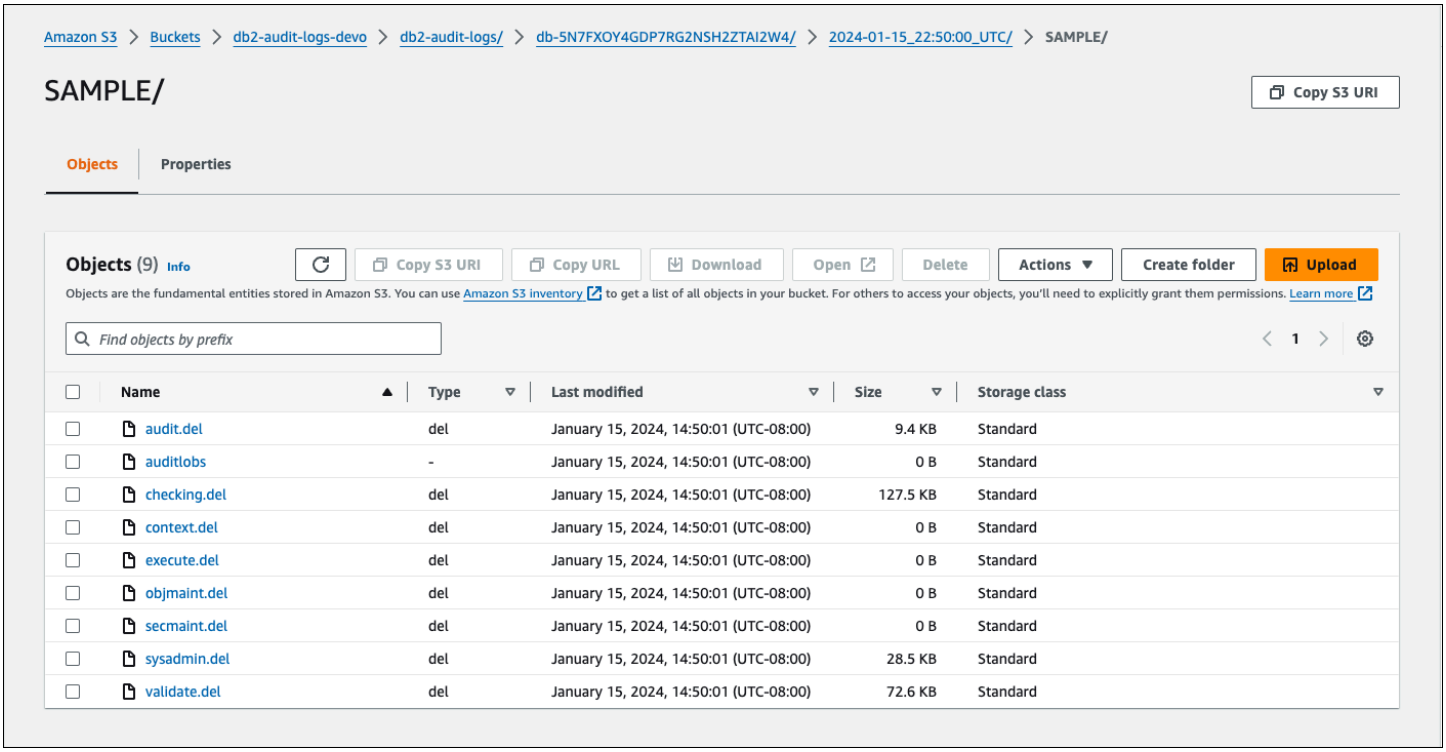
Nachdem Sie die Db2-Audit-Protokollierung aktiviert haben, warten Sie mindestens eine Stunde, bevor Sie sich die Audit-Daten in Ihrem Amazon S3 S3-Bucket ansehen. Amazon RDS sendet die Protokolle automatisch von Ihrer RDS for Db2-DB-Instance an die folgenden Speicherorte:

- Protokolle auf DB-Instance-Ebene — *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`
- Protokolle auf Datenbankebene — *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`*db_name*`/`

Der folgende Beispiel-Screenshot der Amazon S3 S3-Konsole zeigt eine Liste von Ordnern für RDS für Protokolldateien auf DB2-DB-Instance-Ebene.



Der folgende Beispiel-Screenshot der Amazon S3 S3-Konsole zeigt Protokolldateien auf Datenbankebene für die RDS for Db2-DB-Instance.



Fehlerbehebung bei der Db2-Audit-Protokollierung

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme mit der Db2-Auditprotokollierung zu beheben.

Die Überwachungsrichtlinie kann nicht konfiguriert werden

Wenn beim Aufrufen der gespeicherten Prozedur ein Fehler `rdsadmin.configure_db_audit` zurückgegeben wird, kann es sein, dass die Optionsgruppe mit der `DB2_AUDIT` Option nicht mit der DB-Instance RDS for Db2 verknüpft ist. Ändern Sie die DB-Instance, um die Optionsgruppe hinzuzufügen, und versuchen Sie dann erneut, die gespeicherte Prozedur aufzurufen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Keine Daten im Amazon S3 S3-Bucket

Wenn Protokolldaten im Amazon S3 S3-Bucket fehlen, überprüfen Sie Folgendes:

- Der Amazon S3 S3-Bucket befindet sich in derselben Region wie Ihre RDS for Db2-DB-Instance.
- Die Rolle, die Sie in der `IAM_ROLE_ARN` Optionseinstellung angegeben haben, ist mit den erforderlichen Berechtigungen zum Hochladen von Protokollen in Ihren Amazon S3 S3-Bucket konfiguriert. Weitere Informationen finden Sie unter Eine IAM-Richtlinie erstellen.
- Die Einstellungen ARNs für die `S3_BUCKET_ARN` Optionen `IAM_ROLE_ARN` und sind in der Optionsgruppe, die Ihrer RDS for Db2-DB-Instance zugeordnet ist, korrekt. Weitere Informationen finden Sie unter Konfigurieren Sie eine Optionsgruppe.

Sie können den Aufgabenstatus Ihrer Audit-Logging-Konfiguration überprüfen, indem Sie eine Verbindung zur Datenbank herstellen und eine SQL-Anweisung ausführen. Weitere Informationen finden Sie unter Überprüfen Sie die Audit-Konfiguration.

Sie können auch Ereignisse überprüfen, um mehr darüber zu erfahren, warum Protokolle möglicherweise fehlen. Informationen zum Anzeigen von Ereignissen finden Sie unterthe section called “Protokolle, Ereignisse und Streams in der RDS Amazon-Konsole anzeigen”.

Externe gespeicherte Prozeduren für Amazon RDS for Db2

Sie können externe Routinen erstellen und sie bei Ihren Amazon RDS for Db2-Datenbanken als externe gespeicherte Prozeduren registrieren. Derzeit unterstützt RDS for Db2 nur Java-basierte Routinen für externe gespeicherte Prozeduren.

Java-basierte externe gespeicherte Prozeduren

Java-basierte externe gespeicherte Prozeduren sind externe Java-Routinen, die Sie bei Ihrer RDS for Db2-Datenbank als externe gespeicherte Prozeduren registrieren.

Themen

- Einschränkungen für Java-basierte externe gespeicherte Prozeduren
- Konfiguration von Java-basierten externen gespeicherten Prozeduren

Einschränkungen für Java-basierte externe gespeicherte Prozeduren

Bevor Sie Ihre externe Routine entwickeln, sollten Sie die folgenden Einschränkungen und Einschränkungen berücksichtigen.

Stellen Sie sicher, dass Sie das von Db2 bereitgestellte Java Development Kit (JDK) verwenden, um Ihre externe Routine zu erstellen. Weitere Informationen finden Sie unter Java-Softwareunterstützung für Db2-Datenbankprodukte.

Ihr Java-Programm kann Dateien nur in dem `/tmp` Verzeichnis erstellen, und Amazon RDS unterstützt nicht die Aktivierung von ausführbaren Dateien oder Berechtigungen zum Festlegen von Benutzer-IDs (SUID) für diese Dateien. Ihr Java-Programm kann auch keine Socket-Systemaufrufen oder die folgenden Systemaufrufen verwenden:

- _sysctl
- acct
- afs_syscall
- bpf
- capset
- chown
- chroot

- create_module
- delete_module
- fanotify_init
- fanotify_mark
- finit_module
- fsconfig
- fsopen
- fspick
- get_kernel_syms
- getpmsg
- init_module
- mount
- move_mount
- nfsservctl
- open_by_handle_at
- open_tree
- pivot_root
- putpmsg
- query_module
- quotactl
- reboot
- security
- setdomainname
- setfsuid
- sethostname
- sysfs
- tuxcall
- umount2
- uselib
- ustat

- vhangup
- vserver

Weitere Einschränkungen für externe Routinen für Db2 finden Sie in der Dokumentation unter Einschränkungen für externe Routinen. IBM Db2

Konfiguration von Java-basierten externen gespeicherten Prozeduren

Um eine externe gespeicherte Prozedur zu konfigurieren, erstellen Sie eine JAR-Datei mit Ihrer externen Routine, installieren Sie sie in Ihrer RDS for Db2-Datenbank und registrieren Sie sie dann als externe gespeicherte Prozedur.

Themen

- Schritt 1: Aktivieren Sie externe gespeicherte Prozeduren
- Schritt 2: Installieren Sie die JAR-Datei mit Ihrer externen Routine
- Schritt 3: Registrieren Sie die externe gespeicherte Prozedur
- Schritt 4: Überprüfen Sie die externe gespeicherte Prozedur

Schritt 1: Aktivieren Sie externe gespeicherte Prozeduren

Um externe gespeicherte Prozeduren zu aktivieren, setzen Sie den Parameter in einer benutzerdefinierten Parametergruppe, die Ihrer DB-Instance zugeordnet ist, `db2_alternate_authz_behaviour` auf einen der folgenden Werte:

- `EXTERNAL_ROUTINE_DBADM`— Erteilt implizit jedem Benutzer, jeder Gruppe oder Rolle mit entsprechenden Rechten `DBADM` die `CREATE_EXTERNAL_ROUTINE` entsprechende Berechtigung.
- `EXTERNAL_ROUTINE_DBAUTH`— Erlaubt einem Benutzer mit `DBADM` Befugnis, jedem Benutzer, jeder Gruppe oder Rolle die `CREATE_EXTERNAL_ROUTINE` Berechtigung zu erteilen. In diesem Fall wird keinem Benutzer, keiner Gruppe oder Rolle diese Berechtigung implizit erteilt, nicht einmal einem Benutzer mit entsprechenden Rechten`DBADM`.

Weitere Informationen zu dieser Einstellung finden Sie in der IBM Db2 Dokumentation unter GRANT-Anweisung (Datenbankautoritäten).

Sie können eine benutzerdefinierte Parametergruppe mithilfe der AWS Management Console, der oder der Amazon RDS-API erstellen und ändern. AWS CLI

Konsole

Um den Parameter db2_alternate_authz_behavior in einer benutzerdefinierten Parametergruppe zu konfigurieren

1. Wenn Sie eine andere benutzerdefinierte DB-Parametergruppe als die verwenden möchten, die Ihre DB-Instance verwendet, erstellen Sie eine neue DB-Parametergruppe. Wenn Sie das BYOL-Modell (Bring Your Own License) verwenden, stellen Sie sicher, dass die neue benutzerdefinierte Parametergruppe die IBM IDs enthält. Informationen zu diesen finden Sie IDs unterthe section called “IBM IDs für Bring Your Own License (BYOL) für Db2”. Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.
2. Legen Sie den Wert für den `db2_alternate_authz_behaviour` Parameter in Ihrer benutzerdefinierten Parametergruppe fest. Weitere Informationen zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

AWS CLI

So konfigurieren Sie den Parameter db2_alternate_authz_behavior in einer benutzerdefinierten Parametergruppe

1. Wenn Sie eine andere benutzerdefinierte DB-Parametergruppe als die verwenden möchten, die Ihre DB-Instance verwendet, erstellen Sie eine benutzerdefinierte Parametergruppe, indem Sie den Befehl ausführen. create-db-parameter-group Wenn Sie das BYOL-Modell (Bring Your Own License) verwenden, stellen Sie sicher, dass die neue benutzerdefinierte Parametergruppe die IBM IDs enthält. Informationen zu diesen finden Sie IDs unterthe section called “IBM IDs für Bring Your Own License (BYOL) für Db2”.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - `--db-parameter-group-name`— Ein Name für die Parametergruppe, die Sie erstellen.
   - `--db-parameter-group-family`— Die Db2-Engine-Edition und die Hauptversion. Gültige Werte sind `db2-se-11.5` und `db2-ae-11.5`.
   - `--description`— Eine Beschreibung für diese Parametergruppe.

Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.

Das folgende Beispiel zeigt Ihnen, wie Sie eine benutzerdefinierte Parametergruppe erstellen, die MY_EXT_SP_PARAM_GROUP nach der Parametergruppenfamilie benannt istdb2-se-11.5.

Für LinuxmacOS, oderUnix:

```
aws rds create-db-parameter-group \
--region us-east-1 \
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
--db-parameter-group-family db2-se-11.5 \
--description "test db2 external routines"
```

Windows:

```
aws rds create-db-parameter-group ^
--region us-east-1 ^
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
--db-parameter-group-family db2-se-11.5 ^
--description "test db2 external routines"
```

2. Ändern Sie den db2_alternate_authz_behaviour Parameter in Ihrer benutzerdefinierten Parametergruppe, indem Sie den modify-db-parameter-groupBefehl ausführen.

   Verwenden Sie den folgenden erforderlichen Parameter:

   - --db-parameter-group-name— Der Name der Parametergruppe, die Sie erstellt haben.
   - --parameters— Eine Reihe von Parameternamen, Werten und Anwendungsmethoden für die Parameteraktualisierung.

   Weitere Hinweise zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

   Das folgende Beispiel zeigt, wie Sie die Parametergruppe ändern, MY_EXT_SP_PARAM_GROUP indem Sie den Wert db2_alternate_authz_behaviour auf festlegenEXTERNAL_ROUTINE_DBADM.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

RDS-API

Um den Parameter db2_alternate_authz_behavior in einer benutzerdefinierten Parametergruppe zu konfigurieren

1. Wenn Sie eine andere benutzerdefinierte DB-Parametergruppe als die verwenden möchten, die Ihre DB-Instance verwendet, erstellen Sie mithilfe der Amazon CreateDBParameterGroupRDS-API-Operation eine neue DB-Parametergruppe. Wenn Sie das Modell „Bring Your Own License“ (BYOL) verwenden, stellen Sie sicher, dass die neue benutzerdefinierte Parametergruppe die IBM Db2 IDs enthält. Informationen zu diesen finden Sie IDs unterthe section called “IBM IDs für Bring Your Own License (BYOL) für Db2”.

   Nutzen Sie die folgenden erforderlichen Parameter:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.
2. Ändern Sie den `db2_alternate_authz_behaviour` Parameter in Ihrer benutzerdefinierten Parametergruppe, die Sie mithilfe des ModifyDBParameterGroupRDS-API-Vorgangs erstellt haben.

Nutzen Sie die folgenden erforderlichen Parameter:

- `DBParameterGroupName`
- `Parameters`

Weitere Informationen zum Ändern einer Parametergruppe finden Sie unterÄndern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Schritt 2: Installieren Sie die JAR-Datei mit Ihrer externen Routine

Nachdem Sie Ihre Java-Routine erstellt haben, erstellen Sie die JAR-Datei und führen Sie sie anschließend aus, `db2 "call sqlj.install_jar('file:file_path',jar_ID)"` um sie in Ihrer RDS for Db2-Datenbank zu installieren.

Das folgende Beispiel zeigt Ihnen, wie Sie eine Java-Routine erstellen und sie in einer RDS for Db2-Datenbank installieren. Das Beispiel enthält Beispielcode für eine einfache Routine, mit der Sie den Prozess testen können. In diesem Beispiel werden die folgenden Annahmen getroffen:

- Der Java-Code wird auf einem Server kompiliert, auf dem Db2 installiert ist. Dies ist eine bewährte Methode, da die Nichtkompilierung mit dem von IBM bereitgestellten JDK zu unerklärlichen Fehlern führen kann.
- Auf dem Server ist die Datenbank RDS for Db2 lokal katalogisiert.

Wenn Sie den Vorgang mit dem folgenden Beispielcode ausprobieren möchten, kopieren Sie ihn und speichern Sie ihn dann in einer Datei mit dem Namen. `MYJAVASP.java`

```
import java.sql.*;
public class MYJAVASP
{
public static void my_JAVASP (String inparam) throws SQLException, Exception
{
try
{
// Obtain the calling context's connection details.
Connection myConn = DriverManager.getConnection("jdbc:default:connection");
String myQuery = "INSERT INTO TEST.TEST_TABLE VALUES (?, CURRENT DATE)";
PreparedStatement myStmt = myConn.prepareStatement(myQuery);
myStmt.setString(1, inparam);
```

```
myStmt.executeUpdate();
}
catch (SQLException sql_ex)
{
throw sql_ex;
}
catch (Exception ex)
{
throw ex;
}
}
```

Der folgende Befehl kompiliert die Java-Routine.

```
~/sqllib/java/jdk64/bin/javac MYJAVASP.java
```

Der folgende Befehl erstellt die JAR-Datei.

```
~/sqllib/java/jdk64/bin/jar cvf MYJAVASP.jar MYJAVASP.class
```

Mit den folgenden Befehlen wird eine Verbindung zur genannten Datenbank hergestellt `MY_DB2_DATABASE` und die JAR-Datei installiert.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

db2 "call sqlj.install_jar('file:/tmp/MYJAVASP.jar','MYJAVASP')"
db2 "call sqlj.refresh_classes()"
```

Schritt 3: Registrieren Sie die externe gespeicherte Prozedur

Nachdem Sie die JAR-Datei in Ihrer RDS for Db2-Datenbank installiert haben, registrieren Sie sie als gespeicherte Prozedur, indem Sie den Befehl `db2 CREATE PROCEDURE` or `db2 REPLACE PROCEDURE` ausführen.

Das folgende Beispiel zeigt Ihnen, wie Sie eine Verbindung zur Datenbank herstellen und die im vorherigen Schritt erstellte Java-Routine als gespeicherte Prozedur registrieren.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

create procedure TESTSP.MYJAVASP (in input char(6))
```

```
specific myjavasp
dynamic result sets 0
deterministic
language java
parameter style java
no dbinfo
fenced
threadsafe
modifies sql data
program type sub
external name 'MYJAVASP!my_JAVASP';
```

Schritt 4: Überprüfen Sie die externe gespeicherte Prozedur

Verwenden Sie die folgenden Schritte, um das Beispiel einer externen gespeicherten Prozedur zu testen, die im vorherigen Schritt registriert wurde.

Um die externe gespeicherte Prozedur zu validieren

1. Erstellen Sie eine Tabelle wie `TEST.TEST_TABLE` im folgenden Beispiel.

   ```
   db2 "create table TEST.TEST_TABLE(C1 char(6), C2 date)"
   ```

2. Rufen Sie die neue externe gespeicherte Prozedur auf. Der Aufruf gibt den Status zurück`0`.

   ```
   db2 "call TESTSP.MYJAVASP('test')"
   Return Status = 0
   ```

3. Fragen Sie die Tabelle ab, die Sie in Schritt 1 erstellt haben, um die Ergebnisse des Aufrufs der gespeicherten Prozedur zu überprüfen.

   ```
   db2 "SELECT * from TEST.TEST_TABLE"
   ```

   Die Abfrage erzeugt eine Ausgabe, die dem folgenden Beispiel ähnelt:

   ```
   C1     C2
   ------ ----------
   test   02/05/2024
   ```

Bekannte Probleme und Einschränkungen für Amazon RDS for Db2

Die folgenden Punkte sind bekannte Probleme und Einschränkungen bei der Arbeit mit Amazon RDS for Db2:

Themen

- Beschränkung der Authentifizierung
- Routinen ohne Umzäunung
- Nichtautomatische Speicher-Tablespaces während der Migration
- Einstellung des Parameters db2_compatibility_vector
- Migrieren von Datenbanken, die UNGÜLTIGE Pakete enthalten

Beschränkung der Authentifizierung

Amazon RDS ist `JCC_ENFORCE_SECMEC` standardmäßig `db2auth` auf eingestellt. Wenn Sie jedoch nicht möchten, dass Benutzer-ID und Passwort drahtgebunden verschlüsselt werden, können Sie diese Einstellung überschreiben, indem Sie den Parameter `CLEAR_TEXT` in der `db2auth` Parametergruppe auf ändern. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Routinen ohne Umzäunung

RDS for Db2 unterstützt nicht die Erstellung von Routinen ohne Fencing und die Migration dieser Routinen durch Sicherung und Wiederherstellung von Daten. Führen Sie den folgenden SQL-Befehl aus, um zu überprüfen, ob Ihre Datenbank Routinen ohne Umzäunung enthält:

```
SELECT 'COUNT:' || count(*) FROM SYSCAT.ROUTINES where fenced='N' and routineschema not
 in ('SQLJ','SYSCAT','SYSFUN','SYSIBM','SYSIBMADM','SYSPROC','SYSTOOLS')
```

Nichtautomatische Speicher-Tablespaces während der Migration

RDS for Db2 unterstützt die Erstellung neuer nichtautomatischer Speicher-Tablespaces nicht. Wenn Sie die native Wiederherstellung für eine einmalige Migration Ihrer Datenbank verwenden, konvertiert RDS for Db2 Ihre nicht automatischen Speicher-Tablespaces automatisch in automatische

Tablespaces und stellt dann Ihre Datenbank in RDS for Db2 wieder her. Hinweise zu einmaligen Migrationen finden Sie unter und. Migration von Linux to Linux für Amazon RDS für Db2 Migration von AIX or Windows to Linux für Amazon RDS für Db2

Einstellung des Parameters db2_compatibility_vector

Mit Amazon RDS können Sie beim Erstellen der DB-Instance eine erste Datenbank erstellen und dann Parameter in einer zugehörigen Parametergruppe ändern. Wenn Sie jedoch für Db2 den `db2_compatibility_vector` Parameter in einer Parametergruppe festlegen möchten, müssen Sie zuerst den Parameter in einer benutzerdefinierten Parametergruppe ändern, die DB-Instance ohne Datenbank erstellen und dann mithilfe der `rdsadmin.create_database` gespeicherten Prozedur eine Datenbank erstellen.

Um den Parameter festzulegen **`db2_compatibility_vector`**

1. Erstellen Sie eine benutzerdefinierte Parametergruppe. (Sie können Parameter in Standardparametergruppen nicht ändern.)
2. Ändern Sie den Parameter.
3. Erstellen Sie eine DB-Instance.
4. Erstellen Sie eine Datenbank mithilfe der `rdsadmin.create_database` gespeicherten Prozedur.
5. Ordnen Sie die Parametergruppe der DB-Instance zu, die die Datenbank enthält.

Migrieren von Datenbanken, die UNGÜLTIGE Pakete enthalten

Wenn Sie Db2-Datenbanken, die `INVALID` Pakete enthalten, mithilfe des `RESTORE` Befehls zu RDS für Db2 migrieren, können Probleme auftreten, wenn Sie die Datenbanken verwenden. `INVALID`Pakete können aufgrund der Einrichtung der Autorisierung für den DB-Instance-Benutzer `rdsdb` und der Entfernung der Autorisierung von Probleme verursachen. `PUBLIC` `INVALID`Pakete führen dazu, dass die folgenden Befehle fehlschlagen:

- `db2updv115`
- `db2 "call SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS()"`

Bevor Sie Ihre Datenbank mit dem `RESTORE` Befehl migrieren, stellen Sie sicher, dass Ihre Datenbank keine `INVALID` Pakete enthält, indem Sie den folgenden Befehl ausführen:

```
db2 "SELECT 'COUNT:' || count(*) FROM SYSCAT.INVALIDOBJECTS"
```

Wenn der Befehl eine Anzahl von mehr als Null zurückgibt, rufen Sie den folgenden Befehl auf:

```
db2 "call SYSPROC.ADMIN_REVALIDATE_DB_OBJECTS()"
```

Rufen Sie anschließend den vorherigen Befehl auf, um zu bestätigen, dass Ihre Datenbank keine `INVALID` Pakete mehr enthält.

```
db2 "SELECT 'COUNT:' || count(*) FROM SYSCAT.INVALIDOBJECTS"
```

Jetzt sind Sie bereit, eine Sicherungskopie Ihrer Datenbank zu erstellen und sie auf Ihrer RDS for Db2-DB-Instance wiederherzustellen.

Referenz für gespeicherte Prozeduren in Amazon RDS für Db2

Sie können Ihre Amazon RDS for Db2-DB-Instances verwalten, auf denen die Db2-Engine ausgeführt wird, indem Sie integrierte gespeicherte Prozeduren aufrufen.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.activate_database”	Datenbanken	Verwenden Sie die `rdsadmin.activate_database` gespeicherte Prozedur, um eine Datenbank auf einer eigenständigen RDS for Db2-DB-Instance zu aktivieren.
the section called “rdsadmin.add_groups”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.add_groups` gespeicherte Prozedur, um einem Benutzer für eine Datenbank auf einer RDS for Db2-DB-Instance eine oder mehrere Gruppen hinzuzufügen.
the section called “rdsadmin.add_user”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.add_user` gespeicherte Prozedur, um einen Benutzer zu einer Autorisierungsliste für eine Datenbank auf einer RDS for Db2-DB-Instance hinzuzufügen.
the section called “rdsadmin.alter_bufferpool”	Pufferpools	Verwenden Sie die `rdsadmin.alter_bufferpool` gespeicherte Prozedur, um einen Pufferpool für eine Datenbank auf einer RDS for Db2-DB-Instance zu ändern.
the section called “rdsadmin.alter_tablespace”	Tablespaces	Verwenden Sie die `rdsadmin.alter_tablespace` gespeicherte Prozedur, um einen Tablespace für eine Datenbank auf einer RDS for Db2-DB-Instance zu ändern.
the section called “rdsadmin.catalog_storage_access”	Speicherzugriff	Verwenden Sie die `rdsadmin.catalog_storage_access` gespeicherte Prozedur, um einen Speicheralias für den Zugriff auf einen Amazon S3 S3-Bucket mit Db2-Daten

Gespeicherte Prozedur	Kategorie	Beschreibung
		dateien für eine Datenbank auf einer RDS for Db2-DB-Instance zu katalogisieren.
the section called “rdsadmin.change_password”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.change_password` gespeicherte Prozedur, um das Passwort eines Benutzers für eine Datenbank auf einer RDS for Db2-DB-Instance zu ändern.
the section called “rdsadmin.complete_rollforward”	Datenbanken	Verwenden Sie die `rdsadmin.complete_rollforward` gespeicherte Prozedur, um eine Datenbank auf einer RDS for Db2-DB-Instance von einem `ROLL-FORWARD PENDING` Status aus online zu schalten. Ein `ROLL-FORWARD PENDING` Status tritt ein, wenn Sie aufgerufen haben, den `complete_rollforward` Parameter the section called “rdsadmin.rollforward_database” aber auf `FALSE` gesetzt haben.
the section called “rdsadmin.configure_db_audit”	Richtlinien für die Prüfung	Verwenden Sie die `rdsadmin.configure_db_audit` gespeicherte Prozedur, um eine Überwachungsrichtlinie für eine Datenbank auf einer RDS for Db2-DB-Instance zu ändern. Wenn keine Überwachungsrichtlinie vorhanden ist, wird durch die Ausführung dieser gespeicherten Prozedur eine Überwachungsrichtlinie erstellt.
the section called “rdsadmin.create_bufferpool”	Pufferpools	Verwenden Sie die `rdsadmin.create_bufferpool` gespeicherte Prozedur, um einen Pufferpool für eine Datenbank auf einer RDS for Db2-DB-Instance zu erstellen.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.create_database”	Datenbanken	Verwenden Sie die `rdsadmin.create_database` gespeicherte Prozedur, um eine Datenbank auf einer RDS for Db2-DB-Instance zu erstellen.
the section called “rdsadmin.create_role”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.create_role` gespeicherte Prozedur, um eine Rolle zu erstellen, die an eine Datenbank auf einer RDS for Db2-DB-Instance angehängt wird.
the section called “rdsadmin.create_tablespace”	Tablespaces	Verwenden Sie die `rdsadmin.create_tablespace` gespeicherte Prozedur, um einen Tablespace für eine Datenbank auf einer RDS for Db2-DB-Instance zu erstellen.
the section called “rdsadmin.db2pd_command”	Datenbanken	Verwenden Sie die `rdsadmin.db2pd_command` gespeicherte Prozedur, um Informationen über eine Datenbank auf einer RDS for Db2-DB-Instance zu sammeln. Diese Informationen können bei der Überwachung und Problembehandlung von Datenbanken in RDS for Db2 helfen.
the section called “rdsadmin.dbadm_grant”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.dbadm_grant` gespeicherte Prozedur, um einer oder mehreren Rollen `DBADMACCESSCTRL` , Benutzern oder Gruppen für eine Datenbank auf einer RDS for Db2-DB-Instance einen oder mehrere Autorisierungstypen (,, oder`DATAACCESS`) zu erteilen.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.dbadm_revoke”	Berechtigungen gewähren und entziehen	Verwenden Sie die `rdsadmin.dbadm_revoke` gespeicherte Prozedur, um einer oder mehreren Rollen `DBADMACCESSCTRL` , Benutzern oder Gruppen einen oder mehrere Autorisierungstypen (, oder`DATAACCESS`) für eine Datenbank auf einer RDS for Db2-DB-Instance zu entziehen.
the section called “rdsadmin.deactivate_database”	Datenbanken	Verwenden Sie die `rdsadmin.deactivate_database` gespeicherte Prozedur, um eine Datenbank auf einer RDS for Db2-DB-Instance zu deaktivieren. Sie können Datenbanken deaktivieren, um Speicherressourcen zu schonen.
the section called “rdsadmin.disable_db_audit”	Richtlinien für die Prüfung	Verwenden Sie die `rdsadmin.disable_db_audit` gespeicherte Prozedur, um die Auditprotokollierung zu beenden und eine Überwachungsrichtlinie aus einer Datenbank auf einer RDS for Db2-DB-Instance zu entfernen.
the section called “rdsadmin.drop_bufferpool”	Pufferpools	Verwenden Sie die `rdsadmin.drop_bufferpool` gespeicherte Prozedur, um einen Pufferpool aus einer Datenbank auf einer RDS for Db2-DB-Instance zu löschen.
the section called “rdsadmin.drop_database”	Datenbanken	Verwenden Sie die `rdsadmin.drop_database` gespeicherte Prozedur, um eine Datenbank aus einer RDS for Db2-DB-Instance zu löschen.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.drop_role”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.drop_role` gespeicherte Prozedur, um eine Rolle aus einer Datenbank auf einer RDS for Db2-DB-Instance zu löschen.
the section called “rdsadmin.drop_tablespace”	Tablespaces	Verwenden Sie die `rdsadmin.drop_tablespace` gespeicherte Prozedur, um einen Tablespace aus einer Datenbank auf einer RDS for Db2-DB-Instance zu löschen.
the section called “rdsadmin.force_application”	Datenbanken	Verwenden Sie die `rdsadmin.force_application` gespeicherte Prozedur, um Anwendungen aus einer Datenbank auf einer RDS for Db2-DB-Instance zu entfernen, um Wartungsarbeiten durchzuführen.
the section called “rdsadmin.grant_role”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.grant_role` gespeicherte Prozedur, um einer Rolle, einem Benutzer oder einer Gruppe des Empfängers in einer Datenbank auf einer RDS for Db2-DB-Instance eine Rolle zuzuweisen. Sie können diese gespeicherte Prozedur auch verwenden, um dem Empfänger die Autorisierung zur Zuweisung von Rollen zu `DBADM` erteilen.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.list_archive_log_information”	Datenbanken	Verwenden Sie die `rdsadmin.list_archive_log_information` gespeicherte Prozedur, um Informationen über Archivprotokolle für eine Datenbank auf einer RDS for Db2-DB-Instance zurückzugeben. Zu diesen Informationen gehören Details wie Größe und Erstellungsdatum einzelner Protokolldateien sowie der Gesamtspeicher, der von den Archivprotokolldateien belegt wird.
the section called “rdsadmin.list_sid_group_mapping”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.list_sid_group_mapping` gespeicherte Prozedur, um eine Liste aller Sicherheits-ID (SID) und Active Directory-Gruppenzuordnungen zurückzugeben, die auf einer RDS-für-DB2-DB-Instance konfiguriert sind.
the section called “rdsadmin.list_users”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.list_users` gespeicherte Prozedur, um eine Liste von Benutzern auf einer Autorisierungsliste für eine Datenbank auf einer RDS for Db2-DB-Instance zurückzugeben.
the section called “rdsadmin.reactivate_database”	Datenbanken	Verwenden Sie die `rdsadmin.reactivate_database` gespeicherte Prozedur, um eine Datenbank auf einer RDS for Db2-DB-Instance zu reaktivieren, nachdem Sie Änderungen an der Datenbankkonfiguration vorgenommen haben. Für eine Datenbank auf einer eigenständigen DB-Instance können Sie entweder diese gespeicherte Prozedur oder die rdsadmin.activate_database gespeicherte Prozedur verwenden.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.remove_groups”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.remove_groups` gespeicherte Prozedur, um eine oder mehrere Gruppen von einem Benutzer für eine Datenbank auf einer RDS for Db2-DB-Instance zu entfernen.
the section called “rdsadmin.remove_sid_group_mapping”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.remove_sid_group_mapping` gespeicherte Prozedur, um eine Sicherheits-ID (SID) und die entsprechende Active Directory-Gruppenzuordnung aus einer RDS-für-DB2-DB-Instance zu entfernen.
the section called “rdsadmin.remove_user”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.remove_user` gespeicherte Prozedur, um einen Benutzer aus einer Autorisierungsliste für eine Datenbank auf einer RDS for Db2-DB-Instance zu entfernen.
the section called “rdsadmin.rename_tablespace”	Tablespaces	Verwenden Sie die `rdsadmin.rename_tablespace` gespeicherte Prozedur, um einen Tablespace für eine Datenbank auf einer RDS for Db2-DB-Instance umzubenennen.
the section called “rdsadmin.restore_database”	Datenbanken	Verwenden Sie die `rdsadmin.restore_database` gespeicherte Prozedur, um eine Datenbank auf einer RDS for Db2-DB-Instance aus einem Amazon S3 S3-Bucket wiederherzustellen.
the section called “rdsadmin.revoke_role”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.revoke_role` gespeicherte Prozedur, um einer Rolle, einem Benutzer oder einer Gruppe des Empfängers in einer Datenbank auf einer RDS for Db2-DB-Instance eine Rolle zu entziehen.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.rollforward_database”	Datenbanken	Verwenden Sie die `rdsadmin.rollforward_database` gespeicherte Prozedur, um eine Datenbank auf einer RDS for Db2-DB-Instance online zu schalten und Transaktionsprotokolle anzuwenden, nachdem Sie eine Datenbank auf einer RDS for Db2-DB-Instance durch einen Aufruf wiederhergestellt haben. the section called “rdsadmin.restore_database”
the section called “rdsadmin.rollforward_status”	Datenbanken	Verwenden Sie die `rdsadmin.rollforward_status` gespeicherte Prozedur, um den Rollforward-Status des Aufrufs der the section called “rdsadmin.rollforward_database” oder der the section called “rdsadmin.complete_rollforward” gespeicherten Prozedur auf einer RDS for Db2-DB-Instance abzufragen.
the section called “rdsadmin.set_archive_log_retention”	Datenbanken	Verwenden Sie die `rdsadmin.set_archive_log_retention` gespeicherte Prozedur, um zu konfigurieren, wie lange Archivprotokolldateien für eine Datenbank auf einer RDS for Db2-DB-Instance aufbewahrt werden sollen. Sie können diese gespeicherte Prozedur auch verwenden, um die Aufbewahrung von Archivprotokollen zu deaktivieren.
the section called “rdsadmin.set_configuration”	Datenbanken	Verwenden Sie die `rdsadmin.set_configuration` gespeicherte Prozedur, um bestimmte Einstellungen für eine Datenbank auf einer RDS for Db2-DB-Instance zu konfigurieren.

Gespeicherte Prozedur	Kategorie	Beschreibung
the section called “rdsadmin.set_sid_group_mapping”	Erteilen und Widerrufen von Rechten	Verwenden Sie die `rdsadmin.set_sid_group_mapping` gespeicherte Prozedur, um eine Zuordnung zwischen einer Sicherheits-ID (SID) und der entsprechenden Active Directory -Gruppe auf einer RDS for Db2-DB-Instance zu erstellen.
the section called “rdsadmin.show_archive_log_retention”	Datenbanken	Verwenden Sie die `rdsadmin.show_archive_log_retention` gespeicherte Prozedur, um die aktuelle Einstellung für die Aufbewahrung von Archivprotokollen für eine Datenbank auf einer RDS for Db2-DB-Instance zurückzugeben.
the section called “rdsadmin.show_configuration”	Datenbanken	Verwenden Sie die `rdsadmin.show_configuration` gespeicherte Prozedur, um eine oder mehrere Einstellungen zurückzugeben, die für eine Datenbank auf einer RDS for Db2-DB-Instance geändert werden können.
the section called “rdsadmin.uncatalog_storage_access”	Speicherzugriff	Verwenden Sie die `rdsadmin.uncatalog_storage_access` gespeicherte Prozedur, um einen Speicheralias für den Zugriff auf einen Amazon S3 S3-Bucket mit Db2-Daten dateien zu entfernen.
the section called “rdsadmin.update_db_param”	Datenbanken	Verwenden Sie die `rdsadmin.update_db_param` gespeicherte Prozedur, um Datenbankparameter für eine Datenbank auf einer RDS for Db2-DB-Instance zu aktualisieren.

Themen

- Überlegungen zu gespeicherten Amazon RDS for Db2-Prozeduren

- Gespeicherte Prozeduren zum Erteilen und Widerrufen von Rechten für RDS for Db2
- Gespeicherte Prozeduren für Überwachungsrichtlinien für RDS for Db2
- Gespeicherte Prozeduren für Pufferpools RDS für Db2
- Gespeicherte Prozeduren für Datenbanken für RDS for Db2
- Gespeicherte Prozeduren für den Speicherzugriff RDS für Db2
- Gespeicherte Prozeduren für Tablespaces für RDS for Db2

Überlegungen zu gespeicherten Amazon RDS for Db2-Prozeduren

Bevor Sie die gespeicherten Systemprozeduren von Amazon RDS für RDS für Db2-DB-Instances verwenden, auf denen die Db2-Engine ausgeführt wird, überprüfen Sie die folgenden Informationen:

- Bevor Sie die gespeicherten Prozeduren ausführen, müssen Sie sich zunächst als Masterbenutzer für Ihre RDS for Db2-DB-Instance mit der `rdsadmin` Datenbank verbinden. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre eigenen Informationen.

  ```
  db2 "connect to rdsadmin user master_username using master_password"
  ```

- Die gespeicherten Prozeduren geben den `ERR_MESSAGE` Parameter zurück, der angibt, ob die gespeicherte Prozedur erfolgreich ausgeführt wurde oder nicht und warum sie nicht erfolgreich ausgeführt wurde.

  Beispiele

  Das folgende Beispiel zeigt, dass die gespeicherte Prozedur erfolgreich ausgeführt wurde.

  ```
  Parameter Name : ERR_MESSAGE
  Parameter Value : -
  Return Status = 0
  ```

  Das folgende Beispiel zeigt, dass die gespeicherte Prozedur nicht erfolgreich ausgeführt wurde, weil der in der gespeicherten Prozedur verwendete Amazon S3 S3-Bucket-Name nicht gültig war.

  ```
  Parameter Name : ERR_MESSAGE
  Parameter Value : Invalid S3 bucket name
  Return Status = -1006
  ```

Informationen zu Fehlermeldungen, die beim Aufrufen von gespeicherten Prozeduren zurückgegeben werden, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Hinweise zur Überprüfung des Status einer gespeicherten Prozedur finden Sie unterrdsadmin.get_task_status.

Gespeicherte Prozeduren zum Erteilen und Widerrufen von Rechten für RDS for Db2

Die in diesem Thema beschriebenen integrierten gespeicherten Prozeduren verwalten Benutzer, Rollen, Gruppen und Autorisierungen für Amazon RDS for Db2-Datenbanken. Um diese Prozeduren auszuführen, muss der Masterbenutzer zuerst eine Verbindung zur `rdsadmin` Datenbank herstellen.

Aufgaben, die diese gespeicherten Prozeduren verwenden, finden Sie unter Erteilen und Widerrufen von Rechten undAuthentifizierung einrichten Kerberos.

Informationen zu deren Syntax, Parametern, Verwendungshinweisen und Beispielen finden Sie in den folgenden integrierten gespeicherten Prozeduren.

Gespeicherte Prozeduren

- rdsadmin.create_role
- rdsadmin.grant_role
- rdsadmin.revoke_role
- rdsadmin.drop_role
- rdsadmin.add_user
- rdsadmin.change_password
- rdsadmin.list_users
- rdsadmin.remove_user
- rdsadmin.add_groups
- rdsadmin.remove_groups
- rdsadmin.dbadm_grant
- rdsadmin.dbadm_revoke
- rdsadmin.set_sid_group_mapping
- rdsadmin.list_sid_group_mapping
- rdsadmin.remove_sid_group_mapping

rdsadmin.create_role

Erzeugt eine Rolle.

Syntax

```
db2 "call rdsadmin.create_role(
    'database_name',
    'role_name')"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, in der der Befehl ausgeführt wird. Der Datentyp ist`varchar`.

role_name

Der Name der Rolle, die Sie erstellen möchten. Der Datentyp ist`varchar`.

Nutzungshinweise

Hinweise zur Überprüfung des Status beim Erstellen einer Rolle finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird eine Rolle namens `MY_ROLE` Datenbank erstelltDB2DB.

```
db2 "call rdsadmin.create_role(
    'DB2DB',
    'MY_ROLE')"
```

rdsadmin.grant_role

Weist einer Rolle, einem Benutzer oder einer Gruppe eine Rolle zu.

Syntax

```
db2 "call rdsadmin.grant_role(
    ?,
    'database_name',
    'role_name',
```

```
    'grantee',
    'admin_option')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die den eindeutigen Bezeichner für die Aufgabe ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Datenbank, auf der der Befehl ausgeführt werden soll. Der Datentyp istvarchar.

role_name

Der Name der Rolle, die Sie erstellen möchten. Der Datentyp istvarchar.

grantee

Die Rolle, der Benutzer oder die Gruppe, die die Autorisierung erhalten soll. Der Datentyp istvarchar. Zulässige Werte: ROLE, USER, GROUP, PUBLIC.

Das Format muss ein Wert gefolgt von einem Namen sein. Trennen Sie mehrere Werte und Namen durch Kommas. Beispiel: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ersetzen Sie die Namen durch Ihre eigenen Informationen.

Der folgende Eingabeparameter ist optional:

admin_option

Gibt an, ob der Empfänger DBADM berechtigt ROLE ist, Rollen zuzuweisen. Der Datentyp istchar. Der Standardwert ist N.

Nutzungshinweise

Hinweise zur Überprüfung des Status der Rollenzuweisung finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Rolle, Benutzer und Gruppe zuweisen und Autorisierung erteilen

Im folgenden Beispiel wird der `ROLE_TEST` aufgerufenen Rolle, dem `TESTDB` aufgerufenen Benutzer und der `role1` aufgerufenen Gruppe eine Rolle namens `user1` Datenbank zugewiesen. `group1` `ROLE_TEST`erhält die Administratorberechtigung zum Zuweisen von Rollen.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1',
    'Y')"
```

Beispiel 2: Eine Rolle zuweisen **`PUBLIC`** und keine Autorisierung gewähren

Im folgenden Beispiel wird eine Rolle zugewiesen, die als Datenbank `TESTDB` bezeichnet wird`ROLE_TEST`. `PUBLIC` `ROLE_TEST`That keine Administratorberechtigung zum Zuweisen von Rollen.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'PUBLIC')"
```

rdsadmin.revoke_role

Widerruft einer Rolle, einem Benutzer oder einer Gruppe eine Rolle.

Syntax

```
db2 "call rdsadmin.revoke_role(
    ?,
    'database_name',
    'role_name',
    'grantee')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die den eindeutigen Bezeichner für die Aufgabe ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Datenbank, auf der der Befehl ausgeführt werden soll. Der Datentyp ist`varchar`.

role_name

Der Name der Rolle, die Sie widerrufen möchten. Der Datentyp ist`varchar`.

grantee

Die Rolle, der Benutzer oder die Gruppe, für die die Autorisierung verloren gehen soll. Der Datentyp ist`varchar`. Zulässige Werte: `ROLE`, `USER`, `GROUP`, `PUBLIC`.

Das Format muss ein Wert gefolgt von einem Namen sein. Trennen Sie mehrere Werte und Namen durch Kommas. Beispiel: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ersetzen Sie die Namen durch Ihre eigenen Informationen.

Nutzungshinweise

Informationen zur Überprüfung des Status beim Widerrufen einer Rolle finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Rolle, Benutzer und Gruppe entziehen

Im folgenden Beispiel wird der aufgerufenen Rolle, dem `ROLE_TEST` `TESTDB` aufgerufenen Benutzer und der `role1` aufgerufenen Gruppe eine Rolle für die Datenbank entzogen. `user1` `group1`

```
db2 "call rdsadmin.revoke_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1')"
```

Beispiel 2: Sperren der Rolle von **`PUBLIC`**

Das folgende Beispiel widerruft eine Rolle, die `ROLE_TEST` für die Datenbank `TESTDB` von aufgerufen wurde. `PUBLIC`

```
db2 "call rdsadmin.revoke_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'PUBLIC')"
```

rdsadmin.drop_role

Löscht eine Rolle.

Syntax

```
db2 "call rdsadmin.drop_role(
    ?,
    'database_name',
    'role_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die den eindeutigen Bezeichner für die Aufgabe ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

`database_name`

Der Name der Datenbank, auf der der Befehl ausgeführt werden soll. Der Datentyp ist`varchar`.

`role_name`

Der Name der Rolle, die Sie löschen möchten. Der Datentyp ist`varchar`.

Nutzungshinweise

Hinweise zur Überprüfung des Status beim Löschen einer Rolle finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird eine Rolle gelöscht, `ROLE_TEST` die als Datenbank bezeichnet wird`TESTDB`.

```
db2 "call rdsadmin.drop_role(
    ?,
    'TESTDB',
    'ROLE_TEST')"
```

rdsadmin.add_user

Fügt einen Benutzer zu einer Autorisierungsliste hinzu.

Syntax

```
db2 "call rdsadmin.add_user(
    'username',
    'password',
    'group_name,group_name')"
```

Parameter

Die folgenden Parameter sind erforderlich:

`username`

Der Nutzername eines Benutzers. Der Datentyp ist`varchar`.

`password`

Das Passwort eines Benutzers. Der Datentyp ist`varchar`.

Der folgende Parameter ist optional:

group_name

Der Name einer Gruppe, zu der Sie den Benutzer hinzufügen möchten. Der Datentyp ist`varchar`. Die Standardeinstellung ist eine leere Zeichenfolge oder Null.

Nutzungshinweise

Sie können einen Benutzer zu einer oder mehreren Gruppen hinzufügen, indem Sie die Gruppennamen durch Kommas trennen.

Sie können eine Gruppe erstellen, wenn Sie einen neuen Benutzer erstellen, oder wenn Sie einem vorhandenen Benutzer eine Gruppe hinzufügen. Sie können eine Gruppe nicht alleine erstellen.

> **Note**
>
> Die maximale Anzahl von Benutzern, die Sie per Anruf hinzufügen können, `rdsadmin.add_user` beträgt 5.000.

Informationen zum Überprüfen des Status beim Hinzufügen eines Benutzers finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird ein Benutzer mit dem Namen `jorge_souza` und erstellt und der Benutzer wird den Gruppen mit dem Namen `sales` und `inside_sales` zugewiesen.

```
db2 "call rdsadmin.add_user(
    'jorge_souza',
    '*******',
    'sales,inside_sales')"
```

rdsadmin.change_password

Ändert das Passwort eines Benutzers.

Syntax

```
db2 "call rdsadmin.change_password(
    'username',
```

```
    'new_password')"
```

Parameter

Die folgenden Parameter sind erforderlich:

username

Der Nutzername eines Benutzers. Der Datentyp ist`varchar`.

new_password

Ein neues Passwort für den Benutzer. Der Datentyp ist`varchar`.

Nutzungshinweise

Hinweise zur Überprüfung des Status einer Kennwortänderung finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird das Passwort für geändert`jorge_souza`.

```
db2 "call rdsadmin.change_password(
    'jorge_souza',
    '*******')"
```

rdsadmin.list_users

Führt Benutzer in einer Autorisierungsliste auf.

Syntax

```
db2 "call rdsadmin.list_users()"
```

Nutzungshinweise

Hinweise zur Überprüfung des Status von Benutzerlisten finden Sie unterrdsadmin.get_task_status.

rdsadmin.remove_user

Entfernt den Benutzer aus der Autorisierungsliste.

Syntax

```
db2 "call rdsadmin.remove_user('username')"
```

Parameter

Der folgende Parameter ist erforderlich:

username

Der Nutzername eines Benutzers. Der Datentyp ist`varchar`.

Nutzungshinweise

Hinweise zur Überprüfung des Status beim Entfernen eines Benutzers finden Sie unter[rdsadmin.get_task_status](#).

Beispiele

Im folgenden Beispiel wird `jorge_souza` der Zugriff auf Datenbanken in RDS für Db2-DB-Instances verhindert.

```
db2 "call rdsadmin.remove_user('jorge_souza')"
```

rdsadmin.add_groups

Fügt einem Benutzer Gruppen hinzu.

Syntax

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Parameter

Die folgenden Parameter sind erforderlich:

username

Der Nutzername eines Benutzers. Der Datentyp ist`varchar`.

group_name

Der Name einer Gruppe, zu der Sie den Benutzer hinzufügen möchten. Der Datentyp ist`varchar`. Der Standardwert ist eine leere Zeichenfolge.

Nutzungshinweise

Sie können einem Benutzer eine oder mehrere Gruppen hinzufügen, indem Sie die Gruppennamen durch Kommas trennen. Informationen zur Überprüfung des Status beim Hinzufügen von Gruppen finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel werden die `b2b_sales` Gruppen `direct_sales` und zum Benutzer hinzugefügt`jorge_souza`.

```
db2 "call rdsadmin.add_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.remove_groups

Entfernt Gruppen aus einem Benutzer.

Syntax

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Parameter

Die folgenden Parameter sind erforderlich:

username

Der Nutzername eines Benutzers. Der Datentyp ist`varchar`.

group_name

Der Name einer Gruppe, aus der Sie den Benutzer entfernen möchten. Der Datentyp ist`varchar`.

Nutzungshinweise

Sie können eine oder mehrere Gruppen aus einem Benutzer entfernen, indem Sie die Gruppennamen durch Kommas trennen.

Informationen zur Überprüfung des Status beim Entfernen von Gruppen finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel werden die `b2b_sales` Gruppen `direct_sales` und aus dem Benutzer entfernt`jorge_souza`.

```
db2 "call rdsadmin.remove_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.dbadm_grant

Erteilt einer Rolle `DBADMACCESSCTRL`, einem Benutzer oder einer Gruppe eine `DATAACCESS` Autorisierung oder Autorisierung.

Syntax

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die den eindeutigen Bezeichner für die Aufgabe ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Datenbank, auf der der Befehl ausgeführt werden soll. Der Datentyp ist`varchar`.

authorization

Die Art der zu erteilenden Autorisierung. Der Datentyp ist`varchar`. Zulässige Werte: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Trennen Sie mehrere Typen durch Kommas.

grantee

Die Rolle, der Benutzer oder die Gruppe, die die Autorisierung erhalten soll. Der Datentyp ist`varchar`. Zulässige Werte: `ROLE`, `USER`, `GROUP`.

Das Format muss ein Wert gefolgt von einem Namen sein. Trennen Sie mehrere Werte und Namen durch Kommas. Beispiel: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ersetzen Sie die Namen durch Ihre eigenen Informationen.

Nutzungshinweise

Die Rolle, um Zugriff zu erhalten, muss vorhanden sein.

Hinweise zur Überprüfung des Status der Gewährung von Datenbankadministratoren finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Datenbankadministratorzugriff auf eine Rolle gewähren

Im folgenden Beispiel wird dem Datenbankadministrator Zugriff auf die `TESTDB` nach der Rolle benannte Datenbank gewährt`ROLE_DBA`.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

Beispiel 2: Benutzern und Gruppen Datenbankadministratorzugriff gewähren

Im folgenden Beispiel wird dem Datenbankadministrator Zugriff auf die Datenbank mit dem Namen `TESTDB` `user1` und gewährt`group1`.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

Beispiel 3: Gewährung von Datenbankadministratorzugriff für mehrere Benutzer und Gruppen

Im folgenden Beispiel wird Datenbankadministratoren Zugriff auf die Datenbank mit dem Namen `TESTDB` `user1`, `user2` `group1`, und gewährt `group2`.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

rdsadmin.dbadm_revoke

Widerruft `DBADM` `ACCESSCTRL`, oder die `DATAACCESS` Autorisierung für eine Rolle, einen Benutzer oder eine Gruppe.

Syntax

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Die eindeutige Kennung für die Aufgabe. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Datenbank, auf der der Befehl ausgeführt werden soll. Der Datentyp ist`varchar`.

authorization

Die Art der zu widerrufenden Autorisierung. Der Datentyp ist`varchar`. Zulässige Werte: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Trennen Sie mehrere Typen durch Kommas.

grantee

Die Rolle, der Benutzer oder die Gruppe, für die die Autorisierung widerrufen werden soll. Der Datentyp ist`varchar`. Zulässige Werte: `ROLE`, `USER`, `GROUP`.

Das Format muss ein Wert gefolgt von einem Namen sein. Trennen Sie mehrere Werte und Namen durch Kommas. Beispiel: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Ersetzen Sie die Namen durch Ihre eigenen Informationen.

Nutzungshinweise

Hinweise zur Überprüfung des Status des Widerrufs des Datenbankadministratorzugriffs finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Widerrufen des Datenbankadministratorzugriffs für die Rolle

Im folgenden Beispiel wird der Datenbankadministratorzugriff auf die nach der Rolle benannte `TESTDB` Datenbank widerrufen. `ROLE_DBA`

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

Beispiel 2: Widerrufen des Datenbankadministratorzugriffs für Benutzer und Gruppen

Im folgenden Beispiel wird dem Datenbankadministrator der Zugriff auf die Datenbank mit dem Namen `TESTDB` und entzogen. `user1` `group1`

```
db2 "call rdsadmin.dbadm_revoke(
```

```
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

Beispiel 3: Widerrufen des Datenbankadministratorzugriffs für mehrere Benutzer und Gruppen

Im folgenden Beispiel wird der Datenbankadministratorzugriff auf die Datenbank mit dem Namen `TESTDB` `user1`, `user2` `group1`, und entzogen. `group2`

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

rdsadmin.set_sid_group_mapping

Erstellt eine Zuordnung zwischen einer Sicherheits-ID (SID) und der entsprechenden Active Directory-Gruppe.

Syntax

```
db2 "call rdsadmin.set_sid_group_mapping(
    ?,
    'SID',
    'group_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

SID

Die Sicherheits-ID (SID). Der Datentyp ist`varchar`.

group_name

Der Name der Active Directory-Gruppe, die der SID zugeordnet werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Verwenden Sie diese gespeicherte Prozedur, um die Kerberos-Authentifizierung mit Active Directory-Gruppen zu aktivieren. Wenn das `SID` oder `group_name` bereits in der Zuordnung vorhanden ist, schlägt diese gespeicherte Prozedur fehl.

Hinweise zum Ermitteln der SID für eine Gruppe finden Sie unterSchritt 8: Rufen Sie die Active Directory-Gruppen-SID ab in PowerShell.

Informationen zum Überprüfen des Status beim Erstellen einer Zuordnung finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird eine SID einer Gruppe mit dem Namen zugeordnetmy_`group`.

```
db2 "call rdsadmin.set_sid_group_mapping(
    ?,
    'S-1-5-21-9146495592-531070549-834388463-513',
    'my_group')"
```

rdsadmin.list_sid_group_mapping

Listet alle Sicherheits-ID (SID) und Active Directory-Gruppenzuordnungen auf, die auf der DB-Instance konfiguriert sind.

Syntax

```
db2 "call rdsadmin.list_sid_group_mapping()"
```

Nutzungshinweise

Hinweise zur Überprüfung des Status von Listenzuordnungen finden Sie unter. rdsadmin.get_task_status

rdsadmin.remove_sid_group_mapping

Entfernt eine Sicherheits-ID (SID) und die entsprechende Active Directory-Gruppenzuordnung aus einer DB-Instance.

Syntax

```
db2 "call rdsadmin.remove_sid_group_mapping(
    ?,
    'SID')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

SID

Die Sicherheits-ID (SID). Der Datentyp ist`varchar`.

Nutzungshinweise

Informationen zum Ermitteln der SID für eine Gruppe finden Sie unterSchritt 8: Rufen Sie die Active Directory-Gruppen-SID ab in PowerShell.

Informationen zur Überprüfung des Status beim Entfernen von Zuordnungen finden Sie unter. rdsadmin.get_task_status

Beispiele

Im folgenden Beispiel wird eine SID-Zuordnung aus der Gruppe entfernt, der sie zugeordnet wurde.

```
db2 "call rdsadmin.remove_sid_group_mapping(
    ?,
    'S-1-5-21-9146495592-531070549-834388463-513')"
```

Gespeicherte Prozeduren für Überwachungsrichtlinien für RDS for Db2

Die in diesem Thema beschriebenen integrierten gespeicherten Prozeduren verwalten Überwachungsrichtlinien für Amazon RDS for Db2-Datenbanken, die Auditprotokollierung verwenden. Weitere Informationen finden Sie unter the section called “Db2-Auditprotokollierung”. Um diese Prozeduren auszuführen, muss der Masterbenutzer zuerst eine Verbindung zur `rdsadmin` Datenbank herstellen.

Informationen zu deren Syntax, Parametern, Verwendungshinweisen und Beispielen finden Sie in den folgenden integrierten gespeicherten Prozeduren.

Gespeicherte Prozeduren

- rdsadmin.configure_db_audit
- rdsadmin.disable_db_audit

rdsadmin.configure_db_audit

Konfiguriert die Überwachungsrichtlinie für die RDS for Db2-Datenbank, angegeben von. *db_name* Wenn die Richtlinie, die Sie konfigurieren, nicht existiert, wird sie beim Aufrufen dieser gespeicherten Prozedur erstellt. Wenn diese Richtlinie existiert, wird sie beim Aufrufen dieser gespeicherten Prozedur mit den von Ihnen angegebenen Parameterwerten geändert.

Syntax

```
db2 "call rdsadmin.configure_db_audit(
    'db_name',
    'category',
    'category_setting',
    '?')"
```

Parameter

Die folgenden Parameter sind erforderlich.

db_name

Der DB-Name der RDS for Db2-Datenbank, für die die Überwachungsrichtlinie konfiguriert werden soll. Der Datentyp ist`varchar`.

category

Der Name der Kategorie, für die diese Überwachungsrichtlinie konfiguriert werden soll. Der Datentyp ist`varchar`. Die folgenden Werte sind für diesen Parameter gültig:

- `ALL`— Bei `ALL` Amazon RDS sind die `ERROR` Kategorien `CONTEXTEXECUTE`, oder nicht enthalten.
- `AUDIT`
- `CHECKING`
- `CONTEXT`
- `ERROR`
- `EXECUTE`— Sie können diese Kategorie mit oder ohne Daten konfigurieren. Mit Daten bedeutet, auch Eingabedatenwerte zu protokollieren, die für beliebige Hostvariablen und Parametermarkierungen bereitgestellt wurden. Die Standardeinstellung ist ohne Daten. Weitere Informationen finden Sie in der Beschreibung des *category_setting* Parameters und desthe section called “Beispiele”.
- `OBJMAINT`
- `SECMAINT`
- `SYSADMIN`
- `VALIDATE`

Weitere Informationen zu diesen Kategorien finden Sie im IBM Db2 Dokumentation.

category_setting

Die Einstellung für die angegebene Audit-Kategorie. Der Datentyp ist`varchar`.

Die folgende Tabelle zeigt die gültigen Kategorieeinstellungswerte für jede Kategorie.

Kategorie	Gültige Kategorieeinstellungen
`ALL`	`BOTH`|`FAILURE`|`SUCCESS`|`NONE`
`AUDIT`	
`CHECKING`	
`CONTEXT`	

Kategorie	Gültige Kategorieeinstellungen
OBJMAINT SECMAINT SYSADMIN VALIDATE	
ERROR	AUDIT|NORMAL . Die Standardeinstellung istNORMAL.
EXECUTE	BOTH,WITH|BOTH,WITHOUT|FAILURE,WITH| FAILURE,WITHOUT|SUCCESS,WITH|SUCCESS ,WITHOUT|NONE

Nutzungshinweise

Stellen Sie vor dem Aufrufen sicherrdsadmin.configure_db_audit, dass die RDS for Db2-DB-Instance mit der Datenbank, für die Sie die Überwachungsrichtlinie konfigurieren, einer Optionsgruppe zugeordnet ist, für die diese DB2_AUDIT Option verfügbar ist. Weitere Informationen finden Sie unter the section called “Einrichtung der Db2-Auditprotokollierung”.

Nachdem Sie die Überwachungsrichtlinie konfiguriert haben, können Sie den Status der Überwachungskonfiguration für die Datenbank überprüfen, indem Sie die Schritte unter befolgen. Überprüfen Sie die Audit-Konfiguration

Die Angabe ALL für den category Parameter schließt die ERROR Kategorien CONTEXTEXECUTE, oder nicht mit ein. Um diese Kategorien zu Ihrer Überwachungsrichtlinie hinzuzufügen, rufen Sie jede Kategorie, die Sie hinzufügen möchten, rdsadmin.configure_db_audit separat an. Weitere Informationen finden Sie unter the section called “Beispiele”.

Beispiele

In den folgenden Beispielen wird die Überwachungsrichtlinie für eine Datenbank mit dem Namen erstellt oder geändertTESTDB. Wenn die ERROR Kategorie in den Beispielen 1 bis 5 nicht zuvor konfiguriert wurde, ist diese Kategorie auf NORMAL (Standard) gesetzt. Gehen Sie wie folgt vorAUDIT, um diese Einstellung zu ändernExample 6: Specifying the ERROR category.

Beispiel 1: Angabe der **ALL** Kategorie

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ALL', 'BOTH', ?)"
```

In diesem Beispiel konfiguriert der Aufruf die `VALIDATE` Kategorien`AUDIT`,`CHECKING`,`OBJMAINT`, `SECMAINTSYSADMIN`, und in der Prüfrichtlinie. Die Angabe `BOTH` bedeutet, dass sowohl erfolgreiche als auch fehlgeschlagene Ereignisse für jede dieser Kategorien geprüft werden.

Beispiel 2: Angabe der **`EXECUTE`** Kategorie anhand von Daten

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'SUCCESS,WITH', ?)"
```

In dem Beispiel konfiguriert der Aufruf die `EXECUTE` Kategorie in der Prüfrichtlinie. Die Angabe `SUCCESS,WITH` bedeutet, dass die Protokolle für diese Kategorie nur erfolgreiche Ereignisse und Eingabedatenwerte enthalten, die für Hostvariablen und Parametermarkierungen bereitgestellt wurden.

Beispiel 3: Angabe der **`EXECUTE`** Kategorie ohne Daten

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'FAILURE,WITHOUT', ?)"
```

In dem Beispiel konfiguriert der Aufruf die `EXECUTE` Kategorie in der Prüfrichtlinie. Die Angabe `FAILURE,WITHOUT` bedeutet, dass die Protokolle für diese Kategorie nur fehlgeschlagene Ereignisse und keine Eingabedatenwerte enthalten, die für Hostvariablen und Parametermarkierungen bereitgestellt wurden.

Beispiel 4: Angabe der **`EXECUTE`** Kategorie ohne Statusereignisse

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'NONE', ?)"
```

In dem Beispiel konfiguriert der Aufruf die `EXECUTE` Kategorie in der Audit-Richtlinie. Eine Angabe `NONE` bedeutet, dass keine Ereignisse in dieser Kategorie geprüft werden.

Beispiel 5: Angabe der Kategorie **`OBJMAINT`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'OBJMAINT', 'NONE', ?)"
```

In dem Beispiel konfiguriert der Aufruf die `OBJMAINT` Kategorie in der Prüfrichtlinie. Eine Angabe `NONE` bedeutet, dass keine Ereignisse in dieser Kategorie geprüft werden.

Beispiel 6: Angabe der Kategorie **`ERROR`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ERROR', 'AUDIT', ?)"
```

In dem Beispiel konfiguriert der Aufruf die ERROR Kategorie in der Prüfrichtlinie. Spezifizieren AUDIT bedeutet, dass alle Fehler, einschließlich der Fehler, die in der Auditprotokollierung selbst auftreten, in den Protokollen erfasst werden. Der Standardfehlertyp istNORMAL. Bei werden die durch die Prüfung generierten Fehler ignoriertNORMAL, und es werden nur die SQLCODE s für Fehler im Zusammenhang mit dem ausgeführten Vorgang erfasst.

rdsadmin.disable_db_audit

Beendet die Überwachungsprotokollierung für die von angegebene RDS for Db2-Datenbank *db_name* und entfernt die dafür konfigurierte Überwachungsrichtlinie.

> **Note**
>
> Diese gespeicherte Prozedur entfernt nur Überwachungsrichtlinien, die durch einen Aufruf the section called “rdsadmin.configure_db_audit” konfiguriert wurden.

Syntax

```
db2 "call rdsadmin.disable_db_audit('db_name', ?)"
```

Parameter

Die folgenden Parameter sind erforderlich.

db_name

Der DB-Name der RDS for Db2-Datenbank, für die die Audit-Protokollierung deaktiviert werden soll. Der Datentyp istvarchar.

Nutzungshinweise

Durch rdsadmin.disable_db_audit das Aufrufen wird die Audit-Protokollierung für die RDS for Db2-DB-Instance nicht deaktiviert. Um die Audit-Protokollierung auf DB-Instance-Ebene zu deaktivieren, entfernen Sie die Optionsgruppe aus der DB-Instance. Weitere Informationen finden Sie unter Deaktivierung der Db2-Auditprotokollierung.

Beispiele

Im folgenden Beispiel wird die Auditprotokollierung für eine Datenbank mit dem Namen `TESTDB` deaktiviert.

```
db2 "call rdsadmin.disable_db_audit('TESTDB', ?)"
```

Gespeicherte Prozeduren für Pufferpools RDS für Db2

Die in diesem Thema beschriebenen integrierten gespeicherten Prozeduren verwalten Pufferpools für Amazon RDS for Db2-Datenbanken. Um diese Prozeduren auszuführen, muss der Masterbenutzer zuerst eine Verbindung zur `rdsadmin` Datenbank herstellen.

Diese gespeicherten Prozeduren werden für eine Vielzahl von Aufgaben verwendet. Diese Liste ist nicht umfassend.

- Allgemeine Aufgaben für Pufferpools
- Generieren von Leistungsberichten
- Datenbank-Metadaten mit db2look kopieren
- Erstellen einer Repository-Datenbank für IBM Db2 Data Management Console

Informationen zu deren Syntax, Parametern, Verwendungshinweisen und Beispielen finden Sie in den folgenden integrierten gespeicherten Prozeduren.

Gespeicherte Prozeduren

- rdsadmin.create_bufferpool
- rdsadmin.alter_bufferpool
- rdsadmin.drop_bufferpool

rdsadmin.create_bufferpool

Erzeugt einen Pufferpool.

Syntax

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, in der der Befehl ausgeführt werden soll. Der Datentyp ist`varchar`.

buffer_pool_name

Der Name des zu erstellenden Pufferpools. Der Datentyp ist`varchar`.

Die folgenden Parameter sind optional:

buffer_pool_size

Die Größe des Pufferpools in Seitenanzahl. Der Datentyp ist`integer`. Der Standardwert ist `-1`.

immediate

Gibt an, ob der Befehl sofort ausgeführt wird. Der Datentyp ist`char`. Der Standardwert ist `Y`.

automatic

Gibt an, ob der Pufferpool auf automatisch gesetzt werden soll. Der Datentyp ist`char`. Der Standardwert ist `Y`.

page_size

Die Seitengröße des Pufferpools. Der Datentyp ist`integer`. Zulässige Werte: `4096`, `8192`, `16384`, `32768`. Der Standardwert ist `8192`.

number_block_pages

Die Anzahl der Blockseiten in den Pufferpools. Der Datentyp ist`integer`. Der Standardwert ist `0`.

block_size

Die Blockgröße für die Blockseiten. Der Datentyp ist`integer`. Gültige Werte: `2` to `256`. Der Standardwert ist `32`.

Nutzungshinweise

Hinweise zur Überprüfung des Status der Erstellung eines Pufferpools finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Pufferpool mit Standardparametern erstellen

Das folgende Beispiel erstellt einen Pufferpool, der BP8 für eine Datenbank aufgerufen wird, die TESTDB mit Standardparametern aufgerufen wird, sodass der Pufferpool eine Seitengröße von 8 KB verwendet.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP8')"
```

Beispiel 2: Erstellen eines Pufferpools, der sofort mit automatischer Zuweisung ausgeführt werden kann

Das folgende Beispiel erstellt einen Pufferpool, der BP16 für eine Datenbank aufgerufen wirdTESTDB, die eine Seitengröße von 16 KB mit einer anfänglichen Seitenanzahl von 1.000 verwendet und auf automatisch eingestellt ist. Db2 führt den Befehl sofort aus. Wenn Sie eine anfängliche Seitenanzahl von -1 verwenden, verwendet Db2 die automatische Seitenzuweisung.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    1000,
    'Y',
    'Y',
    16384)"
```

Beispiel 3: Erstellen eines Pufferpools zur sofortigen Ausführung mithilfe von Blockseiten

Das folgende Beispiel erstellt einen Pufferpool, der BP16 für eine Datenbank namens aufgerufen wirdTESTDB. Dieser Pufferpool hat eine Seitengröße von 16 KB mit einer anfänglichen Seitenanzahl von 10.000. Db2 führt den Befehl sofort aus und verwendet dabei 500 Blockseiten mit einer Blockgröße von 512.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
```

```
    'Y',
    16384,
    500,
    512)"
```

rdsadmin.alter_bufferpool

Ändert einen Pufferpool.

Syntax

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
    block_size)"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, in der der Befehl ausgeführt werden soll. Der Datentyp ist`varchar`.

buffer_pool_name

Der Name des zu ändernden Pufferpools. Der Datentyp ist`varchar`.

buffer_pool_size

Die Größe des Pufferpools in Seitenanzahl. Der Datentyp ist`integer`.

Die folgenden Parameter sind optional:

immediate

Gibt an, ob der Befehl sofort ausgeführt wird. Der Datentyp ist`char`. Der Standardwert ist `Y`.

automatic

Gibt an, ob der Pufferpool auf automatisch gesetzt werden soll. Der Datentyp ist`char`. Der Standardwert ist `N`.

change_number_blocks

Gibt an, ob sich die Anzahl der Blockseiten im Pufferpool geändert hat. Der Datentyp ist`char`. Der Standardwert ist `N`.

number_block_pages

Die Anzahl der Blockseiten in den Pufferpools. Der Datentyp ist`integer`. Der Standardwert ist `0`.

block_size

Die Blockgröße für die Blockseiten. Der Datentyp ist`integer`. Gültige Werte: `2` to `256`. Der Standardwert ist `32`.

Nutzungshinweise

Hinweise zur Überprüfung des Status von Änderungen an einem Pufferpool finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird ein Pufferpool, der `BP16` für eine Datenbank aufgerufen wird, `TESTDB` auf „nicht automatisch“ geändert und die Größe auf 10.000 Seiten geändert. Db2 führt diesen Befehl sofort aus.

```
db2 "call rdsadmin.alter_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
    'N')"
```

rdsadmin.drop_bufferpool

Löscht einen Pufferpool.

Syntax

```
db2 "call rdsadmin.drop_bufferpool(
```

```
    'database_name',
    'buffer_pool_name'"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, zu der der Pufferpool gehört. Der Datentyp ist`varchar`.

buffer_pool_name

Der Name des Pufferpools, der gelöscht werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Hinweise zur Überprüfung des Status beim Löschen eines Pufferpools finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird ein Pufferpool gelöscht, der BP16 für eine Datenbank namens aufgerufen wurdeTESTDB.

```
db2 "call rdsadmin.drop_bufferpool(
    'TESTDB',
    'BP16')"
```

Gespeicherte Prozeduren für Datenbanken für RDS for Db2

Die in diesem Thema beschriebenen integrierten gespeicherten Prozeduren verwalten Datenbanken für Amazon RDS for Db2. Um diese Prozeduren auszuführen, muss der Masterbenutzer zuerst eine Verbindung zur `rdsadmin` Datenbank herstellen.

Diese gespeicherten Prozeduren werden für eine Vielzahl von Aufgaben verwendet. Diese Liste ist nicht umfassend.

- Allgemeine Aufgaben für Datenbanken
- Datenbanken mit EBCDIC-Kollation erstellen
- Sammeln von Informationen über Datenbanken
- Ändern der Datenbankkonfigurationsparameter
- Migration von Linux zu Linux
- Migration von Linux zu Linux mit nahezu null Ausfallzeiten

Informationen zu deren Syntax, Parametern, Nutzungshinweisen und Beispielen finden Sie in den folgenden integrierten gespeicherten Prozeduren.

Gespeicherte Prozeduren

- rdsadmin.create_database
- rdsadmin.deactivate_database
- rdsadmin.activate_database
- rdsadmin.reactivate_database
- rdsadmin.drop_database
- rdsadmin.update_db_param
- rdsadmin.set_configuration
- rdsadmin.show_configuration
- rdsadmin.restore_database
- rdsadmin.rollforward_database
- rdsadmin.rollforward_status
- rdsadmin.complete_rollforward
- rdsadmin.db2pd_command

- rdsadmin.force_application
- rdsadmin.set_archive_log_retention
- rdsadmin.show_archive_log_retention
- rdsadmin.list_archive_log_information

rdsadmin.create_database

Erstellt eine Datenbank.

Syntax

```
db2 "call rdsadmin.create_database('database_name')"
```

Parameter

> **Note**
>
> Diese gespeicherte Prozedur validiert die Kombination der erforderlichen Parameter nicht. Beim Aufrufen rdsadmin.get_task_status kann die benutzerdefinierte Funktion aufgrund einer Kombination von`database_codeset`,, einen Fehler zurückgeben`database_territory`, `database_collation` die nicht gültig ist. Weitere Informationen finden Sie in der Dokumentation unter Codepage, Gebiet und Sortierung für Ihre Datenbank auswählen. IBM Db2

Der folgende Parameter ist erforderlich:

database_name

Der Name der zu erstellenden Datenbank. Der Datentyp ist`varchar`.

Die folgenden Parameter sind optional:

database_page_size

Die Standard-Seitengröße der Datenbank. Zulässige Werte: `4096`, `8192`, `16384`, `32768`. Der Datentyp ist`integer`. Der Standardwert ist `8192`.

> **Important**
>
> Amazon RDS unterstützt Schreibatomizität für Seiten mit 4 KiB, 8 KiB und 16 KiB. Im Gegensatz dazu besteht bei Seiten mit 32 KiB die Gefahr, dass Schreibvorgänge abgebrochen werden oder dass unvollständige Daten auf den Schreibtisch geschrieben werden. Wenn Sie Seiten mit 32 KiB verwenden, empfehlen wir Ihnen, die point-in-time Wiederherstellung und automatische Backups zu aktivieren. Andernfalls laufen Sie Gefahr, dass Sie zerrissene Seiten nicht wiederherstellen können. Weitere Informationen erhalten Sie unter the section called “Einführung in Backups” und the section called “Point-in-time Wiederherstellung”.

database_code_set

Der Codesatz für die Datenbank. Der Datentyp ist`varchar`. Der Standardwert ist `UTF-8`.

database_territory

Der aus zwei Buchstaben bestehende Ländercode für die Datenbank. Der Datentyp ist`varchar`. Der Standardwert ist `US`.

database_collation

Die Sortierreihenfolge, die bestimmt, wie in der Datenbank gespeicherte Zeichenketten sortiert und verglichen werden. Der Datentyp ist`varchar`.

Zulässige Werte:

- `COMPATIBILITY`— Eine IBM Db2 Version 2-Kollationssequenz.
- `EBCDIC_819_037`— Lateinische ISO-Codepage, Sortierung; CCSID 037 (EBCDIC, US-Englisch).
- `EBCDIC_819_500`— Lateinische ISO-Codepage, Sortierung; CCSID 500 (EBCDIC International).
- `EBCDIC_850_037`— Lateinische ASCII-Codepage, Sortierung; CCSID 037 (EBCDIC, US-Englisch).
- `EBCDIC_850_500`— Lateinische ASCII-Codepage, Sortierung; CCSID 500 (EBCDIC International).
- `EBCDIC_932_5026`— Japanische ASCII-Codepage, Sortierung; CCSID 5026 (EBCDIC, US-Englisch).

- `EBCDIC_932_5035`— Japanische ASCII-Codepage, Sortierung; CCSID 5035 (EBCDIC International).
- `EBCDIC_1252_037`— Lateinische Windows-Codepage, Sortierung; CCSID 037 (EBCDIC, US-Englisch).
- `EBCDIC_1252_500`— Lateinische Windows-Codepage, Sortierung; CCSID 500 (EBCDIC International).
- `IDENTITY`— Standardsortierung. Zeichenketten werden Byte für Byte verglichen.
- `IDENTITY_16BIT`— Das Kompatibilitätskodierungsschema für UTF-16:8-Bit-Kollationssequenz (CESU-8). Weitere Informationen finden Sie im technischen Unicode-Bericht #26 auf der Website des Unicode-Konsortiums.
- `NLSCHAR`— Nur zur Verwendung mit der thailändischen Codepage (CP874).
- `SYSTEM`— Wenn Sie verwenden`SYSTEM`, verwendet die Datenbank die Sortierreihenfolge automatisch für `database_codeset` und`database_territory`.

Der Standardwert ist `IDENTITY`.

Darüber hinaus unterstützt RDS for Db2 die folgenden Sortierungsgruppen: und. `language-aware-collation` `locale-sensitive-collation` Weitere Informationen finden Sie in der Dokumentation unter Auswählen einer Kollation für eine Unicode-Datenbank. IBM Db2

`database_autoconfigure_str`

Die `AUTOCONFIGURE` Befehlssyntax, zum Beispiel. `'AUTOCONFIGURE APPLY DB'` Der Datentyp ist`varchar`. Die Standardeinstellung ist eine leere Zeichenfolge oder Null.

Weitere Informationen finden Sie unter AUTOCONFIGUREBefehl in der IBM Db2 Dokumentation.

Nutzungshinweise

Wenn Sie den `db2_compatibility_vector` Parameter ändern möchten, ändern Sie den Parameter, bevor Sie eine Datenbank erstellen. Weitere Informationen finden Sie unter Einstellung des Parameters db2_compatibility_vector.

Besondere Überlegungen:

- Der an die Db2-Instanz gesendete `CREATE DATABASE` Befehl verwendet die `RESTRICTIVE` Option.

- RDS für Db2 verwendet `AUTOMATIC STORAGE` nur Tablespaces.
- RDS für Db2 verwendet die Standardwerte für und. `NUMSEGS` `DFT_EXTENT_SZ`
- RDS für Db2 verwendet Speicherverschlüsselung und unterstützt keine Datenbankverschlüsselung.

Weitere Informationen zu diesen Überlegungen finden Sie in der IBM Db2 Dokumentation unter CREATE DATABASEBefehl.

Vor dem Aufrufen `rdsadmin.create_database` müssen Sie eine Verbindung zur `rdsadmin` Datenbank herstellen. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre RDS for Db2-DB-Instance-Informationen:

```
db2 connect to rdsadmin user master_username using master_password
```

Hinweise zur Überprüfung des Status beim Erstellen einer Datenbank finden Sie unterrdsadmin.get_task_status.

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.create_database`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Im folgenden Beispiel wird eine Datenbank erstellt, die `TESTJP` mit einer korrekten Kombination der *`database_collation`* Parameter *`database_code_setdatabase_territory`*, und für Japan aufgerufen wird:

```
db2 "call rdsadmin.create_database('TESTJP', 4096, 'IBM-437', 'JP', 'SYSTEM')"
```

rdsadmin.deactivate_database

Deaktiviert eine Datenbank.

Syntax

```
db2 "call rdsadmin.deactivate_database(
    ?,
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der Datenbank, die deaktiviert werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Sie können Datenbanken deaktivieren, um Speicherressourcen zu sparen oder andere Änderungen an der Datenbankkonfiguration vorzunehmen. Rufen Sie die rdsadmin.activate_database gespeicherte Prozedur auf, um deaktivierte Datenbanken wieder online zu schalten.

Hinweise zur Überprüfung des Status der Deaktivierung einer Datenbank finden Sie unter. rdsadmin.get_task_status

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.deactivate_database`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Im folgenden Beispiel wird eine `TESTDB` aufgerufene Datenbank deaktiviert.

```
db2 "call rdsadmin.deactivate_database(?, 'TESTDB')"
```

rdsadmin.activate_database

Aktiviert eine Datenbank.

Informationen zu den Unterschieden zwischen rdsadmin.reactivate_database und `rdsadmin.activate_database` finden Sie unterNutzungshinweise.

Syntax

```
db2 "call rdsadmin.activate_database(
    ?,
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der zu aktivierenden Datenbank. Der Datentyp istvarchar.

Nutzungshinweise

Alle Datenbanken sind standardmäßig aktiviert, wenn sie erstellt werden. Wenn Sie eine Datenbank auf einer eigenständigen DB-Instance deaktivieren, um Speicherressourcen zu sparen oder andere Änderungen an der Datenbankkonfiguration vorzunehmen, rufen Sie die `rdsadmin.activate_database` gespeicherte Prozedur auf, um die Datenbank erneut zu aktivieren.

Hinweise zur Überprüfung des Status der Aktivierung einer Datenbank finden Sie unterrdsadmin.get_task_status.

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.activate_database`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Das folgende Beispiel aktiviert eine Datenbank namens`TESTDB`.

```
db2 "call rdsadmin.activate_database(?, 'TESTDB')"
```

rdsadmin.reactivate_database

Reaktiviert eine Datenbank.

Hinweise zu den Unterschieden zwischen rdsadmin.activate_database und finden Sie `rdsadmin.reactivate_database` unterNutzungshinweise.

Syntax

```
db2 "call rdsadmin.reactivate_database(
    ?,
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der Datenbank, die reaktiviert werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Wenn Sie die `rdsadmin.reactivate_database` gespeicherte Prozedur aufrufen, deaktiviert die gespeicherte Prozedur zuerst die Datenbank, indem sie die rdsadmin.deactivate_database gespeicherte Prozedur aufruft, und aktiviert dann die Datenbank, indem sie die rdsadmin.activate_database gespeicherte Prozedur aufruft.

Nachdem Sie Änderungen an den Datenbankkonfigurationen vorgenommen haben, müssen Sie möglicherweise eine Datenbank auf einer RDS for Db2-DB-Instance reaktivieren. Um festzustellen, ob Sie eine Datenbank reaktivieren müssen, stellen Sie eine Verbindung mit der Datenbank her und führen Sie den Befehl aus. `db2 get db cfg show detail`

Für eine Datenbank auf einer eigenständigen DB-Instance können Sie die `rdsadmin.reactivate_database` Store-Prozedur verwenden, um die Datenbank zu reaktivieren.

Oder, wenn Sie die rdsadmin.deactivate_database gespeicherte Prozedur bereits aufgerufen haben, können Sie stattdessen die rdsadmin.activate_database gespeicherte Prozedur aufrufen.

Hinweise zur Überprüfung des Status der Reaktivierung einer Datenbank finden Sie unterrdsadmin.get_task_status.

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.reactivate_database`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Im folgenden Beispiel wird eine `TESTDB` aufgerufene Datenbank reaktiviert.

```
db2 "call rdsadmin.reactivate_database(?, 'TESTDB')"
```

rdsadmin.drop_database

Entfernt eine Datenbank.

Syntax

```
db2 "call rdsadmin.drop_database('database_name')"
```

Parameter

Der folgende Parameter ist erforderlich:

`database_name`

Der Name der Datenbank, die gelöscht werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Sie können eine Datenbank `rdsadmin.drop_database` nur löschen, indem Sie sie aufrufen, wenn die folgenden Bedingungen erfüllt sind:

- Sie haben den Namen der Datenbank nicht angegeben, als Sie Ihre RDS for Db2-DB-Instance mit der Amazon RDS-Konsole oder der AWS CLI erstellt haben. Weitere Informationen finden Sie unter Erstellen einer DB-Instance.

- Sie haben die Datenbank erstellt, indem Sie die the section called “rdsadmin.create_database” gespeicherte Prozedur aufgerufen haben.
- Sie haben die Datenbank aus einem Offline- oder Sicherungsabbild wiederhergestellt, indem Sie die the section called “rdsadmin.restore_database” gespeicherte Prozedur aufgerufen haben.

Vor dem Aufrufen `rdsadmin.drop_database` müssen Sie eine Verbindung mit der `rdsadmin` Datenbank herstellen. Ersetzen Sie im folgenden Beispiel *`master_username`* und *`master_password`* durch Ihre RDS for Db2-DB-Instance-Informationen:

```
db2 connect to rdsadmin user master_username using master_password
```

Hinweise zur Überprüfung des Status beim Löschen einer Datenbank finden Sie unterrdsadmin.get_task_status.

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.drop_database`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Im folgenden Beispiel wird eine Datenbank mit dem Namen gelöscht`TESTDB`:

```
db2 "call rdsadmin.drop_database('TESTDB')"
```

rdsadmin.update_db_param

Aktualisiert Datenbankparameter.

Syntax

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value',
    'restart_database')"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, für die die Aufgabe ausgeführt werden soll. Der Datentyp ist`varchar`.

parameter_to_modify

Der Name des zu ändernden Parameters. Der Datentyp ist`varchar`. Weitere Informationen finden Sie unter Amazon RDS für Db2-Parameter.

changed_value

Der Wert, auf den der Parameterwert geändert werden soll. Der Datentyp ist`varchar`.

Der folgende Parameter ist optional:

restart_database

Gibt an, ob RDS die Datenbank neu startet, falls ein Neustart erforderlich ist. Der Datentyp ist`varchar`. Um `logprimary` und zu ändern`logfilsiz`, setzen Sie diesen Parameter auf`'YES'`.

Nutzungshinweise

Hinweise zur Überprüfung des Status der Aktualisierung von Datenbankparametern finden Sie unterrdsadmin.get_task_status.

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.update_db_param`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Beispiel 1: Einen Parameter aktualisieren

Im folgenden Beispiel `archretrydelay` wird der Parameter `100` für eine Datenbank mit dem Namen auf aktualisiert`TESTDB`:

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'archretrydelay',
    '100')"
```

Beispiel 2: Die Validierung von Objekten verzögern

Im folgenden Beispiel wird die Validierung von erstellten Objekten in einer Datenbank verschoben, die aufgerufen wird, um eine Überprüfung der Abhängigkeiten TESTDB zu vermeiden:

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'auto_reval',
    'deferred_force')"
```

rdsadmin.set_configuration

Konfiguriert spezifische Einstellungen für die Datenbank.

Syntax

```
db2 "call rdsadmin.set_configuration(
    'name',
    'value')"
```

Parameter

Die folgenden Parameter sind erforderlich:

name

Der Name der Konfigurationseinstellung. Der Datentyp istvarchar.

value

Der Wert für die Konfigurationseinstellung. Der Datentyp istvarchar.

Nutzungshinweise

Die folgende Tabelle zeigt die Konfigurationseinstellungen, mit denen Sie steuern könnenrdsadmin.set_configuration.

Name	Beschreibung
RESTORE_DATABASE_NUM_BUFFERS	Die Anzahl der Puffer, die während eines Wiederherstellungsvorgangs erstellt werden sollen. Dieser Wert muss kleiner als die

Name	Beschreibung
	Gesamtspeichergröße der DB-Instance-Klasse sein. Wenn diese Einstellung nicht konfiguriert ist, bestimmt Db2 den Wert, der während des Wiederherstellungsvorgangs verwendet werden soll. Weitere Informationen finden Sie in der IBM Db2-Dokumentation.
`RESTORE_DATABASE_PARALLELISM`	Die Anzahl der Puffermanipulatoren, die während eines Wiederherstellungsvorgangs erstellt werden sollen. Dieser Wert muss kleiner als das Doppelte der Zahl von v CPUs für die DB-Instance sein. Wenn diese Einstellung nicht konfiguriert ist, bestimmt Db2 den Wert, der während des Wiederherstellungsvorgangs verwendet werden soll. Weitere Informationen finden Sie in der IBM Db2-Dokumentation.

Beispiele

Beispiel 1: Angeben der Anzahl der zu erstellenden Puffermanipulatoren

Im folgenden Beispiel wird die `RESTORE_DATABASE_PARALLELISM` Konfiguration auf festgelegt. 8

```
db2 "call rdsadmin.set_configuration(
    'RESTORE_DATABASE_PARALLELISM',
    '8')"
```

Beispiel 2: Angeben der Anzahl der zu erstellenden Puffer

Im folgenden Beispiel wird die `RESTORE_DATABASE_NUM_BUFFERS` Konfiguration auf `150` festgelegt.

```
db2 "call rdsadmin.set_configuration(
    'RESTORE_DATABASE_NUM_BUFFERS',
    '150')"
```

rdsadmin.show_configuration

Gibt die aktuellen Einstellungen zurück, die Sie mithilfe der gespeicherten Prozedur festlegen können`rdsadmin.set_configuration`.

Syntax

```
db2 "call rdsadmin.show_configuration(
    'name')"
```

Parameter

Der folgende Parameter ist optional:

name

Der Name der Konfigurationseinstellung, über die Informationen zurückgegeben werden sollen. Der Datentyp ist`varchar`.

Die folgenden Konfigurationsnamen sind gültig:

- RESTORE_DATABASE_NUM_BUFFERS — Die Anzahl der Puffer, die während eines Wiederherstellungsvorgangs erstellt werden sollen.
- RESTORE_DATABASE_PARALLELISM — Die Anzahl der Puffermanipulatoren, die während eines Wiederherstellungsvorgangs erstellt werden sollen.

Nutzungshinweise

Wenn Sie den Namen einer Konfigurationseinstellung nicht angeben, werden Informationen für alle Konfigurationseinstellungen `rdsadmin.show_configuration` zurückgegeben, die Sie mithilfe der gespeicherten Prozedur festlegen können. `rdsadmin.set_configuration`

Beispiele

Im folgenden Beispiel werden Informationen zur aktuellen `RESTORE_DATABASE_PARALLELISM` Konfiguration zurückgegeben.

```
db2 "call rdsadmin.show_configuration(
    'RESTORE_DATABASE_PARALLELISM')"
```

rdsadmin.restore_database

Stellt eine Datenbank aus einem Amazon S3 S3-Bucket auf Ihrer RDS for Db2-DB-Instance wieder her.

Syntax

```
db2 "call rdsadmin.restore_database(
    ?,
    'database_name',
    's3_bucket_name',
    's3_prefix',
    restore_timestamp,
    'backup_type')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Zieldatenbank, die in RDS for Db2 wiederhergestellt werden soll. Der Datentyp ist`varchar`.

Wenn der Quelldatenbankname beispielsweise lautete `TESTDB` und Sie *database_name* auf eingestellt habe`NEWDB`, stellt Amazon RDS `NEWDB` als Quelldatenbank wieder her.

s3_bucket_name

Der Name des Amazon S3 S3-Buckets, in dem sich Ihr Backup befindet. Der Datentyp ist`varchar`.

s3_prefix

Das Präfix, das für den Dateiabgleich beim Herunterladen verwendet werden soll. Der Datentyp ist`varchar`.

Wenn dieser Parameter leer ist, werden alle Dateien im Amazon S3 S3-Bucket heruntergeladen. Das Folgende ist ein Beispiel für ein Präfix:

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

restore_timestamp

Der Zeitstempel des Datenbank-Backup-Images. Der Datentyp ist`varchar`.

Der Zeitstempel ist im Namen der Sicherungsdatei enthalten. `20230615010101`Ist zum Beispiel der Zeitstempel für den Dateinamen. `SAMPLE.0.rdsdb.DBPART000.20230615010101.001`

backup_type

Die Art der Sicherung. Der Datentyp ist`varchar`. Zulässige Werte: `OFFLINE`, `ONLINE`.

Wird `ONLINE` für Migrationen verwendet, bei denen fast keine Ausfallzeiten auftreten. Weitere Informationen finden Sie unter Migration von Linux to Linux mit nahezu null Ausfallzeiten für Amazon for Db2 RDS.

Nutzungshinweise

Sie können diese gespeicherte Prozedur verwenden, um eine Db2-Datenbank auf eine RDS for Db2-DB-Instance zu migrieren. Weitere Informationen finden Sie unter Nutzung von AWS Services zur Migration von Daten von Db2 zu Amazon RDS for Db2. Sie können diese gespeicherte Prozedur auch verwenden, um mehrere Kopien derselben Datenbank mit unterschiedlichen Datenbanknamen zu erstellen, die dasselbe Wiederherstellungsabbild verwenden.

Bevor Sie die gespeicherte Prozedur aufrufen, sollten Sie sich die folgenden Überlegungen ansehen:

- Bevor Sie eine Datenbank wiederherstellen, müssen Sie Speicherplatz für Ihre RDS for Db2-DB-Instance bereitstellen, der mindestens der Summe der Größe Ihres Backups und der ursprünglichen Db2-Datenbank auf der Festplatte entspricht. Weitere Informationen finden Sie unter Insufficient disk space.
- Wenn Sie das Backup wiederherstellen, extrahiert Amazon RDS die Sicherungsdatei auf Ihrer RDS for Db2-DB-Instance. Jede Backup-Datei muss 5 TB oder kleiner sein. Wenn eine Sicherungsdatei größer als 5 TB ist, müssen Sie die Sicherungsdatei in kleinere Dateien aufteilen.
- Um alle Dateien mithilfe der `rdsadmin.restore_database` gespeicherten Prozedur wiederherzustellen, geben Sie das Dateinummersuffix nach dem Zeitstempel nicht in die Dateinamen ein. Das stellt beispielsweise die folgenden *s3_prefix* `backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101` Dateien wieder her:

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.001
SAMPLE.0.rdsdb.DBPART000.20230615010101.002
```

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.003
SAMPLE.0.rdsdb.DBPART000.20230615010101.004
SAMPLE.0.rdsdb.DBPART000.20230615010101.005
```

- RDS for Db2 unterstützt keine nichtautomatische Speicherung. Weitere Informationen finden Sie unter Tablespaces not restored.
- RDS für Db2 unterstützt keine Routinen ohne Fencing. Weitere Informationen finden Sie unter Non-fenced routines not allowed.
- Um die Leistung von Datenbankwiederherstellungsvorgängen zu verbessern, können Sie die Anzahl der von RDS zu verwendenden Puffer und Puffermanipulatoren konfigurieren. Um die aktuelle Konfiguration zu überprüfen, verwenden Sie. the section called “rdsadmin.show_configuration” Um die Konfiguration zu ändern, verwenden Siethe section called “rdsadmin.set_configuration”.

Informationen zum Onlineschalten der Datenbank und zum Anwenden zusätzlicher Transaktionslogs nach dem Wiederherstellen der Datenbank finden Sie unterrdsadmin.rollforward_database.

Hinweise zur Überprüfung des Status der Datenbankwiederherstellung finden Sie unterrdsadmin.get_task_status.

Informationen zu Fehlermeldungen, die beim Aufrufen zurückgegeben werden`rdsadmin.restore_database`, finden Sie unterthe section called “Fehler bei gespeicherten Prozeduren”.

Beispiele

Das folgende Beispiel stellt eine Offline-Sicherung mit einer einzelnen Datei oder mehreren Dateien wieder her, die Folgendes aufweisen *`s3_prefix`*`backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101`:

```
db2 "call rdsadmin.restore_database(
    ?,
    'SAMPLE',
    'amzn-s3-demo-bucket',
    'backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101',
    20230615010101,
    'OFFLINE')"
```

rdsadmin.rollforward_database

Bringt die Datenbank online und wendet zusätzliche Transaktionsprotokolle an, nachdem eine Datenbank durch einen Aufruf wiederhergestellt wurderdsadmin.restore_database.

Syntax

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'database_name',
    's3_bucket_name',
    s3_prefix,
    'rollforward_to_option',
    'complete_rollforward')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Datenbank, für die der Vorgang ausgeführt werden soll. Der Datentyp istvarchar.

s3_bucket_name

Der Name des Amazon S3 S3-Buckets, in dem sich Ihr Backup befindet. Der Datentyp istvarchar.

s3_prefix

Das Präfix, das für den Dateiabgleich beim Herunterladen verwendet werden soll. Der Datentyp istvarchar.

Wenn dieser Parameter leer ist, werden alle Dateien im S3-Bucket heruntergeladen. Das folgende Beispiel ist ein Beispielpräfix:

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

Die folgenden Eingabeparameter sind optional:

rollforward_to_option

Der Punkt, zu dem Sie weiterrollen möchten. Der Datentyp istvarchar. Gültige Werte:END_OF_LOGS, END_OF_BACKUP mit dem Zeitstempel im FormatYYYY-MM-DD-HH.MM.SS. Der Standardwert ist END_OF_LOGS.

complete_rollforward

Gibt an, ob der Roll-Forward-Prozess abgeschlossen werden soll. Der Datentyp ist. varchar Der Standardwert ist TRUE.

WennTRUE, dann ist die Datenbank nach Abschluss online und zugänglich. WennFALSE, dann bleibt die Datenbank in einem ROLL-FORWARD PENDING Zustand.

Nutzungshinweise

Sie können es rds.rollforward_database für eine Online-Sicherung mit Include-Protokollen verwenden, die in vielen verschiedenen Szenarien vor Ort erstellt werden.

Szenario 1: Wiederherstellung der Datenbank, Rollforward der enthaltenen Protokolle und Online-Aktivierung der Datenbank

Verwenden Sie nach rdsadmin.restore_database() Abschluss des Vorgangs die Syntax in Beispiel 1, um die Datenbank mit den Transaktionsprotokollen online zu schalten.

Szenario 2: Die Datenbank wird online geschaltet, aber die enthaltenen Protokolle werden nicht weitergeleitet.

Verwenden Sie nach rdsadmin.restore_database() Abschluss des Vorgangs die Syntax in Beispiel 2, um die Datenbank ohne die Transaktionsprotokolle online zu schalten.

Szenario 3: Weiterleitung der in der Sicherung enthaltenen Protokolle und Anwenden zusätzlicher Transaktionsprotokolle, wenn sie vor Ort erstellt werden

Verwenden Sie nach rdsadmin.restore_database() Abschluss des Vorgangs die Syntax in Beispiel 3 oder Beispiel 4, um Protokolle weiterzuleiten, ohne die Datenbank online zu schalten.

Wenn Sie complete_rollforward auf einstellenFALSE, befindet sich Ihre Datenbank in einem ROLL-FORWARD PENDING Zustand und ist offline. Um die Datenbank online zu schalten, müssen Sie anrufenrdsadmin.complete_rollforward.

Hinweise zur Überprüfung des Status beim Rollforward der Datenbank finden Sie unterrdsadmin.rollforward_status.

Beispiele

Beispiel 1: Die Datenbank mit Transaktionsprotokollen online schalten

Im folgenden Beispiel wird ein Rollforward zu einer Online-Sicherung der Datenbank mit Transaktionsprotokollen durchgeführt und die Datenbank anschließend online geschaltet:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    null,
    null,
    'END_OF_LOGS',
    'TRUE')"
```

Beispiel 2: Eine Datenbank ohne Transaktionsprotokolle online schalten

Im folgenden Beispiel wird ein Rollforward zu einer Online-Sicherung der Datenbank ohne Transaktionsprotokolle durchgeführt und die Datenbank anschließend online geschaltet:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'amzn-s3-demo-bucket',
    'logsfolder/,
    'END_OF_BACKUP',
    'TRUE')"
```

Beispiel 3: Datenbank mit Transaktionsprotokollen wird nicht online geschaltet

Im folgenden Beispiel wird ein Rollforward zu einer Online-Sicherung der Datenbank mit Transaktionsprotokollen durchgeführt, ohne dass die Datenbank online geschaltet wird:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    null,
    'onlinebackup/TESTDB',
    'END_OF_LOGS',
    'FALSE')"
```

Beispiel 4: Die Datenbank mit zusätzlichen Transaktionsprotokollen wird nicht online geschaltet

Im folgenden Beispiel wird ein Rollforward zu einer Online-Sicherung der Datenbank mit zusätzlichen Transaktionsprotokollen durchgeführt, ohne dass die Datenbank online geschaltet wird:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'amzn-s3-demo-bucket',
    'logsfolder/S0000155.LOG',
    'END_OF_LOGS',
    'FALSE')"
```

rdsadmin.rollforward_status

Gibt die Ausgabe von zurückROLLFORWARD DATABASE *database_name* QUERY STATUS.

Syntax

```
db2 "call rdsadmin.rollforward_status(
    ?,
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der Datenbank, für die der Vorgang ausgeführt werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Nach dem Anruf können Sie anrufenrdsadmin.rollforward_database, `rdsadmin.rollforward_status` um den Status des Rollforward in der Datenbank zu überprüfen.

Hinweise zur Überprüfung des Status dieser gespeicherten Prozedur finden Sie unterrdsadmin.get_task_status.

rdsadmin.complete_rollforward

Bringt die Datenbank aus einem `ROLL-FORWARD PENDING` Status online.

Syntax

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der Datenbank, die Sie online stellen möchten. Der Datentyp ist`varchar`.

Nutzungshinweise

Wenn Sie rdsadmin.rollforward_database mit der `complete_rollforward` Einstellung auf aufgerufen haben`FALSE`, befindet sich Ihre Datenbank in einem `ROLL-FORWARD PENDING` Status und ist offline. Rufen Sie an, um den Roll-Forward-Vorgang abzuschließen und die Datenbank online zu schalten. `rdsadmin.complete_rollforward`

Hinweise zur Überprüfung des Status des Abschlusses des Rollforward-Vorgangs finden Sie unter. rdsadmin.rollforward_status

Beispiele

Im folgenden Beispiel wird die TESTDB Datenbank online geschaltet:

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'TESTDB')"
```

rdsadmin.db2pd_command

Sammelt Informationen über eine RDS for Db2-Datenbank.

Syntax

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

Parameter

Der folgende Eingabeparameter ist erforderlich:

db2pd_cmd

Der Name des db2pd Befehls, den Sie ausführen möchten. Der Datentyp ist`varchar`.

Der Parameter muss mit einem Bindestrich beginnen. Eine Liste der Parameter finden Sie in der Dokumentation unter db2pd — Db2-Datenbankbefehl überwachen und beheben. IBM Db2

Die folgenden Optionen werden nicht unterstützt:

- `-addnode`
- `-alldatabases`
- `-alldbp`
- `-alldbs`
- `-allmembers`
- `-alm_in_memory`
- `-cfinfo`
- `-cfpool`
- `-command`
- `-dbpartitionnum`
- `-debug`

- `-dump`
- `-everything`
- `-file | -o`
- `-ha`
- `-interactive`
- `-member`
- `-pages`

  > **Note**
  >
  > `-pages summary` wird unterstützt.

- `-pdcollection`
- `-repeat`
- `-stack`
- `-totalmem`

Die `file` Unteroption wird beispielsweise nicht unterstützt. `db2pd -db testdb -tcbstats file=tcbstat.out`

Die Verwendung der `stacks` Option wird beispielsweise nicht unterstützt. `db2pd -edus interval=5 top=10 stacks`

Nutzungshinweise

Diese gespeicherte Prozedur sammelt Informationen, die bei der Überwachung und Problembehandlung von RDS for Db2-Datenbanken helfen können.

Die gespeicherte Prozedur verwendet das IBM db2pd Hilfsprogramm, um verschiedene Befehle auszuführen. Das db2pd Hilfsprogramm erfordert eine SYSADM Autorisierung, über die der RDS for Db2-Masterbenutzer nicht verfügt. Mit der gespeicherten Amazon RDS-Prozedur kann der Masterbenutzer das Hilfsprogramm jedoch verwenden, um verschiedene Befehle auszuführen. Weitere Informationen über das Hilfsprogramm finden Sie in der Dokumentation unter db2pd — Db2-Datenbankbefehl überwachen und beheben. IBM Db2

Die Ausgabe ist auf maximal 2 GB beschränkt.

Hinweise zur Überprüfung des Status der Erfassung von Informationen über die Datenbank finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Rückgabe der Verfügbarkeit der DB-Instance

Das folgende Beispiel gibt die Verfügbarkeit einer RDS für eine DB2-DB-Instance zurück:

```
db2 "call rdsadmin.db2pd_command('-')"
```

Beispiel 2: Rückgabe der Verfügbarkeit der Datenbank

Das folgende Beispiel gibt die Verfügbarkeit einer Datenbank mit dem Namen zurück: TESTDB

```
db2 "call rdsadmin.db2pd_command('-db TESTDB -')"
```

Beispiel 3: Rückgabe der Speichernutzung einer DB-Instance

Das folgende Beispiel gibt die Speichernutzung einer RDS für eine DB2-DB-Instance zurück:

```
db2 "call rdsadmin.db2pd_command('-dbptnmem')"
```

Beispiel 4: Rückgabe von Speichersätzen von DB-Instance und Datenbank

Das folgende Beispiel gibt die Speichersätze einer RDS for Db2-DB-Instance und einer Datenbank namens TESTDB zurück:

```
db2 "call rdsadmin.db2pd_command('-inst -db TESTDB -memsets')"
```

rdsadmin.force_application

Zwingt Anwendungen aus einer RDS for Db2-Datenbank.

Syntax

```
db2 "call rdsadmin.force_application(
    ?,
    'applications')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

applications

Die Anwendungen, die Sie aus einer RDS for Db2-Datenbank ausschalten möchten. Der Datentyp ist`varchar`. Gültige Werte: `ALL` oder *application_handle*.

Trennen Sie die Namen mehrerer Anwendungen durch Kommas. Beispiel: '*application_handle_1*,*application_handle_2*'.

Nutzungshinweise

Diese gespeicherte Prozedur zwingt alle Anwendungen aus einer Datenbank, sodass Sie Wartungsarbeiten durchführen können.

Die gespeicherte Prozedur verwendet den IBM `FORCE APPLICATION` Befehl. Für den `FORCE APPLICATION` Befehl ist eine `SYSCTRL` Autorisierung oder Autorisierung erforderlich `SYSADMSYSMAINT`, über die der RDS for Db2-Masterbenutzer nicht verfügt. Mit der gespeicherten Amazon RDS-Prozedur kann der Masterbenutzer den Befehl jedoch verwenden. Weitere Informationen finden Sie in der IBM Db2 Dokumentation unter dem Befehl FORCE APPLICATION.

Hinweise zur Überprüfung des Status beim Abschalten von Anwendungen aus einer Datenbank finden Sie unterrdsadmin.get_task_status.

Beispiele

Beispiel 1: Alle Anwendungen angeben

Im folgenden Beispiel werden alle Anwendungen aus einer RDS for Db2-Datenbank entfernt:

```
db2 "call rdsadmin.force_application(
    ?,
```

```
    'ALL')"
```

Beispiel 2: Angabe mehrerer Anwendungen

Im folgenden Beispiel werden Anwendungs-Handles 99918891, und 1192 aus einer RDS for Db2-Datenbank deaktiviert:

```
db2 "call rdsadmin.force_application(
    ?,
    '9991, 8891, 1192')"
```

rdsadmin.set_archive_log_retention

Konfiguriert den Zeitraum (in Stunden) für die Aufbewahrung von Archivprotokolldateien für die angegebene RDS for Db2-Datenbank.

Syntax

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'database_name',
    'archive_log_retention_hours')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Die folgenden Eingabeparameter sind erforderlich:

database_name

Der Name der Datenbank, für die die Aufbewahrung von Archivprotokollen konfiguriert werden soll. Der Datentyp ist`varchar`.

archive_log_retention_hours

Die Anzahl der Stunden, für die die Archiv-Protokolldateien aufbewahrt werden sollen. Der Datentyp ist`smallint`. Die Standardeinstellung ist`0`, und das Maximum ist 168 (7 Tage).

Wenn der Wert ist`0`, speichert Amazon RDS die Archiv-Protokolldateien nicht.

Nutzungshinweise

Standardmäßig speichert RDS for Db2 Protokolle für 5 Minuten. Wenn Sie Replikationstools wie AWS DMS Change Data Capture (CDC) oder verwenden, empfehlen wirIBM Q Replication, die Aufbewahrung von Protokollen in diesen Tools für mehr als 5 Minuten festzulegen.

Sie können die aktuellen Einstellungen für die Aufbewahrung von Archivprotokollen einsehen, indem Sie anrufenthe section called “rdsadmin.show_archive_log_retention”.

Sie können die Einstellung für die Aufbewahrung von Archivprotokollen in der `rdsadmin` Datenbank nicht konfigurieren.

Beispiele

Beispiel 1: Einstellung der Aufbewahrungszeit

Im folgenden Beispiel wird die Aufbewahrungszeit des Archivprotokolls für eine aufgerufene Datenbank `TESTDB` auf 24 Stunden festgelegt.

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '24')"
```

Beispiel 2: Deaktivieren der Aufbewahrungszeit

Im folgenden Beispiel wird die Aufbewahrung von Archivprotokollen für eine Datenbank mit dem Namen deaktiviert. `TESTDB`

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '0')"
```

rdsadmin.show_archive_log_retention

Gibt die aktuelle Aufbewahrungseinstellung für das Archivprotokoll für die angegebene Datenbank zurück.

Syntax

```
db2 "call rdsadmin.show_archive_log_retention(
    ?,
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der Datenbank, für die die Einstellung zur Aufbewahrung von Archivprotokollen angezeigt werden soll. Der Datentyp istvarchar.

Beispiele

Das folgende Beispiel zeigt die Einstellung zur Aufbewahrung von Archivprotokollen für eine Datenbank namensTESTDB.

```
db2 "call rdsadmin.show_archive_log_retention(
    ?
    'TESTDB')"
```

rdsadmin.list_archive_log_information

Gibt Details zu den Archivprotokolldateien zurück, z. B. Größe, Datum und Uhrzeit der Erstellung sowie den Namen der einzelnen Protokolldateien für die angegebene Datenbank. Außerdem wird die Gesamtspeichermenge zurückgegeben, die von den Protokolldateien in der Datenbank belegt wird.

Syntax

```
db2 "call rdsadmin.list_archive_log_information(
    ?,
```

```
    'database_name')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

Der folgende Eingabeparameter ist erforderlich:

database_name

Der Name der Datenbank, für die Archivprotokollinformationen aufgelistet werden sollen. Der Datentyp istvarchar.

Beispiele

Das folgende Beispiel gibt Archivprotokollinformationen für eine Datenbank mit dem Namen zurückTESTDB.

```
db2 "call rdsadmin.list_archive_log_information(
    ?
    'TESTDB')"
```

Gespeicherte Prozeduren für den Speicherzugriff RDS für Db2

Die in diesem Thema beschriebenen integrierten gespeicherten Prozeduren verwalten den Speicherzugriff RDS für Db2-Datenbanken, die Amazon S3 für die Datenmigration verwenden. Weitere Informationen finden Sie unter the section called “Migration mit Amazon S3”.

Informationen zu deren Syntax, Parametern, Nutzungshinweisen und Beispielen finden Sie in den folgenden integrierten gespeicherten Prozeduren.

Gespeicherte Prozeduren

- rdsadmin.catalog_storage_access
- rdsadmin.uncatalog_storage_access

rdsadmin.catalog_storage_access

Katalogisiert einen Speicheralias für den Zugriff auf einen Amazon S3 S3-Bucket mit Db2-Datendateien.

Syntax

```
db2 "call rdsadmin.catalog_storage_access(
    ?,
    'alias',
    's3_bucket_name',
    'grantee_type',
    'grantee'
    )"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Der Datentyp ist. `varchar`

Die folgenden Eingabeparameter sind erforderlich:

alias

Der Aliasname für den Zugriff auf Remotespeicher in einem Amazon S3 S3-Bucket. Der Datentyp ist. `varchar`

s3_bucket_name

Der Name des Amazon S3 S3-Buckets, in dem sich Ihre Daten befinden. Der Datentyp ist`varchar`.

grantee_type

Der Typ des Empfängers, der eine Autorisierung erhalten soll. Der Datentyp ist`varchar`. Zulässige Werte: `USER`, `GROUP`.

grantee

Der Benutzer oder die Gruppe, der die Autorisierung erhalten soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Amazon nimmt RDS den katalogisierten Alias in die IAM Rolle auf, die Sie Ihrer RDS for Db2-DB-Instance hinzugefügt haben. Wenn Sie die IAM Rolle aus Ihrer DB-Instance entfernen, RDS löscht Amazon den Alias. Weitere Informationen finden Sie unter the section called “Migration mit Amazon S3”.

Informationen zur Überprüfung des Status der Katalogisierung Ihres Alias finden Sie unter. rdsadmin.get_task_status

Beispiele

Im folgenden Beispiel wird ein Alias namens `SAMPLE` registriert. Dem Benutzer `jorge_souza` wird Zugriff auf den aufgerufenen Amazon S3 S3-Bucket gewährt`amzn-s3-demo-bucket`.

```
db2 "call rdsadmin.catalog_storage_access(
    ?,
    'SAMPLE',
    'amzn-s3-demo-bucket',
    'USER',
    'jorge_souza')"
```

rdsadmin.uncatalog_storage_access

Entfernt einen Alias für den Speicherzugriff.

Syntax

```
db2 "call rdsadmin.uncatalog_storage_access(
    ?,
    'alias')"
```

Parameter

Der folgende Ausgabeparameter ist erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Der Datentyp ist. `varchar`

Der folgende Eingabeparameter ist erforderlich:

alias

Der Name des Speicheralias, der entfernt werden soll. Der Datentyp ist. `varchar`

Nutzungshinweise

Hinweise zur Überprüfung des Status der Entfernung Ihres Alias finden Sie unter. rdsadmin.get_task_status

Beispiele

Im folgenden Beispiel wird ein Alias namens entferntSAMPLE. Dieser Alias bietet keinen Zugriff mehr auf den Amazon S3 S3-Bucket, mit dem er verknüpft war.

```
db2 "call rdsadmin.uncatalog_storage_access(
    ?,
    'SAMPLE')"
```

Gespeicherte Prozeduren für Tablespaces für RDS for Db2

Die in diesem Thema beschriebenen integrierten gespeicherten Prozeduren verwalten Tablespaces für Amazon RDS for Db2-Datenbanken. Um diese Prozeduren auszuführen, muss der Masterbenutzer zuerst eine Verbindung zur Datenbank herstellen. `rdsadmin`

Diese gespeicherten Prozeduren werden für eine Vielzahl von Aufgaben verwendet. Diese Liste ist nicht umfassend.

- Allgemeine Aufgaben für Tablespaces
- Generieren von Leistungsberichten
- Datenbank-Metadaten mit db2look kopieren
- Erstellen einer Repository-Datenbank für IBM Db2 Data Management Console

Informationen zu deren Syntax, Parametern, Verwendungshinweisen und Beispielen finden Sie in den folgenden integrierten gespeicherten Prozeduren.

Gespeicherte Prozeduren

- rdsadmin.create_tablespace
- rdsadmin.alter_tablespace
- rdsadmin.rename_tablespace
- rdsadmin.drop_tablespace

rdsadmin.create_tablespace

Erzeugt einen Tablespace.

Syntax

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_page_size,
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type',
```

```
    'tablespace_prefetch_size')"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, in der der Tablespace erstellt werden soll. Der Datentyp ist`varchar`.

tablespace_name

Der Name des zu erstellenden Tablespaces. Der Datentyp ist`varchar`.

Für den Tablespace-Namen gelten die folgenden Einschränkungen:

- Er darf nicht mit dem Namen eines vorhandenen Tablespaces in dieser Datenbank identisch sein.
- Er kann nur die Zeichen `_$#@a-zA-Z0-9` enthalten.
- Es kann nicht mit _ oder beginnen$.
- Es kann nicht beginnen mitSYS.

Die folgenden Parameter sind optional:

buffer_pool_name

Der Name des Pufferpools, dem der Tablespace zugewiesen werden soll. Der Datentyp ist`varchar`. Der Standardwert ist eine leere Zeichenfolge.

> **Important**
>
> Sie müssen bereits über einen Pufferpool mit derselben Seitengröße verfügen, der dem Tablespace zugeordnet werden kann.

tablespace_page_size

Die Seitengröße des Tablespaces in Byte. Der Datentyp ist`integer`. Zulässige Werte: `4096`, `8192`, `16384`, `32768`. Die Standardeinstellung ist die Seitengröße, die verwendet wurde, als Sie die Datenbank durch Aufrufen erstellt habenrdsadmin.create_database.

> **Important**
>
> Amazon RDS unterstützt Schreibatomizität für Seiten mit 4 KiB, 8 KiB und 16 KiB. Im Gegensatz dazu besteht bei Seiten mit 32 KiB die Gefahr, dass Schreibvorgänge abgebrochen werden oder dass unvollständige Daten auf den Schreibtisch geschrieben werden. Wenn Sie Seiten mit 32 KiB verwenden, empfehlen wir Ihnen, die point-in-time Wiederherstellung und automatische Backups zu aktivieren. Andernfalls laufen Sie Gefahr, dass Sie zerrissene Seiten nicht wiederherstellen können. Weitere Informationen erhalten Sie unter the section called “Einführung in Backups” und the section called “Point-in-time Wiederherstellung”.

tablespace_initial_size

Die ursprüngliche Größe des Tablespace in Kilobyte (KB). Der Datentyp ist. `integer` Gültige Werte: 48 oder höher. Der Standardwert ist „null“.

Wenn Sie keinen Wert angeben, legt Db2 einen geeigneten Wert für Sie fest.

> **Note**
>
> Dieser Parameter gilt nicht für temporäre Tablespaces, da das System temporäre Tablespaces verwaltet.

tablespace_increase_size

Der Prozentsatz, um den der Tablespace erhöht werden soll, wenn er voll ist. Der Datentyp ist`integer`. Gültige Werte: 1 —`100`. Der Standardwert ist „null“.

Wenn Sie keinen Wert angeben, legt Db2 einen geeigneten Wert für Sie fest.

> **Note**
>
> Dieser Parameter gilt nicht für temporäre Tablespaces, da das System temporäre Tablespaces verwaltet.

tablespace_type

Der Typ des Tablespaces. Der Datentyp ist`char`. Gültige Werte: `U` (für Benutzerdaten), `T` (für temporäre Benutzerdaten) oder `S` (für temporäre Systemdaten). Der Standardwert ist `U`.

tablespace_prefetch_size

Die Prefetch-Seitengröße des Tablespaces. Der Datentyp ist. `char` Gültige Werte: `AUTOMATIC` (Groß- und Kleinschreibung wird nicht berücksichtigt) oder positive Ganzzahlen ungleich Null, die kleiner oder gleich 32767 sind.

Nutzungshinweise

RDS for Db2 erstellt immer eine große Datenbank für Daten.

Hinweise zur Überprüfung des Status beim Erstellen eines Tablespaces finden Sie unter. rdsadmin.get_task_status

Beispiele

Beispiel 1: Einen Tablespace erstellen und einen Pufferpool zuweisen

Das folgende Beispiel erstellt einen Tablespace namens `SP8` und weist einen Pufferpool zu, der `BP8` für eine Datenbank namens aufgerufen wird. `TESTDB` Der Tablespace hat eine anfängliche Tablespace-Seitengröße von 4.096 Byte, einen anfänglichen Tablespace von 1.000 KB und eine Erhöhung der Tabellengröße ist auf 50% festgelegt.

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    4096,
    1000,
    50)"
```

Beispiel 2: Einen temporären Tablespace erstellen und einen Pufferpool zuweisen

Das folgende Beispiel erstellt einen temporären Tablespace namens. `SP8` Es weist einer `BP8` aufgerufenen Datenbank einen Pufferpool mit einer Größe von 8 KiB zu. `TESTDB`

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
```

```
    'SP8',
    'BP8',
    8192,
    NULL,
    NULL,
    'T')"
```

Beispiel 3: Einen Tablespace erstellen und eine Prefetch-Seitengröße zuweisen

Das folgende Beispiel erstellt einen Tablespace, der für eine Datenbank namens aufgerufen wird`SP8`. `TESTDB` Der Tablespace hat eine anfängliche Tablespace-Größe von `50` und eine Prefetch-Seitengröße von. `800`

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    NULL,
    NULL,
    NULL,
    50,
    NULL,
    '800')"
```

rdsadmin.alter_tablespace

Ändert einen Tablespace.

Syntax

```
db2 "call rdsadmin.alter_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_increase_size,
    'max_size',
    'reduce_max',
    'reduce_stop',
    'reduce_value',
    'lower_high_water',
    'lower_high_water_stop',
    'switch_online',
    'tablespace_prefetch_size')"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, die den Tablespace verwendet. Der Datentyp ist`varchar`.

tablespace_name

Der Name des Tablespaces, der geändert werden soll. Der Datentyp ist`varchar`.

Die folgenden Parameter sind optional:

buffer_pool_name

Der Name des Pufferpools, dem der Tablespace zugewiesen werden soll. Der Datentyp ist`varchar`. Der Standardwert ist eine leere Zeichenfolge.

> **Important**
>
> Sie müssen bereits über einen Pufferpool mit derselben Seitengröße verfügen, der dem Tablespace zugeordnet werden kann.

tablespace_increase_size

Der Prozentsatz, um den der Tablespace vergrößert werden soll, wenn er voll ist. Der Datentyp ist`integer`. Gültige Werte: `1` —`100`. Der Standardwert ist `0`.

max_size

Die maximale Größe für den Tablespace. Der Datentyp ist`varchar`. Gültige Werte: *integer* `K` | `M` |`G`, oder`NONE`. Der Standardwert ist `NONE`.

reduce_max

Gibt an, ob der Höchstwert auf den Höchstwert reduziert werden soll. Der Datentyp ist`char`. Der Standardwert ist `N`.

reduce_stop

Gibt an, ob ein vorhergehender `reduce_value` Befehl `reduce_max` oder ein Befehl unterbrochen werden soll. Der Datentyp ist`char`. Der Standardwert ist `N`.

reduce_value

Die Zahl oder der Prozentsatz, um den die Höchstwassermarke im Tablespace reduziert werden soll. Der Datentyp istvarchar. Gültige Werte: *integer* K M | | G oder 1 —100. Der Standardwert ist N.

lower_high_water

Gibt an, ob der ALTER TABLESPACE LOWER HIGH WATER MARK Befehl ausgeführt werden soll. Der Datentyp istchar. Der Standardwert ist N.

lower_high_water_stop

Gibt an, ob der ALTER TABLESPACE LOWER HIGH WATER MARK STOP Befehl ausgeführt werden soll. Der Datentyp istchar. Der Standardwert ist N.

switch_online

Gibt an, ob der ALTER TABLESPACE SWITCH ONLINE Befehl ausgeführt werden soll. Der Datentyp istchar. Der Standardwert ist N.

tablespace_prefetch_size

Die Prefetch-Seitengröße des Tablespaces. Der Datentyp ist. char Gültige Werte: AUTOMATIC (Groß- und Kleinschreibung wird nicht berücksichtigt) oder positive Ganzzahlen ungleich Null, die kleiner oder gleich 32767 sind.

> **Note**
>
> Dieser Parameter funktioniert nur mitbuffer_pool_name,, und. table_increase_size max_size switch_online Er funktioniert nicht mitreduce_max,reduce_stop, reduce_valuelower_high_water, undlower_high_water_stop.

Nutzungshinweise

Bevor Sie die gespeicherte Prozedur aufrufen, sollten Sie sich die folgenden Überlegungen ansehen:

- Die rdsadmin.alter_tablespace gespeicherte Prozedur funktioniert nicht in einem Tablespace, in dem die tablespace_type Einstellung T für temporäre Benutzerdaten auf gesetzt ist.

- Die optionalen Parameter`reduce_max`,`reduce_stop`, `reduce_value` `lower_high_water`lower_high_water_stop, und schließen `switch_online` sich gegenseitig aus. Sie können sie nicht mit anderen optionalen Parametern kombinierenbuffer_pool_name, z. B. im `rdsadmin.alter_tablespace` Befehl. Weitere Informationen finden Sie unter Statement not valid.

Hinweise zur Überprüfung des Status der Änderung eines Tablespaces finden Sie unter. rdsadmin.get_task_status

Fehlermeldungen, die beim Aufrufen von gespeicherten Prozeduren zurückgegeben werden, finden Sie unter. the section called “Fehler bei gespeicherten Prozeduren”

Beispiele

Beispiel 1: Senkung der Hochwassermarke

Im folgenden Beispiel wird ein Tablespace geändert, der aufgerufen wird, SP8 und es wird ein Pufferpool zugewiesen, der BP8 für eine Datenbank aufgerufen wird, die aufgerufen wirdTESTDB, um den Grenzwert zu senken.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    NULL,
    NULL,
    'Y')"
```

Beispiel 2: Reduzierung der Höchstwassermarke

Im folgenden Beispiel wird der REDUCE MAX Befehl in einem Tablespace ausgeführt, der TBSP_TEST in der Datenbank TESTDB aufgerufen wird.

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    'Y')"
```

Beispiel 3: Befehle werden unterbrochen, um den Wasserstand zu reduzieren

Im folgenden Beispiel wird der `REDUCE STOP` Befehl in einem Tablespace ausgeführt, der `TBSP_TEST` in der Datenbank aufgerufen wird. `TESTDB`

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    NULL,
    'Y')"
```

Beispiel 4: Ändern der vorhandenen Prefetch-Seitengröße

Im folgenden Beispiel wird der `ALTER TABLESPACE SWITCH ONLINE` Befehl für einen Tablespace ausgeführt, der aufgerufen wird, `TSBP_TEST` und die bestehende Prefetch-Seitengröße wird auf geändert. `64`

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    ‘Y’,
    ‘64’)"
```

rdsadmin.rename_tablespace

Benennt einen Tablespace um.

Syntax

```
db2 "call rdsadmin.rename_tablespace(
    ?,
```

```
    'database_name',
    'source_tablespace_name',
    'target_tablespace_name')"
```

Parameter

Die folgenden Parameter sind erforderlich:

?

Eine Parametermarkierung, die eine Fehlermeldung ausgibt. Dieser Parameter akzeptiert nur?.

database_name

Der Name der Datenbank, zu der der Tablespace gehört. Der Datentyp ist`varchar`.

source_tablespace_name

Der Name des Tablespaces, der umbenannt werden soll. Der Datentyp ist`varchar`.

target_tablespace_name

Der neue Name des Tablespace. Der Datentyp ist`varchar`.

Der neue Name hat die folgenden Einschränkungen:

- Er darf nicht mit dem Namen eines vorhandenen Tablespaces identisch sein.
- Er kann nur die Zeichen `_$#@a-zA-Z0-9` enthalten.
- Es kann nicht mit _ oder beginnen$.
- Es kann nicht beginnen mitSYS.

Nutzungshinweise

Hinweise zur Überprüfung des Status beim Umbenennen eines Tablespaces finden Sie unter. rdsadmin.get_task_status

Sie können Tablespaces, die zur Datenbank gehören, nicht umbenennen. `rdsadmin`

Beispiele

Im folgenden Beispiel wird ein aufgerufener Tablespace `SP9` in einer Datenbank mit `SP8` dem Namen umbenannt. `TESTDB`

```
db2 "call rdsadmin.rename_tablespace(
    ?,
    'TESTDB',
    'SP8',
    'SP9')"
```

rdsadmin.drop_tablespace

Löscht einen Tablespace.

Syntax

```
db2 "call rdsadmin.drop_tablespace(
    'database_name',
    'tablespace_name')"
```

Parameter

Die folgenden Parameter sind erforderlich:

database_name

Der Name der Datenbank, zu der der Tablespace gehört. Der Datentyp ist`varchar`.

tablespace_name

Der Name des Tablespaces, der gelöscht werden soll. Der Datentyp ist`varchar`.

Nutzungshinweise

Hinweise zur Überprüfung des Status beim Löschen eines Tablespaces finden Sie unterrdsadmin.get_task_status.

Beispiele

Im folgenden Beispiel wird ein Tablespace gelöscht, der SP8 aus einer Datenbank namens aufgerufen wurde. TESTDB

```
db2 "call rdsadmin.drop_tablespace(
    'TESTDB',
    'SP8')"
```

Referenz RDS für benutzerdefinierte Funktionen von Amazon for Db2

Die folgenden benutzerdefinierten Funktionen sind für RDS Amazon-DB-Instances verfügbar, auf denen die Db2-Engine ausgeführt wird.

Themen

- rdsadmin.get_task_status
- rdsadmin.list_databases

rdsadmin.get_task_status

Gibt den Status einer Aufgabe zurück.

Syntax

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(task_id,'database_name','task_type'))"
```

Parameter

Die folgenden Parameter sind optional. Wenn Sie keine Parameter angeben, gibt die benutzerdefinierte Funktion den Status aller Aufgaben für alle Datenbanken zurück. Amazon RDS speichert den Aufgabenverlauf 35 Tage lang.

task_id

Die ID der Aufgabe, die gerade ausgeführt wird. Diese ID wird zurückgegeben, wenn Sie eine Aufgabe ausführen. Standard: `0`.

database_name

Der Name der Datenbank, für die die Aufgabe ausgeführt wird.

task_type

Der Typ der abzufragenden Aufgabe. Gültige Werte: `ADD_GROUPSADD_USER,ALTER_BUFFERPOOL,ALTER_TABLESPACE,CHANGE_PASSWORD,COMPLETE_RO`

Nutzungshinweise

Sie können die `rdsadmin.get_task_status` benutzerdefinierte Funktion verwenden, um den Status der folgenden Aufgaben für Amazon RDS for Db2 zu überprüfen. Diese Liste ist nicht umfassend.

- Einen Pufferpool erstellen, ändern oder löschen
- Einen Tablespace erstellen, ändern oder löschen
- Eine Datenbank erstellen oder löschen
- Wiederherstellung eines Datenbank-Backups aus Amazon S3
- Datenbankprotokolle von Amazon S3 weiterleiten

Beispiele

Im folgenden Beispiel werden die Spalten angezeigt, die beim Aufruf `rdsadmin.get_task_status` zurückgegeben werden.

```
db2 "describe select * from table(rdsadmin.get_task_status())"
```

Im folgenden Beispiel wird der Status aller Aufgaben aufgeführt.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,null,null))"
```

Im folgenden Beispiel wird der Status einer bestimmten Aufgabe aufgeführt.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(1,null,null))"
```

Im folgenden Beispiel wird der Status einer bestimmten Aufgabe und Datenbank aufgeführt.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(2,'SAMPLE',null))"
```

Im folgenden Beispiel wird der Status aller ADD_GROUPS Aufgaben aufgeführt.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,null,'add_groups'))"
```

Im folgenden Beispiel wird der Status aller Aufgaben für eine bestimmte Datenbank aufgeführt.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,'testdb', null))"
```

Im folgenden Beispiel werden die JSON Werte als Spalten ausgegeben.

```
db2 "select varchar(r.task_type,25) as task_type, varchar(r.lifecycle,10) as lifecycle,
 r.created_at, u.* from
    table(rdsadmin.get_task_status(null,null,'restore_db')) as r,
 json_table(r.task_input_params, 'strict $' columns(s3_prefix varchar(500)
    null on empty, s3_bucket_name varchar(500) null on empty) error on error ) as U"
```

Antwort

Die `rdsadmin.get_task_status` benutzerdefinierte Funktion gibt die folgenden Spalten zurück:

`TASK_ID`

Die ID der Aufgabe.

`TASK_TYPE`

Hängt von den Eingabeparametern ab.

- `ADD_GROUPS`— Fügt Gruppen hinzu.
- `ADD_USER`— Fügt einen Benutzer hinzu.
- `ALTER_BUFFERPOOL`— Ändert einen Pufferpool.
- `ALTER_TABLESPACE`— Ändert einen Tablespace.
- `CHANGE_PASSWORD` — Ändert das Passwort eines Benutzers.
- `COMPLETE_ROLLFORWARD`— Schließt eine `rdsadmin.rollforward_database` Aufgabe ab und aktiviert eine Datenbank.

- CREATE_BUFFERPOOL— Erzeugt einen Pufferpool.
- CREATE_DATABASE— Erzeugt eine Datenbank.
- CREATE_ROLE— Erzeugt eine Db2-Rolle für einen Benutzer.
- CREATE_TABLESPACE— Erzeugt einen Tablespace.
- DROP_BUFFERPOOL— Löscht einen Pufferpool.
- DROP_DATABASE— Löscht eine Datenbank.
- DROP_TABLESPACE— Löscht einen Tablespace.
- LIST_USERS— Listet alle Benutzer auf.
- REMOVE_GROUPS— Entfernt Gruppen.
- REMOVE_USER— Löscht einen Benutzer.
- RESTORE_DB— Stellt eine vollständige Datenbank wieder her.
- ROLLFORWARD_DB_LOG— Führt eine `rdsadmin.rollforward_database` Aufgabe mit Datenbankprotokollen aus.
- ROLLFORWARD_STATUS — Gibt den Status einer `rdsadmin.rollforward_database` Aufgabe zurück.
- UPDATE_DB_PARAM— Aktualisiert die Datenparameter.

`DATABASE_NAME`

Der Name der Datenbank, mit der die Aufgabe verknüpft ist.

`COMPLETED_WORK_BYTES`

Die Anzahl der Byte, die durch die Aufgabe wiederhergestellt wurden.

`DURATION_MINS`

Die Zeit, die zum Abschließen der Aufgabe benötigt wurde.

`LIFECYCLE`

Der Status der Aufgabe. Mögliche Status:

- CREATED— Nachdem eine Aufgabe an Amazon übermittelt wurdeRDS, RDS setzt Amazon den Status aufCREATED.
- IN_PROGRESS— Nach dem Start einer Aufgabe RDS setzt Amazon den Status aufIN_PROGRESS. Es kann bis zu 5 Minuten dauern, bis sich ein Status von CREATED zu ändertIN_PROGRESS.

- SUCCESS— Nach Abschluss einer Aufgabe RDS setzt Amazon den Status aufSUCCESS.
- ERROR— Wenn eine Wiederherstellungsaufgabe fehlschlägt, RDS setzt Amazon den Status aufERROR. Weitere Informationen zu dem Fehler finden Sie unterTASK_OUPUT.

CREATED_BY

authidDerjenige, der den Befehl erstellt hat.

CREATED_AT

Datum und Uhrzeit der Erstellung der Aufgabe.

LAST_UPDATED_AT

Datum und Uhrzeit der letzten Aktualisierung der Aufgabe.

TASK_INPUT_PARAMS

Die Parameter unterscheiden sich je nach Aufgabentyp. Alle Eingabeparameter werden als JSON Objekt dargestellt. Die JSON Schlüssel für die RESTORE_DB Aufgabe lauten beispielsweise wie folgt:

- DBNAME
- RESTORE_TIMESTAMP
- S3_BUCKET_NAME
- S3_PREFIX

TASK_OUTPUT

Zusätzliche Informationen über die Aufgabe. Wenn bei der systemeigenen Wiederherstellung ein Fehler auftritt, enthält diese Spalte Informationen zu dem Fehler.

Beispiele für Antworten

Das folgende Antwortbeispiel zeigt, dass eine aufgerufene Datenbank erfolgreich erstellt TESTJP wurde. Weitere Informationen finden Sie in der the section called “rdsadmin.create_database” gespeicherten Prozedur.

```
`1 SUCCESS CREATE_DATABASE RDSDB 2023-10-24-18.32.44.962689 2023-10-24-18.34.50.038523
 1 TESTJP { "CODESET" : "IBM-437", "TERRITORY" : "JP", "COLLATION" : "SYSTEM",
 "AUTOCONFIGURE_CMD" : "", "PAGESIZE" : 4096 }
```

```
2023-10-24-18.33.30.079048 Task execution has started.

2023-10-24-18.34.50.038523 Task execution has completed successfully`.
```

Das folgende Antwortbeispiel erklärt, warum das Löschen einer Datenbank fehlgeschlagen ist. Weitere Informationen finden Sie in der the section called “rdsadmin.drop_database” gespeicherten Prozedur.

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

Das folgende Antwortbeispiel zeigt die erfolgreiche Wiederherstellung einer Datenbank. Weitere Informationen finden Sie in der the section called “rdsadmin.restore_database” gespeicherten Prozedur.

```
1 RESTORE_DB  SAMPLE  SUCCESS

{ "S3_BUCKET_NAME" : "amzn-s3-demo-bucket", "S3_PREFIX" :
 "SAMPLE.0.rdsdb3.DBPART000.20230413183211.001", "RESTORE_TIMESTAMP" :
 "20230413183211", "BACKUP_TYPE" : "offline" }




2023-11-06-18.31.03.115795 Task execution has started.
2023-11-06-18.31.04.300231 Preparing to download
2023-11-06-18.31.08.368827 Download complete. Starting Restore
2023-11-06-18.33.13.891356 Task Completed Successfully
```

rdsadmin.list_databases

Gibt eine Liste aller Datenbanken zurück, die auf einer RDS for Db2-DB-Instance laufen.

Syntax

```
db2 "select * from table(rdsadmin.list_databases())"
```

Nutzungshinweise

Diese benutzerdefinierte Funktion gibt nicht an, ob sich Datenbanken in einem aktivierten oder deaktivierten Zustand befinden.

Wenn Sie Ihre Datenbanken nicht in der Liste sehen, rufen Sie die the section called “rdsadmin.get_task_status” benutzerdefinierte Funktion auf und suchen Sie nach Fehlermeldungen.

Antwort

Die `rdsadmin.list_databases` benutzerdefinierte Funktion gibt die folgenden Spalten zurück:

`DATABASE_NAME`

Der Name einer Datenbank.

`CREATE_TIME`

Datum und Uhrzeit der Erstellung der Datenbank.

Beispiele für Antworten

Das folgende Antwortbeispiel zeigt eine Liste von Datenbanken und die Zeiten, zu denen sie erstellt wurden. `rdsadmin`ist eine Datenbank, die Amazon RDS verwaltet und die immer in der Ausgabe erscheint.

```
DATABASE_NAME   CREATE_TIME
--------------- --------------------------
rdsadmin        2024-10-22-03.37.48.535671
TEST            2024-10-22-03.39.36.818679
TEST1           2024-10-22-03.57.15.218009
TEST2           2024-10-22-03.59.28.029556
```

Fehlerbehebung für Amazon RDS for Db2

Der folgende Inhalt kann Ihnen bei der Behebung von Problemen helfen, die bei RDS for Db2 auftreten.

Weitere Informationen zu allgemeinen Problemen mit Amazon RDS zur Fehlerbehebung finden Sie unterFehlerbehebung für Amazon RDS.

Themen

- Fehler bei der Datenbankverbindung
- Ein Fehler in der Datei I/O
- Fehler bei gespeicherten Prozeduren

Fehler bei der Datenbankverbindung

Die folgende Fehlermeldung weist darauf hin, dass eine Datenbank keine Verbindung herstellen konnte, weil der Server nicht über ausreichend Arbeitsspeicher verfügt.

```
SQL1643C The database manager failed to allocate shared memory because the
database manager instance memory limit has been reached.
```

Erhöhen Sie den Arbeitsspeicher für Ihre DB-Instance und versuchen Sie dann erneut, eine Verbindung zu Ihrer Datenbank herzustellen. Hinweise zur Speichernutzung und Empfehlungen für Datenbanken finden Sie unterthe section called “Mehrere Db2-Datenbanken”. Hinweise zum Aktualisieren des Speichers für eine RDS for Db2-Datenbank finden Sie unterthe section called “rdsadmin.update_db_param”.

Ein Fehler in der Datei I/O

Ein I/O Dateifehler kann aus verschiedenen Gründen auftreten, z. B. wenn Sie den LOAD Befehl verwenden oder die `rdsadmin.restore_database` gespeicherte Prozedur aufrufen.

In diesem Beispiel führen Sie den folgenden LOAD Befehl aus.

```
db2 "call sysproc.admin_cmd('load from "DB2REMOTE://s3test//public/datapump/t6.del" of
 del lobs from "DB2REMOTE://s3test/public/datapump/" modified by lobsinfile MESSAGES ON
 SERVER insert INTO RDSDB.t6 nonrecoverable ')"
```

Der LOAD Befehl gibt die folgende Meldung zurück:

```
 Result set 1
 --------------

 ROWS_READ            ROWS_SKIPPED         ROWS_LOADED          ROWS_REJECTED
ROWS_DELETED         ROWS_COMMITTED       ROWS_PARTITIONED     NUM_AGENTINFO_ENTRIES
MSG_RETRIEVAL




MSG_REMOVAL
 -------------------- -------------------- -------------------- --------------------
-------------------- -------------------- -------------------- --------------------
--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
                     -                    -                    -                    -
                    -                    -                    -                    -
SELECT SQLCODE, MSG FROM TABLE(SYSPROC.ADMIN_GET_MSGS('1594987316_285548770')) AS MSG




                                                                                    CALL
SYSPROC.ADMIN_REMOVE_MSGS('1594987316_285548770')

 1 record(s) selected.

 Return Status = 0

SQL20397W  Routine "SYSPROC.ADMIN_CMD" execution has completed, but at least
one error, "SQL1652", was encountered during the execution. More information
is available.  SQLSTATE=01H52
```

Um die Fehlermeldung anzuzeigen, führen Sie den SQL-Befehl wie in der vorherigen Antwort vorgeschlagen aus. `SELECT SQLCODE, MSG FROM TABLE(SYSPROC.ADMIN_GET_MSGS('1594987316_285548770')) AS MSG`gibt die folgende Meldung zurück:

```
SQLCODE   MSG
```

```
---------
 ------------------------------------------------------------------------------------------------
SQL2025N  An I/O error occurred. Error code “438”. Media on which this error occurred:
 “DB2REMOTE://s3test//public/datapump/t6.del”

SQL3500W The utility is beginning the LOAD phase at time “07/05/2024 21:21:48.082954”

SQL1652N File I/O error occurred
```

Die Db2-Diagnoseprotokolle enthalten eine Protokolldatei, die der folgenden ähnelt:

```
2024-07-05-21.20.09.440609+000 I1191321E864          LEVEL: Error
PID     : 2710                 TID : 139619509200640 PROC : db2sysc 0
INSTANCE: rdsdb                NODE : 000            DB   : NTP
APPHDL  : 0-12180              APPID: xxx.xx.x.xxx.xxxxx.xxxxxxxxxxxx
UOWID   : 5                    ACTID: 1
AUTHID  : ADMIN                HOSTNAME: ip-xx-xx-x-xx
EDUID   : 147                  EDUNAME: db2lmr 0
FUNCTION: DB2 UDB, oper system services, sqloS3Client_GetObjectInfo, probe:219
MESSAGE : ZRC=0x870F01B6=-2029059658=SQLO_FAILED
          "An unexpected error is encountered"
DATA #1 : String, 29 bytes
S3:HeadObject request failed.
DATA #2 : signed integer, 4 bytes
99
DATA #3 : String, 0 bytes
Object not dumped: Address: 0x00007EFC08A9AE38 Size: 0 Reason: Zero-length data
DATA #4 : String, 33 bytes
curlCode: 28, Timeout was reached
```

Dieser I/O Dateifehler kann auf verschiedene Szenarien zurückzuführen sein. Beispielsweise fehlt der VPC, die der Sicherheitsgruppe zugeordnet ist, die zur Erstellung Ihrer RDS for Db2-DB-Instance verwendet wurde, möglicherweise kein Amazon S3 S3-Gateway-Endpunkt. Dieser Endpunkt ist wichtig, damit RDS for Db2 auf Amazon S3 zugreifen kann. Wenn sich Ihre RDS for Db2-DB-Instance in privaten Subnetzen befindet, ist ein Amazon S3 S3-Gateway-Endpunkt erforderlich. Sie können angeben, ob Ihre DB-Instance private oder öffentliche Subnetze verwendet, indem Sie Amazon RDS-Subnetzgruppen konfigurieren. Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.

Themen

- Schritt 1: Erstellen Sie einen VPC-Gateway-Endpunkt für Amazon S3

- Schritt 2: Stellen Sie sicher, dass Ihr VPC-Gateway-Endpunkt für Amazon S3 existiert

Schritt 1: Erstellen Sie einen VPC-Gateway-Endpunkt für Amazon S3

Damit Ihre RDS for Db2-DB-Instance mit Amazon S3 interagieren kann, erstellen Sie eine VPC und dann einen Amazon S3 S3-Gateway-Endpunkt, den private Subnetze verwenden können.

So erstellen Sie einen VPC-Gateway-Endpunkt für S3

1. Erstellen Sie eine VPC. Weitere Informationen finden Sie unter Erstellen einer VPC im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
2. Erstellen Sie einen Amazon S3 S3-Gateway-Endpunkt, den private Subnetze verwenden können. Weitere Informationen finden Sie im Handbuch unter Gateway-Endpunkte.AWS PrivateLink

Schritt 2: Stellen Sie sicher, dass Ihr VPC-Gateway-Endpunkt für Amazon S3 existiert

Bestätigen Sie, dass Sie erfolgreich einen Amazon S3 S3-Gateway-Endpunkt erstellt haben, indem Sie den AWS Management Console oder den verwenden AWS CLI.

Konsole

Um einen Amazon S3 S3-Gateway-Endpunkt zu bestätigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon VPC-Konsole unterhttps://console.aws.amazon.com/vpc.
2. Wählen Sie in der oberen rechten Ecke der Konsole die Ihrer VPC AWS-Region aus.
3. Wählen Sie die VPC aus, die Sie erstellt haben.
4. Vergewissern Sie sich auf der Registerkarte Ressourcenübersicht unter Netzwerkverbindungen, dass ein Amazon S3 S3-Gateway-Endpunkt aufgeführt ist.

AWS CLI

Führen Sie den describe-vpc-endpointsBefehl aus, um einen Amazon S3 S3-Gateway-Endpunkt zu bestätigen. Ersetzen Sie im folgenden Beispiel *`vpc_id`* durch die VPC-ID, *`region`* durch Ihre AWS-Region und *`profile`* durch Ihren Profilnamen.

Für LinuxmacOS, oderUnix:

```
aws ec2 describe-vpc-endpoints \
    --filters "Name=vpc-id,Values=$vpc_id" \
    "Name=service-name,\
    Values=com.amazonaws.${region}.s3" \
    --region $region --profile=$profile \
    --query "VpcEndpoints[*].VpcEndpointId" --output text
```

Windows:

```
aws ec2 describe-vpc-endpoints ^
    --filters "Name=vpc-id,Values=$vpc_id" ^
    "Name=service-name,^
    Values=com.amazonaws.${region}.s3" ^
    --region $region --profile=$profile ^
    --query "VpcEndpoints[*].VpcEndpointId" --output text
```

Dieser Befehl erzeugt eine Ausgabe ähnlich dem folgenden Beispiel, wenn ein Amazon S3 S3-Gateway-Endpunkt vorhanden ist.

```
[
    "vpce-0ea810434ff0b97e4"
]
```

Dieser Befehl erzeugt eine Ausgabe ähnlich dem folgenden Beispiel, wenn kein Amazon S3 S3-Gateway-Endpunkt existiert.

```
[]
```

Wenn kein Amazon S3 S3-Gateway-Endpunkt aufgeführt ist, dannSchritt 1: Erstellen Sie einen VPC-Gateway-Endpunkt für Amazon S3.

Fehler bei gespeicherten Prozeduren

In diesem Abschnitt werden verschiedene Fehler beschrieben, die beim Aufrufen von gespeicherten Prozeduren auftreten, und wie sie behoben werden können.

Kategorie	Fehler bei gespeicherten Prozeduren
Datenbanken	rdsadmin.activate_database-Fehler

Kategorie	Fehler bei gespeicherten Prozeduren
Datenbanken	rdsadmin.create_database-Fehler
Datenbanken	rdsadmin.deactivate_database-Fehler
Datenbanken	rdsadmin.drop_database-Fehler
Datenbanken	rdsadmin.reactivate_database-Fehler
Datenbanken	rdsadmin.restore_database-Fehler
Datenbanken	rdsadmin.update_db_param Fehler
Tablespaces	rdsadmin.alter_tablespace-Fehler

rdsadmin.activate_database-Fehler

Die folgenden Fehler können auftreten, wenn Sie die gespeicherte Prozedur aufrufen. the section called “rdsadmin.activate_database”

Fehler	Fehlermeldung
Failed to allocate shared memory	`SQL1643C The database manager failed to allocate shared memory because the database manager instance memory limit has been reached.`
Unable to activate because of running processes	`The database can’t be activated because it's in the process of being created or restored.`

Gemeinsamer Speicher konnte nicht zugewiesen werden

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur eine Datenbank nicht aktivieren konnte, da die DB-Instance nicht über ausreichend Arbeitsspeicher verfügt.

```
SQL1643C The database manager failed to allocate shared memory because the database
 manager instance memory limit has been reached.
```

Erhöhen Sie den Speicher für Ihre DB-Instance und rufen Sie dann die `rdsadmin.activate_database` gespeicherte Prozedur erneut auf. Hinweise zur Speichernutzung und Empfehlungen für Datenbanken finden Sie unterthe section called “Mehrere Db2-Datenbanken”.

Die Aktivierung ist aufgrund laufender Prozesse nicht möglich

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur eine Datenbank nicht aktivieren konnte, da die `rdsadmin.create_database` oder die `rdsadmin.restore_database` gespeicherte Prozedur gerade läuft.

```
The database can’t be activated because it's in the process of being created or
 restored.
```

Warten Sie einige Minuten, und rufen Sie dann die `rdsadmin.activate_database` gespeicherte Prozedur erneut auf.

rdsadmin.alter_tablespace-Fehler

Die folgenden Fehler können auftreten, wenn Sie die gespeicherte Prozedur aufrufen. the section called “rdsadmin.alter_tablespace”

Fehler	Fehlermeldung

Statement not valid	`DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned:`
	
	`SQL1763N Invalid ALTER TABLESPACE statement for table space "TBSP_TEST" due to reason "12"`

tablespace_prefetch_size value not valid	`Invalid tablespace_prefetch_size. Set value to AUTOMATIC or to a non-zero positive numerical value.`
tablespace_prefetch_size numerical value not valid	`Invalid tablespace_prefetch_size. The number of pages can't be greater than 32767.`
Parameter can't be used with tablespace_prefetch_size	`You can't use tablespace_prefetch_size with {parameter }.`

Fehler	Fehlermeldung
Tablespace change failed	`The change to tablespace { tablespace_name}  failed because you can only alter LARGE or REGULAR tablespaces.`

Die Aussage ist nicht gültig

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur sich gegenseitig ausschließende optionale Parameter mit anderen optionalen Parametern kombiniert hat. Die optionalen Parameter`reduce_max, reduce_stop`,`reduce_value`, `lower_high_waterlower_high_water_stop`, und `switch_online` für die `rdsadmin.alter_tablespace` gespeicherte Prozedur schließen sich gegenseitig aus. Sie können sie nicht mit anderen optionalen Parametern kombinieren, z. B. `buffer_pool_name` in der `rdsadmin.alter_tablespace` gespeicherten Prozedur. Wenn Sie sie kombinieren, gibt Db2 beim Aufrufen der `rdsadmin.get_task_status` benutzerdefinierten Funktion diese Fehlermeldung zurück.

```
DB21034E The command was processed as an SQL statement because it was not a valid
 Command Line Processor command. During SQL processing it returned:
SQL1763N Invalid ALTER TABLESPACE statement for table space "TBSP_TEST" due to reason
 "12"
```

Rufen Sie die `rdsadmin.alter_tablespace` gespeicherte Prozedur erneut auf, ohne sich gegenseitig ausschließende optionale Parameter mit anderen optionalen Parametern zu kombinieren. Rufen Sie dann die `rdsadmin.get_task_status` benutzerdefinierte Funktion auf. Weitere Informationen finden Sie unter rdsadmin.alter_tablespace und rdsadmin.get_task_status.

Der Wert für tablespace_prefetch_size ist nicht gültig

Die folgende Fehlermeldung weist darauf hin, dass Sie keinen Wert auf oder einen nicht positiven numerischen Wert festgelegt haben. `tablespace_prefetch_size` `AUTOMATIC` Sie haben beispielsweise versucht, ihn auf zu setzen. `testinput`

```
Invalid tablespace_prefetch_size. Set value to AUTOMATIC or to a non-zero positive
 numerical value.
```

Rufen Sie die `rdsadmin.alter_tablespace` gespeicherte Prozedur erneut auf und setzen Sie `tablespace_prefetch_size` sie auf `AUTOMATIC` oder einen nicht positiven numerischen Wert.

Der numerische Wert von tablespace_prefetch_size ist nicht gültig

Die folgende Fehlermeldung weist darauf hin, dass Sie einen numerischen Wert angegeben haben, der größer als 32767 ist. `tablespace_prefetch_size`

```
Invalid tablespace_prefetch_size. The number of pages can't be greater than 32767.
```

Rufen Sie die `rdsadmin.alter_tablespace` gespeicherte Prozedur erneut auf und setzen Sie `tablespace_prefetch_size` sie auf einen positiven numerischen Wert ungleich Null, der kleiner oder gleich 32767 ist.

Der Parameter kann nicht mit tablespace_prefetch_size verwendet werden

Die folgende Fehlermeldung weist darauf hin, dass Sie versucht haben, die Datei mit einem inkompatiblen Parameter zu verwenden. `tablespace_prefetch_size`

```
You can't use tablespace_prefetch_size with {parameter}.
```

Rufen Sie die `rdsadmin.alter_tablespace` gespeicherte Prozedur erneut auf und verwenden Sie sie nur `tablespace_prefetch_size` mit kompatiblen Parametern. Hinweise zu Parametern, die Sie mit verwenden können`tablespace_prefetch_size`, finden Sie unter[rdsadmin.alter_tablespace](#).

Die Tablespace-Änderung ist fehlgeschlagen

Die folgende Fehlermeldung weist darauf hin, dass Sie versucht haben, einen Tablespace zu ändern.

```
The change to tablespace {tablespace_name} failed because you can only alter LARGE or
 REGULAR tablespaces.
```

rdsadmin.create_database-Fehler

Der folgende Fehler kann auftreten, wenn Sie die gespeicherte Prozedur aufrufen. [the section called “rdsadmin.create_database”](#)

Fehler	Fehlermeldung
Failed to allocate shared memory	`SQL1643C The database manager failed to allocate shared memory because the database manager instance memory limit has been reached.`

Gemeinsamer Speicher konnte nicht zugewiesen werden

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur keine Datenbank erstellen konnte, da die DB-Instance nicht über ausreichend Arbeitsspeicher verfügt.

```
SQL1643C The database manager failed to allocate shared memory because the database
 manager instance memory limit has been reached.
```

Erhöhen Sie den Speicher für Ihre DB-Instance und rufen Sie dann die `rdsadmin.create_database` gespeicherte Prozedur erneut auf. Hinweise zur Speichernutzung und Empfehlungen für Datenbanken finden Sie unterthe section called “Mehrere Db2-Datenbanken”.

Um zu überprüfen, ob die Datenbank erstellt wurde, rufen Sie die the section called “rdsadmin.list_databases” benutzerdefinierte Funktion auf und überprüfen Sie, ob die neue Datenbank aufgeführt ist.

rdsadmin.deactivate_database-Fehler

Der folgende Fehler kann auftreten, wenn Sie die gespeicherte Prozedur aufrufen. the section called “rdsadmin.deactivate_database”

Fehler	Fehlermeldung
Unable to deactivate because of running processes	`The database can’t be deactivated because it's in the process of being created or restored.`

Die Deaktivierung ist aufgrund laufender Prozesse nicht möglich

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur eine Datenbank nicht deaktivieren konnte, da die `rdsadmin.create_database` oder die `rdsadmin.restore_database` gespeicherte Prozedur gerade läuft.

```
The database can’t be deactivated because it's in the process of being created or
 restored.
```

Warten Sie einige Minuten, und rufen Sie dann die `rdsadmin.deactivate_database` gespeicherte Prozedur erneut auf.

rdsadmin.drop_database-Fehler

Die folgenden Fehler können auftreten, wenn Sie die gespeicherte Prozedur aufrufen. the section called “rdsadmin.drop_database”

Fehler	Fehlermeldung
Database name doesn't exist	`SQL0438N Application raised error or warning with diagnostic text: "Cannot drop database. Database with provided name does not exist". SQLSTATE=99993`
Return status = 0	`Return Status = 0`
Dropping database not allowed	`1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.0 3.744122 2023-10-10-16.33.30.143797 - 2023-10-1 0-16.33.30.098857 Task execution has started. 2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task. Reason Dropping database created via rds CreateDBInstance api is not allowed. Only database created using rdsadmin. create_database can be dropped`

Der Datenbankname ist nicht vorhanden

Die folgende Fehlermeldung weist darauf hin, dass Sie in der `rdsadmin.drop_database` gespeicherten Prozedur einen falschen Datenbanknamen angegeben haben.

```
SQL0438N Application raised error or warning with diagnostic text: "Cannot
drop database. Database with provided name does not exist". SQLSTATE=99993
```

Rufen Sie die `rdsadmin.drop_database` gespeicherte Prozedur erneut mit einem korrekten Datenbanknamen auf. Um zu bestätigen, dass die Datenbank gelöscht wurde, rufen Sie die the

section called “rdsadmin.list_databases” benutzerdefinierte Funktion auf und überprüfen Sie, ob die gelöschte Datenbank nicht aufgeführt ist.

Rückgabestatus = 0

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur nicht abgeschlossen werden konnte.

```
Return Status = 0
```

Rufen Sie nach dem Empfang die `Return Status = 0` benutzerdefinierte Funktion rdsadmin.get_task_status auf.

Das Löschen der Datenbank ist nicht zulässig

Die folgende Fehlermeldung weist darauf hin, dass Sie die Datenbank entweder mit der Amazon RDS-Konsole oder mit dem erstellt haben AWS CLI. Sie können die `rdsadmin.drop_database` gespeicherte Prozedur nur verwenden, wenn Sie die Datenbank durch Aufrufen der the section called “rdsadmin.create_database” gespeicherten Prozedur erstellt haben.

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

Um eine Datenbank zu löschen, die Sie entweder mit der Amazon RDS-Konsole oder der erstellt haben AWS CLI, stellen Sie über einen Client eine Verbindung zur Datenbank her und führen Sie dann den entsprechenden Befehl aus.

rdsadmin.reactivate_database-Fehler

Der folgende Fehler kann auftreten, wenn Sie die gespeicherte Prozedur aufrufen. the section called “rdsadmin.reactivate_database”

Fehler	Fehlermeldung
Failed to allocate shared memory	`SQL1643C The database manager failed to allocate shared memory because the database manager instance memory limit has been reached.`

Fehler	Fehlermeldung
Unable to reactivate because of running processes	`The database can’t be reactivated because it's in the process of being created or restored.`

Gemeinsamer Speicher konnte nicht zugewiesen werden

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur eine Datenbank nicht aktivieren konnte, da die DB-Instance nicht über ausreichend Arbeitsspeicher verfügt.

```
SQL1643C The database manager failed to allocate shared memory because the database
 manager instance memory limit has been reached.
```

Erhöhen Sie den Speicher für Ihre DB-Instance und rufen Sie dann die `rdsadmin.activate_database` gespeicherte Prozedur erneut auf. Hinweise zur Speichernutzung und Empfehlungen für Datenbanken finden Sie unterthe section called “Mehrere Db2-Datenbanken”.

Reaktivierung aufgrund laufender Prozesse nicht möglich

Die folgende Fehlermeldung weist darauf hin, dass die gespeicherte Prozedur eine Datenbank nicht reaktivieren konnte, da die `rdsadmin.create_database` oder die `rdsadmin.restore_database` gespeicherte Prozedur gerade läuft.

```
The database can’t be reactivated because it's in the process of being created or
 restored.
```

Warten Sie einige Minuten, und rufen Sie dann die `rdsadmin.reactivate_database` gespeicherte Prozedur erneut auf.

rdsadmin.restore_database-Fehler

Die folgenden Fehler können auftreten, wenn Sie die gespeicherte Prozedur aufrufen: the section called “rdsadmin.restore_database”

Fehler	Fehlermeldung
Insufficient disk space	`Aborting task. Reason Restoring your database failed because of insufficient disk space. Increase the`

Fehler	Fehlermeldung
	`storage for your DB instance and rerun the rdsadmin.restore_database stored procedure.`
Internal error	`Caught exception during executing task id 104, Aborting task. Reason Internal Error`
Non-fenced routines not allowed	`Caught exception during executing task id 2, Aborting task. Reason Non fenced routines are not allowed. Please delete the routines and retry the restore.`
Tablespaces not restored	`Reason SQL0970N The system attempted to write to a read-only file. Reason SQL2563W The Restore process has completed successfully. However one or more table spaces from the backup were not restored.`

Nicht genügend Festplattenspeicher

Die folgende Fehlermeldung weist darauf hin, dass Ihre DB-Instance nicht über ausreichend Speicherplatz verfügt, um Ihre Datenbank wiederherzustellen:

```
Aborting task. Reason Restoring your database failed because of insufficient
 disk space. Increase the storage for your DB instance and rerun the
 rdsadmin.restore_database stored procedure.
```

Der freie Speicherplatz auf Ihrer DB-Instance muss mehr als doppelt so groß sein wie Ihr Backup-Image. Wenn Ihr Backup-Image komprimiert ist, muss der freie Speicherplatz auf Ihrer DB-Instance mehr als das Dreifache der Größe Ihres Backup-Images betragen. Weitere Informationen finden Sie unter the section called “Steigern der DB-Instance-Speicherkapazität”.

Erhöhen Sie Ihren Festplattenspeicher und rufen Sie dann die `rdsadmin.restore_database` gespeicherte Prozedur erneut auf. Um zu überprüfen, ob die Datenbank wiederhergestellt wurde, rufen Sie die the section called “rdsadmin.list_databases” benutzerdefinierte Funktion auf und überprüfen Sie, ob die wiederhergestellte Datenbank aufgeführt ist.

Interner Fehler

Die folgende Fehlermeldung weist darauf hin, dass bei der gespeicherten Prozedur ein interner Fehler aufgetreten ist:

```
Caught exception during executing task id 104, Aborting task. Reason Internal Error
```

AWS Support kontaktieren.

Routinen ohne Umzäunung sind nicht zulässig

Die folgende Fehlermeldung weist darauf hin, dass Ihre Datenbank Routinen ohne Fencing enthält:

```
Caught exception during executing task id 2, Aborting task. Reason Non fenced routines
 are not allowed. Please delete the routines and retry the restore.
```

RDS für Db2 unterstützt keine Routinen ohne Fencing. Entfernen Sie die Routinen ohne Fencing aus der Quelldatenbank und rufen Sie sie dann erneut auf. `rdsadmin.restore_database` Um zu überprüfen, ob die Datenbank wiederhergestellt wurde, rufen Sie die the section called “rdsadmin.list_databases” benutzerdefinierte Funktion auf und überprüfen Sie, ob die wiederhergestellte Datenbank aufgeführt ist. Weitere Informationen finden Sie unter the section called “Routinen ohne Umzäunung”.

Tablespaces wurden nicht wiederhergestellt

Die folgende Fehlermeldung weist darauf hin, dass RDS for Db2 Ihre Datenbank erfolgreich wiederhergestellt hat, aber einen oder mehrere Tablespaces nicht wiederherstellen konnten:

```
Reason SQL0970N The system attempted to write to a read-only file.
Reason SQL2563W The Restore process has completed successfully. However one or more
 table spaces from the backup were not restored.
```

RDS für Db2 unterstützt keine nichtautomatische Speicherung. Wandeln Sie nichtautomatischen Speicher in automatischen Speicher um und rufen Sie dann erneut auf. `rdsadmin.restore_database` Weitere Informationen finden Sie in der Dokumentation unter Konvertierung einer nichtautomatischen Speicherdatenbank zur Verwendung von automatischem Speicher. IBM Db2

Datenbanken mit nicht automatischem SMS-Speicher müssen manuell wiederhergestellt werden. Wenn Ihre Datenbank über einen nicht automatischen SMS-Speicher verfügt, wenden Sie sich an den AWS Support.

Hinweise zu nichtautomatischem Speicher und einmaligen Migrationen finden Sie unter. the section called “Nichtautomatische Speicher-Tablespaces während der Migration”

rdsadmin.update_db_param-Fehler

Der folgende Fehler kann auftreten, wenn Sie die gespeicherte Prozedur aufrufen. the section called “rdsadmin.update_db_param”

Fehler	Fehlermeldung
Parameter not supported or modifiable	`SQL0438N Application raised error or warning with diagnostic text: "Parameter is either not supported or not modifiable to customers". SQLSTATE=99993`

Der Parameter wird nicht unterstützt oder kann nicht geändert werden

Die folgende Fehlermeldung weist darauf hin, dass Sie versucht haben, einen Datenbankkonfigurationsparameter zu ändern, der entweder nicht unterstützt wird oder nicht geändert werden kann.

```
SQL0438N Application raised error or warning with diagnostic text: "Parameter
is either not supported or not modifiable to customers". SQLSTATE=99993
```

Sie können anhand Ihrer Parametergruppen sehen, welche Parameter geändert werden können. Weitere Informationen finden Sie unter the section called “Parameterwerte für eine DB-Parametergruppe anzeigen”.

Amazon RDS für MariaDB

Amazon RDS unterstützt mehrere Versionen von MariaDB für DB-Instances. Vollständige Informationen zu den unterstützten Versionen finden Sie unter. MariaDB auf Amazon-RDS-Versionen

Verwenden Sie die Amazon-RDS-Management-Tools bzw. -Schnittstellen zum Erstellen einer MariaDB-DB-Instance. Sie können dann die Amazon-RDS-Tools verwenden, um Verwaltungsaktionen für die DB-Instance durchzuführen. Dazu gehören u. a. folgende Aktionen:

- Neukonfiguration oder Größenänderung der DB-Instance
- Autorisieren von Verbindungen mit der DB-Instance
- Erstellen und Wiederherstellen aus Backups oder Snapshots
- Erstellen von sekundären Multi-AZ-Instances
- Erstellen von Read Replicas
- Überwachen der Leistung Ihrer DB-Instance

Verwenden Sie die standardmäßigen MariaDB-Dienstprogramme und -Anwendungen zum Speichern und Aufrufen der Daten in der DB-Instance.

MariaDB ist in allen AWS-Regionen verfügbar. Weitere Informationen zu finden AWS-Regionen Sie unterRegionen, Availability Zones und Local Zones.

Sie können Amazon RDS for MariaDB-Datenbanken für die Erstellung von HIPAA-kompatiblen Anwendungen verwenden. Mit können Sie Gesundheitsdaten, darunter geschützte patientenbezogene Daten (protected health information, PHI), im Rahmen eines Geschäftspartnervertrags (Business Associate Agreement, BAA) speicher AWS. Weitere Informationen finden Sie unter HIPAA-Compliance. AWS Die Services in Scope wurden von einem externen Prüfer umfassend bewertet und führten zu einer Zertifizierung, Konformitätsbescheinigung oder Authority to Operate (ATO). Weitere Informationen finden Sie unterAWS -Services im Rahmen des Compliance-Programms.

Bevor Sie eine DB-Instance erstellen, führen Sie die Schritte in Einrichtung Ihrer Amazon RDS-Umgebung aus. Wenn Sie eine DB-Instance erstellen, erhält das RDS-Hauptbenutzerkonto DBA-Berechtigungen mit einigen Einschränkungen. Verwenden Sie dieses Konto für administrative Aufgaben wie das Erstellen zusätzlicher Datenbankkonten.

Sie können das folgende erstellen:

- DB-Instances
- DB-Snapshots
- Point-in-time stellt wieder her
- Automatische Backups
- Manuelle Backups

Sie können DB-Instances verwenden, auf denen MariaDB in einer Virtual Private Cloud (VPC) auf der Basis von Amazon VPC ausgeführt wird. Sie können auch Funktionen zu Ihrer MariaDB-DB-Instance hinzufügen, indem Sie verschiedene Optionen aktivieren. Amazon RDS unterstützt Multi-AZ-Bereitstellungen für MariaDB als eine Lösung mit hoher Verfügbarkeit und Failover.

> **Important**
>
> Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Shell-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind. Sie können mit Standard-SQL-Clients wie mysql auf Ihre Datenbank zugreifen. Sie können jedoch nicht direkt auf den Host zugreifen, indem Sie Telnet oder Secure Shell (SSH) verwenden.

Themen

- Unterstützung MariaDB MariaDB-Funktionen bei Amazon RDS
- MariaDB auf Amazon-RDS-Versionen
- Verbindung zu Ihrer MariaDB-DB-Instance herstellen
- Sichern von Verbindungen von MariaDB-DB-Instances
- Verbesserung der Abfrageleistung für RDS für MariaDB mit Amazon RDS Optimized Reads
- Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB
- Upgrades der MariaDB-DB-Engine
- Aktualisierung einer MariaDB DB-Snapshot-Engine-Version
- Daten in eine Amazon RDS for MariaDB-DB-Instance importieren
- Arbeiten mit der MariaDB-Replikation in Amazon RDS
- Optionen für MariaDB-Datenbank-Engine

- Parameter für MariaDB
- Daten von einem My SQL DB-Snapshot zu einer MariaDB-DB-Instance migrieren
- MariaDB auf Amazon RDS – SQL-Referenz
- Lokale Zeitzone für MariaDB DB-Instances
- Bekannte Probleme und Einschränkungen für RDS für MariaDB

Unterstützung MariaDB MariaDB-Funktionen bei Amazon RDS

RDSfor MariaDB unterstützt die meisten Funktionen und Fähigkeiten von MariaDB. Einige Funktionen werden möglicherweise nur begrenzt unterstützt oder haben eingeschränkte Berechtigungen.

Sie können neue RDS Amazon-Funktionen in der Liste Was ist neu in der Datenbank? filtern. Seite. Wählen Sie für Produkte Amazon RDS. Suchen Sie dann mit Schlüsselwörtern wie **`MariaDB 2023`**.

> **Note**
>
> Die folgenden Listen sind nicht vollständig.

Weitere Informationen zur Unterstützung von MariaDB-Funktionen bei Amazon RDS finden Sie in den folgenden Themen.

Themen

- Unterstützte Speicher-Engines für MariaDB bei Amazon RDS
- Cache-Erwärmung für MariaDB bei Amazon RDS
- MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS

Unterstützung MariaDB MariaDB-Funktionen bei Amazon RDS für MariaDB-Hauptversionen

In den folgenden Abschnitten finden Sie Informationen zur Unterstützung von MariaDB-Funktionen bei Amazon RDS für MariaDB-Hauptversionen:

Themen

- MariaDB 11.4-Unterstützung bei Amazon RDS

- MariaDB 10.11-Unterstützung bei Amazon RDS
- MariaDB 10.6-Unterstützung bei Amazon RDS
- MariaDB 10.5-Unterstützung bei Amazon RDS
- MariaDB 10.4-Unterstützung bei Amazon RDS

Informationen zu unterstützten Nebenversionen von Amazon RDS für MariaDB finden Sie unter. MariaDB auf Amazon-RDS-Versionen

MariaDB 11.4-Unterstützung bei Amazon RDS

Amazon RDS unterstützt die folgenden neuen Funktionen für Ihre DB-Instances, auf denen MariaDB Version 11.4 oder höher ausgeführt wird.

- Kryptografische Bibliothek — RDS für MariaDB wurde Open durch AWS Libcrypto (AWS-LC) ersetzt, was SSL 140-3-zertifiziert ist. FIPS
- Simple Password Check Plugin — Mit dem MariaDB Simple Password Check Plugin können Sie überprüfen, ob ein Passwort mindestens eine bestimmte Anzahl von Zeichen eines bestimmten Typs enthält. Weitere Informationen finden Sie unter the section called “Plugins zur Passwortvalidierung”.
- Cracklib Password Check Plugin — Sie können das MariaDB Cracklib Password Check Plugin verwenden, um die Stärke neuer Passwörter zu überprüfen. Weitere Informationen finden Sie unter the section called “Plugins zur Passwortvalidierung”.
- InnoDB-Verbesserungen – Diese Verbesserungen umfassen Folgendes:
  - Der Änderungspuffer wurde entfernt. Weitere Informationen finden Sie unter InnoDB Change Buffering.
  - Die InnoDB-Defragmentierung wurde entfernt. Weitere Informationen finden Sie unter InnoDB-Defragmentierung.
- Neues Recht — Der Admin-Benutzer hat jetzt auch das Recht. `SHOW CREATE ROUTINE` Dieses Recht ermöglicht es dem Empfänger, die `SHOW CREATE` Definitionsanweisung einer Routine einzusehen, die einem anderen Benutzer gehört. Weitere Informationen finden Sie unter Datenbankberechtigungen.
- Verbesserung der Replikation — DB-Instances der MariaDB-Version 11.4 unterstützen die Binlog-Indizierung. Sie können für jede Binlog-Datei einen GTID Index erstellen. Diese Indizes verbessern die Leistung der Replikation, indem sie die Zeit reduzieren, die benötigt wird, um eine zu finden. GTID Weitere Informationen finden Sie unter Binlog-Indizierung.

- Veraltete oder entfernte Parameter — Die folgenden Parameter wurden für MariaDB-Instanzen der Version 11.4 als veraltet oder entfernt:
  - `engine_condition_pushdown`wurde aus optimizer_switch entfernt
  - innodb_change_buffer_max_size
  - innodb_defragmentieren
  - `TLSv1.0`und werden aus tls_version entfernt `TLSv1.1`
- Neue Standardwerte für einen Parameter — Der Standardwert des Parameters innodb_undo_tablespaces wurde von zu geändert. `0` `3`
- Neue gültige Werte für Parameter — Die folgenden Parameter haben neue gültige Werte für DB-Instances der Version 11.4 von MariaDB:
  - Zu den gültigen Werten für den Parameter binlog_row_image gehören jetzt. `FULL_NODUP`
  - Zu den gültigen Werten für den Parameter _ gehören jetztOLD. MODE `NO_NULL_COLLATION_IDS`
- Neue Parameter — Die folgenden Parameter sind neu für DB-Instances der Version 11.4 von MariaDB:
  - Der Parameter transaction_isolation ersetzt den Parameter tx_isolation.
  - Der Parameter transaction_read_only ersetzt den Parameter tx_read_only.
  - Der Parameter block_encryption_mode definiert den Standard-Blockverschlüsselungsmodus für die Funktionen _ () und _ (). AES ENCRYPT AES DECRYPT
  - Der character_set_collations definiert Überschreibungen für Zeichensatz-Standardkollationen.
  - binlog_gtid_index, binlog_gtid_index_page_size und binlog_gtid_index_span_min definieren die Eigenschaften des Binlog-Indexes. GTID Weitere Informationen finden Sie unter Binlog-Indizierung.

Eine Liste aller Funktionen von MariaDB 11.4 und deren Dokumentation finden Sie unter Änderungen und Verbesserungen in MariaDB 11.4 und Versionshinweise — MariaDB 11.4-Serie auf der MariaDB-Website.

Eine Liste der nicht unterstützten Funktionen finden Sie unter MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS.

MariaDB 10.11-Unterstützung bei Amazon RDS

Amazon RDS unterstützt die folgenden neuen Funktionen für Ihre DB-Instances, auf denen MariaDB Version 10.11 oder höher ausgeführt wird.

- Password Reuse Check Plugin – Sie können das MariaDB-Plug-in Password Reuse Check verwenden, um zu verhindern, dass Benutzer Passwörter wiederverwenden, und um den Aufbewahrungszeitraum für Passwörter festzulegen. Weitere Informationen finden Sie unter Password Reuse Check Plugin.
- GRANTPUBLICTO-Autorisierung — Sie können allen Benutzern, die Zugriff auf Ihren Server haben, Rechte gewähren. Weitere Informationen finden Sie unter GRANTTO PUBLIC.
- Trennung von READ ONLY ADMIN Rechten SUPER und Rechten — Sie können allen Benutzern READ ONLY ADMIN Rechte entziehen, auch Benutzern, die zuvor SUPER über Berechtigungen verfügten.
- Sicherheit – Sie können jetzt die Option `--ssl` als Standard für Ihren MariaDB-Client festlegen. MariaDB wird nicht mehr stillschweigend deaktiviert, SSL wenn die Konfiguration falsch ist.
- SQLBefehle und Funktionen — Sie können jetzt den `SHOW ANALYZE FORMAT=JSON` Befehl und die Funktionen`ROW_NUMBER`, und verwenden. `SFORMAT` `RANDOM_BYTES` `SFORMAT`ermöglicht die Formatierung von Zeichenketten und ist standardmäßig aktiviert. Sie können mit einem einzigen Befehl Partitionen in Tabellen und Tabellen in Partitionen konvertieren. Es gibt auch mehrere Verbesserungen der `JSON_*()`-Funktionen. Die Funktionen `DES_ENCRYPT` und `DES_DECRYPT` wurden für Version 10.10 und höher als veraltet eingestuft. Weitere Informationen finden Sie unter SFORMAT.
- InnoDB-Verbesserungen – Diese Verbesserungen umfassen Folgendes:
  - Leistungsverbesserungen im Redo-Protokoll, um die Write Amplification zu reduzieren und die Gleichzeitigkeit zu verbessern
  - Die Möglichkeit, den Undo-Tablespace zu ändern, ohne das Datenverzeichnis neu zu initialisieren. Diese Verbesserung reduziert den Overhead auf der Steuerebene. Nach Änderung des Undo-Tablespace ist ein Neustart, jedoch keine Neuinitialisierung erforderlich.
  - Support für `CHECK TABLE … EXTENDED` und intern für absteigende Indizes
  - Verbesserungen bei Masseneinfügungen
- Binlog-Änderungen – Diese Änderungen umfassen Folgendes:
  - Protokollierung von `ALTER` in zwei Phasen, um die Replikationslatenz zu verringern. Der Parameter `binlog_alter_two_phase` ist standardmäßig deaktiviert, kann aber über Parametergruppen aktiviert werden.
  - Protokollierung von `explicit_defaults_for_timestamp`
  - Keine Protokollierung mehr von `INCIDENT_EVENT`, wenn die Transaktion sicher rückgängig gemacht werden kann

- Verbesserungen bei der Replikation s — MariaDB Version 10.11 DB-Instances verwenden standardmäßig die GTID Replikation, wenn der Master sie unterstützt. Darüber hinaus ist `Seconds_Behind_Master` genauer.
- Clients – Sie können neue Befehlszeilenoptionen für `mysqlbinglog` und `mariadb-dump` verwenden. Sie können `mariadb-dump` verwenden, um historische Daten zu speichern und wiederherzustellen.
- System-Versionsverwaltung – Sie können den Verlauf ändern. MariaDB erstellt automatisch neue Partitionen.
- Atomar DDL — **`CREATE OR REPLACE`** ist jetzt atomar. Entweder ist die Aussage erfolgreich oder sie wird komplett rückgängig gemacht.
- Wiederholungsprotokoll-Schreibvorgang – Das Wiederholungsprotokoll schreibt asynchron.
- Gespeicherte Funktionen – Gespeicherte Funktionen unterstützen jetzt die gleichen `IN`-, `OUT`- und `INOUT`-Parameter wie in gespeicherten Prozeduren.
- Veraltete oder entfernte Parameter – Die folgenden Parameter wurden für DB-Instances in MariaDB Version 10.11 als veraltet eingestuft oder entfernt:
  - innodb_change_buffering
  - innodb_disallow_writes
  - innodb_log_write_ahead_size
  - innodb_prefix_index_cluster_optimization
  - keep_files_on_create
  - old
- Dynamische Parameter – Die folgenden Parameter sind jetzt für MariaDB-Instances der Version 10.11 dynamisch:
  - innodb_log_file_size
  - innodb_write_io_threads
  - innodb_read_io_threads
- Neue Standardwerte für Parameter – Die folgenden Parameter haben neue Standardwerte für MariaDB-Instances der Version 10.11:
  - Der Standardwert für den Parameter explicit_defaults_for_timestamp wurde von `OFF` in `ON` geändert.
  - Der Standardwert für den Parameter optimizer_prune_level wurde von 1 in 2 geändert.

- Neue gültige Werte für Parameter – Die folgenden Parameter haben neue gültige Werte für MariaDB-Instances der Version 10.11:
  - Die gültigen Werte für den Parameter old wurden mit denen für den Parameter old_mode zusammengeführt.
  - Die gültigen Werte für den Parameter histogram_type beinhalten jetzt `JSON_HB`.
  - Der gültige Wertebereich für den Parameter innodb_log_buffer_size liegt jetzt zwischen `262144` und `4294967295` (256 KB bis 4 096 MB).
  - Der gültige Wertebereich für den Parameter innodb_log_file_size liegt jetzt zwischen `4194304` und `512GB` (4 MB bis 512 GB).
  - Die gültigen Werte für den Parameter optimizer_prune_level beinhalten jetzt `2`.
- Neue Parameter – Die folgenden Parameter sind für MariaDB-Instances der Version 10.11 neu:
  - Der Parameter binlog_alter_two_phase kann die Replikationsleistung verbessern.
  - Der Parameter log_slow_min_examined_row_limit kann die Leistung verbessern.
  - Der Parameter log_slow_query und der Parameter log_slow_query_file sind Aliase für `slow_query_log` bzw. `slow_query_log_file`.
  - optimizer_extra_pruning_depth
  - system_versioning_insert_history

Eine Liste aller Funktionen von MariaDB 10.11 und deren Dokumentation finden Sie unter Änderungen und Verbesserungen in MariaDB 10.11 und Versionshinweise — MariaDB 10.11-Serie auf der MariaDB-Website.

Eine Liste der nicht unterstützten Funktionen finden Sie unter MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS.

MariaDB 10.6-Unterstützung bei Amazon RDS

Amazon RDS unterstützt die folgenden neuen Funktionen für Ihre DB-Instances, auf denen MariaDB Version 10.6 oder höher ausgeführt wird:

- MyRocks Speicher-Engine — Sie können die MyRocks Speicher-Engine mit RDS für MariaDB verwenden, um den Speicherverbrauch Ihrer schreibintensiven, leistungsstarken Webanwendungen zu optimieren. Weitere Informationen erhalten Sie unter Unterstützte Speicher-Engines für MariaDB bei Amazon RDS und MyRocks.

- AWS Identity and Access Management (IAM) DB-Authentifizierung — Sie können die IAM DB-Authentifizierung für mehr Sicherheit und zentrale Verwaltung der Verbindungen zu Ihren MariaDB-DB-Instances verwenden. Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.
- Upgrade-Optionen — Sie können jetzt von jeder früheren Hauptversion (10.3, 10.4, 10.5) auf MariaDB Version 10.6 aktualisieren. RDS Sie können auch einen Snapshot einer vorhandenen My SQL 5.6- oder 5.7-DB-Instance auf einer MariaDB 10.6-Instance wiederherstellen. Weitere Informationen finden Sie unter Upgrades der MariaDB-DB-Engine.
- Verzögerte Replikation – Sie können jetzt einen konfigurierbaren Zeitraum festlegen, für den ein Lesereplikat hinter der Quelldatenbank zurückbleibt. In einer Standard-MariaDB-Replikationskonfiguration gibt es eine minimale Replikationsverzögerung zwischen der Quelle und dem Replikat. Bei verzögerter Replikation können Sie eine absichtliche Verzögerung als Strategie für die Notfallwiederherstellung festlegen. Weitere Informationen finden Sie unter Konfigurieren der verzögerten Replikation mit MariaDB.
- Oracle SQL PL/Kompatibilität — Durch die Verwendung von Version 10.6 RDS für MariaDB können Sie Ihre älteren Oracle-Anwendungen einfacher zu Amazon migrieren. RDS Weitere Informationen finden Sie unter _ =. SQL MODE ORACLE
- Atomic DDL — Ihre Dynamic Data Language (DDL) -Anweisungen können mit MariaDB Version 10.6 relativ RDS absturzsicher sein. `CREATE TABLE`,, `ALTER TABLERENAME TABLE`, und verwandte Anweisungen sind `DROP TABLE` jetzt `DROP DATABASE` atomar. DDL Entweder ist die Aussage erfolgreich oder sie ist komplett rückgängig gemacht. Weitere Informationen finden Sie unter Atomic DDL.
- Weitere Verbesserungen — Zu diesen Verbesserungen gehören eine `JSON_TABLE` Funktion zur Umwandlung von JSON Daten in ein relationales Format innerhalb von SQL Innodb und ein schnelleres Laden leerer Tabellendaten. Dazu gehören auch das neue `sys_schema` für die Analyse und Fehlerbehebung, die Optimierungserweiterung für das Ignorieren nicht verwendeter Indizes und Leistungsverbesserungen. Weitere Informationen finden Sie unter JSON _. TABLE
- Neue Standardwerte für Parameter – Die folgenden Parameter haben neue Standardwerte für MariaDB-Instances der Version 10.6:
  - Der Standardwert für die folgenden Parameter hat sich von `utf8` in `utf8mb3` geändert:
    - character_set_client
    - character_set_connection
    - character_set_results
    - character_set_system

Obwohl sich die Standardwerte für diese Parameter geändert haben, gibt es keine funktionale Änderung. Weitere Informationen finden Sie unter Unterstützte Zeichensätze und Sortierungen in der MariaDB-Dokumentation.

- Der Standardwert des Parameters `collation_connection` hat sich von `utf8_general_ci` in `utf8mb3_general_ci` geändert. Obwohl sich der Standardwert für diesen Parameter geändert hat, gibt es keine funktionale Änderung.
- Der Standardwert des `old_mode`-Parameters wurde von unset in `UTF8_IS_UTF8MB3` geändert. Obwohl sich der Standardwert für diesen Parameter geändert hat, gibt es keine funktionale Änderung.

Eine Liste aller Funktionen von MariaDB 10.6 und die entsprechende Dokumentation finden Sie unter Changes and improvements in MariaDB 10.6 (Änderungen und Verbesserungen in MariaDB 10.6) und unter Release notes – MariaDB 10.6 (Versionshinweise – MariaDB 10.6) auf der MariaDB-Website.

Eine Liste der nicht unterstützten Funktionen finden Sie unter MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS.

MariaDB 10.5-Unterstützung bei Amazon RDS

Amazon RDS unterstützt die folgenden neuen Funktionen für Ihre DB-Instances, auf denen MariaDB Version 10.5 oder höher ausgeführt wird:

- InnoDB-Verbesserungen – MariaDB-Version 10.5 enthält Verbesserungen von InnoDB. Weitere Informationen finden Sie unter InnoDB: Performance-Verbesserungen usw. in der MariaDB-Dokumentation.
- Aktualisierungen des Leistungsschemas – MariaDB Version 10.5 enthält Aktualisierungen des Leistungsschemas. Weitere Informationen finden Sie unter Performance Schema Updates to Match My SQL 5.7 Instrumentation and Tables in der MariaDB-Dokumentation.
- Eine Datei im Inno DB-Redo-Log – In MariaDB-Versionen vor Version 10.5 wurde der Wert des Parameters `innodb_log_files_in_group` auf 2 festgelegt. In MariaDB Version 10.5 ist der Wert dieses Parameters auf festgeleg 1.

  Wenn Sie von einer früheren Version auf MariaDB Version 10.5 upgraden und die Parameter nicht ändern, bleibt der Parameterwert `innodb_log_file_size` unverändert. Es gilt jedoch für eine Protokolldatei statt für zwei. Das Ergebnis ist, dass Ihre aktualisierte MariaDB-Instance

der Version 10.5 die Hälfte der Redo-Protokollgröße verwendet, die sie vor dem Upgrade verwendet hat. Diese Änderung kann sich spürbar auf die Leistung auswirken. Um dieses Problem anzugehen, können Sie den Wert des Parameters `innodb_log_file_size` verdoppeln. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

- SHOWSLAVESTATUSBefehl nicht unterstützt — In Versionen von MariaDB vor Version 10.5 erforderte der `SHOW SLAVE STATUS` Befehl die Berechtigung. `REPLICATION SLAVE` In MariaDB Version 10.5 benötigt der äquivalente`SHOW REPLICA STATUS` Befehl das `REPLICATION REPLICA ADMIN`-Privileg. Dieses neue Recht wird dem RDS Masterbenutzer nicht gewährt.

  Anstatt den Befehl `SHOW REPLICA STATUS` zu verwenden, führen Sie den neuen gespeicherten Prozess `mysql.rds_replica_status` aus, um ähnliche Informationen zurückzugeben. Weitere Informationen finden Sie unter mysql.rds_replica_status.
- SHOWRELAYLOGEVENTSBefehl nicht unterstützt — In Versionen von MariaDB vor Version 10.5 erforderte der `SHOW RELAYLOG EVENTS` Befehl die Berechtigung. `REPLICATION SLAVE` In MariaDB Version 10.5 benötigt dieser Befehl das Privileg `REPLICATION REPLICA ADMIN`. Dieses neue Recht wird dem RDS Masterbenutzer nicht gewährt.
- Neue Standardwerte für Parameter – Die folgenden Parameter haben neue Standardwerte für MariaDB-Instances der Version 10.5:
  - Der Standardwert des Parameters max_connections hat sich in `LEAST({DBInstanceClassMemory/25165760},12000)` geändert. Hinweise zur Parameterfunktion `LEAST` finden Sie unter DB-Parameter-Funktionen.
  - Der Standardwert des Parameters innodb_adaptive_hash_index wurde in `OFF` (`0`) geändert.
  - Der Standardwert des Parameters innodb_checksum_algorithm hat sich in `full_crc32` geändert.
  - Der Standardwert des Parameters innodb_log_file_size wurde auf 2 GB geändert.

Eine Liste aller Funktionen von MariaDB 10.5 und die entsprechende Dokumentation finden Sie unter Changes and improvements in MariaDB 10.5 (Änderungen und Verbesserungen in MariaDB 10.5) und unter Release notes – MariaDB 10.5 (Versionshinweise – MariaDB 10.5) auf der MariaDB-Website.

Eine Liste der nicht unterstützten Funktionen finden Sie unter MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS.

MariaDB 10.4-Unterstützung bei Amazon RDS

Amazon RDS unterstützt die folgenden neuen Funktionen für Ihre DB-Instances, auf denen MariaDB Version 10.4 oder höher ausgeführt wird:

- Verbesserungen bei der Sicherheit von Benutzerkonten – Verbesserungen beim Passwortablauf und bei der Kontosperrung
- Optimizer-Verbesserungen – Optimizer-Ablaufverfolgungsfunktion
- InnoDB-Verbesserungen — Sofortige DROP COLUMN Unterstützung und sofortige `VARCHAR` Erweiterung für `ROW_FORMAT=DYNAMIC` und `ROW_FORMAT=COMPACT`
- Neue Parameter – Einschließlich tcp_nodedelay, tls_version und gtid_cleanup_batch_size

Eine Liste aller Funktionen von MariaDB 10.4 und die entsprechende Dokumentation finden Sie unter Änderungen und Verbesserungen in MariaDB 10.4 und Versionshinweise – MariaDB 10.4 Serie auf der MariaDB-Website.

Eine Liste der nicht unterstützten Funktionen finden Sie unter MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS.

Unterstützte Speicher-Engines für MariaDB bei Amazon RDS

RDSfor MariaDB unterstützt die folgenden Speicher-Engines.

Themen

- Die InnoDB-Speicher-Engine
- Die Speicher-Engine MyRocks

Andere Speicher-Engines werden derzeit nicht von RDS for MariaDB unterstützt.

Die InnoDB-Speicher-Engine

MariaDB unterstützt zwar mehrere Speicher-Engines mit unterschiedlichen Fähigkeiten und Kapazitäten, jedoch sind nicht alle von ihnen für die Wiederherstellung und für Datenbeständigkeit optimiert. InnoDB ist die empfohlene Speicher-Engine für MariaDB-DB-Instances bei Amazon. RDS RDSAmazon-Funktionen wie point-in-time Wiederherstellung und Snapshot-Wiederherstellung erfordern eine wiederherstellbare Speicher-Engine und werden nur für die empfohlene Speicher-Engine für die MariaDB-Version unterstützt.

Weitere Informationen finden Sie unter InnoDB.

Die Speicher-Engine MyRocks

Die MyRocks Speicher-Engine ist RDS für MariaDB Version 10.6 und höher verfügbar. Bevor Sie die MyRocks Speicher-Engine in einer Produktionsdatenbank verwenden, empfehlen wir Ihnen, gründliche Benchmarks und Tests durchzuführen, um alle potenziellen Vorteile gegenüber InnoDB für Ihren Anwendungsfall zu überprüfen.

Die Standardparametergruppe für MariaDB Version 10.6 umfasst Parameter. MyRocks Weitere Informationen erhalten Sie unter Parameter für MariaDB und Parametergruppen für Amazon RDS.

Um eine Tabelle zu erstellen, die die MyRocks Speicher-Engine verwendet, geben Sie `ENGINE=RocksDB` in der Anweisung Folgendes an. `CREATE TABLE` Im folgenden Beispiel wird eine Tabelle erstellt, die die MyRocks Speicher-Engine verwendet.

```
CREATE TABLE test (a INT NOT NULL, b CHAR(10)) ENGINE=RocksDB;
```

Wir empfehlen dringend, keine Transaktionen auszuführen, die sich sowohl auf InnoDB als auch auf MyRocks Tabellen erstrecken. MariaDB garantiert nicht ACID (Atomarität, Konsistenz, Isolierung, Haltbarkeit) für Transaktionen zwischen Speicher-Engines. Obwohl es möglich ist, sowohl InnoDB als auch MyRocks Tabellen in einer DB-Instance zu haben, empfehlen wir diesen Ansatz nur bei einer Migration von einer Speicher-Engine zur anderen. Wenn sowohl InnoDB als auch MyRocks Tabellen in einer DB-Instance vorhanden sind, hat jede Speicher-Engine ihren eigenen Pufferpool, was zu Leistungseinbußen führen kann.

MyRocks unterstützt keine `SERIALIZABLE` Isolations- oder Gap-Locks. Daher ist die Verwendung bei der anweisungsbasierten MyRocks Replikation im Allgemeinen nicht möglich. Weitere Informationen finden Sie unter MyRocks und Replikation.

Derzeit können Sie nur die folgenden MyRocks Parameter ändern:

- `rocksdb_block_cache_size`
- `rocksdb_bulk_load`
- `rocksdb_bulk_load_size`
- `rocksdb_deadlock_detect`
- `rocksdb_deadlock_detect_depth`
- `rocksdb_max_latest_deadlocks`

Die MyRocks Speicher-Engine und die InnoDB-Speicher-Engine können auf der Grundlage der Einstellungen für die `innodb_buffer_pool_size` Parameter `rocksdb_block_cache_size` und um Speicher konkurrieren. In einigen Fällen beabsichtigen Sie möglicherweise, die MyRocks Speicher-Engine nur auf einer bestimmten DB-Instance zu verwenden. Wenn dies der Fall ist, empfehlen wir, den `innodb_buffer_pool_size minimal`-Parameter auf einen minimalen Wert und das `rocksdb_block_cache_size` so hoch wie möglich zu setzen.

Sie können mithilfe der `DownloadDBLogFilePortion`Operationen `DescribeDBLogFiles`und auf MyRocks Protokolldateien zugreifen.

Weitere Informationen zu MyRocks finden Sie MyRocksauf der MariaDB-Website.

Cache-Erwärmung für MariaDB bei Amazon RDS

InnoDB Cache-Warming kann Leistungssteigerungen für Ihre MariaDB-DB-Instance zur Verfügung stellen, indem es den aktuellen Zustand des Zwischenspeicher-Pools speichert, wenn die DB-Instance deaktiviert ist und dann den Zwischenspeicher-Pool von den gespeicherten Informationen lädt, wenn die DB-Instance gestartet wird. Dieser Ansatz umgeht die Notwendigkeit, dass sich der Zwischenspeicher-Pool für den normalen Datenbankbetrieb „erwärmen“ muss und stattdessen wird der Zwischenspeicher-Pool mit den Seiten für bekannte häufige Anfragen geladen. Weitere Informationen über die Cache-Initialisierung finden Sie unter Dumping und Wiederherstellung des Buffer-Pools in der MariaDB-Dokumentation.

Die Cache-Initialisierung ist standardmäßig für MariaDB 10.3 und höhere DB-Instances aktiviert. Um die Cache-Initialisierung zu aktivieren, setzen Sie die Parameter `innodb_buffer_pool_dump_at_shutdown` und `innodb_buffer_pool_load_at_startup` in der Parametergruppe für Ihre DB-Instance auf 1. Das Ändern dieser Parameterwerte in einer Parametergruppe betrifft alle MariaDB DB-Instancen, die die gleiche Parametergruppe verwenden. Um die Cache-Initialisierung für spezifische MariaDB-DB-Instances zu aktivieren, müssen Sie möglicherweise eine neue Parametergruppe für diese DB-Instances erstellen. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Cache-Warming erzeugt vor allem einen Leistungsvorteil für DB-Instances, die Standardspeicher verwenden. Wenn Sie PIOPS Speicher verwenden, stellen Sie in der Regel keinen signifikanten Leistungsvorteil fest.

Important

Wenn Ihre MariaDB DB-Instance nicht normal herunterfährt, wie bei einem Failover, dann ist der Zwischenspeicher-Pool nicht auf der Festplatte gespeichert. In diesem Fall lädt MariaDB einen verfügbaren Zwischenspeicher-Pool, wenn die DB-Instance gestartet wird. Das ist nicht schlimm, aber der wiederhergestellte Zwischenspeicher-Pool spiegelt möglicherweise nicht den aktuellsten Stand des Zwischenspeicher-Pools vor dem Neustart dar. Wir empfehlen Ihnen Ihren Bufferpool in regelmäßigen Abschnitten in Ihrem Interesse zu verwerfen, um sicherzustellen, dass Sie immer den aktuellsten Zustand in Ihrem Bufferpool für die Initialisierung des Cache beim Starten von haben. Sie können den Zwischenspeicher-Pool auf Abruf laden oder entladen.
Sie können ein Ereignis zum automatischen Entladen des Zwischenspeicher-Pools in regelmäßigen Abständen erstellen. Beispielsweise erstellt das folgende Statement ein Ereignis mit dem Namen `periodic_buffer_pool_dump`, das den Bufferpool stündlich verwirft.

```
CREATE EVENT periodic_buffer_pool_dump
   ON SCHEDULE EVERY 1 HOUR
   DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Weitere Informationen finden Sie unter Events in der MariaDB-Dokumentation.

Entladen und Laden des Zwischenspeicher-Pools auf Abruf

Sie können den Cache auf Abruf mit den folgenden gespeicherten Prozeduren speichern und laden:

- Rufen Sie die gespeicherte Prozedur mysql.rds_innodb_buffer_pool_dump_now auf, um den aktuellen Zustand des Bufferpools auf der Festplatte zu verwerfen.
- Rufen Sie die gespeicherte Prozedur mysql.rds_innodb_buffer_pool_load_now auf, um den Zustand des Bufferpools auf der Festplatte zu laden oder zu speichern.
- Rufen Sie die gespeicherte Prozedur mysql.rds_innodb_buffer_pool_load_abort auf, um eine ladende Operation abzubrechen.

MariaDB-Funktionen, die von Amazon nicht unterstützt werden RDS

Die folgenden MariaDB-Funktionen werden auf Amazon nicht unterstützt: RDS

- S3-Speicher-Engine
- Authentifizierungs-Plugin — GSSAPI
- Authentifizierungs-Plugin – Unix Socket
- AWS Verschlüsselungs-Plugin für die Schlüsselverwaltung
- Verzögerte Replikation für MariaDB-Versionen unter 10.6
- Native MariaDB-Verschlüsselung im Ruhezustand für InnoDB und Aria

  Sie können die Verschlüsselung von Daten im Ruhezustand für eine MariaDB-DB-Instance durch Befolgen der Anleitungen unter aktiviere Verschlüsseln von Amazon RDS-Ressourcen.
- HandlerSocket
- JSONTabellentyp für MariaDB-Versionen unter 10.6
- MariaDB ColumnStore
- MariaDB Galera-Cluster
- Replikation aus mehreren Quellen
- MyRocks Speicher-Engine für MariaDB-Versionen unter 10.6
- Plugin zur Passwortvalidierung und `cracklib_password_check` für MariaDB-Versionen unter 11.4 `simple_password_check`
- Spider-Speicher-Engine
- Sphinx-Speicher-Engine
- TokuDB-Speicher-Engine
- Speicher-Engine-spezifische Objektattribute, wie in Engine-definierte neue Table/Field/Index Attribute in der MariaDB-Dokumentation beschrieben
- Tabellen- und Tablespace-Verschlüsselung
- Hashicorp Key Management Plugin
- Parallele Ausführung von zwei Upgrades

Um ein Managed-Service-Erlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances und schränkt den Zugriff auf bestimmte Systemprozeduren und Tabellen ein, für die erweiterte Rechte erforderlich sind. Amazon RDS unterstützt den Zugriff auf Datenbanken in einer DB-Instance mit jeder SQL Standard-Client-Anwendung. Amazon erlaubt RDS keinen direkten Hostzugriff auf eine DB-Instance mithilfe von Telnet, Secure Shell (SSH) oder Windows Remote Desktop Connection.

MariaDB auf Amazon-RDS-Versionen

In MariaDB werden die Versionsnummern als Version X.Y.Z organisiert. In der Amazon RDS-Terminologie bezeichnet X.Y die Hauptversion und Z ist die Nummer der Unterversion. Bei Amazon RDS-Implementierungen gilt ein Versionswechsel als wesentlich, wenn sich die Hauptversionsnummer ändert, z. B. von Version 10.5 auf 10.6. Eine Versionsänderung gilt als geringfügig, wenn sich nur die Nebenversionsnummer ändert, z. B. wenn von Version 10.6.14 auf 10.6.16 umgestellt wird.

Themen

- Unterstützte MariaDB-Nebenversionen in Amazon RDS
- Unterstützte MariaDB-Hauptversionen in Amazon RDS
- Arbeiten mit der Datenbank-Vorschauumgebung
- MariaDB Version 11.8 in der Database Preview-Umgebung
- MariaDB Version 11.7 in der Database Preview-Umgebung
- MariaDB Version 11.4 in der Database Preview-Umgebung
- Veraltete Versionen für Amazon RDS for MariaDB

Unterstützte MariaDB-Nebenversionen in Amazon RDS

Amazon RDS unterstützt derzeit die folgenden MariaDB-Nebenversionen.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.

Die folgende Tabelle zeigt die Nebenversionen von MariaDB 11.4, die Amazon RDS derzeit unterstützt.

MariaDB-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
11.4.8	06. August 2025	13. August 2025	September 2026

MariaDB-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
11.4.7	22. Mai 2025	4. Juni 2025	September 2026
11.4.5	4. Februar 2025	24. Februar 2025	März 2026
11.4.4	1. November 2024	20. Dezember 2024	März 2026
11.4.3	8. August 2024	15. Oktober 2024	März 2026

Die folgende Tabelle zeigt die Nebenversionen von MariaDB 10.11, die Amazon RDS derzeit unterstützt.

MariaDB-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
10.11.14	06. August 2025	13. August 2025	September 2026
10.11.13	22. Mai 2025	4. Juni 2025	September 2026
10.11.11	4. Februar 2025	24. Februar 2025	März 2026
10.11.10	1. November 2024	20. Dezember 2024	März 2026
10.11.9	8. August 2024	4. September 2024	März 2026
10.11.8	16. Mai 2024	14. Juni 2024	Oktober 2025

Die folgende Tabelle zeigt die Nebenversionen von MariaDB 10.6, die Amazon RDS derzeit unterstützt.

MariaDB-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
10.6.23	6. August 2025	13. August 2025	August 2026
10.6.22	6. Mai 2025	20. Mai 2025	Juli 2026
10.6.21	4. Februar 2025	24. Februar 2025	März 2026
10.6.20	1. November 2024	20. Dezember 2024	März 2026
10.6.19	8. August 2024	4. September 2024	März 2026
10.6.18	16. Mai 2024	14. Juni 2024	Oktober 2025

Die folgende Tabelle zeigt die Nebenversionen von MariaDB 10.5, die Amazon RDS derzeit unterstützt.

MariaDB-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
10.5.29	6. Mai 2025	20. Mai 2025	Februar 2026
10.5.28	4. Februar 2025	24. Februar 2025	Februar 2026
10.5.27	1. November 2024	20. Dezember 2024	Februar 2026
10.5.26	8. August 2024	4. September 2024	Oktober 2025
10.5.25	16. Mai 2024	14. Juni 2024	Oktober 2025

Die folgende Tabelle zeigt die Nebenversionen von MariaDB 10.4, die Amazon RDS derzeit unterstützt.

MariaDB-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
10.4.34	16. Mai 2024	14. Juni 2024	28. Februar 2025
10.4.33	7. Februar 2024	26. Februar 2024	28. Februar 2025
10.4.32	13. November 2023	12. Dezember 2023	28. Februar 2025
10.4.31	14. August 2023	7. September 2023	28. Februar 2025
10.4,30	7. Juni 2023	22. Juni 2023	28. Februar 2025
10.4,29	10. Mai 2023	15. Juni 2023	28. Februar 2025

Sie können alle aktuell unterstützten MariaDB-Versionen beim Erstellen einer neuen DB-Instanz angeben. Sie können die Hauptversionen (wie z. B. MariaDB 10.5) sowie eine beliebige unterstützte Unterversion für die festgelegte Hauptversion festlegen. Wenn keine Version angegeben wird, verwendet Amazon RDS standardmäßig eine unterstützte Version - in der Regel die aktuelle Version. Wenn die Hauptversion, jedoch nicht die Unterversion, festgelegt ist, verwendet Amazon RDS standardmäßig den letzten Release der Hauptversion, die Sie festgelegt haben. Verwenden Sie den Befehl, um eine Liste der unterstützten Versionen sowie die Standardeinstellungen für neu erstellte DB-Instances anzuzeigen. `describe-db-engine-versions` AWS CLI

Um beispielsweise die unterstützten Engine-Versionen für RDS für MariaDB aufzulisten, führen Sie den folgenden CLI-Befehl aus:

```
aws rds describe-db-engine-versions --engine mariadb --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Die Standard-MariaDB-Version kann je nach AWS-Region variieren. Um eine DB-Instance mit einer bestimmten Unterversion zu erstellen, geben Sie die Unterversion bei der Erstellung der DB-Instance an. Sie können die standardmäßige Nebenversion für eine ermitteln, AWS-Region indem Sie den folgenden AWS CLI Befehl ausführen:

```
aws rds describe-db-engine-versions --default-only --engine mariadb
 --engine-version major_engine_version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

*major_engine_version*Ersetzen Sie durch die Hauptversion der Engine und *region* ersetzen Sie sie durch die AWS-Region. Der folgende AWS CLI Befehl gibt beispielsweise die Standardversion der MariaDB-Nebenengine für die Hauptversion 10.5 und die USA West (Oregon) AWS-Region (us-west-2) zurück:

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version
 10.5 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --
output text
```

MariaDB-Nebenversionen auf Amazon RDS

Nebenversionen

- MariaDB versie 11.4.8
- MariaDB versie 11.4.7
- MariaDB versie 11.4.5
- MariaDB versie 11.4.4
- MariaDB versie 10.11.14
- MariaDB versie 10.11.13
- MariaDB versie 10.11.11
- MariaDB versie 10.11.10
- MariaDB versie 10.6.23
- MariaDB versie 10.6.22
- MariaDB versie 10.6.21
- MariaDB versie 10.6.20
- MariaDB versie 10.5.29
- MariaDB versie 10.5.28
- MariaDB versie 10.5.27

MariaDB versie 11.4.8

MariaDB Version 11.4.8 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Das `rds_security_audit` Datenbank-Plugin wurde hinzugefügt, ein internes Amazon RDS MariaDB-Plugin, das sicherheitsrelevante Metriken sammelt. Es ist kein Eingreifen des Kunden erforderlich.

MariaDB versie 11.4.7

MariaDB Version 11.4.7 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025b`.

MariaDB versie 11.4.5

MariaDB Version 11.4.5 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025a`.

MariaDB versie 11.4.4

MariaDB Version 11.4.4 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Zwei Änderungen der MariaDB-Community, die dazu führen, dass die point-in-time Wiederherstellung (PITR) fehlschlägt, wurden rückgängig gemacht. Weitere Informationen finden Sie in der Jira-Ausgabe von MariaDB Server MDEV-35528.

MariaDB versie 10.11.14

MariaDB Version 10.11.14 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

MariaDB versie 10.11.13

MariaDB Version 10.11.13 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025b`.

MariaDB versie 10.11.11

MariaDB Version 10.11.11 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025a`.

MariaDB versie 10.11.10

MariaDB Version 10.11.10 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Zwei Änderungen der MariaDB-Community, die dazu führen, dass die point-in-time Wiederherstellung (PITR) fehlschlägt, wurden rückgängig gemacht. Weitere Informationen finden Sie in der Jira-Ausgabe von MariaDB Server MDEV-35528.

MariaDB versie 10.6.23

MariaDB Version 10.6.23 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

MariaDB versie 10.6.22

MariaDB Version 10.6.22 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025b`.

MariaDB versie 10.6.21

MariaDB Version 10.6.21 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025a`.

MariaDB versie 10.6.20

MariaDB Version 10.6.20 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Zwei Änderungen der MariaDB-Community, die dazu führen, dass die point-in-time Wiederherstellung (PITR) fehlschlägt, wurden rückgängig gemacht. Weitere Informationen finden Sie in der Jira-Ausgabe von MariaDB Server MDEV-35528.

MariaDB versie 10.5.29

MariaDB Version 10.5.29 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025b`.

MariaDB versie 10.5.28

MariaDB Version 10.5.28 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren`tzdata2025a`.

MariaDB versie 10.5.27

MariaDB Version 10.5.27 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MariaDB-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Zwei Änderungen der MariaDB-Community, die dazu führen, dass die point-in-time Wiederherstellung (PITR) fehlschlägt, wurden rückgängig gemacht. Weitere Informationen finden Sie in der Jira-Ausgabe von MariaDB Server MDEV-35528.

Unterstützte MariaDB-Hauptversionen in Amazon RDS

Die Hauptversionen von RDS für MariaDB stehen mindestens bis zum Ende des Lebenszyklus der Community für die entsprechende Community-Version zur Verfügung. Sie können die folgenden Daten verwenden, um Ihre Test- und Upgrade-Zyklen zu planen. Wenn Amazon die Unterstützung für eine RDS für MariaDB-Version länger als ursprünglich geplant erweitert, planen wir, diese Tabelle zu aktualisieren, um das spätere Datum widerzuspiegeln.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.
> Sie können auch Informationen zu den Support-Daten für wichtige Engine-Versionen anzeigen, indem Sie den AWS CLI Befehl describe-db-major-engine-versions ausführen oder den API-Vorgang Describe RDS verwenden. DBMajor EngineVersions

MariaDB Hauptversion	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	Datum des Lebensendes der Gemeinschaft	RDS-Ende des Standard-Supportdatums
MariaDB 11.4	8. August 2024	15. Oktober 2024	Mai 2029	Mai 2029
MariaDB 10.11	16. Februar 2023	21. August 2023	16. Februar 2028	Februar 2028
MariaDB 10.6	6. Juli 2021	3. Februar 2022	6. Juli 2026	Juli 2026
MariaDB 10.5	24. Juni 2020	21. Januar 2021	24. Juni 2025	Februar 2026
MariaDB 10.4	18. Juni 2019	6. April 2020	18. Juni 2024	Februar 2025

Arbeiten mit der Datenbank-Vorschauumgebung

RDS für MariaDB-DB-Instances in der Database Preview-Umgebung ähneln funktionell anderen RDS für MariaDB-DB-Instances. Sie können die Datenbank-Vorschauumgebung jedoch nicht für Produktions-Workloads nutzen.

Für Vorschauumgebungen gelten folgende Einschränkungen:

- Amazon RDS löscht alle DB-Instances 60 Tage nach Erstellung zusammen mit allen Backups und Snapshots.
- Sie können nur Allzweck-SSD und bereitgestellte IOPS-SSD als Speicher verwenden.
- Bei DB-Instances erhalten Sie keine Hilfe. Support Stattdessen können Sie Ihre Fragen in der von uns AWS verwalteten Q&A-Community re:POST stellen.AWS
- Sie können einen Snapshot einer DB-Instance nicht in eine Produktionsumgebung kopieren.

Die folgenden Optionen werden von der Vorschauversion unterstützt.

- Sie können DB-Instances mithilfe von db.m6i-, db.r6i-, db.m6g-, db.m5-, db.t3-, db.r6g- und db.r5-DB-Instance-Klassen erstellen. Weitere Informationen zu RDS-Instance-Klassen erhalten Sie unter .
- Sie können sowohl Single-AZ- als auch Multi-AZ-Bereitstellungen verwenden.

- Sie können die standardmäßigen Dump- und Ladefunktionen von MariaDB verwenden, um Datenbanken aus der Database Preview-Umgebung zu exportieren oder Datenbanken in diese zu importieren.

Nicht in der Datenbank-Vorschauumgebung unterstützte Features

Die folgenden Features sind in der Datenbank-Vorschauumgebung nicht verfügbar:

- Regionsübergreifende Snapshot-Kopie
- Regionsübergreifende Lesereplikate
- RDS-Proxy

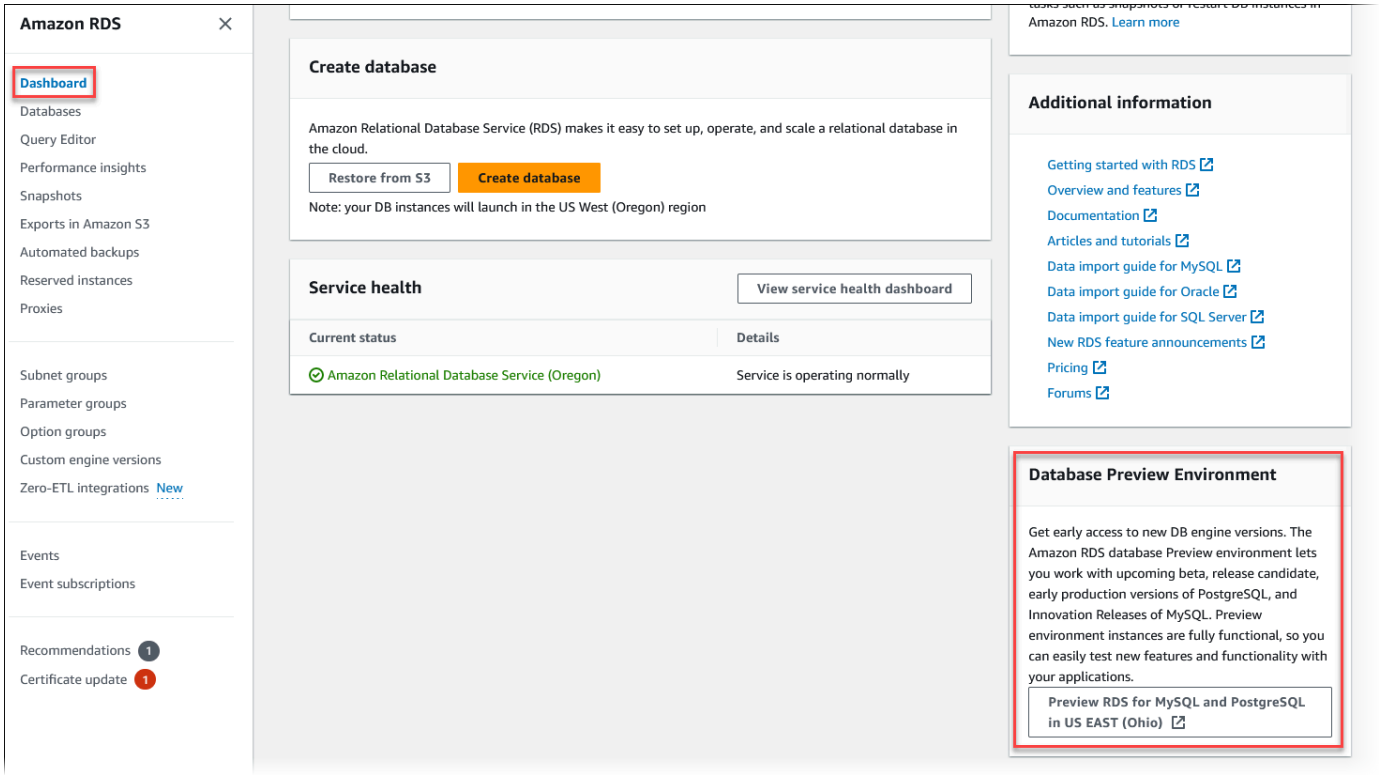
Erstellen einer neuen DB-Instance in der Datenbank-Vorschauumgebung

Sie können eine DB-Instance in der Database Preview-Umgebung mithilfe der AWS Management Console AWS CLI, oder RDS-API erstellen.

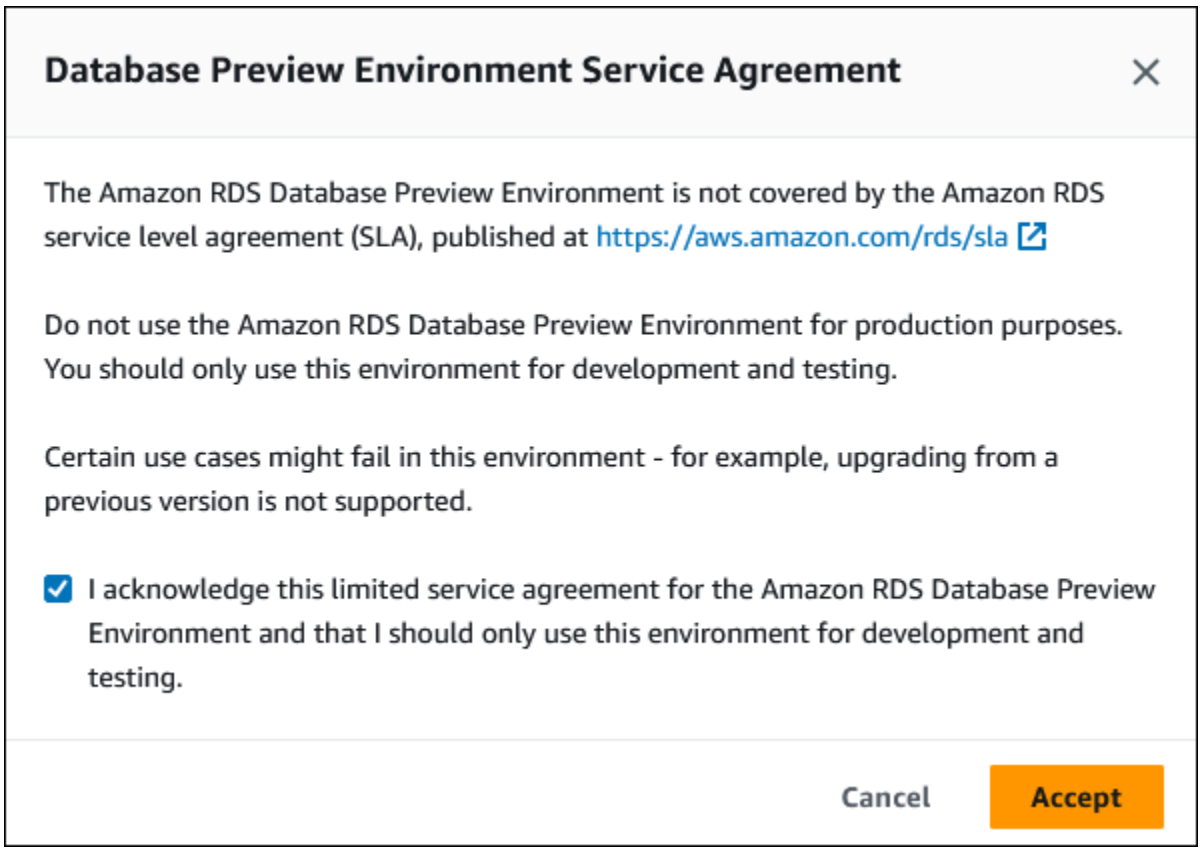
Konsole

So erstellen Sie eine DB-Instance in der Datenbank-Vorschauumgebung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Dashboard aus.
3. Suchen Sie auf der Seite Dashboard nach Datenbank-Preview-Umgebung, wie in der folgenden Abbildung gezeigt.



Sie können auch direkt zu Datenbank-Preview-Umgebung navigieren. Bevor Sie fortfahren können, müssen Sie die Einschränkungen bestätigen und akzeptieren.



4. Um die RDS for MariaDB-DB-Instance zu erstellen, gehen Sie genauso vor wie bei der Erstellung einer beliebigen Amazon RDS-DB-Instance. Weitere Informationen finden Sie im Verfahren Konsole unter Erstellen einer DB-Instance.

AWS CLI

Verwenden Sie den folgenden Endpunkt, um eine DB-Instance in der Database Preview-Umgebung mit dem AWS CLI zu erstellen.

```
rds-preview.us-east-2.amazonaws.com
```

Um die RDS for MariaDB-DB-Instance zu erstellen, gehen Sie genauso vor wie bei der Erstellung einer beliebigen Amazon RDS-DB-Instance. Weitere Informationen finden Sie im Verfahren AWS CLI unter Erstellen einer DB-Instance.

RDS-API

Verwenden Sie den folgenden Endpunkt, um eine DB-Instance in der Datenbank-Vorschauumgebung über die RDS-API zu erstellen.

```
rds-preview.us-east-2.amazonaws.com
```

Um die RDS for MariaDB-DB-Instance zu erstellen, gehen Sie genauso vor wie bei der Erstellung einer beliebigen Amazon RDS-DB-Instance. Weitere Informationen finden Sie im Verfahren RDS-API unter Erstellen einer DB-Instance.

MariaDB Version 11.8 in der Database Preview-Umgebung

MariaDB Version 11.8 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. MariaDB Version 11.8 enthält mehrere Verbesserungen, die in Änderungen und Verbesserungen in MariaDB 11.8 beschrieben sind. Diese Version beinhaltet auch Unterstützung für den Vektordatentyp, die Indizierung und die Suche. Weitere Informationen finden Sie unter Vector Overview in der MariaDB-Dokumentation.

Sie können die Database Preview-Umgebung verwenden, um Ihre Workloads anhand dieser Version zu testen, bevor sie in allen AWS-Regionen für Produktions-Workloads verfügbar ist. Weitere Informationen zur Database-Vorschauumgebung finden Sie unter the section called “Die Datenbank-Vorschauumgebung”. Um von der Konsole auf die Preview-Umgebung zuzugreifen, wählen Sie rds-preview/ aus.

MariaDB Version 11.7 in der Database Preview-Umgebung

MariaDB Version 11.7 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. MariaDB Version 11.7 enthält mehrere Verbesserungen, die in Änderungen und Verbesserungen in MariaDB 11.7 beschrieben sind. Diese Version beinhaltet auch Unterstützung für den Vektordatentyp, die Indizierung und die Suche. Weitere Informationen finden Sie unter Vector Overview in der MariaDB-Dokumentation.

Sie können die Database Preview-Umgebung verwenden, um Ihre Workloads anhand dieser Version zu testen, bevor sie in allen AWS-Regionen für Produktions-Workloads verfügbar ist. Weitere Informationen zur Database-Vorschauumgebung finden Sie unter the section called “Die Datenbank-Vorschauumgebung”. Um von der Konsole auf die Preview-Umgebung zuzugreifen, wählen Sie rds-preview/ aus.

MariaDB Version 11.4 in der Database Preview-Umgebung

MariaDB Version 11.4 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. MariaDB Version 11.4 enthält mehrere Verbesserungen, die in Änderungen und Verbesserungen in MariaDB

11.4 beschrieben sind. Sie können die Database Preview-Umgebung verwenden, um Ihre Workloads anhand dieser Version zu testen, bevor sie in allen für Produktions-Workloads verfügbar ist. AWS-Regionen

Weitere Informationen zur Database-Vorschauumgebung finden Sie unter the section called “Die Datenbank-Vorschauumgebung”. Um von der Konsole auf die Preview-Umgebung zuzugreifen, wählen Sie rds-preview/ aus.

Veraltete Versionen für Amazon RDS for MariaDB

Die Versionen 10.0, 10.1, 10.2 und 10.3 von Amazon RDS for MariaDB sind veraltet.

Informationen zur Amazon RDS-Verfallsrichtlinie für MariaDB finden Sie unter Amazon RDS. FAQs

Verbindung zu Ihrer MariaDB-DB-Instance herstellen

Nachdem Amazon RDS Ihre DB-Instance bereitgestellt hat, können Sie jede Standard-MariaDB-Client-Anwendung und jedes -Hilfsprogramm verwenden, um eine Verbindung mit der Instance herzustellen. In der Verbindungszeichenfolge geben Sie die DNS-Adresse (Domain Name System) vom DB-Instance-Endpunkt als Host-Parameter an. Sie geben auch die Portnummer vom DB-Instance-Endpunkt als Port-Parameter an.

Sie können sich mit einer Amazon-RDS-for-MariaDB-DB-Instance verbinden, indem Sie die MySQL-Befehlszeilenfunktion verwenden. Weitere Informationen zur Verwendung der MySQL-Befehlszeile finden Sie unter mysql Command-line Client in der MariaDB-Dokumentation. Eine GUI-basierte Anwendung, die Sie zum Herstellen einer Verbindung verwenden können, ist Heidi. Weitere Informationen finden Sie auf der Seite Download HeidiSQL. Weitere Informationen zum Installieren von MySQL (einschließlich des MySQL-Befehlszeile-Clients) finden Sie unter Installation und Aktualisierung von MySQL.

Die meisten Linux-Verteilung enthalten den MariaDB Client anstelle des Oracle MySQL Clients. Führen Sie den folgenden Befehl aus, um den MySQL-Befehlszeilenclient auf Amazon Linux 2023 zu installieren:

```
sudo dnf install mariadb105
```

Führen Sie den folgenden Befehl aus, um den MySQL-Befehlszeilenclient auf Amazon Linux 2 zu installieren:

```
sudo yum install mariadb
```

Führen Sie den folgenden Befehl aus, um den MySQL-Befehlszeilenclient auf den meisten DEB-basierten Linux-Distributionen zu installieren.

```
apt-get install mariadb-client
```

Zum Überprüfen der Version des Befehlszeilenclients von MySQL führen Sie den folgenden Befehl aus.

```
mysql --version
```

Zum Lesen der MySQL-Dokumentation für Ihre aktuelle Clientversion führen Sie den folgenden Befehl aus.

```
man mysql
```

Um eine Verbindung zu einer DB-Instance von außerhalb einer Virtual Private Cloud (VPC) basierend auf Amazon VPC herzustellen, muss die DB-Instance öffentlich zugänglich sein. Außerdem muss der Zugriff unter Verwendung der eingehenden Regeln der Sicherheitsgruppe der DB-Instance gewährt werden, und andere Anforderungen müssen erfüllt sein. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Sie können SSL-Verschlüsselung für Verbindungen mit einer MariaDB-DB-Instance verwenden. Weitere Informationen finden Sie unter SSL/TLSUnterstützung für MariaDB-DB-Instances auf Amazon RDS.

Informationen zum Suchen und Herstellen einer Verbindung zu einer RDS for MariaDB-DB-Instance finden Sie in den folgenden Themen.

Themen

- Finden der Verbindungsinformationen für eine MariaDB-DB-Instance
- Verbindung über den SQL Befehlszeilenclient My (unverschlüsselt) für MariaDB herstellen RDS
- Verbindung RDS zu MariaDB mit dem AWS JDBC Treiber und dem AWS Python-Treiber herstellen;
- Fehlerbehebung bei Verbindungen zu Ihrer MariaDB-DB-Instance

Finden der Verbindungsinformationen für eine MariaDB-DB-Instance

Die Verbindungsinformationen für eine DB-Instance umfassen ihren Endpunkt, ihren Port und einen gültigen Datenbankbenutzer, z. B. den Masterbenutzer. Nehmen wir zum Beispiel an, dass ein Endpunktwert laute `mydb.123456789012.us-east-1.rds.amazonaws.com`. In diesem Fall ist `3306` der Port-Wert und der Datenbankbenutzer ist `admin`. Angesichts dieser Informationen geben Sie die folgenden Werte in einer Verbindungszeichenfolge an:

- Geben Sie für Host oder DNS Hostname oder -namen an`mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Als Port `3306`.

- Geben Sie für Benutzer a `admin`.

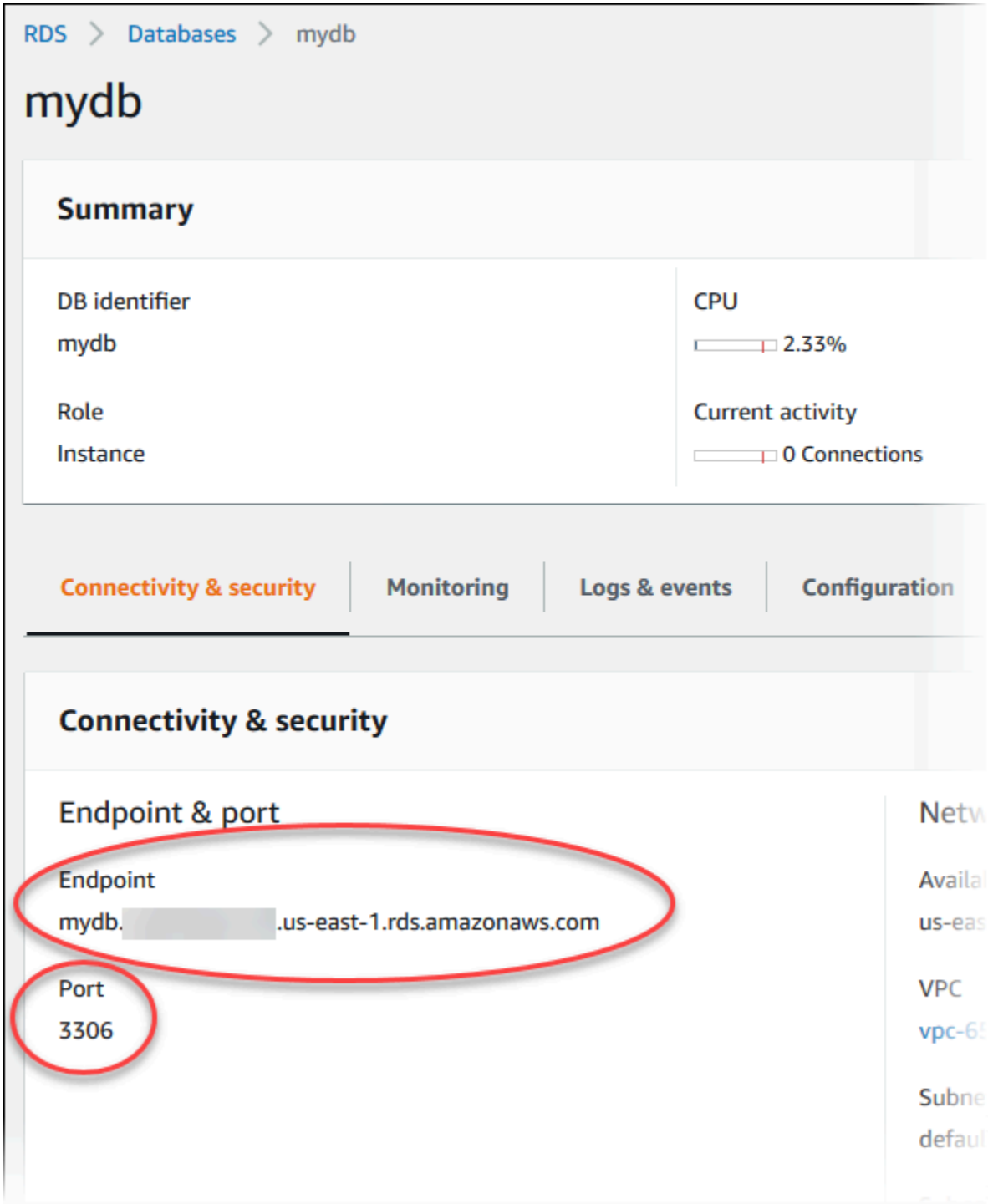
Um eine Verbindung zu einer DB-Instance herzustellen, verwenden Sie einen beliebigen Client für die MariaDB-DB-Engine. Sie können beispielsweise den SQL Befehlszeilenclient My oder My SQL Workbench verwenden.

Um die Verbindungsinformationen für eine DB-Instance zu finden, können Sie den Befehl AWS Management Console, den describe-db-instancesBefehl AWS Command Line Interface (AWS CLI) oder die Amazon RDS API escribeDBInstancesD-Operation verwenden, um deren Details aufzulisten.

Konsole

Um die Verbindungsinformationen für eine DB-Instance zu finden, finden Sie im AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie im Navigationsbereich auf Datenbanken, um eine Liste Ihrer DB-Instances anzuzeigen.
3. Wählen Sie den Namen der MariaDB-DB-Instance, um deren Details anzuzeigen.
4. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



5. Wenn Sie den Masterbenutzernamen finden müssen, wählen Sie die Registerkarte Konfiguration und den Wert für den Masterbenutzernamen an.

AWS CLI

Um die Verbindungsinformationen für eine MariaDB-DB-Instance mithilfe von zu finden AWS CLI, führen Sie den describe-db-instancesBefehl aus. Fragen Sie beim Aufruf die DB-Instance-ID, den Endpunkt, den Port und den Masterbenutzernamen ab.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mariadb" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mariadb" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Die Ausgabe sollte in etwa wie folgt aussehen.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

RDS API

Rufen Sie die escribeDBInstances Operation D auf, um die Verbindungsinformationen für eine DB-Instance mithilfe von Amazon RDS API zu ermitteln. Suchen Sie in der Ausgabe die Werte für die Endpunktadresse, den Endpunktport und den Masterbenutzernamen.

Verbindung über den SQL Befehlszeilenclient My (unverschlüsselt) für MariaDB herstellen RDS

> **Important**
>
> Verwenden Sie eine unverschlüsselte SQL My-Verbindung nur, wenn sich Client und Server im selben Netzwerk befinden VPC und das Netzwerk vertrauenswürdig ist. Weitere IInformationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Herstellen einer Verbindung zu Ihrer MariaDB-DB-Instance auf Amazon RDS SSL mit/über den TLS SQL Befehlszeilenclient My (verschlüsselt).

Um mit dem SQL Befehlszeilenclient My eine Verbindung zu einer DB-Instance herzustellen, geben Sie den folgenden Befehl an einer Befehlszeile auf einem Client-Computer ein. Dadurch werden Sie mit einer Datenbank auf einer MariaDB-DB-Instance verbunden. Ersetzen Sie den DNS Namen (Endpunkt) für Ihre DB-Instance durch *<endpoint>* und den Master-Benutzernamen, den Sie für verwendet haben *<mymasteruser>*. Geben Sie das Master-Passwort ein, das Sie verwendet haben, als Sie zur Eingabe eines Kennworts aufgefordert wurden.

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, wird eine Ausgabe wie die folgende angezeigt.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Verbindung RDS zu MariaDB mit dem AWS JDBC Treiber und dem AWS Python-Treiber herstellen;

Stellen Sie mit dem RDS Treiber und dem AWS JDBC AWS Python-Treiber eine Connect zu MariaDB-DB-Instances her. Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- Mit dem Amazon Web Services (AWS) -Treiber eine Verbindung zu RDS for MariaDB herstellen JDBC
- Herstellen einer Verbindung zu RDS für MariaDB mit dem Amazon Web Services (AWS) Python-Treiber

Mit dem Amazon Web Services (AWS) -Treiber eine Verbindung zu RDS for MariaDB herstellen JDBC

Der Amazon Web Services (AWS) JDBC -Treiber ist als erweiterter JDBC Wrapper konzipiert. Dieser Wrapper ergänzt einen vorhandenen JDBC Treiber und erweitert dessen Funktionalität. Der Treiber ist Drop-In-kompatibel mit dem Community-Treiber My SQL Connector/J und dem Community-Treiber MariaDB Connector/J.

Um den Treiber zu installieren, hängen Sie die AWS JDBC Treiber-.jar-Datei an (befindet sich in der Anwendung`CLASSPATH`) und behalten Sie die Verweise auf den jeweiligen AWS JDBC Community-Treiber bei. Aktualisieren Sie das jeweilige URL Verbindungspräfix wie folgt:

- `jdbc:mysql://` auf `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` auf `jdbc:aws-wrapper:mariadb://`

Weitere Informationen zum AWS JDBC Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im JDBC GitHub Treiber-Repository von Amazon Web Services (AWS).

Herstellen einer Verbindung zu RDS für MariaDB mit dem Amazon Web Services (AWS) Python-Treiber

Der Amazon Web Services (AWS) Python-Treiber ist als fortschrittlicher Python-Wrapper konzipiert. Dieser Wrapper ergänzt den Open-Source-Treiber Psycopg und erweitert dessen Funktionalität. Der AWS Python-Treiber unterstützt Python-Versionen 3.8 und höher. Sie können das `aws-advanced-`

`python-wrapper` Paket zusammen mit den `psycopg` Open-Source-Paketen mit dem `pip` Befehl installieren.

Weitere Informationen zum AWS Python-Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im GitHub Python-Treiber-Repository von Amazon Web Services (AWS).

Fehlerbehebung bei Verbindungen zu Ihrer MariaDB-DB-Instance

Zwei häufige Ursachen für Fehler bei Verbindungen mit einer neuen DB-Instance sind die folgenden:

- Die DB-Instance wurde mit einer Sicherheitsgruppe erstellt, die keine Verbindungen von dem Gerät oder der EC2 Amazon-Instance autorisiert, auf dem die MariaDB-Anwendung oder das MariaDB-Hilfsprogramm ausgeführt wird. Die DB-Instance muss über eine VPC Sicherheitsgruppe verfügen, die die Verbindungen autorisiert. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.

  Sie können eine Regel für eingehenden Datenverkehr in der Sicherheitsgruppe hinzufügen oder ändern. Wählen Sie für Source (Quelle) die Option My IP (Meine IP) aus. Dies ermöglicht Zugriff auf die DB-Instance von der IP-Adresse, die in Ihrem Browser erkannt wird.
- Die DB-Instance wurde mithilfe des Standard-Port 3306 erstellt; die Firewall Ihres Unternehmens blockiert jedoch Verbindungen zu diesem Port von Geräten aus Ihrem Unternehmensnetzwerk. Erstellen Sie die Instance erneut mit einem andern Port, um diesen Fehler zu beheben.

Weitere Informationen zu Verbindungsproblemen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Sichern von Verbindungen von MariaDB-DB-Instances

Sie können die Sicherheit Ihrer DB-Instances von RDS for MariaDB verwalten.

Themen

- MariaDB-Sicherheit bei Amazon RDS
- Verwendung der Passwortvalidierungs-Plugins RDS für MariaDB
- Verschlüsselung von Client-Verbindungen SSL TLS mit/zu MariaDB-DB-Instances auf Amazon RDS
- Aktualisierung von Anwendungen für die Verbindung zu MariaDB-Instanzen mithilfe neuer /-Zertifikate SSL TLS

MariaDB-Sicherheit bei Amazon RDS

Die Sicherheit von MariaDB DB-Instancen wird auf drei Ebenen verwaltet:

- AWS Identity and Access Management steuert, wer RDS Amazon-Verwaltungsaktionen auf DB-Instances ausführen kann. Wenn Sie AWS mithilfe von IAM Anmeldeinformationen eine Verbindung herstellen, muss Ihr IAM Konto über IAM Richtlinien verfügen, die die für die Durchführung von RDS Amazon-Verwaltungsvorgängen erforderlichen Berechtigungen gewähren. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.
- Wenn Sie eine DB-Instance erstellen, verwenden Sie eine VPC Sicherheitsgruppe, um zu steuern, welche Geräte und EC2 Amazon-Instances Verbindungen zum Endpunkt und Port der DB-Instance öffnen können. Diese Verbindungen können mit Secure Socket Layer (SSL) und Transport Layer Security (TLS) hergestellt werden. Zusätzlich können Firewall-Regeln in Ihrem Unternehmen steuern, ob Geräte in Ihrem Unternehmen bestehende Verbindungen zur DB-Instance öffnen können.
- Sobald eine Verbindung zu einer MariaDB DB-Instance geöffnet worden ist, erfolgt die Authentifizierung der Anmeldung und Berechtigungen werden auf die gleiche Weise, wie bei einer eigenständigen Instance von MariaDB angewendet. Befehle wie `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE`, und `SET PASSWORD` arbeiten genauso wie es bei eigenständigen Datenbanken der Fall ist, wie auch das direkte Ändern der Datenbankschematabellen.

Wenn Sie eine RDS Amazon-DB-Instance erstellen, hat der Master-Benutzer die folgenden Standardberechtigungen:

- `alter`
- `alter routine`
- `create`
- `create routine`
- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`
- `event`
- `execute`
- `grant option`
- `index`
- `insert`
- `lock tables`
- `process`
- `references`
- `reload`

  Dieses Privileg ist auf MariaDB DB-Instancen begrenzt. Es gewährt keinen Zugriff auf die `FLUSH LOGS` `FLUSH TABLES WITH READ LOCK` OR-Operationen.
- `replication client`
- `replication slave`
- `select`
- `show create routine`

  Dieses Recht gilt nur für MariaDB-DB-Instances, auf denen Version 11.4 und höher ausgeführt wird.
- `show databases`
- `show view`
- `trigger`

- `update`

Weitere Informationen zu diesen Berechtigungen finden Sie unter Benutzerkontenmanagement in der MariaDB Dokumentation.

> **Note**
>
> Obwohl Sie der Master-Benutzer einer DB-Instance löschen können, empfehlen wir, dies nicht zu tun. Um den Masterbenutzer neu zu erstellen, verwenden Sie `ModifyDBInstance` API oder `modify-db-instance` AWS CLI und geben Sie ein neues Masterbenutzerpasswort mit dem entsprechenden Parameter an. Wenn der Masterbenutzer nicht bereits in der Instance vorhanden ist, wird der Masterbenutzer mit dem angegebenen Passwort erstellt.

Um Verwaltungsdienste für jede DB-Instance bereitzustellen, wird der `rdsadmin`-Benutzer erstellt, wenn die DB-Instance erstellt wird. Ein Versuch, das Passwort auszulassen oder umzubenennen, oder Berechtigungen für das `rdsadmin` Konto zu ändern, führt zu einem Fehler.

Um die Verwaltung der DB-Instance zu erlauben, wurden die Befehle `kill` und `kill_query` beschränkt. Die RDS Amazon-Befehle `mysql.rds_killmysql.rds_kill_query`, und `mysql.rds_kill_query_id` werden für die Verwendung in MariaDB und auch in My bereitgestellt, SQL sodass Sie Benutzersitzungen oder Abfragen auf DB-Instances beenden können.

Verwendung der Passwortvalidierungs-Plugins RDS für MariaDB

Ab MariaDB Version 11.4 können Sie die folgenden Plugins RDS zur Passwortvalidierung verwenden, um die Sicherheit Ihrer Datenbankverbindungen zu erhöhen:

- simple_password_check — prüft, ob ein Passwort mindestens eine bestimmte Anzahl von Zeichen eines bestimmten Typs enthält.
- cracklib_password_check — prüft, ob ein Passwort in einer Wörterbuchdatei der Bibliothek vorkommt. CrackLib

Um diese Plugins zu aktivieren, setzen Sie den Wert des Parameters `simple_password_check` oder auf `FORCE_PLUS_PERMANENT` in der DB-Parametergruppe`cracklib_password_check`, die der DB-Instance zugeordnet ist. Wenn dieser Wert gesetzt ist, kann das Plugin nicht mithilfe der `UNINSTALL PLUGIN` Anweisung zur Laufzeit deinstalliert werden.

Um diese Plugins zu deaktivieren, setzen Sie den Wert des Parameters `simple_password_check` oder `cracklib_password_check` auf `OFF` in der DB-Parametergruppe, die der DB-Instance zugeordnet ist. Wenn dieser Wert festgelegt ist, gelten die Plugin-Validierungsregeln nicht mehr für neue Passwörter.

Hinweise zum Einstellen der Werte von Parametern in Parametergruppen finden Sie unterthe section called “Ändern von Parametern in einer DB-Parametergruppe”.

Verschlüsselung von Client-Verbindungen SSL TLS mit/zu MariaDB-DB-Instances auf Amazon RDS

Secure Sockets Layer (SSL) ist ein Industriestandardprotokoll zur Sicherung von Netzwerkverbindungen zwischen Client und Server. Nach SSL Version 3.0 wurde der Name in Transport Layer Security (TLS) geändert. Amazon RDS unterstützt SSL TLS /Verschlüsselung für MariaDB-DB-Instances. MitSSL/TLSkönnen Sie eine Verbindung zwischen Ihrem Anwendungsclient und Ihrer MariaDB-DB-Instance verschlüsseln. SSL/TLSSupport ist in allen verfügbar. AWS-Regionen

Mit Amazon können Sie Daten während der Übertragung sichernRDS, indem Sie Client-Verbindungen zu MariaDB-DB-Instances mitSSL/verschlüsselnTLS, TLS für alle Verbindungen zu einer MariaDB-DB-InstanceSSL/verlangen und vom SQL My-Befehlszeilenclient aus eine Verbindung mit/(verschlüsselt) herstellen. SSL TLS Die folgenden Abschnitte enthalten Anleitungen zur Konfiguration und Verwendung von SSL TLS /encryption für MariaDB-DB-Instances auf Amazon. RDS

Themen

- SSL/TLSUnterstützung für MariaDB-DB-Instances auf Amazon RDS
- SSL//TLSfür bestimmte Benutzerkonten für eine MariaDB-DB-Instance auf Amazon erforderlich RDS
- SSL//TLSfür alle Verbindungen zu einer MariaDB-DB-Instance auf Amazon erforderlich RDS
- Herstellen einer Verbindung zu Ihrer MariaDB-DB-Instance auf Amazon RDS SSL mit/über den TLS SQL Befehlszeilenclient My (verschlüsselt)

SSL/TLSUnterstützung für MariaDB-DB-Instances auf Amazon RDS

Amazon RDS erstellt ein SSL/TLS certificate and installs the certificate on the DB instance when Amazon RDS provisions the instance. These certificates are signed by a certificate authority. The

SSL/TLS certificate includes the DB instance endpoint as the Common Name (CN) for the SSL/TLS Zertifikat zum Schutz vor Spoofing-Angriffen.

Ein von Amazon erstelltesSSL/TLS-Zertifikat RDS ist die vertrauenswürdige Root-Entität und sollte in den meisten Fällen funktionieren, kann aber fehlschlagen, wenn Ihre Anwendung keine Zertifikatsketten akzeptiert. Wenn Ihre Anwendung keine Zertifikatsketten akzeptiert, versuchen Sie, ein Zwischenzertifikat zu verwenden, um eine Verbindung zu Ihrem herzustellen AWS-Region. Sie müssen beispielsweise ein Zwischenzertifikat verwenden, um eine Verbindung zu den AWS GovCloud (US) Regionen mitSSL/herzustellenTLS.

Informationen zum Herunterladen von Zertifikaten finden Sie unter . Weitere Hinweise zur Verwendung vonSSL/TLSmit My SQL finden Sie unterAktualisierung von Anwendungen für die Verbindung zu MariaDB-Instanzen mithilfe neuer /-Zertifikate SSL TLS.

Amazon RDS für MariaDB unterstützt die Transport Layer Security (TLS) -Versionen 1.3, 1.2, 1.1 und 1.0. TLSDie Unterstützung hängt von der MariaDB-Nebenversion ab. Die folgende Tabelle zeigt die TLS Unterstützung für MariaDB-Nebenversionen.

TLS-Version	MariaDB 11.4	MariaDB 10.11	MariaDB 10.6	MariaDB 10.5	MariaDB 10.4
TLS1.3	Alle Nebenversionen	Alle Nebenversionen	Alle Nebenversionen	Alle Nebenversionen	Alle Nebenversionen
TLS1.2	Alle Nebenversionen	Alle Nebenversionen	Alle Nebenversionen	Alle Nebenversionen	Alle Nebenversionen
TLS1.1	Nicht unterstützt	10.11.6 und niedriger	10.6.16 und niedriger	10.5.23 und niedriger	10.4.32 und niedriger
TLS 1.0	Nicht unterstützt	10.11.6 und niedriger	10.6.16 und niedriger	10.5.23 und niedriger	10.4.32 und niedriger

SSL//TLSfür bestimmte Benutzerkonten für eine MariaDB-DB-Instance auf Amazon erforderlich RDS

Sie können für bestimmte Benutzerkontoverbindungen zu Ihren MariaDB-DB-Instances bei Amazon eine SSL TLS /Verschlüsselung verlangen. RDS Der Schutz vertraulicher Informationen vor unbefugtem Zugriff oder Abfangen ist entscheidend für die Durchsetzung von Sicherheitsrichtlinien, wenn es um die Vertraulichkeit von Daten geht.

UmSSL/TLS-Verbindungen für bestimmte Benutzerkonten vorzuschreiben, verwenden Sie je nach Ihrer SQL My-Version eine der folgenden Anweisungen, umSSL/TLS-Verbindungen für das Benutzerkonto `encrypted_user` vorzuschreiben.

Verwenden Sie dazu die folgende Anweisung.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Weitere Informationen zuSSL/TLS-Verbindungen mit MariaDB finden Sie unter Securing Connections for Client and Server in der MariaDB-Dokumentation.

SSL//TLSfür alle Verbindungen zu einer MariaDB-DB-Instance auf Amazon erforderlich RDS

Verwenden Sie den `require_secure_transport` Parameter, um zu verlangen, dass alle Benutzerverbindungen zu Ihrer MariaDB-DB-Instance/verwendenSSL. TLS Standardmäßig ist der `require_secure_transport`-Parameter auf `OFF` festgelegt. Sie können den `require_secure_transport` Parameter auf „SSL/“ TLS für Verbindungen `ON` zu Ihrer DB-Instance setzen.

> **Note**
>
> Der Parameter `require_secure_transport` wird nur für MariaDB Version 10.5 und höher unterstützt.

Sie können den Parameterwert `require_secure_transport` festlegen, indem Sie die DB-Parametergruppe für Ihre DB-Instance aktualisieren. Sie müssen Ihre DB-Instance nicht neu starten, damit die Änderung wirksam wird.

Wenn der `require_secure_transport`-Parameter auf `ON` für eine DB-Instance festgelegt ist, kann ein Datenbank-Client eine Verbindung zu ihr herstellen, wenn er eine verschlüsselte Verbindung aufbauen kann. Andernfalls wird eine Fehlermeldung ähnlich der folgenden an den Client zurückgegeben:

```
ERROR 1045 (28000): Access denied for user 'USER'@'localhost' (using password: YES |
 NO)
```

Weitere Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Weitere Informationen zum Parameter `require_secure_transport` finden Sie in der MariaDB-Dokumentation.

Herstellen einer Verbindung zu Ihrer MariaDB-DB-Instance auf Amazon RDS SSL mit/über den TLS SQL Befehlszeilenclient My (verschlüsselt)

Die Parameter des `mysql` Client-Programms unterscheiden sich geringfügig, wenn Sie die Version My SQL 5.7, die Version My SQL 8.0 oder die MariaDB-Version verwenden.

Um herauszufinden, welche Version Sie haben, führen Sie`mysql`-Befehl mit `--version`-Option aus. Im folgenden Beispiel zeigt die Ausgabe, dass das Client-Programm von MariaDB stammt.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

Die meisten Linux-Distributionen wie Amazon Linux, CentOS und Debian haben My durch MariaDB SQL ersetzt, und die darin enthaltene `mysql` Version stammt von MariaDB. SUSE

Gehen Sie folgendermaßen vor, um mit/eine Verbindung zu Ihrer DB-Instance herzustellenSSL: TLS

So stellen Sie mit dem SQL Befehlszeilenclient My eine Verbindung zu einer DB-Instance TLS mitSSL/her

1. Laden Sie ein Stammzertifikat herunter, das für alle funktioniert. AWS-Regionen

   Informationen zum Herunterladen von Zertifikaten finden Sie unter .
2. Verwenden Sie einen SQL My-Befehlszeilenclient, um eine Verbindung zu einer DB-Instance mit SSL TLS /-Verschlüsselung herzustellen. Ersetzen Sie für den `-h` Parameter den DNS Namen

(Endpunkt) für Ihre DB-Instance. Ersetzen Sie für den `--ssl-ca` Parameter den Namen der TLS ZertifikatsdateiSSL//. Ersetzen Sie für den `-P`-Parameter den Port für Ihre DB-Instance. Ersetzen Sie für den `-u`-Parameter den Benutzernamen eines gültigen Datenbankbenutzers, z. B. des Masterbenutzers. Geben Sie bei Aufforderung das Passwort für den Masterbenutzer ein.

Im folgenden Beispiel sehen Sie für MariaDB, wie der Client mit dem Parameter `--ssl-ca` gestartet wird.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
```

Geben Sie den folgenden Befehl ein, um zu verlangen, dass die TLS VerbindungSSL/den Endpunkt der DB-Instance mit dem Endpunkt im TLS ZertifikatSSL/vergleicht:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-verify-server-cert -P 3306 -u myadmin -p
```

Das folgende Beispiel zeigt, wie der Client mithilfe des `--ssl-ca` Parameters mit dem My SQL 5.7-Client oder höher gestartet wird:

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
```

3. Geben Sie bei Aufforderung das Passwort für den Masterbenutzer ein.

Die Ausgabe sollte in etwa wie folgt aussehen:

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Aktualisierung von Anwendungen für die Verbindung zu MariaDB-Instanzen mithilfe neuer /-Zertifikate SSL TLS

Seit dem 13. Januar 2023 RDS hat Amazon neue Certificate Authority (CA) -Zertifikate für die Verbindung zu Ihren RDS DB-Instances über Secure Socket Layer oder Transport Layer Security (SSL/TLS) veröffentlicht. Im Folgenden finden Sie Informationen dazu, wie Sie Ihre Anwendungen aktualisieren, um die neuen Zertifikate verwenden zu können.

In diesem Thema finden Sie Informationen dazu, wie Sie ermitteln, ob Ihre Anwendungen für die Herstellung von Verbindungen mit Ihren DB-Instances eine Zertifikatverifizierung erfordern.

> **Note**
>
> Einige Anwendungen sind so konfiguriert, dass sie nur dann Verbindungen mit MariaDB nur herstellen, wenn sie das Zertifikat auf dem Server erfolgreich identifizieren können. Für solche Anwendungen müssen Sie die Trust Stores Ihrer Client-Anwendung aktualisieren, damit diese die neuen CA-Zertifikate enthalten.
> Sie können die folgenden SSL Modi angeben: `disabled`, `preferred`, und`required`. Wenn Sie den `preferred` SSL Modus verwenden und das CA-Zertifikat nicht existiert oder nicht aktuell ist, wird die Verbindung wieder nicht verwendet SSL und die Verbindung wird trotzdem erfolgreich hergestellt.
> Wir empfehlenden `preferred`-Modus. Wenn die Verbindung im `preferred`-Modus auf ein ungültiges Zertifikat stößt, wird die Verschlüsselung beendet und unverschlüsselt fortgesetzt.

Nach der Aktualisierung der CA-Zertifikate in den Trust Stores Ihrer Client-Anwendung können Sie die Zertifikate auf Ihren DB-Instances rotieren. Es wird nachdrücklich empfohlen, diese Verfahren vor der Implementierung in Produktionsumgebungen in einer Entwicklungs- oder Testumgebung zu testen.

Weitere Informationen zur Zertifikatrotation finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate. Weitere Informationen zum Herunterladen von Zertifikaten finden Sie unter . Hinweise zur Verwendung vonSSL/TLSmit MariaDB-DB-Instances finden Sie unter. SSL/TLSUnterstützung für MariaDB-DB-Instances auf Amazon RDS

Themen

- Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert

- Aktualisieren des Trust Stores Ihrer Anwendung
- Java-Beispielcode zum Herstellen von Verbindungen SSL

Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert

Sie können überprüfen, ob JDBC Clients und Meine SQL Clients für die Verbindung eine Zertifikatsüberprüfung benötigen.

JDBC

Das folgende Beispiel mit My SQL Connector/J 8.0 zeigt eine Möglichkeit, die JDBC Verbindungseigenschaften einer Anwendung zu überprüfen, um festzustellen, ob für erfolgreiche Verbindungen ein gültiges Zertifikat erforderlich ist. Weitere Informationen zu allen JDBC Verbindungsoptionen für My SQL finden Sie unter Konfigurationseigenschaften in der Dokumentation My. SQL

Wenn Sie My SQL Connector/J 8.0 verwenden, muss eine SSL Verbindung anhand des CA-Zertifikats des Servers überprüft werden, wenn Ihre Verbindungseigenschaften auf `VERIFY_CA` oder `sslMode` eingestellt sind`VERIFY_IDENTITY`, wie im folgenden Beispiel.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Wenn Sie entweder My SQL Java Connector v5.1.38 oder höher oder My SQL Java Connector v8.0.9 oder höher verwenden, um eine Verbindung zu Ihren Datenbanken herzustellen, verwenden diese Clienttreiber standardmäßig/, auch wenn Sie Ihre Anwendungen nicht explizit so konfiguriert haben, dass TLS sieSSL/verwenden, wenn Sie eine Verbindung zu Ihren Datenbanken herstellen. SSL TLS Wenn SieSSL/TLSverwenden, führen sie außerdem eine teilweise Zertifikatsüberprüfung durch und können keine Verbindung herstellen, wenn das Datenbankserverzertifikat abgelaufen ist.
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Mein SQL

Die folgenden Beispiele mit My SQL Client zeigen zwei Möglichkeiten, die Option Meine SQL Verbindung eines Skripts dahingehend zu überprüfen, ob für erfolgreiche Verbindungen ein gültiges Zertifikat erforderlich ist. Weitere Informationen zu allen Verbindungsoptionen mit My SQL Client finden Sie in der Dokumentation My unter Clientseitige Konfiguration für verschlüsselte Verbindungen. SQL

Wenn Sie den My SQL 5.7- oder My SQL 8.0-Client verwenden, muss eine SSL Verbindung anhand des CA-Zertifikats des Servers überprüft werden, wenn Sie für die `--ssl-mode` Option `VERIFY_CA` oder angeben`VERIFY_IDENTITY`, wie im folgenden Beispiel.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Wenn Sie den My SQL 5.6 Client verwenden, muss eine SSL Verbindung anhand des CA-Zertifikats des Servers verifiziert werden, wenn Sie die `--ssl-verify-server-cert` Option angeben, wie im folgenden Beispiel.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-verify-server-cert
```

Aktualisieren des Trust Stores Ihrer Anwendung

Informationen zum Aktualisieren des Trust Store für Meine SQL Anwendungen finden Sie unter UsingTLS/SSLwith MariaDB Connector/J in der MariaDB-Dokumentation.

Informationen zum Herunterladen des Stammverzeichnisses finden Sie unter .

Beispiele für Skripte, die Zertifikate importieren, finden Sie unter Beispielskript für den Import von Zertifikaten in Ihren Trust Store.

> **Note**
>
> Wenn Sie den Trust Store aktualisieren, können Sie ältere Zertifikate beibehalten und die neuen Zertifikate einfach hinzufügen.

Wenn Sie den MariaDB JDBC Connector/J-Treiber in einer Anwendung verwenden, legen Sie die folgenden Eigenschaften in der Anwendung fest.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Legen Sie während des Startens der Anwendung die folgenden Eigenschaften fest.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Passwörter als hier angegeben an.

Java-Beispielcode zum Herstellen von Verbindungen SSL

Das folgende Codebeispiel zeigt, wie Sie die SSL Verbindung einrichten mitJDBC.

```
private static final String DB_USER = "admin";

        private static final String DB_USER = "user name";
        private static final String DB_PASSWORD = "password";
        // This key store has only the prod root ca.
        private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
        private static final String KEY_STORE_PASS = "keystore-password";

    public static void main(String[] args) throws Exception {
        Class.forName("org.mariadb.jdbc.Driver");

        System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

        Properties properties = new Properties();
```

```
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);


        Connection connection = DriverManager.getConnection("jdbc:mysql://ssl-mariadb-
public.cni62e2e7kwh.us-east-1.rds.amazonaws.com:3306?useSSL=true",properties);
        Statement stmt=connection.createStatement();

        ResultSet rs=stmt.executeQuery("SELECT 1 from dual");

        return;
    }
```

Important

Nachdem Sie festgestellt haben, dass Ihre DatenbankverbindungenSSL/verwenden, TLS und Ihren Anwendungs-Truststore aktualisiert haben, können Sie Ihre Datenbank so aktualisieren, dass sie die rds-ca-rsa 2048-g1-Zertifikate verwendet. Anweisungen hierzu finden Sie in Schritt 3 unter Aktualisierung Ihres CA-Zertifikats durch Änderung Ihrer DB-Instance oder Ihres Clusters.

Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Verbesserung der Abfrageleistung für RDS für MariaDB mit Amazon RDS Optimized Reads

Mit Amazon RDS Optimized Reads können Sie eine schnellere Abfrageverarbeitung für RDS für MariaDB erreichen. Eine DB-Instance von RDS für MariaDB, die RDS Optimized Reads verwendet, kann eine doppelt so schnelle Abfrageverarbeitung erreichen als eine DB-Instance, die diese Funktion nicht verwendet.

Themen

- Übersicht über RDS Optimized Reads
- Anwendungsfälle für RDS Optimized Reads
- Bewährte Methoden für RDS Optimized Reads
- Verwenden von RDS Optimized Reads
- Überwachen von DB-Instances, die RDS Optimized Reads verwenden
- Einschränkungen für RDS Optimized Reads

Übersicht über RDS Optimized Reads

Wenn Sie eine DB-Instance von RDS für MariaDB verwenden, für die RDS Optimized Reads aktiviert ist, erreicht Ihre DB-Instance durch die Verwendung eines Instance-Speichers eine schnellere Abfrageleistung. Ein Instance-Speicher stellt für Ihre DB-Instance temporären Speicher auf Blockebene bereit. Der Speicher befindet sich auf Non-Volatile Memory Express (NVMe)-SSDs, die physisch mit dem Hostserver verbunden sind. Dieser Speicher ist für niedrige Latenzen, eine hohe Random-I/O-Leistung und einen hohen sequentiellen Lesedurchsatz optimiert.

RDS Optimized Reads ist standardmäßig aktiviert, wenn eine DB-Instance eine DB-Instance-Klasse mit einem Instance-Speicher wie db.m5d oder db.m6gd verwendet. Mit RDS Optimized Reads werden einige temporäre Objekte im Instance-Speicher abgelegt. Zu diesen temporären Objekten gehören interne temporäre Dateien, interne temporäre Tabellen auf der Festplatte, Speicherzuordnungsdateien und Binärprotokolldateien (binlog) im Cache. Weitere Informationen zum Instance-Speicher finden Sie unter Instance-Speicher von Amazon EC2 im Benutzerhandbuch für Amazon Elastic Compute Cloud für Linux-Instances.

Die Workloads, die temporäre Objekte in MariaDB für die Abfrageverarbeitung generieren, können den Instance-Speicher für eine schnellere Abfrageverarbeitung nutzen. Diese Art

von Workload umfasst Abfragen mit Sortierungen, Hash-Aggregationen, Joins mit hoher Auslastung, Common Table Expressions (CTEs) und Abfragen für nicht indizierte Spalten. Diese Instance-Speicher-Volumes bieten höhere IOPS und eine höhere Leistung, unabhängig von den Speicherkonfigurationen, die für persistenten Amazon EBS-Speicher verwendet werden. Da RDS Optimized Reads Operationen mit temporären Objekten in den Instance-Speicher auslagert, können die Eingabe-/Ausgabevorgänge pro Sekunde (IOPS) oder der Durchsatz des persistenten Speichers (Amazon EBS) jetzt für Operationen an persistenten Objekten verwendet werden. Zu diesen Vorgängen gehören reguläre Lese- und Schreibvorgänge von Datendateien sowie Engine-Operationen im Hintergrund, wie das Leeren und Zusammenführen von Puffern zum Einfügen.

> **Note**
>
> Sowohl manuelle als auch automatisierte RDS-Snapshots enthalten nur die Engine-Dateien für persistente Objekte. Die im Instance-Speicher erstellten temporären Objekte sind nicht in RDS-Snapshots enthalten.

Anwendungsfälle für RDS Optimized Reads

Wenn Sie Workloads haben, deren Abfrageausführung stark auf temporäre Objekte wie interne Tabellen oder Dateien angewiesen ist, können Sie von der Aktivierung von RDS Optimized Reads profitieren. Die folgenden Anwendungsfälle eignen sich für RDS Optimized Reads:

- Anwendungen, die analytische Abfragen mit komplexen Common Table Expressions (CTEs), abgeleiteten Tabellen und Gruppierungsoperationen ausführen
- Lesereplikate, die einen hohen Leseverkehr mit nicht optimierten Abfragen bewältigen
- Anwendungen, die auf Abruf laufen oder dynamische Berichtsabfragen ausführen, die komplexe Operationen beinhalten, z. B. Abfragen mit `GROUP BY`- und `ORDER BY`-Klauseln
- Workloads, die interne temporäre Tabellen für die Abfrageverarbeitung verwenden

  Sie können die Engine-Statusvariable `created_tmp_disk_tables` überwachen, um die Anzahl der festplattenbasierten temporären Tabellen zu ermitteln, die auf Ihrer DB-Instance erstellt wurden.
- Anwendungen, die direkt oder in Prozeduren große temporäre Tabellen zum Speichern von Zwischenergebnissen erstellen
- Datenbankabfragen, die das Gruppieren oder Sortieren von nicht indizierten Spalten durchführen

Bewährte Methoden für RDS Optimized Reads

Nutzen Sie die folgenden bewährten Methoden für RDS Optimized Reads:

- Fügen Sie Wiederholungslogik für schreibgeschützte Abfragen hinzu, falls diese aufgrund eines Fehlers wegen eines vollen Instance-Speichers während der Ausführung fehlschlagen.
- Überwachen Sie den im Instance-Speicher verfügbaren Speicherplatz anhand der CloudWatch Metrik`FreeLocalStorage`. Wenn der Instance-Speicher aufgrund der Workload der DB-Instance sein Limit erreicht, ändern Sie die DB-Instance, um eine größere DB-Instance-Klasse zu verwenden.
- Wenn Ihre DB-Instance über ausreichend Speicher verfügt, aber immer noch das Speicherlimit für den Instance-Speicher erreicht, erhöhen Sie den `binlog_cache_size`-Wert, um die sitzungsspezifischen Binlog-Einträge im Speicher zu behalten. Diese Konfiguration verhindert, dass die Binlog-Einträge in temporäre Binlog-Cache-Dateien auf der Festplatte geschrieben werden.

  Der `binlog_cache_size`-Parameter ist sitzungsspezifisch. Sie können den Wert für jede neue Sitzung ändern. Die Einstellung für diesen Parameter kann die Speicherauslastung der DB-Instance bei Spitzenauslastung erhöhen. Erwägen Sie daher, den Parameterwert auf der Grundlage des Workload-Musters Ihrer Anwendung und des verfügbaren Speichers in der DB-Instance zu erhöhen.
- Verwenden Sie den Standardwert `MIXED` für `binlog_format`. Abhängig von der Größe der Transaktionen kann die Einstellung von `binlog_format` auf `ROW` zu großen Binlog-Cache-Dateien im Instance-Speicher führen.
- Vermeiden Sie es, Massenänderungen in einer einzigen Transaktion durchzuführen. Diese Arten von Transaktionen können große Binlog-Cache-Dateien im Instance-Speicher generieren und Probleme verursachen, wenn der Instance-Speicher voll ist. Erwägen Sie, Schreibvorgänge in mehrere kleine Transaktionen aufzuteilen, um den Speicherverbrauch für Binlog-Cache-Dateien zu minimieren.

Verwenden von RDS Optimized Reads

Wenn Sie eine DB-Instance von RDS für MariaDB mit einer der folgenden DB-Instance-Klassen in einer Single-AZ-Bereitstellung oder einer Multi-AZ-Bereitstellung der DB-Instance bereitstellen, verwendet die DB-Instance automatisch RDS Optimized Reads.

Führen Sie einen der folgenden Schritte aus, um RDS Optimized Reads zu aktivieren:

- Erstellen Sie eine DB-Instance von RDS für MariaDB mit einer dieser DB-Instance-Klassen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine vorhandene DB-Instance von RDS für MariaDB so, dass sie eine dieser DB-Instance-Klassen verwendet. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

RDS Optimized Reads ist überall verfügbar AWS-Regionen , wo eine oder mehrere DB-Instance-Klassen mit lokalem NVMe-SSD-Speicher unterstützt werden. Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”.

Die Verfügbarkeit der DB-Instance-Klassen unterscheidet sich für AWS-Regionen. Informationen darüber, ob eine DB-Instance-Klasse in einer bestimmten Klasse unterstützt wird AWS-Region, finden Sie unterthe section called “Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen”.

Wenn Sie RDS Optimized Reads nicht verwenden möchten, ändern Sie Ihre DB-Instance so, dass sie keine DB-Instance-Klasse verwendet, die die Funktion unterstützt.

Überwachen von DB-Instances, die RDS Optimized Reads verwenden

Sie können DB-Instances, die RDS-optimierte Lesevorgänge verwenden, anhand der folgenden CloudWatch Metriken überwachen:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Diese Metriken liefern Daten über den verfügbaren Instance-Speicher, die IOPS und den Durchsatz. Weitere Informationen zu diesen Metriken finden Sie unter Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS.

Einschränkungen für RDS Optimized Reads

Für RDS Optimized Reads gelten die folgenden Einschränkungen:

- RDS Optimized Reads wird für die folgenden Versionen von RDS für MariaDB unterstützt:
  - 10.11.4 und höhere 10.11-Versionen
  - 10.6.7 und höhere 10.6-Versionen
  - 10.5.16 und höhere 10.5-Versionen
  - 10.4.25 und höhere 10.4-Versionen

  Informationen zu den Versionen von RDS für MariaDB finden Sie unter MariaDB auf Amazon-RDS-Versionen.
- Sie können den Speicherort temporärer Objekte in den DB-Instance-Klassen, die RDS Optimized Reads unterstützen, nicht auf persistenten Speicher (Amazon EBS) ändern.
- Wenn die binäre Protokollierung auf einer DB-Instance aktiviert ist, ist die maximale Transaktionsgröße durch die Größe des Instance-Speichers begrenzt. Bei MariaDB schreibt jede Sitzung, die mehr Speicherplatz als den Wert `binlog_cache_size` benötigt, Transaktionsänderungen in temporäre Binlog-Cache-Dateien, die im Instance-Speicher erstellt werden.
- Transaktionen können fehlschlagen, wenn der Instance-Speicher voll ist.

Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB

Sie können die Leistung von Schreibtransaktionen mit RDS-optimierten Schreibvorgängen für MariaDB verbessern. Wenn Ihre Datenbank von RDS für MariaDB RDS-optimierte Schreibvorgänge verwendet, kann sie einen bis zu zweimal höheren Durchsatz für Schreibtransaktionen erreichen.

Themen

- Übersicht über RDS Optimized Writes
- Verwenden von RDS Optimized Writes
- Aktivieren von RDS-optimierten Schreibvorgängen in einer vorhandenen Datenbank
- Einschränkungen für RDS Optimized Writes

Übersicht über RDS Optimized Writes

Wenn Sie RDS-optimierte Schreibvorgänge aktivieren, führen Ihre Datenbanken von RDS für MariaDB beim Leeren von Daten in einen dauerhaften Speicher nur einen Schreibvorgang aus, ohne dass der Doublewrite-Puffer erforderlich ist. Die Datenbanken bieten weiterhin ACID-Eigentumsschutzvorkehrungen für zuverlässige Datenbanktransaktionen sowie eine verbesserte Leistung.

Relationale Datenbanken wie MariaDB bieten die ACID-Eigenschaften Atomizität, Konsistenz, Isolation und Beständigkeit für zuverlässige Datenbanktransaktionen. Um diese Eigenschaften bereitzustellen, verwendet MariaDB einen Datenspeicherbereich, den sogenannten Doublewrite-Puffer, der teilweise Schreibfehler von Seiten verhindert. Diese Fehler treten bei einem Hardwarefehler auf, während die Datenbank eine Seite aktualisiert, z. B. bei einem Stromausfall. Eine MariaDB-Datenbank kann teilweise Schreibvorgänge von Seiten erkennen und diese mit einer Kopie der Seite im Doublewrite-Puffer wiederherstellen. Diese Technik bietet zwar Schutz, führt aber auch zu zusätzlichen Schreiboperationen. Weitere Informationen zum Doublewrite-Puffer von MariaDB finden Sie unter InnoDB-Doublewrite-Puffer in der MariaDB-Dokumentation.

Wenn Amazon-RDS-optimierte Schreibvorgänge aktiviert sind, schreiben Ihre Datenbanken von RDS für MariaDB beim Leeren von Daten in einen dauerhaften Speicher nur einmal, ohne den Doublewrite-Puffer zu verwenden. RDS-optimierte Schreibvorgänge sind nützlich, wenn Sie schreibintensive Workloads in Ihren Datenbanken von RDS für MariaDB ausführen. Zu den

Datenbanken mit schreibintensiven Workloads gehören Datenbanken, die digitale Zahlungen, Finanzhandel und Spieleanwendungen unterstützen.

Diese Datenbanken laufen auf DB-Instance-Klassen, die das AWS Nitro-System verwenden. Aufgrund der Hardwarekonfiguration in diesen Systemen kann die Datenbank in einem Schritt zuverlässig und dauerhaft Seiten mit 16 KiB direkt in Datendateien schreiben. Das AWS Nitro-System ermöglicht RDS-optimierte Schreibvorgänge.

Sie können den neuen Datenbankparameter `rds.optimized_writes` festlegen, um die Funktion RDS-optimierte Schreibvorgänge für Datenbanken von RDS für MariaDB zu steuern. Greifen Sie auf diesen Parameter in den DB-Parametergruppen von RDS für MariaDB für die folgenden Versionen zu:

- 11.4.3 und höhere 11.4-Versionen
- 10.11.4 und höhere 10.11-Versionen
- 10.6.10 und höhere 10.6-Versionen

Legen Sie den Parameter anhand der folgenden Werte fest:

- `AUTO` – Aktivieren Sie RDS Optimized Writes, wenn die Datenbank diese Funktion unterstützt. Deaktivieren Sie RDS Optimized Writes, wenn die Datenbank diese Funktion nicht unterstützt. Dies ist die Standardeinstellung.
- `OFF` – Deaktivieren Sie RDS Optimized Writes, auch wenn die Datenbank diese Funktion unterstützt.

Wenn Sie eine Datenbank von RDS für MariaDB, die für die Verwendung von RDS-optimierten Schreibvorgängen konfiguriert ist, zu einer DB-Instance-Klasse migrieren, die die Funktion nicht unterstützt, deaktiviert RDS die Funktion RDS-optimierte Schreibvorgänge für die Datenbank automatisch.

Wenn RDS-optimierte Schreibvorgänge deaktiviert sind, verwendet die Datenbank den MariaDB-Doublewrite-Puffer.

Um festzustellen, ob eine Datenbank von RDS für MariaDB die Funktion RDS-optimierte Schreibvorgänge verwendet, sehen Sie sich den aktuellen Wert des `innodb_doublewrite`-Parameters für die Datenbank an. Wenn die Datenbank RDS Optimized Writes verwendet, ist dieser Parameter auf `FALSE` (`0`) eingestellt.

Verwenden von RDS Optimized Writes

Sie können RDS-optimierte Schreibvorgänge aktivieren, wenn Sie eine RDS for MariaDB-Datenbank mit der RDS-Konsole AWS CLI, der oder der RDS-API erstellen. RDS Optimized Writes wird automatisch aktiviert, wenn bei der Datenbankerstellung die beiden folgenden Bedingungen zutreffen:

- Sie geben eine DB-Engine-Version und eine DB-Instance-Klasse an, die RDS Optimized Writes unterstützt.
  - RDS Optimized Writes wird für die folgenden Versionen von RDS für MariaDB unterstützt:
    - 11.4.3 und höhere 11.4-Versionen
    - 10.11.4 und höhere 10.11-Versionen
    - 10.6.10 und höhere 10.6-Versionen

    Informationen zu den Versionen von RDS für MariaDB finden Sie unter MariaDB auf Amazon-RDS-Versionen.
  - RDS-optimierte Schreibvorgänge werden für Datenbanken von RDS für MariaDB unterstützt, die die folgenden DB-Instance-Klassen verwenden:
    - db.m7i
    - db.m7g
    - db.m6g
    - db.m6gd
    - db.m6i
    - db.m5
    - db.m5d
    - db.r7i
    - db.r7g
    - db.r6g
    - db.r6gd
    - db.r6i
    - db.r5
    - db.r5b
    - db.r5d

  - db.x2idn
  - db.x2iedn

  Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”.

  Die Verfügbarkeit der DB-Instance-Klassen unterscheidet sich für. AWS-Regionen Informationen darüber, ob eine DB-Instance-Klasse in einer bestimmten Klasse unterstützt wird AWS-Region, finden Sie unterthe section called “Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen”.

- In der Parametergruppe, die der Datenbank zugeordnet ist, ist der `rds.optimized_writes`-Parameter auf `AUTO` eingestellt. In Standardparametergruppen ist dieser Parameter immer auf `AUTO` festgelegt.

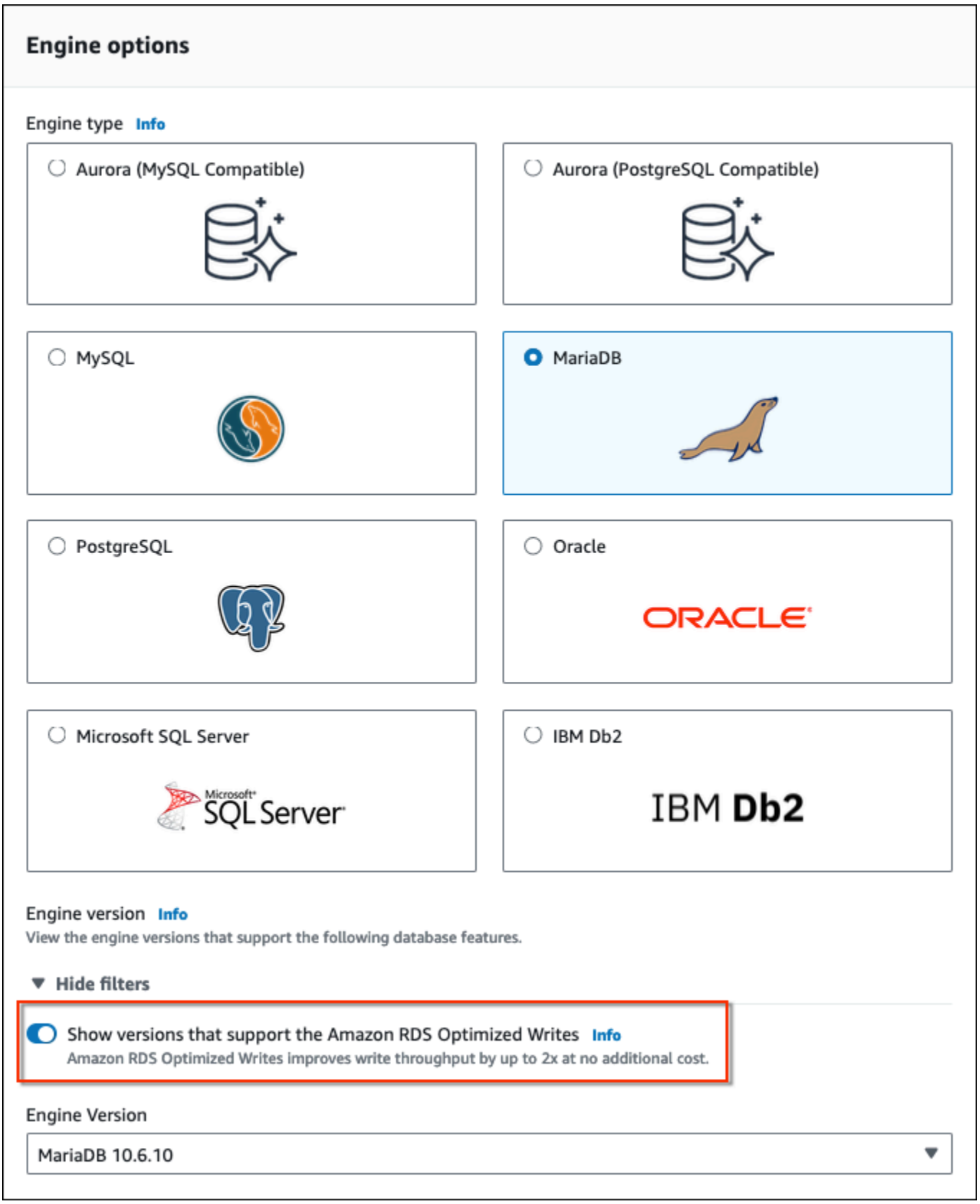
Wenn Sie eine DB-Engine-Version und eine DB-Instance-Klasse verwenden möchten, die RDS-optimierte Schreibvorgänge unterstützen, geben Sie beim Erstellen der Datenbank eine benutzerdefinierte Parametergruppe an. Legen Sie den Parameter `rds.optimized_writes` in dieser Parametergruppe auf `OFF` fest. Wenn Sie möchten, dass die Datenbank später RDS Optimized Writes verwendet, können Sie den Parameter auf `AUTO` einstellen, um ihn zu aktivieren. Weitere Informationen über das Erstellen von benutzerdefinierten DB-Parametergruppen und das Festlegen von Parametern finden Sie unter Parametergruppen für Amazon RDS.

Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

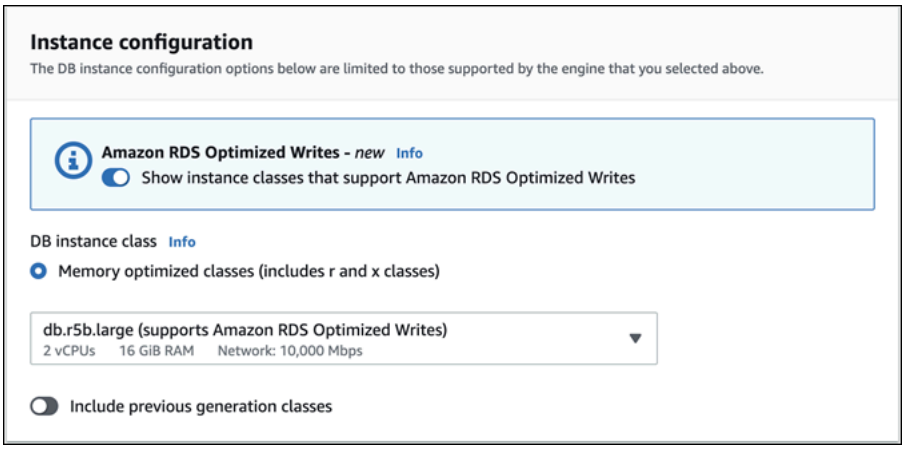
Konsole

Wenn Sie die RDS-Konsole verwenden, um eine Datenbank von RDS für MariaDB zu erstellen, können Sie nach den Versionen der DB-Engine und den DB-Instance-Klassen filtern, die RDS-optimierte Schreibvorgänge unterstützen. Nachdem Sie die Filter aktiviert haben, können Sie aus den verfügbaren DB-Engine-Versionen und DB-Instance-Klassen auswählen.

Wenn Sie eine DB-Engine-Version auswählen möchten, die RDS-optimierte Schreibvorgänge unterstützt, filtern Sie in Engine-Version nach den DB-Engine-Versionen von RDS für MariaDB, die dies unterstützen, und wählen Sie dann eine Version aus.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
IBM Db2
IBM Db2
Engine version Info
View the engine versions that support the following database features.
Hide filters
Show versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Engine Version
MariaDB 10.6.10

Filtern Sie im Abschnitt Instance configuration (Instance-Konfiguration) nach den DB-Instance-Klassen, die RDS Optimized Writes unterstützen, und wählen Sie dann eine DB-Instance-Klasse aus.



Nachdem Sie diese Auswahl getroffen haben, können Sie andere Einstellungen auswählen, die Ihren Anforderungen entsprechen, und die Erstellung der Datenbank von RDS für MariaDB mit der Konsole abschließen.

AWS CLI

Um eine DB-Instance mithilfe von zu erstellen AWS CLI, führen Sie den create-db-instanceBefehl aus. Stellen Sie sicher, dass die Werte `--engine-version` und `--db-instance-class` RDS Optimized Writes unterstützen. Stellen Sie außerdem sicher, dass der `rds.optimized_writes`-Parameter für die Parametergruppe, die der DB-Instance zugeordnet ist, auf AUTO festgelegt ist. Im folgenden Beispiel wird die Standardparametergruppe mit der DB-Instance verknüpft.

Example Erstellen einer DB-Instance, die RDS Optimized Writes verwendet

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mariadb \
    --engine-version 10.6.10 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
```

```
    --db-instance-identifier mydbinstance ^
    --engine mariadb ^
    --engine-version 10.6.10 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

RDS-API

Sie können eine DB-Instance mithilfe der DBInstance Operation Create erstellen. Wenn Sie diese Operation verwenden, stellen Sie sicher, dass die Werte `EngineVersion` und `DBInstanceClass` RDS Optimized Writes unterstützen. Stellen Sie außerdem sicher, dass der `rds.optimized_writes`-Parameter für die Parametergruppe, die der DB-Instance zugeordnet ist, auf `AUTO` festgelegt ist.

Aktivieren von RDS-optimierten Schreibvorgängen in einer vorhandenen Datenbank

Um eine vorhandene Datenbank von RDS für MariaDB zu ändern und RDS-optimierte Schreibvorgänge zu aktivieren, muss die Datenbank mit einer unterstützten DB-Engine-Version und DB-Instance-Klasse erstellt worden sein. Darüber hinaus muss die Datenbank nach der Veröffentlichung der RDS-optimierten Schreibvorgänge am 7. März 2023 erstellt worden sein, da die erforderliche zugrunde liegende Dateisystemkonfiguration nicht mit derjenigen von Datenbanken kompatibel ist, die vor der Veröffentlichung erstellt wurden. Wenn diese Bedingungen erfüllt sind, können Sie RDS-optimierte Schreibvorgänge aktivieren, indem Sie den Parameter `rds.optimized_writes` auf `AUTO` setzen.

Wenn Ihre Datenbank nicht mit einer unterstützten Engine-Version, Instance-Klasse oder Dateisystemkonfiguration erstellt wurde, können Sie die Blue/Green Deployments to migrate to a supported configuration. While creating the blue/green RDS-Bereitstellung verwenden. Gehen Sie dazu wie folgt vor:

- Wählen Sie Optimierte Schreibvorgänge in grüner Datenbank aktivieren aus und geben Sie dann eine Engine-Version und eine DB-Instance-Klasse an, die RDS-optimierte Schreibvorgänge unterstützt. Eine Liste der unterstützten Engine-Versionen und Instance-Klassen finden Sie unter the section called “Verwendung mit einer neuen Datenbank”.

- Wählen Sie unter Speicher die Option Speicherdatei-Systemkonfiguration aktualisieren aus. Mit dieser Option wird die Datenbank auf eine kompatible zugrunde liegende Dateisystemkonfiguration aktualisiert.

Wenn Sie die Blau/Grün-Bereitstellung erstellen und der `rds.optimized_writes`-Parameter auf `AUTO` festgelegt ist, werden RDS-optimierte Schreibvorgänge in der grünen Umgebung automatisch aktiviert. Sie können dann die Blau/Grün-Bereitstellung umstellen, wodurch die grüne Umgebung zur neuen Produktionsumgebung hochgestuft wird.

Weitere Informationen finden Sie unter the section called “Erstellen einer Blau/Grün-Bereitstellung”.

Einschränkungen für RDS Optimized Writes

Wenn Sie eine Datenbank von RDS für MariaDB aus einem Snapshot wiederherstellen, können Sie RDS-optimierte Schreibvorgänge für die Datenbank nur aktivieren, wenn alle nachfolgenden Bedingungen zutreffen:

- Der Snapshot wurde aus einer Datenbank erstellt, die RDS Optimized Writes unterstützt.
- Der Snapshot wurde aus einer Datenbank erstellt, die nach der Veröffentlichung von RDS-optimierten Schreibvorgängen erstellt wurde.
- Der Snapshot wird in einer Datenbank wiederhergestellt, die RDS Optimized Writes unterstützt.
- Die wiederhergestellte Datenbank ist einer Parametergruppe zugeordnet, deren `rds.optimized_writes`-Parameter auf `AUTO` eingestellt ist.

Upgrades der MariaDB-DB-Engine

Wenn Amazon eine neue Version einer Datenbank-Engine RDS unterstützt, können Sie Ihre DB-Instances auf die neue Version aktualisieren. Es gibt zwei Arten von Upgrades für MariaDB-DB-Instances: Hauptversionsupgrades und Unterversionsupgrades.

Hauptversions-Upgrades können Datenbankänderungen enthalten, die nicht mit vorhandenen Anwendungen rückwärts kompatibel sind. Daher müssen Sie Hauptversions-Upgrades Ihrer DB-Instances manuell durchführen. Sie können ein Hauptversions-Upgrade starten, indem Sie Ihre DB-Instance ändern. Bevor Sie ein Hauptversions-Upgrade durchführen, wird jedoch das Befolgen der Schritte unter Hauptversions-Upgrades für RDS für MariaDB empfohlen.

Nebenversions-Upgrades enthalten dagegen nur Änderungen, die mit vorhandenen Anwendungen abwärtskompatibel sind. Sie können ein Nebenversions-Upgrade manuell starten, indem Sie Ihre DB-Instance ändern. Alternativ können Sie auch beim Erstellen oder Ändern einer DB-Instance die Option Auto minor version upgrade (Automatisches Nebenversions-Upgrade) aktivieren. Das bedeutet, dass Ihre DB-Instance automatisch aktualisiert wird, nachdem Amazon die neue Version RDS getestet und genehmigt hat. Weitere Informationen zum Ausführen eines Upgrades finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Wenn Ihre MariaDB-DB-Instance Lesereplikate verwendet, müssen Sie alle Lesereplikate aktualisieren, bevor Sie die Quell-Instance aktualisieren. Wenn sich Ihre DB-Instance in einer Multi-AZ-Bereitstellung befindet, werden sowohl die Writer- als auch die Standby-Replikate aktualisiert. Ihre DB-Instance ist möglicherweise erst verfügbar, wenn das Upgrade abgeschlossen ist.

Weitere Informationen über die unterstützten MariaDB-Versionen und zur Versionsverwaltung finden Sie unter MariaDB auf Amazon-RDS-Versionen.

Datenbank-Engine-Upgrades erfordern Ausfallzeiten. Die Dauer des Nutzungsausfalls ist von der Größe Ihrer DB-Instance abhängig.

> **Tip**
>
> Sie können die für das DB-Instance-Upgrade erforderlichen Ausfallzeiten minimieren, indem Sie eine Blau/Grün-Bereitstellung verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Themen

- Überlegungen zu MariaDB-Upgrades
- Suche nach gültigen Upgrade-Zielen
- MariaDB-Versionsnummern
- RDSVersionsnummern in RDS für MariaDB
- Hauptversions-Upgrades für RDS für MariaDB
- Upgraden einer MariaDB-DB-Instance
- Automatische Upgrades für Nebenversionen RDS für MariaDB
- Verwendung einer Read Replica zur Reduzierung von Ausfallzeiten beim Upgrade einer RDS für MariaDB-Datenbank

Überlegungen zu MariaDB-Upgrades

Amazon erstellt RDS während des Upgrade-Vorgangs zwei oder mehr DB-Snapshots. Amazon RDS erstellt bis zu zwei Snapshots der DB-Instance, bevor Upgrade-Änderungen vorgenommen werden. Wenn das Upgrade bei Ihren Datenbanken nicht funktioniert, können Sie einen dieser Snapshots wiederherstellen, um eine DB-Instance zu erstellen, auf der die alte Version ausgeführt wird. Amazon RDS erstellt nach Abschluss des Upgrades einen weiteren Snapshot der DB-Instance. Amazon erstellt RDS diese Snapshots unabhängig davon, ob die Backups für die DB-Instance AWS Backup verwaltet werden.

> **Note**
>
> Amazon erstellt RDS nur DB-Snapshots, wenn Sie den Aufbewahrungszeitraum für Backups für Ihre DB-Instance auf einen Wert größer als 0 festgelegt haben. Informationen über das Ändern Ihres Aufbewahrungszeitraums für Backups finden Sie unter Ändern einer Amazon RDS DB-Instance.

Nachdem das Upgrade abgeschlossen ist, können Sie nicht zur vorherigen Version der Datenbank-Engine zurückkehren. Wenn Sie zur vorherigen Version zurückkehren möchten, stellen Sie den ersten DB-Snapshot wieder her, um eine neue DB-Instance zu erstellen.

Sie bestimmen, wann Sie Ihre DB-Instance auf eine neue, von Amazon unterstützte Version aktualisieren müssenRDS. Diese Kontrollebene hilft Ihnen, die Kompatibilität mit bestimmten Datenbankversionen aufrechtzuerhalten und neue Versionen mit Ihrer Anwendung zu testen, bevor

sie produktiv bereitgestellt werden. Wenn Sie bereit sind, können Sie Versions-Upgrades zu den Zeiten durchführen, die am besten zu Ihrem Zeitplan passen.

Wenn Ihre DB-Instance die Lesereplikation verwendet, müssen Sie alle Read Replicas aktualisieren, bevor Sie die Quell-Instance aktualisieren.

Wenn sich die DB-Instance in einer Multi-AZ-Bereitstellung befindet, erfolgt das Upgrade sowohl für die primären als auch Standby-DB-Instances. Die primären und die Standby-DB-Instances werden gleichzeitig aktualisiert, und es kommt zu einem Ausfall, bis das Upgrade abgeschlossen ist. Die Dauer des Nutzungsausfalls ist von Ihrer Datenbank-Engine-Version und der Größe Ihrer DB-Instance abhängig.

Suche nach gültigen Upgrade-Zielen

Wenn Sie den verwenden, AWS Management Console um eine DB-Instance zu aktualisieren, werden die gültigen Upgrade-Ziele für die DB-Instance angezeigt. Sie können auch den folgenden AWS CLI Befehl ausführen, um die gültigen Upgrade-Ziele für eine DB-Instance zu identifizieren:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
  --engine mariadb \
  --engine-version version_number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version version_number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Um beispielsweise die gültigen Upgrade-Ziele für eine MariaDB-DB-Instance der Version 10.5.17 zu identifizieren, führen Sie den folgenden Befehl aus: AWS CLI

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
```

```
  --engine mariadb \
  --engine-version 10.5.17 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version 10.5.17 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

MariaDB-Versionsnummern

Die Versionsnummerierungssequenz für die RDS for MariaDB-Datenbank-Engine hat entweder die Form major.minor.patch. YYYYMMDDoder major.minor.patch, zum Beispiel 10.11.5.R2.20231201 oder 10.4.30. Das verwendete Format hängt von der MariaDB-Engine-Version ab.

Major

Die Hauptversionsnummer ist sowohl die Ganzzahl als auch der erste Bruchteil der Versionsnummer, zum Beispiel 10.11. Ein Upgrade der Hauptversion erhöht den Hauptteil der Versionsnummer. Ein Upgrade von 10.5 .20 auf 10.6.12 ist beispielsweise ein Hauptversionsupgrade, wobei 10.5 und 10.6 die Hauptversionsnummern sind.

geringfügig

Die Nebenversionsnummer ist der dritte Teil der Versionsnummer, zum Beispiel die 5 in 10.11.5.

Patch

Der Patch ist der vierte Teil der Versionsnummer, zum Beispiel R2 in 10.11.5.R2. Eine RDS Patch-Version enthält wichtige Bugfixes, die einer Nebenversion nach ihrer Veröffentlichung hinzugefügt wurden.

YYYYMMDD

Das Datum ist der fünfte Teil der Versionsnummer, zum Beispiel 20231201 in 10.11.5.R2.20231201. Eine RDS Datumsversion ist ein Sicherheitspatch, der wichtige Sicherheitsupdates enthält, die einer Nebenversion nach der Veröffentlichung hinzugefügt wurden. Er enthält keine Fixes, die das Verhalten einer Engine ändern könnten.

In der folgenden Tabelle wird das Benennungsschema RDS für MariaDB Version 10.11 erläutert.

Nebenversion 10.11	Benennungsschema
≥5	Neue DB-Instances verwenden major.minor.patch. YYMMDD, zum Beispiel 10.11.5.R2.20231201. Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 10.11.5.R2, bis zu Ihrem nächsten Upgrade der Haupt- oder Nebenversion.
< 5	Bestehende DB-Instances verwenden major.minor.patch, beispielsweise 10.11.4.R2.

In der folgenden Tabelle wird das Benennungsschema RDS für MariaDB Version 10.6 erläutert.

10.6 Nebenversion	Benennungsschema
≥ 14	Neue DB-Instances verwenden major.minor.patch. YYMMDD, zum Beispiel 10.6.14.R2.20231201. Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 10.6.14.R2, bis zu Ihrem nächsten Upgrade der Haupt- oder Nebenversion.
< 14	Bestehende DB-Instances verwenden major.minor.patch, beispielsweise 10.6.13.R2.

Die folgende Tabelle erklärt das Benennungsschema RDS für MariaDB Version 10.5.

10.5 Nebenversion	Benennungsschema
≥ 21	Neue DB-Instances verwenden major.minor.patch. YYMMDD, zum Beispiel 10.5.21.R2.20231201.

10.5 Nebenversion	Benennungsschema
	Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 10.5.21.R2, bis zu Ihrem nächsten Upgrade der Haupt- oder Nebenversion.
< 21	Bestehende DB-Instances verwenden major.minor.patch, beispielsweise 10.5.20.R2.

In der folgenden Tabelle wird das Benennungsschema RDS für MariaDB Version 10.4 erläutert.

10.4 Nebenversion	Benennungsschema
≥ 30	Neue DB-Instances verwenden major.minor.patch. YYMMDD, zum Beispiel 10.4.30.R2.20231201. Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 10.4.30.R2, bis zu Ihrem nächsten Upgrade der Haupt- oder Nebenversion.
< 30	Bestehende DB-Instances verwenden major.minor.patch, beispielsweise 10.4.29.R2.

RDSVersionsnummern in RDS für MariaDB

RDSVersionsnummern verwenden entweder das *major*.*minor*.*patch* oder das *major*.*minor*.*patch*.*YYYYMMDD* Benennungsschema. Eine RDS Patch-Version enthält wichtige Bugfixes, die einer Nebenversion nach ihrer Veröffentlichung hinzugefügt wurden. Eine RDS Datumsversion (*YYMMDD*) ist ein Sicherheitspatch. Ein Sicherheitspatch enthält keine Fixes, die das Verhalten der Engine ändern könnten.

Um die RDS Amazon-Versionsnummer Ihrer Datenbank zu ermitteln, müssen Sie zunächst die `rds_tools` Erweiterung mithilfe des folgenden Befehls erstellen:

```
CREATE EXTENSION rds_tools;
```

Sie können die RDS Versionsnummer Ihrer RDS for MariaDB-Datenbank mit der folgenden SQL Abfrage herausfinden:

```
mysql> select mysql.rds_version();
```

Wenn Sie RDS beispielsweise eine nach MariaDB 10.6.14 Datenbank abfragen, wird die folgende Ausgabe zurückgegeben:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
| 10.6.14.R2.20231201  |
+---------------------+
1 row in set (0.01 sec)
```

Hauptversions-Upgrades für RDS für MariaDB

Hauptversions-Upgrades können Datenbankänderungen enthalten, die nicht mit vorhandenen Anwendungen rückwärts kompatibel sind. Aus diesem Grund wendet Amazon Hauptversions-Upgrades RDS nicht automatisch an. Sie müssen Ihre DB-Instance manuell ändern. Sie sollten jedes Upgrade gründlich testen, bevor Sie es auf Ihre Produktions-Instances anwenden.

Amazon RDS unterstützt die folgenden direkten Upgrades für Hauptversionen der MariaDB-Datenbank-Engine:

- Jede MariaDB-Version bis MariaDB 11.4
- Jede MariaDB-Version auf MariaDB 10.11
- Jede MariaDB-Version zu MariaDB 10.6
- MariaDB 10.4 zu MariaDB 10.5

Wenn Sie eine benutzerdefinierte Parametergruppe verwenden und ein Upgrade auf eine Hauptversion durchführen, müssen Sie entweder eine Standardparametergruppe für die neue DB-Engine-Version angeben oder eine eigene benutzerdefinierte Parametergruppe für die neue DB-Engine-Version erstellen. Die Zuordnung der neuen Parametergruppe zu DB-Instance erfordert einen vom Kunden initiierten Neustart der Datenbank nach Abschluss des Upgrades. Der Parametergruppenstatus der Instance zeigt an, `pending-reboot` ob die Instance neu gestartet werden muss, um die Änderungen der Parametergruppe zu übernehmen. Der Status einer

Parametergruppe einer Instanz kann im AWS Management Console oder durch Ausführen eines „describe“ -Aufrufs wie angezeigt werden. `describe-db-instances`

Upgraden einer MariaDB-DB-Instance

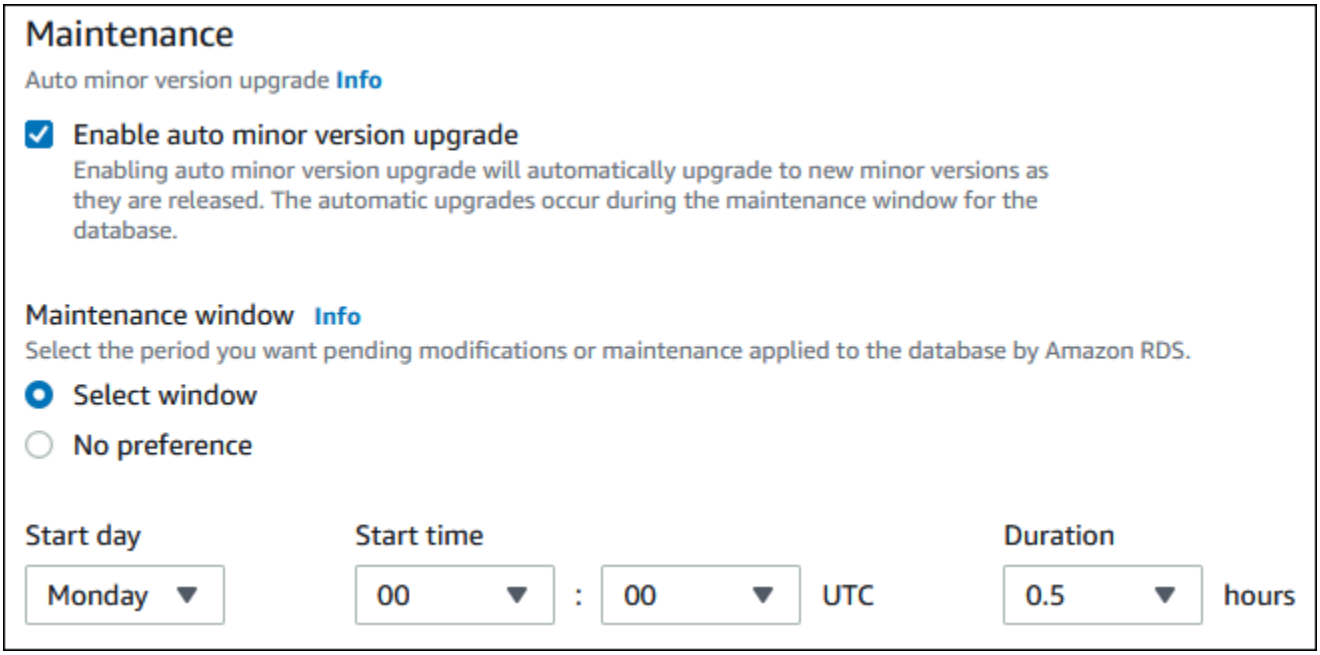
Informationen über das manuelle oder automatische Upgraden einer MariaDB-DB-Instance finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Automatische Upgrades für Nebenversionen RDS für MariaDB

Wenn Sie beim Erstellen oder Ändern einer DB-Instance die folgenden Einstellungen angeben, können Sie Ihre DB-Instance automatisch aktualisieren lassen.

- Die Einstellung Automatisches Upgrade der Nebenversion ist aktiviert.
- Die Einstellung Aufbewahrungszeitraum für Sicherungen beträgt mehr als 0.

In der befinden AWS Management Console sich diese Einstellungen unter Zusätzliche Konfiguration. Die folgende Abbildung zeigt die Auto minor version upgrade (Upgrade einer Unterversion automatisch durchführen)-Einstellung.



Weitere Informationen zu diesen Einstellungen finden Sie unter Einstellungen für DB-Instances.

RDSFür einige MariaDB-Hauptversionen AWS-Regionen, in einigen wird eine Nebenversion RDS als automatische Upgrade-Version bezeichnet. Nachdem eine Nebenversion von Amazon

getestet und genehmigt wurdeRDS, erfolgt das Upgrade der Nebenversion automatisch während Ihres Wartungsfensters. RDSlegt nicht automatisch neuere veröffentlichte Nebenversionen als automatische Upgrade-Version fest. Bevor RDS eine neuere automatische Upgrade-Version bestimmt, werden mehrere Kriterien berücksichtigt, wie beispielsweise die folgenden:

- Bekannte Sicherheitsprobleme
- Fehler in der MariaDB-Community-Version
- Gesamtflottenstabilität seit Erscheinen der Minor-Version

> **Note**
>
> Die Support für die Verwendung von TLS Version 1.0 und 1.1 wurde ab bestimmten Nebenversionen von MariaDB entfernt. Informationen zu unterstützten MariaDB-Nebenversionen finden Sie unter. the section called “SSL/TLSUnterstützung für MariaDB”

Sie können den folgenden AWS CLI Befehl ausführen, um die aktuelle automatische Minor-Upgrade-Zielversion für eine angegebene MariaDB-Nebenversion in einer bestimmten zu ermitteln. AWS-Region

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version minor_version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version minor_version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Der folgende AWS CLI Befehl bestimmt beispielsweise das automatische kleinere Upgrade-Ziel für die MariaDB-Nebenversion 10.5.16 in den USA Ost (Ohio) (us-east-2) AWS-Region .

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version 10.5.16 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version 10.5.16 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Ihre Ausgabe sieht Folgendem ähnlich.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  True        |  10.5.17        |
|  False       |  10.5.18        |
|  False       |  10.5.19        |
|  False       |  10.6.5         |
|  False       |  10.6.7         |
|  False       |  10.6.8         |
|  False       |  10.6.10        |
|  False       |  10.6.11        |
|  False       |  10.6.12        |
+--------------+-----------------+
```

In diesem Beispiel ist der `AutoUpgrade`-Wert `True` für MariaDB-Version 10.5.17. Das automatische Nebenversions-Upgrade-Ziel ist daher die MariaDB-Version 10.5.17, die in der Ausgabe hervorgehoben wird.

Eine MariaDB-DB-Instance wird während Ihres Wartungsfensters automatisch aktualisiert, wenn die folgenden Kriterien erfüllt sind:

- Die Einstellung Automatisches Upgrade der Nebenversion ist aktiviert.
- Die Einstellung Aufbewahrungszeitraum für Sicherungen beträgt mehr als 0.
- Die DB-Instance führt eine Minor-Version der DB-Engine aus, die niedriger ist als die aktuelle Minor-Version des automatischen Upgrades.

Weitere Informationen finden Sie unter Automatisches Upgraden der Engine-Unterversion.

Verwendung einer Read Replica zur Reduzierung von Ausfallzeiten beim Upgrade einer RDS für MariaDB-Datenbank

In den meisten Fällen ist eine Blau/Grün-Bereitstellung die beste Option, um Ausfallzeiten beim Upgrade einer MariaDB-DB-Instance zu reduzieren. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Wenn Sie keine Blau/Grün-Bereitstellung verwenden können und Ihre MariaDB-DB-Instance aktuell von einer Produktionsanwendung genutzt wird, können Sie mit dem folgenden Verfahren die Datenbankversion Ihrer DB-Instance aktualisieren. Dieses Verfahren kann die Ausfallzeiten Ihrer Anwendung reduzieren.

Mithilfe einer Read Replica können Sie die meisten Wartungsschritte im Voraus durchführen und die erforderlichen Änderungen während des tatsächlichen Ausfalls minimieren. Mit dieser Technik können Sie die neue DB-Instance testen und vorbereiten, ohne Änderungen an Ihrer bestehenden DB-Instance vorzunehmen.

Im Folgenden wird ein Beispiel für ein Upgrade von MariaDB Version 10.5 auf MariaDB Version 10.6 gezeigt. Sie können die gleichen allgemeinen Schritte für Upgrades auf andere Hauptversionen durchführen.

So führen Sie ein Upgrade einer MariaDB-Datenbank durch, während eine DB-Instance verwendet wird

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Erstellen Sie ein Lesereplikat Ihrer DB-Instance von MariaDB 10.5. Dieser Prozess erstellt eine aktualisierbare Kopie Ihrer Datenbank. Andere Read Replicas der DB-Instance könnten ebenfalls vorhanden sein.

   a. Wählen Sie in der Konsole Datenbanken und dann die DB-Instance aus, die Sie upgraden möchten.
   b. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.
   c. Geben Sie für das Lesereplikat einen Wert im Feld DB instance identifier (DB-Instance-Kennung) ein und stellen Sie sicher, dass der Eintrag unter DB instance class (DB-Instance-Klasse) und die anderen Einstellungen mit Ihrer DB-Instance von MariaDB 10.5 übereinstimmen.
   d. Wählen Sie Read Replica erstellen aus.
3. (Optional) Wenn die Read Replica erstellt wurde und der Status Verfügbar anzeigt, konvertieren Sie die Read Replica in eine Multi-AZ-Bereitstellung und aktivieren Sie Sicherungen.

   Standardmäßig wird ein Lesereplikat als Single-AZ-Bereitstellung mit deaktivierten Backups erstellt. Da das Lesereplikat letztendlich zur DB-Produktions-Instance wird, ist es eine bewährte Methode, eine Multi-AZ-Bereitstellung zu konfigurieren und Backups jetzt zu aktivieren.

   a. Wählen Sie in der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade erstellt haben.
   b. Wählen Sie Ändern aus.
   c. Für die Multi-AZ-Bereitstellungwählen Sie Standby-Instance erstellen.
   d. Wählen Sie unter Backup Retention Period (Aufbewahrungszeitraum für Backups) einen positiven Wert größer als null aus, z. B. 3 Tage. Klicken Sie anschließend auf Continue (Weiter).
   e. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.
   f. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.
4. Wenn der Status des Lesereplikats Available (Verfügbar) anzeigt, aktualisieren Sie das Lesereplikat auf MariaDB 10.6.

a. Wählen Sie in der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade erstellt haben.

b. Wählen Sie Ändern aus.

c. Wählen Sie im Feld DB engine version (DB-Engine-Version) die gewünschte Version von MariaDB 10.6 für das Upgrade aus und klicken Sie auf Continue (Weiter).

d. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.

e. Wählen Sie Modify DB instance (DB-Instance ändern) aus, um das Upgrade zu starten.

5. Wenn das Upgrade abgeschlossen ist und der Status Verfügbar anzeigt, stellen Sie sicher, dass sich die aktualisierte Read Replica up-to-date mit der Quell-MariaDB 10.5 DB-Instance befindet. Stellen Sie zur Überprüfung eine Verbindung mit dem Lesereplikat her und führen Sie den Befehl `SHOW REPLICA STATUS` aus. Wenn das `Seconds_Behind_Master` Feld lautet`0`, dann ist die Replikation. up-to-date

> **Note**
>
> Frühere Versionen von MariaDB wurden`SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie eine ältere MariaDB-Version nutzen als 10.6, verwenden Sie `SHOW SLAVE STATUS`.

6. (Optional) Erstellen Sie eine Read Replica Ihrer Read Replica.

   Wenn Sie möchten, dass die DB-Instance eine Read Replica hat, nachdem sie auf eine eigenständige DB-Instance hochgestuft wurde, können Sie jetzt die Read Replica erstellen.

   a. Wählen Sie auf der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade aktualisiert haben.

   b. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.

   c. Geben Sie für das Lesereplikat einen Wert im Feld DB instance identifier (DB-Instance-Kennung) ein und stellen Sie sicher, dass der Eintrag unter DB instance class (DB-Instance-Klasse) und die anderen Einstellungen mit Ihrer DB-Instance von MariaDB 10.5 übereinstimmen.

   d. Wählen Sie Read Replica erstellen aus.

7. (Optional) Konfigurieren Sie eine benutzerdefinierte DB-Parametergruppe für die Read Replica.

Wenn Sie möchten, dass die DB-Instance eine benutzerdefinierte Parametergruppe verwendet, nachdem sie zu einer eigenständigen DB-Instance hochgestuft wurde, können Sie die DB-Parametergruppe erstellen und sie jetzt dem Lesereplikat zuordnen kann.

a. Erstellen Sie eine benutzerdefinierte DB-Parametergruppe für MariaDB 10.6. Detaillierte Anweisungen finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.

b. Ändern Sie die Parameter, die Sie in der gerade erstellten DB-Parametergruppe ändern möchten. Detaillierte Anweisungen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

c. Wählen Sie in der Konsole Datenbankenund dann die Read Replica aus.

d. Wählen Sie Ändern aus.

e. Wählen Sie für DB parameter group (DB-Parametergruppe) die soeben erstellte DB-Parametergruppe von MariaDB 10.6 aus, und klicken Sie dann auf Continue (Weiter).

f. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.

g. Wählen Sie Modify DB instance (DB-Instance ändern) aus, um das Upgrade zu starten.

8. Machen Sie Ihr Lesereplikat von MariaDB 10.6 zu einer eigenständigen DB-Instance.

> **Important**
>
> Wenn Sie Ihr Lesereplikat von MariaDB 10.6 zu einer eigenständigen DB-Instance hochstufen, handelt es sich nicht mehr um ein Replikat der DB-Instance von MariaDB 10.5. Wir empfehlen, dass Sie Ihr Lesereplikat von MariaDB 10.6 während eines Wartungsfensters hochstufen, wenn sich Ihre Quell-DB-Instance von MariaDB 10.5 im schreibgeschützten Modus befindet und alle Schreiboperationen ausgesetzt sind. Wenn das Hochstufen abgeschlossen ist, können Sie Ihre Schreiboperationen an die aktualisierte DB-Instance von MariaDB 10.6 weiterleiten, um sicherzustellen, dass keine Schreiboperationen verloren gehen.
> Darüber hinaus empfehlen wir, dass Sie, bevor Sie Ihre MariaDB 10.6-Read Replica bewerben, alle erforderlichen Data Definition Language (DDL) -Operationen auf Ihrer MariaDB 10.6-Read Replica durchführen. Ein Beispiel hierfür ist das Erstellen von Indizes. Mit diesem Ansatz werden negative Auswirkungen auf die Leistung des Lesereplikats von MariaDB 10.6 vermieden, nachdem es hochgestuft wurde. Gehen Sie folgendermaßen vor, um ein Lesereplikat hochzustufen.

a. Wählen Sie auf der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade aktualisiert haben.

b. Wählen Sie für Actions (Aktionen) Promote (Hochstufen) aus.

c. Klicken Sie auf Yes (Ja), um automatische Sicherungen für die Lesereplikat-Instance zu aktivieren. Weitere Informationen finden Sie unter Einführung in Backups.

d. Klicken Sie auf Continue (Fortfahren).

e. Klicken Sie auf Read Replica hochstufen.

9. Sie haben jetzt eine aktualisierte Version Ihrer MariaDB-Datenbank vorliegen. An dieser Stelle können Sie Ihre Anwendungen auf die neue DB-Instance von MariaDB 10.6 umleiten.

Aktualisierung einer MariaDB DB-Snapshot-Engine-Version

Mit Amazon RDS können Sie einen Speichervolume-DB-Snapshot Ihrer MariaDB-DB-Instance erstellen. Wenn Sie einen DB-Snapshot erstellen, basiert der Snapshot auf der Engine-Version, die von Ihrer DB-Instance verwendet wird. Sie können die Engine-Version für Ihre DB-Snapshots aktualisieren.

Für RDS for MariaDB können Sie auf alle verfügbaren Engine-Versionen aktualisieren. Sie können verschlüsselte oder unverschlüsselte DB-Snapshots aktualisieren.

Verwenden Sie das folgende AWS CLI Beispiel, um die verfügbaren Engine-Versionen für Ihren RDS for MariaDB-DB-Snapshot anzuzeigen.

```
aws rds describe-db-engine-versions --engine mariadb --include-all --engine-
version example-engine-version --query "DBEngineVersions[*].ValidUpgradeTarget[*].
{EngineVersion:EngineVersion}" --output text
```

Wenn Sie keine Ergebnisse für Ihren Snapshot sehen, ist Ihre Engine-Version möglicherweise veraltet. Wenn Ihre Engine-Version veraltet ist, empfehlen wir Ihnen, ein Upgrade auf das neueste Upgrade-Ziel für die Hauptversion oder auf eines der anderen verfügbaren Upgrade-Ziele für diese Version durchzuführen. Weitere Informationen finden Sie unter Upgrade-Optionen für DB-Snapshots mit nicht unterstützten Engine-Versionen für RDS for MariaDB.

Wenn Sie einen DB-Snapshot wiederherstellen, der auf eine neue Engine-Version aktualisiert wurde, sollten Sie prüfen, ob das Upgrade erfolgreich durchgeführt wurde. Weitere Informationen zu größeren Versionsaktualisierungen finden Sie unter the section called “Upgrades der MariaDB-DB-Engine”. Informationen zum Wiederherstellen eines DB-Snapshots finden Sie unter the section called “Wiederherstellung auf einer DB-Instance”.

> **Note**
>
> Automatisierte DB-Snapshots, die während des automatisierten Backup-Vorgangs erstellt wurden, können nicht aktualisiert werden.

Sie können einen DB-Snapshot mithilfe der AWS Management Console AWS CLI, oder RDS-API aktualisieren.

Console

Gehen Sie wie folgt vor AWS Management Console, um eine DB-Snapshot-Engine-Version mithilfe von zu aktualisieren.

So führen Sie ein Upgrade eines DB-Snapshots durch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den Snapshot für die Aktualisierung aus.
4. Wählen Sie unter Actions (Aktionen) die Option Upgrade Snapshot (Snapshot aktualisieren). Die Seite Upgrade snapshot (Snapshot aktualisieren) erscheint.
5. Wählen Sie zum Aktualisieren New engine version (Neue Engine-Version).
6. Wählen Sie Save changes (Änderungen speichern), um den Snapshot zu aktualisieren.

   Während des Upgrades werden alle Snapshot-Aktionen für diesen DB-Snapshot deaktiviert. Außerdem ändert sich der Status des DB-Snapshots von „Verfügbar“ zu „Aktualisierung“ und nach Abschluss des Vorgangs zu „Aktiv“. Wenn der DB-Snapshot aufgrund von Problemen mit dem Snapshot nicht aktualisiert werden kann, ändert sich der Status in Nicht verfügbar. Sie können den Snapshot aus diesem Zustand nicht wiederherstellen.

   > **Note**
   >
   > Wenn die Aktualisierung des DB-Snapshots fehlschlägt, wird der Snapshot wieder in seinen ursprünglichen Zustand zurückgebracht.

AWS CLI

Um einen DB-Snapshot auf eine neue Version der Datenbank-Engine zu aktualisieren, führen Sie den AWS CLI modify-db-snapshotBefehl aus.

Optionen

- `--db-snapshot-identifier`: die Kennung des DB-Snapshots, für den das Upgrade durchgeführt werden soll Die Kennung muss ein eindeutiger Amazon-Ressourcenname (ARN) sein. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.

- `--engine-version`: Die Engine-Version, auf die das Upgrade des DB-Snapshots durchgeführt werden soll

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot \

    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

Amazon RDS API

Um einen DB-Snapshot auf eine neue Version der Datenbank-Engine zu aktualisieren, rufen Sie den DBSnapshot RDS-API-Vorgang Modify auf.

Parameter

- `DBSnapshotIdentifier`: die Kennung des DB-Snapshots, für den das Upgrade durchgeführt werden soll Die Kennung muss ein eindeutiger Amazon-Ressourcenname (ARN) sein. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.
- `EngineVersion`: Die Engine-Version, auf die das Upgrade des DB-Snapshots durchgeführt werden soll

Upgrade-Optionen für DB-Snapshots mit nicht unterstützten Engine-Versionen für RDS for MariaDB

Die folgende Tabelle zeigt, auf welche Engine-Versionen Sie von einer nicht unterstützten Engine-Version für RDS for MariaDB DB-Snapshots aktualisieren können.

> **Note**
>
> Möglicherweise müssen Sie Ihren DB-Snapshot mehrmals aktualisieren, um auf die von Ihnen gewählte Engine-Version zu aktualisieren.

Version der DB-Snapshot-Engine	Engine-Versionen sind für ein Upgrade verfügbar
10.0.17, 10.0.24, 10.0.28, 10.0.31, 10.0.32, 10.0.34, 10.0.35	10.0.38, 10.1.48, 10.6.19, 10.6.20, 10,11.9, 10,11.10, 11,4,3, 11,4,4
10.1.14, 10.1.16, 10.1.19, 10.1.23, 10.1.26, 10.1.31, 10.1.34	10.1.48, 10.2.44, 10.6.19, 10.6.20, 10.11.9, 10.11.10, 11,4,3, 11,4,4
10.2.11, 10.2.12, 10.2.15, 10.2.21, 10.2.32, 10.2.37, 10,2.39, 10,2,40, 10.2.41, 10.2.43	10.2.44, 10.3.39, 10.6.19, 10.6.20, 10.6.21, 10.11.9, 10.11.10, 10.11.11, 11.4.3, 11.4.4
10.3.8, 10.3.13, 10.3.20, 10.3.23, 10.3.28, 10.3.31, 10.3.32	10.3.39, 10.4.34, 10.6.19, 10.6.20 10.6.21, 10.11.9, 10.11.10, 10.11.11, 11,4,3, 11,4,4
10.4.8	10.4.34, 10.5.16, 10.5.17, 10.5.18, 10.5.20, 10.5.21, 10.5.22, 10.5.23, 10.5.24, 10.5.25, 10.5.26, 10.5.27, 10.5.28, 10.6.8, 10.6.10, 10.6.11, 10.6.13, 10.6.14, 10.6.15, 10.6.16, 10.6.18, 10.6.19, 10.6.20, 10.6.21, 10,11,4, 10,6.17, 10,11,5, 10,11,6, 10,11,7, 10,11,8, 10,11,9, 10,11,110, 10,11.11

Daten in eine Amazon RDS for MariaDB-DB-Instance importieren

Für den Import von Daten in eine DB-Instance von RDS for MariaDB stehen verschiedene Techniken zur Verfügung. Der beste Ansatz hängt von einer Reihe von Faktoren ab:

- Quelle der Daten
- Datenmenge
- Einmaliger Import oder fortlaufend
- Umfang der Ausfallzeit

Wenn Sie auch eine Anwendung mit den Daten migrieren, ist es wichtig, die Höhe der Ausfallzeit zu berücksichtigen.

In der folgenden Tabelle sind Techniken zum Importieren von Daten in eine RDS for MariaDB-DB-Instance aufgeführt:

> **Note**
>
> Amazon RDS unterstützt nur den Import von Amazon S3 in RDS für MySQL-DB-Instances. Das Importieren von Backups, `mariadb-backup` die mit erstellt wurden, in RDS for MariaDB wird derzeit nicht unterstützt.

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Bestehend MariaDB-Datenbank vor Ort oder bei	Any	Kontinuierlich	Minimal	Konfigurieren Sie die Replikation mit einer vorhandenen MariaDB-Datenbank als Replikationsquelle. Sie können die Replikation in eine MariaDB-DB-Instance mithilfe von MariaDB Global Transaction Identifiers (GTIDs) konfigurieren, wenn die externe Instance MariaDB-Version 10.0.24 oder	Konfigurieren der Replikation der Binärprotokolldateiposition

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Amazon EC2				höher ist, oder mithilfe binärer Protokoll koordinaten für MariaDB-Instances in früheren Versionen als 10.0.24. MariaDB GTIDs werden anders implementiert als MySQL GTIDs, die von Amazon RDS nicht unterstützt werden.	mit einer externen Quell-Instance Import von Daten in eine Amazon RDS for MariaDB-DB-Instance mit reduzierter Ausfallzeit

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Alle vorhandenen Datenbanken	Alle	Einmalig oder kontinuierlich	Minimal	Wird verwendet AWS Database Migration Service , um die Datenbank mit minimalen Ausfallzeiten zu migrieren und bei vielen Datenbank-DB-Engines die fortlaufende Replikation fortzusetzen.	Was ist AWS Database Migration Service und Verwenden einer MySQL-kompatiblen Datenbank als Ziel für AWS DMS im AWS Database Migration Service - Benutzerhandbuch
Vorhandene MariaDB-DB-Instance	Any	Einmalig oder kontinuierlich	Minimal	Erstellen Sie eine Read Replica für die laufende Replikation. Stufen Sie die Read Replica für die einmalige Erstellung einer neuen DB-Instance hoch.	Arbeiten mit DB-Instance-Lesereplikaten

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Bestehende MariaDB-Datenbank	Small	Einmalig	Etwas	Kopieren Sie die Daten mit einem Befehlszeilenprogramm direkt in Ihre MariaDB-DB-Instance.	Daten aus einer externen MariaDB-Datenbank in eine Amazon RDS for MariaDB-DB-Instance importieren

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendung	Technik	Weitere Informationen
Nicht in einer vorhandenen Datenbank gespeicherte Daten	Medium	Einmalig	Etwas	Erstellen Sie Flatfiles und importieren Sie sie mit MariaDB-Anweisungen `LOAD DATA LOCAL INFILE`.	Daten aus einer beliebigen Quelle in eine Amazon RDS for MariaDB-DB-Instance importieren

> **Note**
>
> Die Systemdatenbank `mysql` enthält Authentifizierungs- und Autorisierungsinformationen, die für die Anmeldung bei der DB-Instance und für den Zugriff auf die Daten erforderlich sind. Das Löschen, Ändern, Umbenennen oder Kürzen von Tabellen, Daten oder anderen Inhalten der `mysql` Datenbank in Ihrer DB-Instance kann zu Fehlern führen und dazu führen, dass auf die DB-Instance und Ihre Daten nicht mehr zugegriffen werden kann. In diesem Fall können Sie die DB-Instance aus einem Snapshot wiederherstellen, indem Sie AWS CLI restore-db-instance-from-db-snapshotBefehl. Sie können die DB-Instance wiederherstellen mit restore-db-instance-to-point-in-timeBefehl.

Überlegungen zum Importieren von Daten für MariaDB

Der folgende Inhalt enthält technische Informationen zum Laden von Daten in MariaDB. Dieser Inhalt richtet sich an Benutzer, die mit der MariaDB-Serverarchitektur vertraut sind.

Binäre Protokollierung

Die Aktivierung der binären Protokollierung reduziert die Leistung beim Laden von Daten und erfordert bis zu viermal mehr Festplattenspeicher als bei deaktivierter Protokollierung. Die Transaktionsgröße, die zum Laden der Daten verwendet wird, wirkt sich direkt auf die Systemleistung und den Speicherplatzbedarf aus. Größere Transaktionen erfordern mehr Ressourcen.

Transaktionsgröße

Die Transaktionsgröße beeinflusst die folgenden Aspekte des Ladens von MariaDB-Daten:

- Ressourcenverbrauch
- Nutzung des Festplattenspeichers
- Vorgang fortsetzen
- Zeit, sich zu erholen
- Eingabeformat (Flatfiles oder SQL)

In diesem Abschnitt wird beschrieben, wie sich die Transaktionsgröße auf die binäre Protokollierung auswirkt, und verdeutlicht, welche Vorteile das Deaktivieren der binären Protokollierung bei umfangreichen Datenladevorgängen mit sich bringen kann. Sie können die binäre Protokollierung aktivieren und deaktivieren, indem Sie den Aufbewahrungszeitraum für automatische Amazon RDS-Backups festlegen. Nicht-Null-Werte aktivieren die binäre Protokollierung und Null deaktiviert diese. Weitere Informationen finden Sie unter Aufbewahrungszeitraum für Backups.

In diesem Abschnitt werden auch die Auswirkungen großer Transaktionen auf InnoDB beschrieben und warum es wichtig ist, die Transaktionsgrößen klein zu halten.

Kleine Transaktionen

Bei kleinen Transaktionen sorgt die binäre Protokollierung für eine Verdopplung der Schreibvorgänge auf der Festplatte, die für das Laden der Daten erforderlich sind. Dieser Effekt kann die Leistung für andere Datenbanksitzungen signifikant herabsetzen und die zum Laden der Daten erforderliche Zeit erhöhen. Die eingetretene Verschlechterung hängt teilweise von den folgenden Faktoren ab:

- Upload-Rate
- Andere Datenbankaktivitäten, die während des Ladevorgangs stattfinden
- Kapazität Ihrer Amazon RDS-DB-Instance

Die Binärprotokolle verbrauchen außerdem Festplattenspeicher, der in etwa der Menge der geladenen Daten entspricht, bis die Protokolle gesichert und entfernt werden. Amazon RDS minimiert dies, indem binäre Protokolle häufig gesichert und entfernt werden.

Große Transaktionen

Bei großen Transaktionen verdreifacht die binäre Protokollierung die IOPS und die Festplattennutzung aus den folgenden Gründen:

- Der Binärprotokollcache speichert Transaktionsdaten vorübergehend auf der Festplatte.
- Dieser Cache wächst mit der Transaktionsgröße, wodurch Speicherplatz beansprucht wird.
- Wenn die Transaktion (Commit oder Rollback) abgeschlossen ist, kopiert das System den Cache in das Binärprotokoll.

Bei diesem Vorgang werden drei Kopien der Daten erstellt:

- Die Originaldaten
- Der Cache auf der Festplatte
- Der letzte binäre Protokolleintrag

Jeder Schreibvorgang verursacht zusätzliche I/O, was sich weiter auf die Leistung auswirkt.

Aus diesem Grund benötigt die binäre Protokollierung dreimal so viel Festplattenspeicher wie die deaktivierte Protokollierung. Wenn Sie beispielsweise 10 GiB an Daten als eine einzelne Transaktion laden, werden drei Kopien erstellt:

- 10 GiB für die Tabellendaten
- 10 GiB für den binären Log-Cache
- 10 GiB für die binäre Logdatei

Der gesamte benötigte temporäre Festplattenspeicher beträgt 30 GiB.

Wichtige Überlegungen zum Festplattenspeicher:

- Die Cache-Datei bleibt bestehen, bis entweder die Sitzung endet oder eine neue Transaktion einen weiteren Cache erstellt.
- Das Binärprotokoll bleibt so lange bestehen, bis es gesichert wird, und kann möglicherweise 20 GiB (Cache und Protokoll) für einen längeren Zeitraum enthalten.

Wenn Sie die Daten `LOAD DATA LOCAL INFILE` zum Laden verwenden, erstellt Data Recovery eine vierte Kopie für den Fall, dass die Datenbank aus einer vor dem Ladevorgang erstellten Sicherung wiederhergestellt werden muss. Während der Wiederherstellung extrahiert MariaDB die Daten aus dem Binärprotokoll in eine Flat-Datei. MariaDB läuft dann. `LOAD DATA LOCAL INFILE` Aufbauend auf dem vorherigen Beispiel erfordert diese Wiederherstellung insgesamt einen temporären Festplattenspeicher von 40 GiB oder jeweils 10 GiB für Tabelle, Cache, Protokoll und lokale Datei. Ohne mindestens 40 GiB freien Festplattenspeicher schlägt die Wiederherstellung fehl.

Optimierung großer Datenlasten

Deaktivieren Sie bei großen Datenlasten die binäre Protokollierung, um den Overhead und den Speicherplatzbedarf zu reduzieren. Sie können die binäre Protokollierung deaktivieren, indem Sie den Aufbewahrungszeitraum für Backups auf 0 setzen. Stellen Sie nach Abschluss des Ladevorgangs die Aufbewahrungsdauer für Backups auf den entsprechenden Wert ungleich Null zurück. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance und Aufbewahrungszeitraum für Backup in der Einstellungstabelle.

> **Note**
>
> Wenn es sich bei der DB-Instance um eine Quell-DB-Instance für Read Replicas handelt, können Sie den Aufbewahrungszeitraum für Backups nicht auf 0 setzen.

Wir empfehlen Ihnen, vor dem Laden der Daten einen DB-Snapshot zu erstellen. Weitere Informationen finden Sie unter Verwaltung manueller Backups.

InnoDB

Die folgenden Informationen zu Undo-Logging- und Wiederherstellungsoptionen unterstützen dabei, InnoDB-Transaktionen klein zu halten, um die Datenbankleistung zu optimieren.

Grundlegendes zur InnoDB-Undo-Protokollierung

Undo ist ein Protokollierungsmechanismus, der das Rollback von Transaktionen ermöglicht und Multi-Version Concurrency Control (MVCC) unterstützt.

Für MariaDB 10.11 und niedrigere Versionen werden Undo-Logs im InnoDB-System-Tablespace (normalerweise ibdata1) gespeichert und aufbewahrt, bis der Purge-Thread sie entfernt. Infolgedessen können umfangreiche Datenladetransaktionen dazu führen, dass der System-Tablespace ziemlich groß wird und Speicherplatz beansprucht, den Sie nicht zurückgewinnen können, es sei denn, Sie erstellen die Datenbank neu.

Für alle MariaDB-Versionen muss der Purge-Thread warten, um alle Undo-Logs zu entfernen, bis die älteste aktive Transaktion entweder festgeschrieben oder zurückgesetzt wird. Wenn die Datenbank während des Ladevorgangs andere Transaktionen verarbeitet, sammeln sich auch deren Undo-Logs an und können nicht entfernt werden, selbst wenn die Transaktionen festgeschrieben werden und keine andere Transaktion die Undo-Logs für MVCC benötigt. In dieser Situation werden alle Transaktionen — einschließlich schreibgeschützter Transaktionen — langsamer. Diese Verlangsamung ist darauf zurückzuführen, dass alle Transaktionen auf alle Zeilen zugreifen, die von jeder Transaktion — nicht nur der Ladetransaktion — geändert werden. Tatsächlich müssen Transaktionen Undo-Logs durchsuchen, deren Löschung durch lang andauernde Ladetransaktionen während einer Undo-Log-Bereinigung verhindert wurde. Dies beeinträchtigt die Leistung aller Operationen, die auf geänderte Zeilen zugreifen.

Optionen zur Wiederherstellung von InnoDB-Transaktionen

Obwohl InnoDB Commit-Operationen optimiert, sind große Transaktions-Rollbacks langsam. Führen Sie für eine schnellere Wiederherstellung eine Wiederherstellung durch oder stellen Sie einen point-in-time DB-Snapshot wieder her. Weitere Informationen erhalten Sie unter Point-in-time Wiederherstellung und Wiederherstellung auf einer DB-Instance.

Formate für den Datenimport

MariaDB unterstützt zwei Datenimportformate: Flatfiles und SQL. Lesen Sie die Informationen zu den einzelnen Formaten, um die beste Option für Ihre Bedürfnisse zu ermitteln.

Flache Dateien

Laden Sie für kleine Transaktionen Flatfiles mit`LOAD DATA LOCAL INFILE`. Dieses Datenimportformat bietet gegenüber der Verwendung von SQL die folgenden Vorteile:

- Weniger Netzwerkverkehr
- Niedrigere Datenübertragungskosten
- Geringerer Aufwand für die Datenbankverarbeitung
- Schnellere Verarbeitung

`LOAD DATA LOCAL INFILE`lädt die gesamte Flat-Datei als eine Transaktion. Halten Sie die Größe der einzelnen Dateien klein, um die folgenden Vorteile zu erzielen:

- Wiederaufnahmefunktion — Sie können verfolgen, welche Dateien geladen wurden. Wenn beim Laden ein Problem auftritt, können Sie dort weitermachen, wo Sie aufgehört haben. Möglicherweise müssen Sie einige Daten erneut an Amazon RDS übertragen, aber bei kleinen Dateien ist die Menge der erneuten Übertragung minimal.
- Paralleles Laden von Daten — Wenn Sie über ausreichend IOPS und Netzwerkbandbreite für das Laden einer einzelnen Datei verfügen, kann das parallele Laden Zeit sparen.
- Steuerung der Laderate — Wenn sich Ihre Datenlast negativ auf andere Prozesse auswirkt, können Sie die Laderate steuern, indem Sie das Intervall zwischen den Dateien verlängern.

Große Transaktionen verringern die Vorteile, die sich aus `LOAD DATA LOCAL INFILE` dem Import von Daten ergeben. Wenn Sie eine große Datenmenge nicht in kleinere Dateien aufteilen können, sollten Sie SQL verwenden.

SQL

SQL hat einen Hauptvorteil gegenüber flachen Dateien: Sie können die Transaktionsgröße leicht klein halten. Das Laden von SQL kann jedoch erheblich länger dauern als das Laden von Flatfiles. Außerdem kann es nach einem Fehler schwierig sein, zu bestimmen, wo weitergemacht werden soll — Sie können Mariadb-Dump-Dateien nicht neu starten. Wenn beim Laden der Mariadb-Dump-Datei ein Fehler auftritt, müssen Sie die Datei ändern oder ersetzen, bevor der Ladevorgang fortgesetzt werden kann. Alternativ können Sie, nachdem Sie die Ursache des Fehlers behoben haben, den Zeitpunkt vor dem Laden wiederherstellen und die Datei erneut senden. Weitere Informationen finden Sie unter Point-in-time Wiederherstellung.

Verwenden von Amazon RDS-DB-Snapshots für Datenbank-Checkpoints

Wenn Sie Daten über lange Zeiträume — wie Stunden oder Tage — ohne binäre Protokollierung laden, verwenden Sie DB-Snapshots, um regelmäßige Checkpoints für die Datensicherheit bereitzustellen. Jeder DB-Snapshot erstellt eine konsistente Kopie Ihrer Datenbank-Instance,

die bei Systemausfällen oder Datenbeschädigungen als Wiederherstellungspunkt dient. Da DB-Snapshots schnell sind, hat häufiges Checkpointing nur minimale Auswirkungen auf die Ladeleistung. Sie können frühere DB-Snapshots löschen, ohne die Haltbarkeit der Datenbank oder die Wiederherstellungsmöglichkeiten zu beeinträchtigen. Weitere Hinweise zu DB-Snapshots finden Sie unter. Verwaltung manueller Backups

Verkürzung der Ladezeiten der Datenbank

Die folgenden Punkte sind zusätzliche Tipps zur Verkürzung der Ladezeiten:

- Erstellen Sie alle sekundären Indizes, bevor Sie Daten in MariaDB-Datenbanken laden. Im Gegensatz zu anderen Datenbanksystemen erstellt MariaDB die gesamte Tabelle neu, wenn Sekundärindizes hinzugefügt oder geändert werden. Dieser Prozess erstellt eine neue Tabelle mit Indexänderungen, kopiert alle Daten und löscht die Originaltabelle.
- Lädt Daten in der Reihenfolge der Primärschlüssel. Für InnoDB-Tabellen kann dies die Ladezeiten um 75 bis 80% und die Datendateigröße um 50% reduzieren.
- Deaktivieren Sie Fremdschlüsseleinschränkungen, indem Sie `foreign_key_checks` auf `0` setzen. Dies ist häufig für Flatfiles erforderlich, die mit geladen werden`LOAD DATA LOCAL INFILE`. Bei jedem Ladevorgang beschleunigt die Deaktivierung der Fremdschlüsselprüfungen das Laden der Daten. Aktivieren Sie nach Abschluss des Ladevorgangs die Einschränkungen erneut, indem Sie die Einstellungen `foreign_key_checks` auf festlegen `1` und die Daten überprüfen.
- Laden Sie Daten parallel, sofern Sie sich nicht einem Ressourcenlimit nähern. Verwenden Sie gegebenenfalls partitionierte Tabellen, um das gleichzeitige Laden über mehrere Tabellensegmente hinweg zu ermöglichen.
- Um den Aufwand bei der SQL-Ausführung zu reduzieren, kombinieren Sie mehrere `INSERT` Anweisungen zu einzelnen Operationen mit mehreren `INSERT` Werten. `mariadb-dump`implementiert diese Optimierung automatisch.
- Reduzieren Sie InnoDB-Protokoll-IO-Operationen, indem Sie `innodb_flush_log_at_trx_commit` auf `0` setzen. Stellen Sie nach Abschluss des Ladevorgangs `innodb_flush_log_at_trx_commit` auf `1` wieder her.

> **Warning**
>
> Die Einstellung `innodb_flush_log_at_trx_commit` auf `0` bewirkt, dass InnoDB seine Logs jede Sekunde leert, nicht bei jedem Commit. Diese Einstellung erhöht die Leistung, kann jedoch zu Transaktionsverlusten bei Systemausfällen führen.

- Wenn Sie Daten in eine DB-Instance laden, die keine Read Replicas hat, setzen Sie den Wert `sync_binlog` auf`0`. Stellen Sie nach Abschluss des Ladevorgangs `sync_binlog parameter` auf 1 wieder her.
- Laden Sie Daten in eine Single-AZ-DB-Instance, bevor Sie die DB-Instance in eine Multi-AZ-Bereitstellung konvertieren. Wenn die DB-Instance bereits eine Multi-AZ-Bereitstellung verwendet, empfehlen wir nicht, zum Laden von Daten zu einer Single-AZ-Bereitstellung zu wechseln. Dies führt nur zu geringfügigen Verbesserungen

Daten aus einer externen MariaDB-Datenbank in eine Amazon RDS for MariaDB-DB-Instance importieren

Sie können Daten aus einer vorhandenen MariaDB- oder MySQL-Datenbank in eine RDS for MariaDB- oder RDS for MySQL-DB-Instance importieren. Dazu kopieren Sie die Datenbank mit mysqldump und leiten sie direkt an die DB-Instance RDS for MariaDB oder RDS for MySQL weiter. Das Befehlszeilen-Hilfsprogramm `mysqldump` wird üblicherweise verwendet, um Backups zu erstellen und Daten aus einem MariaDB- oder MySQL-Server in einen anderen zu übertragen. Es ist in der MySQL- und MariaDB-Client-Software enthalten.

In MariaDB 10.5 heißt der Client mariadb-dump. Ab MariaDB 11.0.1 müssen Sie anstelle von verwendenmariadb-dump. `mysqldump`

> **Note**
>
> Wenn Sie große Datenmengen mit einer MySQL-DB-Instance importieren oder exportieren, ist es zuverlässiger und schneller, Daten mithilfe von `xtrabackup` Sicherungsdateien und Amazon S3 in und aus Amazon RDS zu verschieben. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance. Amazon RDS unterstützt `xtrabackup` MariaDB nicht oder. `mariabackup` Amazon RDS unterstützt nur den Import aus Amazon S3 für MySQL.

Ein typischer `mysqldump` Befehl zum Verschieben von Daten aus einer externen Datenbank in eine Amazon RDS-DB-Instance sieht dem folgenden Beispiel ähnlich. Ersetzen Sie Werte durch Ihre eigenen Informationen. Ersetzen Sie für MariaDB 11.0.1 und höhere Versionen durch und `mysqldump` durch`mariadb-dump`. `mysql` `mariadb`

```
mysqldump -u local_user \
```

```
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Lassen Sie ein Leerzeichen zwischen der Option `-p` und dem eingegeben Passwort.
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die in diesem Beispiel gezeigten Eingabeaufforderungen angeben.

Stellen Sie sicher, dass Sie die folgenden Empfehlungen und Überlegungen kennen:

- Schließen Sie die folgenden Schemas aus der Dump-Datei aus:
  - `sys`
  - `performance_schema`
  - `information_schema`

  Das `mariadb-dump` Hilfsprogramm `mysqldump` and schließt diese Schemas standardmäßig aus.
- Wenn Sie Benutzer und Rechte migrieren müssen, sollten Sie in Erwägung ziehen, ein Tool zu verwenden, das die Data Control Language (DCL) für deren Neuerstellung generiert, z. B. das Hilfsprogramm. pt-show-grants
- Um den Import durchzuführen, stellen Sie sicher, dass der Benutzer Zugriff auf die DB-Instance hat. Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Die folgenden Parameter werden verwendet:

- `-u` *`local_user`* – für die Angabe eines Benutzernamens. Geben Sie bei der ersten Verwendung dieses Parameters den Namen eines Benutzerkontos in der lokalen MariaDB- oder MySQL-Datenbank an, das Sie mit dem `--databases` Parameter identifizieren.

- `--databases` *`database_name`* – für die Angabe des Datenbanknamens in der lokalen MariaDB- oder MySQL-Instance, die Sie in Amazon RDS importieren möchten.
- `--single-transaction`: zur Sicherstellung, dass alle aus der lokalen Datenbank geladenen Daten mit einem einzelnen Zeitpunkt übereinstimmen Wenn andere Prozesse die Daten ändern, während diese von `mysqldump` gelesen werden, kann durch die Verwendung dieses Parameters die Datenintegrität erhalten bleiben.
- `--compress`: für die Reduzierung des Verbrauchs der Netzwerkbandbreite, indem Daten vor dem Sendevorgang aus der lokalen Datenbank an Amazon RDS komprimiert werden.
- `--order-by-primary`: für die Reduzierung der Ladezeit durch Sortieren der Daten jeder Tabelle nach entsprechendem Primärschlüssel
- `--routines`— Verwenden Sie diese Option, wenn Routinen wie gespeicherte Prozeduren oder Funktionen in der Datenbank vorhanden sind, die Sie kopieren. Setzen Sie den Parameter auf`0`, wodurch die Routinen während des Importvorgangs ausgeschlossen werden. Erstellen Sie die Routinen später manuell in der Amazon RDS-Datenbank neu.
- `--triggers`— Verwenden Sie diese Option, wenn in der Datenbank, die Sie kopieren, Trigger vorhanden sind. Setzen Sie den Parameter auf`0`, wodurch die Trigger während des Importvorgangs ausgeschlossen werden. Erstellen Sie die Trigger später manuell in der Amazon RDS-Datenbank neu.
- `--events`— Verwenden Sie diese Option, wenn Ereignisse in der Datenbank vorhanden sind, die Sie kopieren. Setzen Sie den Parameter auf`0`, wodurch die Ereignisse während des Importvorgangs ausgeschlossen werden. Erstellen Sie die Ereignisse später manuell in der Amazon RDS-Datenbank neu.
- `-p`*`local_password`* – für die Angabe eines Passworts. Geben Sie bei der ersten Verwendung dieses Parameters das Passwort für das Benutzerkonto an, das Sie mit dem ersten `-u` Parameter identifizieren.
- `-u` *`RDS_user`* – für die Angabe eines Benutzernamens. Bei der zweiten Verwendung dieses Parameters geben Sie den Namen eines Benutzerkontos in der Standarddatenbank für die MariaDB- oder MySQL-DB-Instance an, die Sie mit dem `--host` Parameter identifizieren.
- `--port` *`port_number`* – für die Angabe des Ports für Ihre MariaDB- oder MySQL-DB-Instance. Standardmäßig ist dies 3306, sofern Sie den Wert nicht beim Erstellen der DB-Instance geändert haben.
- `--host` *`host_name`* – für die Angabe des Domain-Name-System(DNS)-Namens aus dem Endpunkt der Amazon-RDS-DB-Instance, zum Beispiel, `myinstance.123456789012.us-`

`east-1.rds.amazonaws.com`. Sie finden den Endpunktwert in den DB-Instance-Details in der Amazon RDS-Konsole.

- `-p`*`RDS_password`* – für die Angabe eines Passworts. Beim zweiten Gebrauch dieses Parameters geben Sie den Namen eines Passworts einer lokalen MySQL- oder MariaDB-Datenbank an, die durch den zweiten Parameter `-u` bezeichnet wird.

Stellen Sie sicher, dass Sie alle gespeicherten Prozeduren, Auslöser, Funktionen oder Ereignisse manuell in Ihrer Amazon-RDS-Datenbank erstellen. Wenn Sie eines dieser Objekte in der Datenbank haben, die Sie kopieren, schließen Sie es aus, wenn Sie `mysqldump` oder ausführen`mariadb-dump`. Fügen Sie dazu Ihrem `mariadb-dump` Befehl `mysqldump` oder die folgenden Parameter hinzu:

- `--routines=0`
- `--triggers=0`
- `--events=0`

Beispiel

Im folgenden Beispiel wird die `world` Beispieldatenbank auf dem lokalen Host in eine RDS for MySQL-DB-Instance kopiert. Ersetzen Sie Werte durch Ihre eigenen Informationen. Um die Beispieldatenbank in eine RDS for MariaDB-DB-Instance zu kopieren, `mysqldump` ersetzen Sie sie durch `mariadb-dump` und `mysql` durch. `mariadb`

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sudo mysqldump -u local_user \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u rds_user \
        --port=3306 \
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com \
        -pRDS_password
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

Führen Sie den folgenden Befehl in einer Befehlszeile aus, die geöffnet wurde, indem Sie im Windows-Programmmenü mit der rechten Maustaste auf Eingabeaufforderung klicken und Als Administrator ausführen wählen. Ersetzen Sie Werte durch Ihre eigenen Informationen. Um die Beispieldatenbank in eine RDS for MariaDB-DB-Instance zu kopieren, `mysqldump` ersetzen Sie sie durch `mariadb-dump` und `mysql` durch. `mariadb`

```
mysqldump -u local_user ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocal_password | mysql -u RDS_user ^
        --port=3306 ^
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com ^
        -pRDS_password
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.

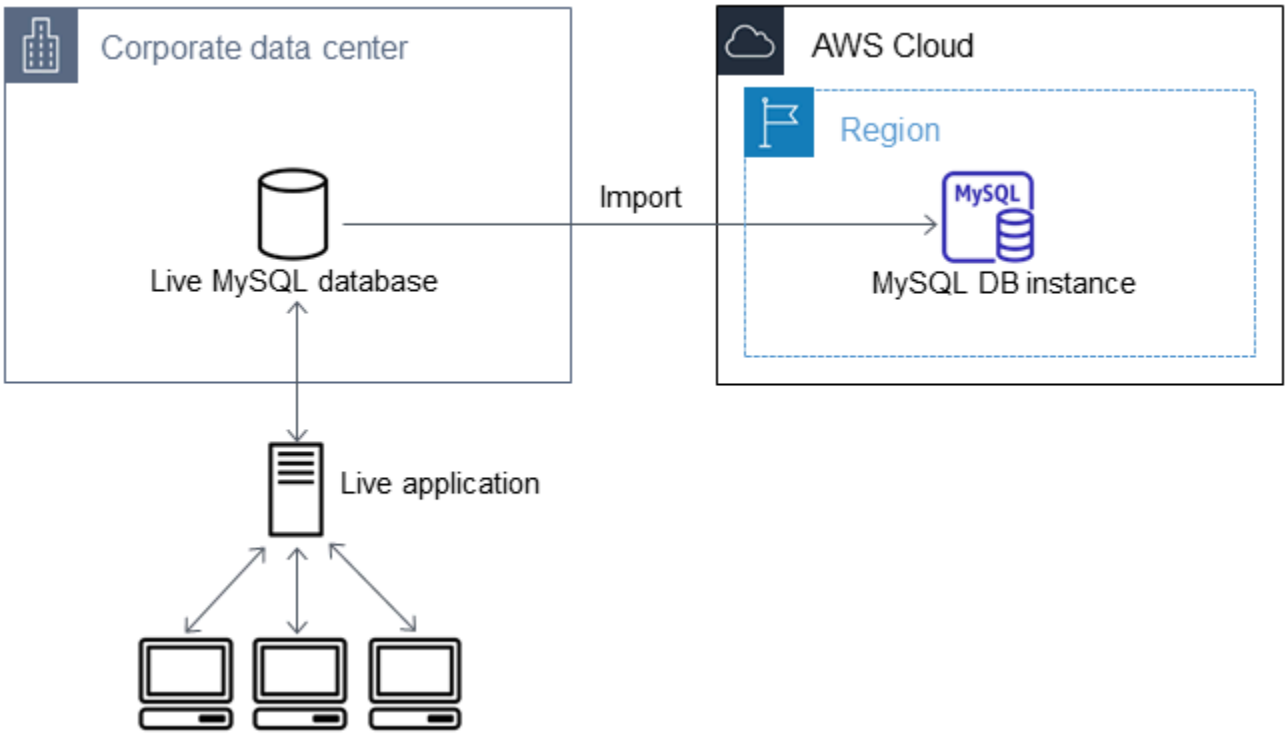
Import von Daten in eine Amazon RDS for MariaDB-DB-Instance mit reduzierter Ausfallzeit

In einigen Fällen müssen Sie möglicherweise Daten aus einer externen MariaDB- oder MySQL-Datenbank, die eine Live-Anwendung unterstützt, in eine RDS for MariaDB-DB-Instance, eine RDS for MySQL-DB-Instance oder einen RDS for MySQL Multi-AZ-DB-Cluster importieren. Nutzen Sie das folgende Verfahren, um die Auswirkungen auf die Verfügbarkeit der Anwendung zu minimieren. Dieses Verfahren kann außerdem hilfreich sein, wenn Sie mit einer sehr großen Datenbank arbeiten. Mit diesem Verfahren können Sie die Importkosten senken, indem Sie die Datenmenge reduzieren, die über das Netzwerk übertragen wird. AWS

In diesem Verfahren übertragen Sie eine Kopie Ihrer Datenbankdaten auf eine EC2 Amazon-Instance und importieren die Daten in eine neue Amazon RDS-Datenbank. Anschließend verwenden Sie die Replikation, um die Amazon RDS-Datenbank up-to-date mit Ihrer externen Live-Instance zu verknüpfen, bevor Sie Ihre Anwendung auf die Amazon RDS-Datenbank umleiten. Wenn die externe Instance MariaDB 10.0.24 oder höher ist und die Zielinstanz RDS for MariaDB ist, konfigurieren Sie die MariaDB-Replikation basierend auf globalen Transaktions-Identifikatoren (). GTIDs Andernfalls konfigurieren Sie die Replikation basierend auf den Binärprotokollkoordinaten. Wir empfehlen die GTID-basierte Replikation, wenn Ihre externe Datenbank diese unterstützt, da diese Methode zuverlässiger ist. Weitere Informationen finden Sie unter Global Transaction ID in der MariaDB-Dokumentation.

> **Note**
>
> Wenn Sie Daten in eine RDS for MySQL-DB-Instance importieren möchten und Ihr Szenario dies unterstützt, empfehlen wir, Daten mithilfe von Sicherungsdateien und Amazon S3 in und aus Amazon RDS zu verschieben. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance.

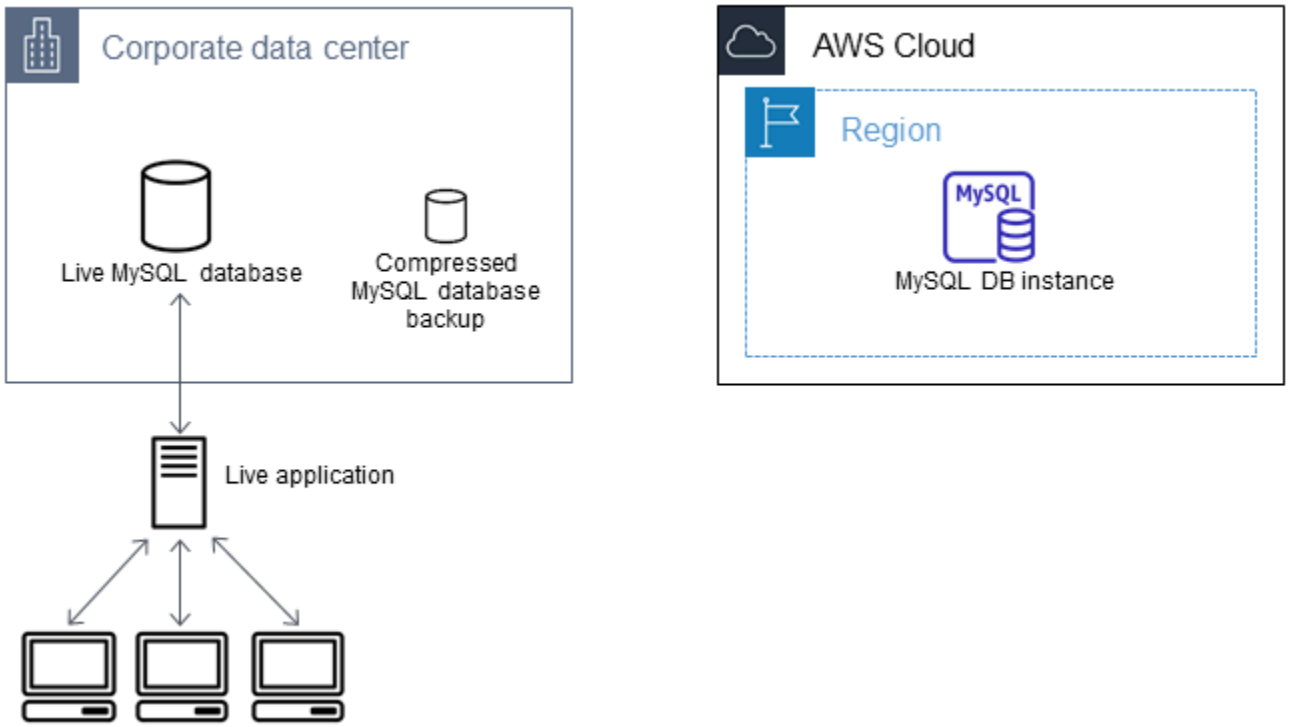
Das folgende Diagramm zeigt den Import einer externen MySQL-Datenbank in eine MySQL-Datenbank auf Amazon RDS.



Aufgabe 1: Erstellen Sie eine Kopie Ihrer vorhandenen Datenbank

Der erste Schritt bei der Migration von großen Datenmengen in eine Datenbank von RDS für MariaDB oder RDS für MySQL mit minimaler Ausfallzeit ist das Erstellen einer Kopie der Quelldaten.

Das folgende Diagramm zeigt die Erstellung einer Sicherungskopie der MySQL-Datenbank.



Sie können das Hilfsprogramm `mysqldump` verwenden, um ein Datenbank-Backup im SQL-Format oder im separierten Textformat zu erstellen. In MariaDB 10.5 heißt der Client mariadb-dump. Ab MariaDB 11.0.1 müssen Sie anstelle von verwenden`mariadb-dump`. `mysqldump` Es wird empfohlen, mit jedem Format in einer Nichtproduktionsumgebung einen Testlauf durchzuführen, um zu sehen, welche Methode die Ausführungsdauer von mysqldump minimiert.

Wir empfehlen außerdem, die `mysqldump` Leistung gegen den Vorteil abzuwägen, der sich aus der Verwendung des Textformats mit Trennzeichen beim Laden ergibt. Ein Backup, das ein separiertes Textformat verwendet, erstellt eine tabulatorseparierte Textdatei für jede verworfene Tabelle. Um den Zeitaufwand für den Import Ihrer Datenbank zu reduzieren, können Sie diese Dateien mit dem Befehl `LOAD DATA LOCAL INFILE` parallel laden. Weitere Informationen finden Sie unter Daten für MariaDB laden oder Daten für MySQL laden.

Bevor Sie mit dem Sicherungsvorgang beginnen, müssen Sie die Optionen für die Replikation in der nach Amazon RDS zu kopierenden MariaDB- oder MySQL-Datenbank einstellen. Die Optionen

für die Replikation schließen die Aktivierung der Binärprotokollierung und das Einstellen einer eindeutigen Server-ID mit ein. Das Einstellen dieser Optionen veranlasst den Server, mit der Protokollierung Ihrer Datenbanktransaktionen zu beginnen, und bereitet ihn darauf vor, später im Vorgang als Quellreplikationsinstance zu agieren.

Stellen Sie sicher, dass Sie die folgenden Empfehlungen und Überlegungen kennen:

- Verwenden Sie die Option `--single-transaction` mit `mysqldump`, da sie einen einheitlichen Zustand der Datenbank speichert. Um eine gültige Dump-Datei sicherzustellen, führen Sie beim Ausführen von `mysqldump` keine DDL-Anweisungen (Data Definition Language) aus. Sie können ein Wartungsfenster für diese Abläufe planen.
- Schließen Sie die folgenden Schemas aus der Dump-Datei aus:
  - `sys`
  - `performance_schema`
  - `information_schema`

  Das `mariadb-dump` Hilfsprogramm `mysqldump` and schließt diese Schemas standardmäßig aus.
- Wenn Sie Benutzer und Rechte migrieren müssen, sollten Sie in Erwägung ziehen, ein Tool zu verwenden, das die Data Control Language (DCL) für deren Neuerstellung generiert, z. B. das Hilfsprogramm. pt-show-grants

So stellen Sie Optionen für die Replikation ein:

1. Bearbeiten Sie die `my.cnf`-Datei. Diese Datei befindet sich normalerweise unter. `/etc`

   ```
   sudo vi /etc/my.cnf
   ```

   Fügen Sie die Optionen `log_bin` und `server_id` zum Abschnitt `[mysqld]` hinzu. Die Option `log_bin` bietet eine Dateinamenkennung für Binärprotokolldateien. Die Option `server_id` stellt eine eindeutige Kennung für den Server für Quelle-Replica-Beziehungen bereit.

   Das folgende Beispiel zeigt den aktualisierten `[mysqld]` Abschnitt einer `my.cnf` Datei:

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

Weitere Informationen finden Sie unter Einstellung der Konfiguration der Replikationsquelle in der MySQL-Dokumentation.

2. Für die Replikation mit einem Multi-AZ-DB-Cluster setzen Sie den `ENFORCE_GTID_CONSISTENCY` und den `GTID_MODE` Parameter auf`ON`.

```
mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

Diese Einstellungen sind für die Replikation mit einer DB-Instance nicht erforderlich.

3. Den Service `mysql` neu starten.

```
sudo service mysqld restart
```

So erstellen Sie eine Sicherungskopie für Ihre bestehende Datenbank:

1. Erstellen Sie mit dem `mysqldump` Hilfsprogramm oder dem Hilfsprogramm eine Sicherungskopie Ihrer Daten und geben Sie dabei entweder das `mariadb-dump` SQL-Format oder das Textformat mit Trennzeichen an.

   Geben Sie für MySQL 8.0.25 und niedrigere Versionen `--master-data=2` an, dass eine Sicherungsdatei erstellt werden soll, mit der die Replikation zwischen Servern gestartet werden kann. Geben Sie für MySQL 8.0.26 und höhere Versionen `--source-data=2` an, dass eine Sicherungsdatei erstellt werden soll, mit der die Replikation zwischen Servern gestartet werden kann. Weitere Informationen finden Sie unter mysqldump — A Database Backup Program in der MySQL-Dokumentation.

   Um die Leistung zu verbessern und die Datenintegrität sicherzustellen, verwenden Sie die Optionen `--order-by-primary` und `--single-transaction` für. `mysqldump`

   Um zu vermeiden, dass die MySQL-Systemdatenbank in das Backup aufgenommen wird, verwenden Sie die `--all-databases` Option mit nicht`mysqldump`. Weitere Informationen finden Sie unter Erstellen eines Datensnapshots mit mysqldump in der MySQL-Dokumentation.

   Stellen Sie `chmod` bei Bedarf sicher, dass das Verzeichnis, in dem die Sicherungsdatei erstellt wird, beschreibbar ist.

> **Important**
>
> Führen Sie unter Windows die Eingabeaufforderung als Administrator aus.

- Verwenden Sie den folgenden Befehl, um eine SQL-Ausgabe zu erzeugen:

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
    --order-by-primary \
    -r backup.sql \
    -u local_user \
    -ppassword
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -ppassword
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.

- Verwenden Sie den folgenden Befehl, um eine Ausgabe mit begrenztem Text zu erzeugen:

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -ppassword
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -ppassword
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.
> Stellen Sie sicher, dass Sie alle gespeicherten Prozeduren, Auslöser, Funktionen oder Ereignisse manuell in Ihrer Amazon-RDS-Datenbank erstellen. Falls Sie eines

dieser Objekte in der Datenbank haben, die Sie kopieren, schließen Sie sie aus, wenn Sie `mysqldump` ausführen. Fügen Sie dazu Ihrem `mysqldump` Befehl die folgenden Argumente hinzu:

- `--routines=0`
- `--triggers=0`
- `--events=0`

Bei MySQL 8.0.22 und niedrigeren Versionen und MariaDB-Versionen wird ein Kommentar zurückgegeben, wenn Sie das Format mit Trennzeichen ausführen `mysqldump` und angeben. `CHANGE MASTER TO` Dieser Kommentar beinhaltet den Namen und die Position der Hauptprotokolldatei. Wenn Sie für MySQL 8.0.23 und höhere Versionen das Format mit getrenntem Text `mysqldump` verwenden, wird ein Kommentar zurückgegeben. `CHANGE REPLICATION SOURCE TO` Dieser Kommentar enthält den Namen und die Position der Quellprotokolldatei. Wenn es sich bei der externen Instanz um etwas anderes als MariaDB 10.0.24 und höhere Versionen oder MySQL 8.0.23 und höhere Versionen handelt, notieren Sie sich die Werte für und. `MASTER_LOG_FILE` `MASTER_LOG_POS` Sie benötigen diese Werte beim Einrichten der Replikation.

Die folgende Ausgabe wird für MariaDB-Versionen und für MySQL 8.0.22 und niedrigere Versionen zurückgegeben:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

Die folgende Ausgabe wird für MySQL 8.0.23 und höhere Versionen zurückgegeben:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE SOURCE TO SOURCE_LOG_FILE='mysql-bin-changelog.000031',
 SOURCE_LOG_POS=107;
```

Für MySQL 8.0.22 und niedrigere Versionen können Sie, wenn Sie das SQL-Format verwenden, den Namen und die Position der Master-Logdatei im `CHANGE MASTER TO` Kommentar in der Backup-Datei abrufen. Für MySQL 8.0.23 und höhere Versionen können Sie, wenn Sie

das SQL-Format verwenden, den Namen und die Position der Quellprotokolldatei im `CHANGE REPLICATION SOURCE TO` Kommentar in der Sicherungsdatei abrufen. Wenn die externe Instance MariaDB Version 10.0.24 oder höher ist, können Sie die GTID im nächsten Schritt abrufen.

2. Wenn es sich bei der von Ihnen verwendeten externen Instanz um MariaDB-Version 10.0.24 oder höher handelt, verwenden Sie die GTID-basierte Replikation. Führen Sie es `SHOW MASTER STATUS` auf der externen MariaDB-Instanz aus, um den Namen und die Position der binären Protokolldatei abzurufen, und konvertieren Sie sie dann in eine GTID, indem Sie sie `BINLOG_GTID_POS` auf der externen MariaDB-Instanz ausführen.

   ```
   SELECT BINLOG_GTID_POS('binary_log_file_name', binary_log_file_position);
   ```

   Beachten Sie die zurückgegebene GTID. Sie benötigen die GTID, um die Replikation zu konfigurieren.

3. Komprimieren Sie die kopierten Daten, um die Menge der Netzwerkressourcen zu reduzieren, die benötigt werden, um Ihre Daten in eine Amazon-RDS-Datenbank zu kopieren. Notieren Sie sich die Größe der Backup-Datei. Sie benötigen diese Informationen, um zu bestimmen, wie groß eine EC2 Amazon-Instance erstellt werden soll. Wenn Sie fertig sind, komprimieren Sie die Sicherungsdatei mithilfe von GZIP oder Ihrem bevorzugten Komprimierungsprogramm.

   - Verwenden Sie den folgenden Befehl, um die SQL-Ausgabe zu komprimieren:

     ```
     gzip backup.sql
     ```

   - Verwenden Sie den folgenden Befehl, um die Ausgabe mit begrenztem Text zu komprimieren:

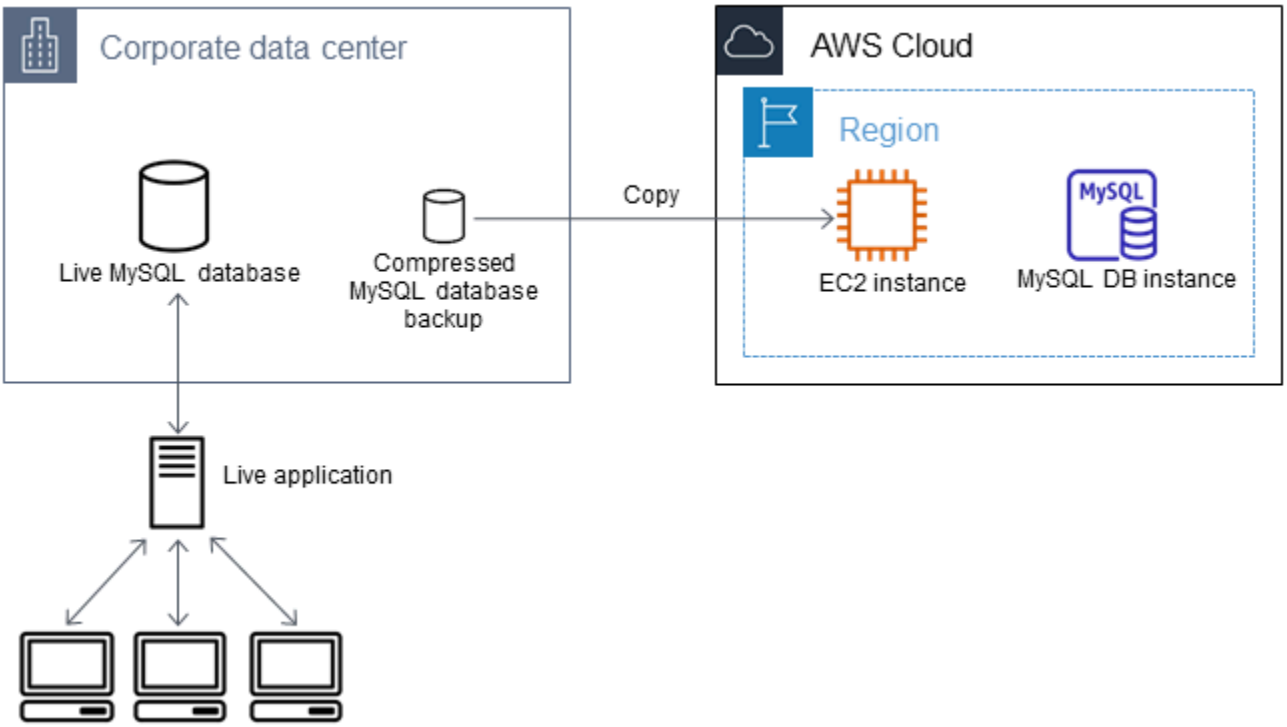
     ```
     tar -zcvf backup.tar.gz target_directory
     ```

Aufgabe 2: Erstellen Sie eine EC2 Amazon-Instance und kopieren Sie die komprimierte Datenbank

Das Kopieren Ihrer komprimierten Datenbank-Backup-Datei auf eine EC2 Amazon-Instance benötigt weniger Netzwerkressourcen als das direkte Kopieren unkomprimierter Daten zwischen Datenbank-Instances. Nachdem sich Ihre Daten in Amazon befinden EC2, können Sie sie von dort direkt in Ihre MariaDB- oder MySQL-Datenbank kopieren. Damit Sie Kosten für Netzwerkressourcen sparen können, muss sich Ihre EC2 Amazon-Instance in derselben Umgebung AWS-Region wie Ihre Amazon RDS-DB-Instance befinden. Wenn sich die EC2 Amazon-Instance in derselben Datenbank

AWS-Region wie Ihre Amazon RDS-Datenbank befindet, verringert sich auch die Netzwerklatenz während des Imports.

Das folgende Diagramm zeigt das Kopieren der Datenbanksicherung auf eine EC2 Amazon-Instance.



Um eine EC2 Amazon-Instance zu erstellen und Ihre Daten zu kopieren

1. Erstellen Sie in AWS-Region dem Bereich, in dem Sie die Amazon RDS-Datenbank erstellen möchten, eine Virtual Private Cloud (VPC), eine VPC-Sicherheitsgruppe und ein VPC-Subnetz. Stellen Sie sicher, dass die eingehenden Regeln für Ihre VPC-Sicherheitsgruppe IP-Adressen zulassen, die für eine Verbindung Ihrer Anwendung mit erforderlich sin AWS. Sie können beispielsweise einen Bereich von IP-Adressen `203.0.113.0/24` oder eine andere VPC-Sicherheitsgruppe angeben. Sie können die Amazon VPC-Konsole verwenden VPCs, um Subnetze und Sicherheitsgruppen zu erstellen und zu verwalten. Weitere Informationen finden Sie unter Erste Schritte mit Amazon VPC im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
2. Öffnen Sie die EC2 Amazon-Konsole und wählen Sie die aus, die sowohl Ihre EC2 Amazon-Instance als auch Ihre Amazon RDS-Datenbank enthalten AWS-Region soll. Starten Sie eine EC2 Amazon-Instance mit der VPC, dem Subnetz und der Sicherheitsgruppe, die Sie in Schritt 1 erstellt haben. Stellen Sie sicher, dass Sie einen Instance-Typ mit genügend Speicherplatz für Ihre unkomprimierte Datenbank-Sicherungsdatei ausgewählt haben. Einzelheiten zu EC2 Amazon-Instances finden Sie unter Erste Schritte mit Amazon EC2 im Amazon Elastic Compute Cloud-Benutzerhandbuch.

3. Um von Ihrer EC2 Amazon-Instance aus eine Verbindung zu Ihrer Amazon RDS-Datenbank herzustellen, bearbeiten Sie Ihre VPC-Sicherheitsgruppe. Fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die die private IP-Adresse Ihrer EC2 Instance angibt. Sie finden die private IP-Adresse auf der Registerkarte „Details“ im Bereich „Instanz“ im EC2 Konsolenfenster. Um die VPC-Sicherheitsgruppe zu bearbeiten und eine Regel für eingehenden Datenverkehr hinzuzufügen, wählen Sie im Navigationsbereich der EC2 Konsole Sicherheitsgruppen, wählen Sie Ihre Sicherheitsgruppe aus und fügen Sie dann eine eingehende Regel für MySQL oder Aurora hinzu, die die private IP-Adresse Ihrer Instance angibt. EC2 Informationen zum Hinzufügen einer Regel für eingehenden Datenverkehr zu einer VPC-Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
4. Kopieren Sie Ihre komprimierte Datenbank-Backup-Datei von Ihrem lokalen System auf Ihre EC2 Amazon-Instance. Verwenden Sie `chmod`, falls erforderlich, um sicherzustellen, dass Sie über Schreibberechtigungen für das Zielverzeichnis der EC2 Amazon-Instance verfügen. Sie können `scp` oder einen Secure-Shell(SSH)-Client verwenden, um die Datei zu kopieren. Der folgende Befehl ist ein `scp` Beispielbefehl:

   ```
   scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
   ```

   > **Important**
   >
   > Achten Sie beim Kopieren vertraulicher Daten darauf, ein sicheres Netzwerkübertragungsprotokoll zu verwenden.

5. Stellen Sie mithilfe der folgenden Befehle Connect zu Ihrer EC2 Amazon-Instance her und installieren Sie die neuesten Updates und die MySQL-Client-Tools:

   ```
   sudo yum update -y
   sudo yum install mysql -y
   ```

   Weitere Informationen finden Sie unter Connect Ihrer Instance für Linux-Instances herstellen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

   > **Important**
   >
   > In diesem Beispiel wird der MySQL-Client auf einem Amazon Machine Image (AMI) für eine Amazon-Linux-Verteilung installiert. In diesem Beispiel wird der MySQL-Client nicht auf einer anderen Distribution wie Ubuntu oder Red Hat Enterprise Linux installiert.

Informationen zur Installation von MySQL finden Sie unter Installation von MySQL in der MySQL-Dokumentation.

6. Während Sie mit Ihrer EC2 Amazon-Instance verbunden sind, dekomprimieren Sie Ihre Datenbank-Backup-Datei. Die folgenden Befehle sind Beispiele.
   - Verwenden Sie den folgenden Befehl, um eine SQL-Ausgabe zu extrahieren:

     ```
     gzip backup.sql.gz -d
     ```

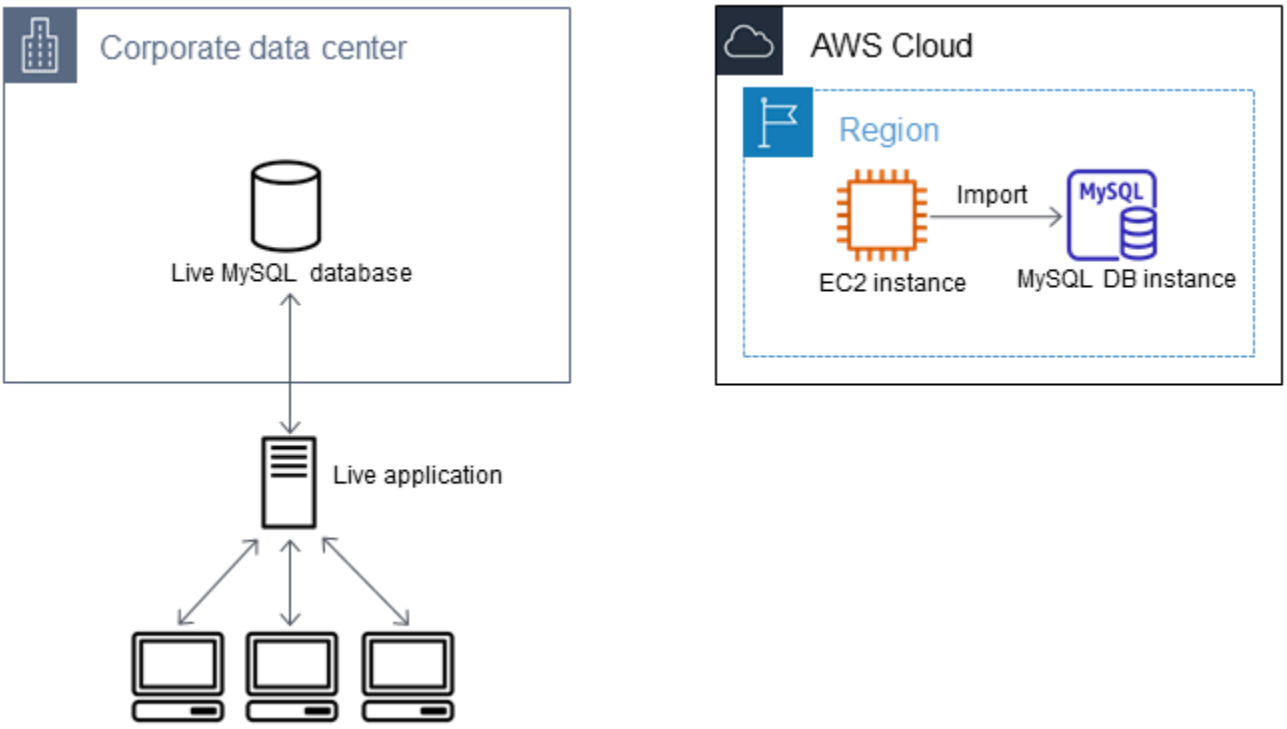
   - Verwenden Sie den folgenden Befehl, um eine separierte Textausgabe zu extrahieren:

     ```
     tar xzvf backup.tar.gz
     ```

Aufgabe 3: Erstellen Sie eine MySQL- oder MariaDB-Datenbank und importieren Sie Daten aus Ihrer Amazon-Instance EC2

Indem Sie eine RDS for MariaDB-DB-Instance, eine RDS for MySQL-DB-Instance oder einen RDS for MySQL Multi-AZ-DB-Cluster in derselben Umgebung AWS-Region wie Ihre EC2 Amazon-Instance erstellen, können Sie die Datenbank-Backup-Datei EC2 schneller als über das Internet von Amazon importieren.

Das folgende Diagramm zeigt den Import des Backups aus einer EC2 Amazon-Instance in eine MySQL-Datenbank.



So erstellen Sie eine MariaDB- oder MySQL-Datenbank und importieren Ihre Daten

1. Bestimmen Sie, welche DB-Instance-Klasse und wie viel Speicherplatz erforderlich sind, um den erwarteten Workload für diese Amazon-RDS-Datenbank unterstützen zu können. Bei diesem Vorgang sollten Sie auch entscheiden, wie viel Speicherplatz und Verarbeitungskapazität für Ihre Datenladevorgänge ausreichen. Entscheiden Sie außerdem, was zur Bewältigung der Produktionslast erforderlich ist. Sie können diese Faktoren anhand der Größe und der Ressourcen Ihrer Quell-MariaDB- oder MySQL-Datenbank einschätzen. Weitere Informationen finden Sie unter .
2. Erstellen Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster in dem AWS-Region , der Ihre EC2 Amazon-Instance enthält.

   Um einen RDS for MySQL Multi-AZ-DB-Cluster zu erstellen, folgen Sie den Anweisungen unterEinen Multi-AZ-DB-Cluster für Amazon RDS erstellen.

   Um eine RDS for MariaDB- oder RDS for MySQL-DB-Instance zu erstellen, folgen Sie den Anweisungen unter Erstellen einer Amazon RDS-DB-Instance und verwenden Sie die folgenden Richtlinien:

   - Geben Sie eine DB-Engine-Version an, die mit Ihrer Quell-DB-Instance kompatibel ist.

- Wenn Ihre Quell-Instance MySQL 5.5.x ist, muss Ihre Amazon-RDS-DB-Instance MySQL sein.
- Wenn Ihre Quell-Instance MySQL 5.6.x oder 5.7.x ist, muss Ihre Amazon RDS-DB-Instance MySQL oder MariaDB sein.
- Wenn Ihre Quell-Instance MySQL 8.0.x ist, muss Ihre Amazon RDS-DB-Instance MySQL 8.0.x sein.
- Wenn Ihre Quell-Instance MySQL 8.4.x ist, muss die Amazon RDS-DB-Instance MySQL 8.4.x sein.
- Wenn Ihre Quell-Instance MariaDB 5.5 oder höher ist, muss Ihre Amazon-RDS-DB-Instance MariaDB sein.

- Geben Sie dieselbe Virtual Private Cloud (VPC) und dieselbe VPC-Sicherheitsgruppe wie für Ihre EC2 Amazon-Instance an. Dieser Ansatz stellt sicher, dass Ihre EC2 Amazon-Instance und Ihre Amazon RDS-Instance im Netzwerk füreinander sichtbar sind. Stellen Sie sicher, dass Ihre DB-Instance öffentlich zugänglich ist. Um die Replikation mit Ihrer Quelldatenbank wie im folgenden Abschnitt beschrieben einzurichten, muss Ihre DB-Instance öffentlich zugänglich sein.
- Konfigurieren Sie nicht mehrere Availability Zones, Backup-Aufbewahrungen oder Lesereplikate, nachdem Sie das Datenbank-Backup importiert haben. Wenn dieser Importvorgang abgeschlossen ist, können Sie Multi-AZ und Backup-Aufbewahrung für die Produktions-Instance konfigurieren.

3. Überprüfen Sie die Optionen der Standardkonfiguration für die Amazon-RDS-Datenbank. Wenn in der Standardparametergruppe für die Datenbank die von Ihnen gewünschten Optionen nicht konfiguriert sind, wählen Sie eine andere aus, die die entsprechenden Konfigurationsoptionen enthält, oder erstellen Sie eine neue Parametergruppe. Weitere Hinweise zum Erstellen einer Parametergruppe finden Sie unterParametergruppen für Amazon RDS.
4. Stellen Sie als Hauptbenutzer eine Verbindung mit der neuen Amazon-RDS-Datenbank her. Erstellen Sie die Benutzer, die zur Unterstützung der Administratoren, Anwendungen und Dienste erforderlich sind, die auf die DB-Instance zugreifen müssen. Der Hostname für die Amazon RDS-Datenbank ist der Endpunktwert für diese DB-Instance ohne die Portnummer, zum Beispiel. `mysampledb.123456789012.us-west-2.rds.amazonaws.com` Sie finden den Endpunktwert in den Datenbankdetails in der Amazon RDS-Konsole.
5. Connect zu Ihrer EC2 Amazon-Instance her. Weitere Informationen finden Sie unter Connect Ihrer Instance für Linux-Instances herstellen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

6. Stellen Sie mit dem `mysql` Befehl eine Connect zu Ihrer Amazon RDS-Datenbank als Remote-Host von Ihrer EC2 Amazon-Instance her. Der folgende Befehl ist ein Beispiel:

   ```
   mysql -h host_name -P 3306 -u db_master_user -p
   ```

   Das *host_name* ist der Amazon RDS-Datenbank-Endpunkt.

7. Führen Sie an der `mysql` Eingabeaufforderung den `source` Befehl aus und übergeben Sie ihm den Namen Ihrer Datenbank-Dump-Datei. Dieser Befehl lädt die Daten in die Amazon RDS-DB-Instance.

   - Verwenden Sie für das SQL-Format den folgenden Befehl:

     ```
     mysql> source backup.sql;
     ```

   - Für das Format mit begrenztem Text erstellen Sie zuerst die Datenbank, falls es sich nicht um die Standarddatenbank handelt, die Sie bei der Einrichtung der Amazon RDS-Datenbank erstellt haben.

     ```
     mysql> create database database_name;
     mysql> use database_name;
     ```

     Erstellen Sie anschließend die Tabellen.

     ```
     mysql> source table1.sql
     mysql> source table2.sql
     etc...
     ```

     Importieren Sie dann die Daten.

     ```
     mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     etc...
     ```

     Zur Verbesserung der Leistung können Sie diese Operationen parallel aus mehreren Verbindungen ausführen, damit alle Ihre Tabellen erstellt und die Daten anschließend gleichzeitig geladen werden.

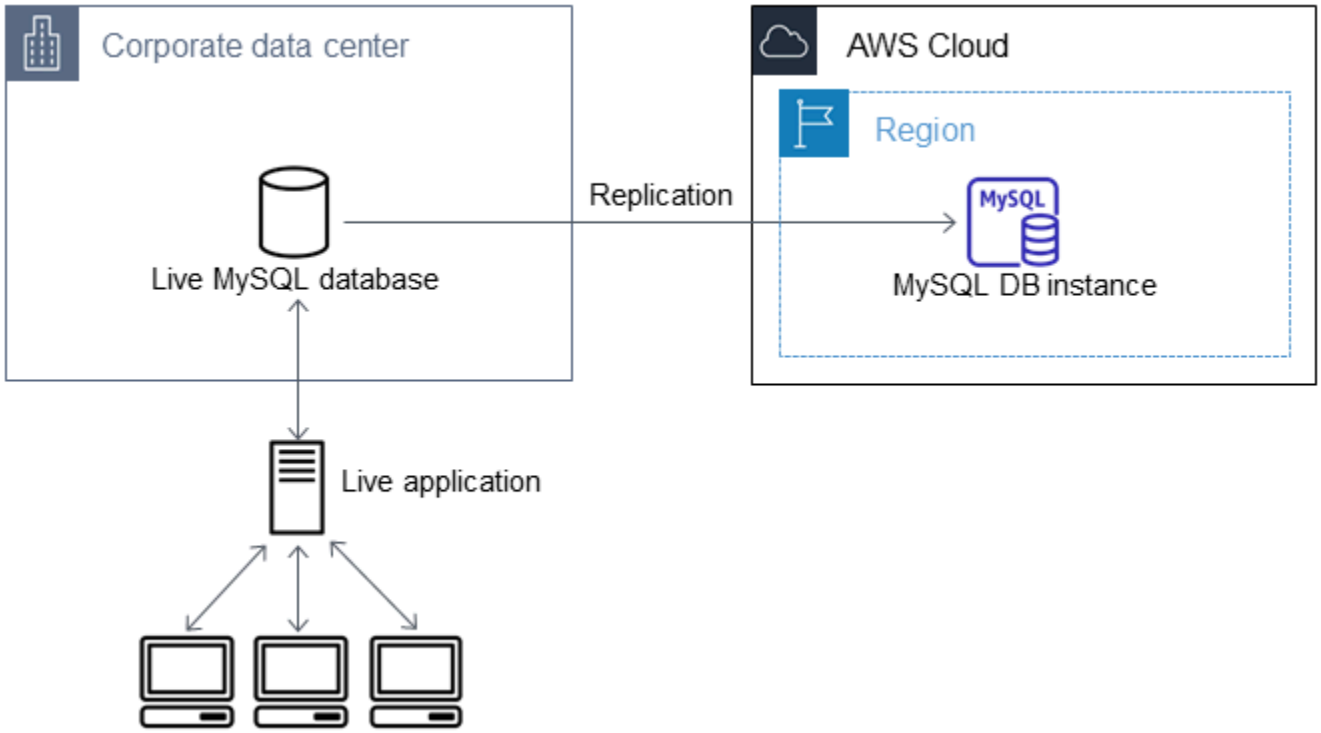
> **Note**
>
> Wenn Sie `mysqldump` beim ersten Speichern der Tabelle irgendwelche Datenformatierungsoptionen verwendet haben, stellen Sie sicher, dass Sie dieselben Optionen mit verwenden, um eine korrekte Interpretation des `LOAD DATA LOCAL INFILE` Inhalts der Datendatei sicherzustellen.

8. Führen Sie eine einfache `SELECT` Abfrage für eine oder zwei der Tabellen in der importierten Datenbank aus, um zu überprüfen, ob der Import erfolgreich war.

Wenn Sie die in diesem Verfahren verwendete EC2 Amazon-Instance nicht mehr benötigen, beenden Sie die EC2 Instance, um Ihren AWS Ressourcenverbrauch zu reduzieren. Informationen zum Beenden einer EC2 Instance finden Sie unter Kündigen einer Instance im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Aufgabe 4: Daten aus Ihrer externen Datenbank in Ihre neue Amazon RDS-Datenbank replizieren

Die Quelldatenbank wurde in der Zeit, in der die Daten in die MariaDB- oder MySQL-Datenbank kopiert und übertragen wurden, wahrscheinlich aktualisiert. Sie können also die Replikation verwenden, um die kopierte Datenbank up-to-date mit der Quelldatenbank zu verbinden.



Die zum Starten der Replikation auf einer Amazon RDS-Datenbank erforderlichen Berechtigungen sind eingeschränkt und stehen Ihrem Amazon RDS-Mastbenutzer nicht zur Verfügung. Verwenden Sie aus diesem Grund die entsprechende gespeicherte Amazon RDS-Prozedur für Ihre Haupt-Engine-Version:

-
-
- mysql.rds_set_external_master_gtidum die Replikation zu konfigurieren und mysql.rds_start_replication die Replikation zu starten

So starten Sie eine Replikation:

In Aufgabe 1 haben Sie bei der Festlegung der Replikationsoptionen die binäre Protokollierung aktiviert und eine eindeutige Server-ID für Ihre Quelldatenbank festgelegt. Jetzt können Sie Ihre Amazon-RDS-Datenbank als Replikat mit Ihrer Live-Datenbank als Quellreplikations-Instance einrichten.

1. Fügen Sie in der Amazon RDS-Konsole die IP-Adresse des Servers, der die Quelldatenbank hostet, zur VPC-Sicherheitsgruppe für die Amazon RDS-Datenbank hinzu. Weitere Informationen

zur Konfiguration einer VPC-Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln konfigurieren im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Möglicherweise müssen Sie auch Ihr lokales Netzwerk so konfigurieren, dass Verbindungen von der IP-Adresse Ihrer Amazon RDS-Datenbank zugelassen werden, damit diese mit Ihrer Quell-Instance kommunizieren kann. Verwenden Sie den `host` folgenden Befehl, um die IP-Adresse der Amazon RDS-Datenbank zu ermitteln:

```
host host_name
```

Das *`host_name`* ist zum Beispiel der DNS-Name vom Amazon RDS-Datenbank-Endpunkt`myinstance.123456789012.us-east-1.rds.amazonaws.com`. Sie finden den Endpunktwert in den DB-Instance-Details in der Amazon RDS-Konsole.

2. Verbinden Sie sich mithilfe eines Clients Ihrer Wahl mit der Quell-Instance und erstellen Sie einen Benutzer, der für die Replikation verwendet werden soll. Dieses Konto wird ausschließlich für die Replikation verwendet und muss auf Ihre Domäne beschränkt sein, um die Sicherheit zu erhöhen. Der folgende Befehl ist ein Beispiel:

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

3. Erteilen Sie für die Quell-Instance die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` für Ihren Replikationsbenutzer. Erteilen Sie beispielsweise die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` in allen Datenbanken für den `repl_user`-Benutzer für Ihre Domäne, mit dem folgenden Befehl:

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

4. Wenn Sie zum Erstellen Ihrer Sicherungsdatei das SQL-Format verwendet haben und die externe Instanz nicht MariaDB 10.0.24 oder höher ist, sehen Sie sich den Inhalt dieser Datei an, indem Sie den folgenden Befehl ausführen:

```
cat backup.sql
```

Die Datei beinhaltet einen `CHANGE MASTER TO`-Kommentar, der den Namen und die Position der Hauptprotokolldatei beinhaltet. Dieser Kommentar ist in der Sicherungsdatei enthalten, wenn Sie die Option `--master-data` mit `mysqldump` verwenden. Beachten Sie die Werte für `MASTER_LOG_FILE` und `MASTER_LOG_POS`.

```
--
-- Position to start replication or point-in-time recovery from
--

-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
```

Wenn Sie zum Erstellen Ihrer Sicherungsdatei ein Textformat mit Trennzeichen verwendet haben und die externe Instanz nicht MariaDB 10.0.24 oder höher ist, sollten Sie bereits über binäre Protokollkoordinaten aus Schritt 1 der Prozedur So erstellen Sie eine Sicherungskopie Ihrer vorhandenen Datenbank unter Aufgabe 1 verfügen.

Wenn es sich bei der externen Instanz um MariaDB 10.0.24 oder höher handelt, sollten Sie bereits über die GTID verfügen, von der aus Sie die Replikation starten können. Gehen Sie dazu aus Schritt 2 der Prozedur So erstellen Sie eine Sicherungskopie Ihrer vorhandenen Datenbank unter Aufgabe 1.

5. Konfigurieren Sie die Amazon-RDS-Datenbank als Replikat. Wenn die externe Instance nicht MariaDB 10.0.24 oder höher ist, stellen Sie als Masterbenutzer eine Verbindung zur Amazon RDS-Datenbank her und identifizieren Sie die Quelldatenbank als Quellreplikationsinstanz, indem Sie die entsprechende gespeicherte Amazon RDS-Prozedur verwenden:

   •

   •

   Wenn Sie über eine Sicherungsdatei im SQL-Format verfügen, verwenden Sie den Namen der Master-Log-Datei und die Master-Log-Position, die Sie in Schritt 4 festgelegt haben. Wenn Sie das Format mit Trennzeichen verwendet haben, verwenden Sie den Namen und die Position, die Sie bei der Erstellung der Sicherungsdateien festgelegt haben. Die folgenden Befehle sind Beispiele:

   MySQL 8.4 und höhere Versionen

```
CALL mysql.rds_set_external_source ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

MariaDB und MySQL 8.0 und niedrigere Versionen

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

Wenn die externe Instance MariaDB 10.0.24 oder höher ausführt, stellen Sie eine Verbindung mit der Amazon-RDS-Datenbank als Hauptbenutzer her und identifizieren Sie die Quelldatenbank mithilfe des Befehls mysql.rds_set_external_master_gtid als Quellreplikations-Instance. Verwenden Sie die GTID, die Sie in Schritt 2 der Prozedur So erstellen Sie eine Sicherungskopie Ihrer vorhandenen Datenbank unter Aufgabe 1 festgelegt haben. Der folgende Befehl ist ein Beispiel:

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 1);
```

Die *source_server_ip_address* ist die IP-Adresse der Quellreplikationsinstance. Eine EC2 private DNS-Adresse wird derzeit nicht unterstützt.

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

6. Führen Sie in der Amazon RDS-Datenbank den folgenden mysql.rds_start_replication Befehl aus, um die Replikation zu starten:

```
CALL mysql.rds_start_replication;
```

7. Führen Sie in der Amazon RDS-Datenbank den Befehl SHOW REPLICA STATUS aus, um zu ermitteln, wann das Replikat mit der Quellreplikationsinstanz auf dem neuesten Stand ist. Zu den Ergebnissen des `SHOW REPLICA STATUS`-Befehls gehört das Feld `Seconds_Behind_Master`. Wenn das `Seconds_Behind_Master` Feld 0 zurückgibt, entspricht das Replikat der Quellreplikationsinstanz.

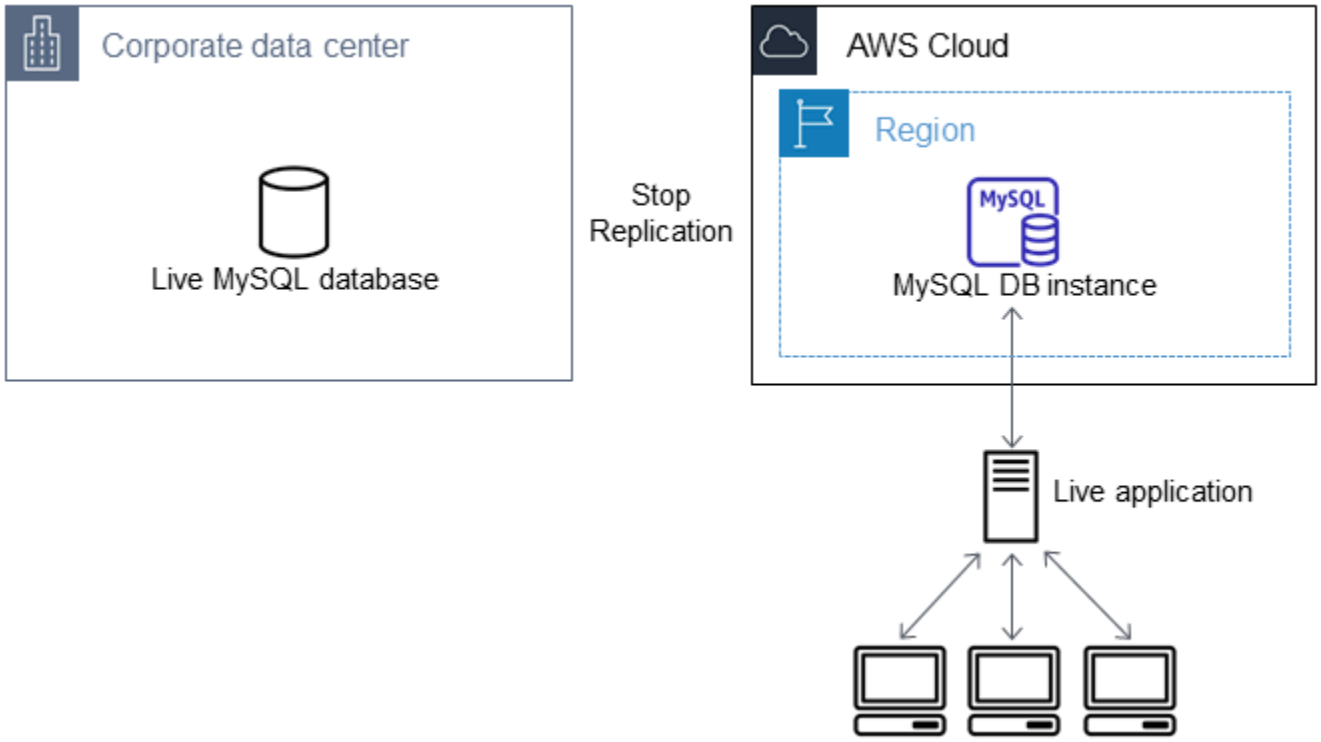
> **Note**
>
> Frühere Versionen von MySQL verwenden `SHOW SLAVE STATUS` anstelle von `SHOW REPLICA STATUS`. Wenn Sie vor 8.0.23 eine MySQL-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

Führen Sie für eine MariaDB 10.5-, 10.6- oder 10.11-DB-Instance die mysql.rds_replica_status gespeicherte Prozedur anstelle des MySQL-Befehls aus.

8. Sobald die Amazon RDS-Datenbank auf dem neuesten Stand ist, aktivieren Sie automatische Backups, damit Sie diese Datenbank bei Bedarf wiederherstellen können. Sie können automatische Backups für Ihre Amazon RDS-Datenbank mithilfe der Amazon RDS-Konsole aktivieren oder ändern. Weitere Informationen finden Sie unter Einführung in Backups.

Aufgabe 5: Leiten Sie Ihre Live-Anwendung auf Ihre Amazon RDS-Instance weiter

Nachdem die MariaDB- oder MySQL-Datenbank mit der Quellreplikationsinstanz auf dem neuesten Stand ist, können Sie jetzt Ihre Live-Anwendung aktualisieren, um die Amazon RDS-Instance zu verwenden.



So leiten Sie Ihre Live-Anwendung an die MariaDB- oder MySQL-Datenbank weiter und halten die Replikation an

1. Fügen Sie die IP-Adresse des Host-Servers der Anwendung hinzu, um die VPC-Sicherheitsgruppe für Ihre Amazon RDS-Datenbank hinzuzufügen. Weitere Informationen zum Ändern einer VPC-Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln konfigurieren im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
2. Stellen Sie sicher, dass das `Seconds_Behind_Master` Feld in den Ergebnissen des Befehls SHOW REPLICA STATUS den Wert 0 hat. Dies bedeutet, dass das Replikat mit der Quellreplikationsinstanz auf dem neuesten Stand ist.

   ```
   SHOW REPLICA STATUS;
   ```

   > **Note**
   >
   > Frühere Versionen von MySQL verwenden `SHOW SLAVE STATUS` anstelle von `SHOW REPLICA STATUS`. Wenn Sie vor 8.0.23 eine MySQL-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

   Bei einer DB-Instance von MariaDB 10.5, 10.6 oder 10.11 führen Sie das Verfahren mysql.rds_replica_status anstelle des MySQL-Befehls aus.
3. Schließen Sie alle Verbindungen zur Quelle, nachdem ihre Transaktionen abgeschlossen sind.
4. Aktualisieren Sie Ihre Anwendung, um die Amazon-RDS-Datenbank zu nutzen. Dieses Update ändert die Verbindungseinstellungen, um den Hostnamen und den Port der Amazon-RDS-Datenbank, das Benutzerkonto und Passwort für die Verbindung und die zu verwendende Datenbank zu bestimmen.
5. Stellen Sie eine Verbindung mit der DB-Instance her.

   Stellen Sie für einen Multi-AZ-DB-Cluster eine Verbindung mit der Writer-DB-Instance her.
6. Beenden Sie die Replikation für die Amazon RDS-Instance mit dem folgenden mysql.rds_stop_replication Befehl:

   ```
   CALL mysql.rds_stop_replication;
   ```

7. Führen Sie die entsprechende gespeicherte Amazon RDS-Prozedur in Ihrer Amazon RDS-Datenbank aus, um die Replikationskonfiguration zurückzusetzen, sodass diese Instance nicht mehr als Replikat identifiziert wird.

   •

   •

   MySQL 8.4 und höhere Versionen

   ```
   CALL mysql.rds_reset_external_source;
   ```

   MariaDB und MySQL 8.0 und niedrigere Versionen

   ```
   CALL mysql.rds_reset_external_master;
   ```

8. Aktivieren Sie zusätzliche Amazon-RDS-Funktionen, wie Multi-AZ-Unterstützung und Lesereplikate. Weitere Informationen erhalten Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS und Arbeiten mit DB-Instance-Lesereplikaten.

Daten aus einer beliebigen Quelle in eine Amazon RDS for MariaDB-DB-Instance importieren

Die folgenden Schritte führen Sie durch den Import von Daten in eine Amazon RDS-DB-Instance:

Schritt 1: Erstellen Sie flache Dateien, die die zu ladenden Daten enthalten

Verwenden Sie ein gängiges Format, z. B. kommagetrennte Werte (CSV), um die zu ladenden Daten zu speichern. Jede Tabelle muss eine eigene Datei haben — Sie können keine Daten für mehrere Tabellen in derselben Datei kombinieren. Geben Sie jeder Datei denselben Namen wie der zugehörigen Tabelle. Die Dateierweiterung können Sie benennen, wie Sie möchten. Wenn der Tabellenname beispielsweise lautet`sales`, könnte der Dateiname oder lauten`sales.csv`. `sales.txt`

Wenn möglich, ordnen Sie die Daten nach dem Primärschlüssel der Tabelle, die geladen wird. Dadurch werden die Ladezeiten drastisch verbessert und die Anforderungen an den Festplattenspeicher minimiert.

Die optimale Geschwindigkeit und Effizienz dieser Prozedur ist auf kleine Dateigrößen ausgelegt. Wenn die Größe einer einzelnen unkomprimierten Datei mehr als 1 GiB beträgt, teilen Sie diese in mehrere Dateien auf, die danach separat geladen werden können.

Verwenden Sie auf Unix-ähnlichen Systemen (einschließlich Linux) den `split`-Befehl. Der folgende Befehl teilt beispielsweise die Datei `sales.csv` in mehrere Dateien auf, die kleiner als 1 GiB sind. Die Teilung findet nur an Zeilenumbrüchen statt (-C 1024m). Die Namen der neuen Dateien enthalten aufsteigende numerische Suffixe. Der folgende Befehl erzeugt Dateien mit Namen wie und. `sales.part_00` `sales.part_01`

```
split -C 1024m -d sales.csv sales.part_
```

Ähnliche Hilfsprogramme sind auch für andere Betriebssysteme verfügbar.

Sie können die Flatfiles überall speichern. Wenn Sie die Daten in Schritt 5 laden, müssen Sie die `mysql` Shell jedoch von demselben Speicherort aufrufen, an dem sich die Dateien befinden, oder Sie müssen bei der Ausführung `LOAD DATA LOCAL INFILE` den absoluten Pfad für die Dateien verwenden.

Schritt 2: Verhindern Sie, dass alle Anwendungen auf die Ziel-DB-Instance zugreifen

Bevor Sie einen großen Ladevorgang starten, beenden Sie den Zugriff aller Anwendungsaktivitäten auf die Ziel-DB-Instance, in die Sie laden möchten. Wir empfehlen dies insbesondere, wenn andere Sitzungen die zu ladenden Tabellen oder die in diesen Tabellen referenzierten Tabellen verändern. Dadurch wird die Gefahr von Verstößen gegen Einschränkungen während des Ladens reduziert und zugleich die Ausführungsgeschwindigkeit des Ladevorgangs erhöht. Zudem wird es möglich, die DB-Instance im Zustand unmittelbar vor dem Ladevorgang wiederherzustellen, ohne die Änderungen durch Prozesse zu verlieren, die nicht am Ladevorgang beteiligt sind.

Unter Umständen ist dies jedoch nicht möglich oder nicht praktikabel. Wenn Sie vor dem Ladevorgang den Zugriff von Anwendungen auf die DB-Instance nicht stoppen können, führen Sie die erforderlichen Schritte aus, um die Verfügbarkeit und Integrität der Daten sicherzustellen. Die jeweiligen erforderlichen Schritte können sich stark unterscheiden, je nachdem welche besonderen Verwendungsfälle der Standortanforderungen vorliegen.

Schritt 3: Erstellen Sie einen DB-Snapshot

Wenn Sie Daten in eine neue DB-Instance laden wollen, die keine Daten enthält, können Sie diesen Schritt überspringen. Andernfalls empfehlen wir, vor und nach dem Laden der Daten DB-

Snapshots der Amazon RDS-DB-Zielinstanz zu erstellen. Amazon RDS-DB-Snapshots sind vollständige Backups Ihrer DB-Instance, mit denen Sie Ihre DB-Instance in einen bekannten Zustand zurückversetzen können. Wenn Sie ein DB-Snapshot-I/O initiieren, werden alle Operationen in Ihrer DB-Instance augenblicklich unterbrochen, während Ihre Datenbank gesichert wird.

Wenn Sie einen DB-Snapshot unmittelbar vor dem Laden erstellen, können Sie die Datenbank bei Bedarf in ihrem Zustand vor dem Laden wiederherstellen. Ein DB-Snapshot, der sofort nach einem Ladevorgang erstellt wird, bewahrt Sie davor, die Daten bei einem Fehler erneut laden zu müssen. Sie können DB-Snapshots auch nach dem Laden verwenden, um Daten in neue Datenbank-Instances zu importieren.

Im folgenden Beispiel wird der AWS CLI create-db-snapshotBefehl ausgeführt, um einen DB-Snapshot der `AcmeRDS` Instance zu erstellen und dem DB-Snapshot die ID `"preload"` zu geben.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Sie können auch die Wiederherstellung aus der DB-Snapshot-Funktionalität verwenden, um Test-DB-Instances für Testversuche zu erstellen, oder, um die Änderungen während eines Ladevorgangs rückgängig zu machen.

Bedenken Sie, dass die Wiederherstellung einer Datenbank aus einem DB-Snapshot eine neue DB-Instance erstellt, die wie alle DB-Instances über eine eindeutige Kennung und einen Endpunkt verfügt. Um die DB-Instance wiederherzustellen, ohne den Endpunkt zu ändern, löschen Sie zuerst die DB-Instance, damit Sie den Endpunkt wiederverwenden können.

Um beispielsweise eine DB-Instance für einen Testlauf oder andere Testzwecke zu erstellen, weisen Sie der DB-Instance eine eigene Kennung zu. Im Beispiel ist `AcmeRDS-2"` die Kennung. Das Beispiel stellt über den Endpunkt, der `AcmeRDS-2` zugeordnet ist, eine Verbindung mit der DB-Instance her. Weitere Informationen finden Sie unter restore-db-instance-from-db-snapshot.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Um einen bestehenden Endpunkt erneut zu verwenden, löschen Sie zuerst die DB-Instance und weisen Sie dann der wiederhergestellten Datenbank dieselbe Kennung zu. Weitere Informationen finden Sie unter delete-db-instance.

Im folgenden Beispiel wird auch ein letzter DB-Snapshot der DB-Instance erstellt, bevor sie gelöscht wird. Dies ist zwar optional, wird aber empfohlen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Schritt 4 (optional): Automatische Amazon RDS-Backups ausschalten

> **Warning**
>
> Schalten Sie automatische Backups nicht aus, wenn Sie eine point-in-time Wiederherstellung durchführen müssen.

Das Ausschalten automatisierter Backups dient der Leistungsoptimierung und ist für das Laden von Daten nicht erforderlich. Durch das Deaktivieren automatisierter Backups werden alle vorhandenen Backups gelöscht. Daher ist eine point-in-time Wiederherstellung nicht möglich, nachdem Sie automatische Backups deaktiviert haben. Beachten Sie, dass manuelle DB-Snapshots nicht von der Deaktivierung automatischer Backups betroffen sind. Alle bestehenden manuellen DB-Snapshots bleiben für eine Wiederherstellung verfügbar.

Das Deaktivieren von automatischen Backups reduziert die Ladezeit um 25 % und verringert zugleich den während des Ladevorgangs erforderlichen Speicherplatz. Wenn Sie Daten in eine neue DB-Instance laden wollen, die keine Daten enthält, ist das Deaktivieren von Backups eine einfache Möglichkeit, die Übertragungsgeschwindigkeit zu erhöhen und zusätzlichen Speicherverbrauch zu vermeiden. In einigen Fällen können Sie jedoch in eine DB-Instance laden, die bereits Daten enthält. Wenn ja, sollten Sie die Vorteile der Deaktivierung von Backups gegen die Auswirkungen eines Verlusts der Leistungsfähigkeit abwägen point-in-time-recovery.

In DB-Instances ist die Funktion für automatische Backups standardmäßig aktiviert (mit einem Aufbewahrungszeitraum von 1 Tag). Setzen Sie den Wert des Aufbewahrungszeitraums für Backups auf 0, um automatische Backups zu deaktivieren. Nach dem Ladevorgang können Sie Backups erneut aktivieren, indem Sie den Aufbewahrungszeitraum für Backups auf einen Nicht-Null-Wert setzen. Um Backups ein- oder auszuschalten, fährt Amazon RDS die DB-Instance herunter und startet sie dann neu, um die MariaDB- oder MySQL-Protokollierung ein- oder auszuschalten.

Führen Sie den AWS CLI `modify-db-instance` Befehl aus, um die Backup-Aufbewahrung auf Null zu setzen und die Änderung sofort zu übernehmen. Das Setzen des Aufbewahrungszeitraums für Backups auf Null erfordert den Neustart einer DB-Instance. Warten Sie daher bitte, bis der Neustart abgeschlossen wurde, bevor Sie fortfahren. Weitere Informationen finden Sie unter modify-db-instance.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
```

```
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Sie können den Status Ihrer DB-Instance mit dem AWS CLI describe-db-instancesBefehl überprüfen. Im folgenden Beispiel wird der DB-Instance-Status der AcmeRDS DB-Instance angezeigt:

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Wenn der DB-Instance-Status lautetavailable, können Sie mit dem nächsten Schritt fortfahren.

Schritt 5: Laden Sie die Daten

Verwenden Sie die LOAD DATA LOCAL INFILE MySQL-Anweisung, um Zeilen aus Ihren Flatfiles in die Datenbanktabellen zu lesen.

> **Note**
>
> Sie müssen die mysql Shell von demselben Ort aufrufen, an dem sich Ihre Flatfiles befinden, oder bei der Ausführung LOAD DATA LOCAL INFILE den absoluten Pfad für die Dateien verwenden.

Das folgende Beispiel zeigt, wie Daten aus einer Datei mit dem Namen sales.txt in eine Sales in der Datenbank benannte Tabelle geladen werden:

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Weitere Informationen zu der `LOAD DATA` Anweisung finden Sie unter LOAD DATA Statement in der MySQL-Dokumentation.

Schritt 6: Automatisierte Amazon RDS-Backups wieder aktivieren

Wenn Sie in Schritt 4 automatische Amazon RDS-Backups deaktiviert haben, schalten Sie nach Abschluss des Ladevorgangs automatische Backups ein, indem Sie die Aufbewahrungsfrist für Backups auf den Wert vor dem Laden zurücksetzen. Wie in Schritt 4 erwähnt, startet Amazon RDS die DB-Instance neu. Seien Sie also auf einen kurzen Ausfall vorbereitet.

Im folgenden Beispiel wird der AWS CLI modify-db-instanceBefehl ausgeführt, um automatische Backups für die `AcmeRDS` DB-Instance zu aktivieren und die Aufbewahrungsfrist auf einen Tag festzulegen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Arbeiten mit der MariaDB-Replikation in Amazon RDS

Normalerweise verwenden Sie Read Replicas, um die Replikation zwischen RDS Amazon-DB-Instances zu konfigurieren. Allgemeine Informationen zu Lesereplikaten finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten. Spezifische Informationen zur Arbeit mit Read Replicas auf Amazon RDS für MariaDB finden Sie unter. Arbeiten mit MariaDB Read Replicas

Sie können die Replikation auch basierend auf den Binärprotokollkoordinaten für eine MariaDB-Instance konfigurieren. Für MariaDB-Instances können Sie die Replikation auch auf der Grundlage der globalen Transaktion IDs (GTIDs) konfigurieren, was eine bessere Absturzsicherheit bietet. Weitere Informationen finden Sie unter Konfiguration der GTID basierten Replikation mit einer externen Quellinstanz.

Nachfolgend finden Sie weitere Replikationsoptionen, die mit RDS for MariaDB verfügbar sind:

- Sie können die Replikation zwischen einer RDS für MariaDB DB-Instance und einer My- SQL oder MariaDB-Instance einrichten, die sich außerhalb von Amazon befindet. RDS Informationen zum Konfigurieren der Replikation mit einer externen Quelle finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.
- Sie können die Replikation so konfigurieren, dass Datenbanken aus einer My SQL - oder MariaDB-Instance importiert werden, die sich außerhalb von Amazon befindetRDS, oder um Datenbanken in solche Instances zu exportieren. Weitere Informationen erhalten Sie unter Import von Daten in eine Amazon RDS for MariaDB-DB-Instance mit reduzierter Ausfallzeit und Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation.

Für jede dieser Replikationsoptionen können Sie die Replikation vom Typ „row-based“, „statement-based“ oder „mixed“ verwenden. Bei der zeilenbasierten Replikation werden nur die geänderten Zeilen repliziert, die sich aus einer Anweisung ergeben. SQL Bei der anweisungsbasierten Replikation wird die gesamte Anweisung repliziert. SQL Bei der gemischten Replikation wird, wenn möglich, die anweisungsbasierte Replikation verwendet, sie wechselt jedoch zur zeilenbasierten Replikation, wenn SQL Anweisungen ausgeführt werden, die für die anweisungsbasierte Replikation unsicher sind. In den meisten Fällen wird eine Replikation vom Typ „mixed“ empfohlen. Das binäre Protokollformat der DB-Instance bestimmt, ob die Replikation vom Typ „row-based“, „statement-based“ oder „mixed“ ist. Weitere Informationen zum Einrichten des binären Protokollformats finden Sie unter Konfiguration der MariaDB-Binärprotokollierung.

Informationen zur Replikationskompatibilität zwischen MariaDB-Versionen finden Sie unter Replikationskompatibilität in der MariaDB-Dokumentation.

Themen

- Arbeiten mit MariaDB Read Replicas
- Konfiguration der GTID basierten Replikation mit einer externen Quellinstanz
- Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Arbeiten mit MariaDB Read Replicas

Im Folgenden finden Sie spezifische Informationen zur Arbeit mit Read Replicas auf Amazon RDS for MariaDB. Allgemeine Informationen zu Lesereplikaten und Anleitungen zu ihrer Verwendung finden Sie in Arbeiten mit DB-Instance-Lesereplikaten.

- Konfigurieren von Replikationsfiltern mit MariaDB
- Konfigurieren der verzögerten Replikation mit MariaDB
- Löschen von Lesereplikaten bei MariaDB
- Arbeiten mit Bereitstellungen von Multi-AZ-Lesereplikaten mit MariaDB
- Verwendung von kaskadierenden Read Replicas mit für MariaDB RDS
- Überwachung von MariaDB Read Replicas
- Starten und Stoppen der Replikation mit MariaDB Read Replicas
- Fehlerbehebung bei Problemen mit einer MariaDB Read Replica

Konfigurieren von Lesereplikaten mit MariaDB

Bevor eine MariaDB-DB-Instance als Replikationsquelle eingesetzt werden kann, müssen Sie automatische Backups für die Quell-DB-Instance aktivieren, indem Sie den Aufbewahrungszeitraum für Backups auf einen anderen Wert als 0 festlegen. Diese Erfordernis gilt auch für ein Lesereplikat, das die Quell-DB-Instance für ein anderes Lesereplikat ist.

Sie können bis zu 15 Lesereplikate von einer DB-Instance innerhalb derselben Region erstellen. Damit die Replikation effektiv durchgeführt werden kann, sollte jedes Lesereplikat über dieselbe Menge an Rechen- und Speicherressourcen wie die Quell-DB-Instance verfügen. Wenn Sie die Quell-DB-Instance skalieren, skalieren Sie auch die Lesereplikate.

RDSfür MariaDB unterstützt kaskadierende Read Replicas. Informationen zum Konfigurieren von kaskadierenden Lesereplikaten finden Sie unter Verwendung von kaskadierenden Read Replicas mit für MariaDB RDS.

Sie können mehrere Erstellungs- und Löschaktionen für Lesereplikate gleichzeitig ausführen, die auf die gleiche Quell-DB-Instance verweisen. Halten Sie sich bei der Ausführung dieser Aktionen an die Grenze von 15 Lesereplikaten für jede Quell-Instance.

Konfigurieren von Replikationsfiltern mit MariaDB

Sie können Replikationsfilter verwenden, um anzugeben, welche Datenbanken und Tabellen mit einem Lesereplikat repliziert werden. Replikationsfilter können Datenbanken und Tabellen in die Replikation einbeziehen oder sie von der Replikation ausschließen.

Im Folgenden finden Sie einige Anwendungsfälle für Replikationsfilter:

- Reduzieren der Größe eines Lesereplikats. Mit Replikationsfiltern können Sie die Datenbanken und Tabellen ausschließen, die für das Lesereplikat nicht benötigt werden.
- Ausschließen von Datenbanken und Tabellen von Lesereplikaten aus Sicherheitsgründen.
- Replizieren verschiedener Datenbanken und Tabellen für spezifische Anwendungsfälle bei verschiedenen Lesereplikaten. Beispielsweise könnten Sie bestimmte Lesereplikate für Analysen oder Sharding verwenden.
- Für eine DB-Instance, die Read Replicas in verschiedenen AWS-Regionen, um verschiedene Datenbanken oder Tabellen in verschiedenen zu replizieren AWS-Regionen.

> **Note**
>
> Sie können Replikationsfilter verwenden, um anzugeben, welche Datenbanken und Tabellen mit einer primären MariaDB-Instance repliziert werden, die als Replikat in einer eingehenden Replikationstopologie konfiguriert ist. Weitere Informationen zu dieser Konfiguration finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

Themen

- Einstellung der Replikationsfilterparameter RDS für MariaDB
- Einschränkungen der Replikationsfilterung RDS für MariaDB
- Beispiele für Replikationsfilterung RDS für MariaDB
- Anzeigen der Replikationsfilter für ein Lesereplikat

Einstellung der Replikationsfilterparameter RDS für MariaDB

Um Replikationsfilter zu konfigurieren, legen Sie die folgenden Filterparameter für die Replikation fest:

- `replicate-do-db` – Repliziert Änderungen der angegebenen Datenbanken. Wenn Sie diesen Parameter für ein Lesereplikat festlegen, werden nur die im Parameter angegebenen Datenbanken repliziert.
- `replicate-ignore-db` – Repliziert keine Änderungen der angegebenen Datenbanken. Wenn der Parameter `replicate-do-db` für ein Lesereplikat festgelegt ist, wird dieser Parameter nicht ausgewertet.
- `replicate-do-table` – Repliziert Änderungen der angegebenen Tabellen. Wenn Sie diesen Parameter für ein Lesereplikat festlegen, werden nur die im Parameter angegebenen Tabellen repliziert. Wenn der Parameter `replicate-do-db` oder `replicate-ignore-db` festgelegt ist, muss die Datenbank, die die angegebenen Tabellen enthält, in die Replikation mit dem Lesereplikat einbezogen werden.
- `replicate-ignore-table` – Repliziert keine Änderungen der angegebenen Tabellen. Wenn der Parameter `replicate-do-table` für ein Lesereplikat festgelegt ist, wird dieser Parameter nicht ausgewertet.
- `replicate-wild-do-table` – Repliziert Tabellen basierend auf den angegebenen Namensmustern für Datenbanken und Tabellen. Die Platzhalterzeichen `%` und `_` werden unterstützt. Wenn der Parameter `replicate-do-db` oder `replicate-ignore-db` festgelegt ist, müssen Sie die Datenbank, welche die angegebenen Tabellen enthält, in die Replikation mit dem Lesereplikat einbeziehen.
- `replicate-wild-ignore-table` – Repliziert keine Tabellen basierend auf den angegebenen Namensmustern für Datenbanken und Tabellen. Die Platzhalterzeichen `%` und `_` werden unterstützt. Wenn die Parameter `replicate-do-table` oder `replicate-wild-do-table` für ein Lesereplikat festgelegt sind, wird dieser Parameter nicht ausgewertet.

Die Parameter werden in der angegebenen Reihenfolge ausgewertet. Weitere Informationen darüber, wie diese Parameter funktionieren, finden Sie in der Dokumentation für MariaDB.

Standardmäßig hat jeder dieser Parameter einen leeren Wert. Sie können diese Parameter auf jedem Lesereplikat verwenden, um Replikationsfilter festzulegen, zu ändern und zu löschen. Wenn Sie einen dieser Parameter festlegen, trennen Sie die einzelnen Filter durch ein Komma voneinander.

Sie können die Platzhalterzeichen % und _ in den Parametern `replicate-wild-do-table` und `replicate-wild-ignore-table` verwenden. Der Platzhalter % entspricht einer beliebigen Anzahl von Zeichen, und der Platzhalter _ entspricht nur einem Zeichen.

Das binäre Protokollierungsformat der Quell-DB-Instance ist wichtig für die Replikation, da es den Datensatz der Datenänderungen bestimmt. Die Einstellung des Parameters `binlog_format` bestimmt, ob die Replikation zeilenbasiert oder anweisungsbasiert ist. Weitere Informationen finden Sie unter Konfiguration der MariaDB-Binärprotokollierung.

> **Note**
>
> Alle Anweisungen der Data Definition Language (DDL) werden unabhängig von der `binlog_format` Einstellung in der Quell-DB-Instance als Anweisungen repliziert.

Einschränkungen der Replikationsfilterung RDS für MariaDB

Die folgenden Einschränkungen gelten für die Replikationsfilterung RDS für MariaDB:

- Jeder Filterparameter für die Replikation hat ein Limit von 2.000 Zeichen.
- Kommas werden in Replikationsfiltern nicht unterstützt.
- Die Optionen `binlog_do_db` und `binlog_ignore_db` von MariaDB für binäre Protokollfilter werden nicht unterstützt.
- Die Replikationsfilterung unterstützt keine XA-Transaktionen.

  Weitere Informationen finden Sie unter Einschränkungen für XA-Transaktionen in der Dokumentation MeineSQL.
- Die Replikationsfilterung wird RDS für MariaDB Version 10.2 nicht unterstützt.

Beispiele für Replikationsfilterung RDS für MariaDB

Um die Replikationsfilter für ein Lesereplikat zu konfigurieren, ändern Sie die Parameter der Replikationsfilter in der Parametergruppe, die dem Lesereplikat zugeordnet ist.

> **Note**
>
> Eine Standard-Parametergruppe kann nicht modifiziert werden. Erstellen Sie eine neue Parametergruppe und ordnen Sie diese der Lesereplika zu, wenn die Lesereplika eine

Standardparametergruppe verwendet. Weitere Informationen zu DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Sie können Parameter in einer Parametergruppe festlegen, indem Sie AWS Management Console, AWS CLI, oder RDSAPI. Weitere Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS. Wenn Sie Parameter in einer Parametergruppe festlegen, verwenden alle DB-Instances, die der Parametergruppe zugeordnet sind, diese Parametereinstellungen. Wenn Sie Parameter der Replikationsfilter in einer Parametergruppe festlegen, stellen Sie sicher, dass die Parametergruppe nur Lesereplikaten zugeordnet ist. Lassen Sie die Parameter der Replikationsfilter für Quell-DB-Instances leer.

In den folgenden Beispielen werden die Parameter mit dem festgelegt AWS CLI. Diese Beispiele sind `immediate` so eingestelltApplyMethod, dass die Parameteränderungen unmittelbar nach Abschluss des CLI Befehls vorgenommen werden. Wenn Sie möchten, dass eine ausstehende Änderung nach dem Neustart des Lesereplikats angewendet wird, legen Sie `ApplyMethod` auf `pending-reboot` fest.

In den folgenden Beispielen werden Replikationsfilter festgelegt:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Escaping wildcard characters in names
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Einschließen von Datenbanken in die Replikation

Das folgende Beispiel schließt die Datenbanken `mydb1` und `mydb2` in die Replikation ein. Wenn Sie `replicate-do-db` für ein Lesereplikat festlegen, werden nur die im Parameter angegebenen Datenbanken repliziert.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
```

```
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Example Einschließen von Tabellen in die Replikation

Das folgende Beispiel schließt die Tabellen `table1` und `table2` in der Datenbank `mydb1` in die Replikation ein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Einschließen von Tabellen in die Replikation mit Platzhalterzeichen

Das folgende Beispiel schließt Tabellen mit Namen, die mit `orders` und `returns` beginnen, in Datenbank `mydb` in die Replikation ein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
```

```
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Example Escape-Platzhalterzeichen in Namen

Das folgende Beispiel zeigt, wie Sie das Escapezeichen \ verwenden, um ein Platzhalterzeichen zu umgehen, das Teil eines Namens ist.

Angenommen, Sie haben mehrere Tabellennamen in der Datenbank `mydb1`, die mit `my_table` beginnen, und Sie möchten diese Tabellen in die Replikation einschließen. Die Tabellennamen enthalten einen Unterstrich, der auch ein Platzhalterzeichen darstellt, sodass in dem Beispiel Unterstriche in den Tabellennamen mit Escapezeichen versehen sind.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Example Ausschließen von Datenbanken von der Replikation

Das folgende Beispiel schließt die Datenbanken `mydb1` und `mydb2` von der Replikation aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Example Ausschließen von Tabellen von der Replikation

Das folgende Beispiel schließt die Tabellen `table1` und `table2` in der Datenbank `mydb1` von der Replikation aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Ausschließen von Tabellen von der Replikation mit Platzhalterzeichen

Das folgende Beispiel schließt Tabellen mit Namen, die mit `orders` und `returns` beginnen, in Datenbank `mydb` von der Replikation aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Anzeigen der Replikationsfilter für ein Lesereplikat

Sie können die Replikationsfilter für ein Lesereplikat wie folgt anzeigen:

- Überprüfen Sie die Einstellungen der Parameter der Replikationsfilter in der dem Lesereplikat zugeordneten Parametergruppe.

  Detaillierte Anweisungen finden Sie unter Parameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen.
- Stellen Sie in einem MariaDB-Client eine Verbindung zum Read-Replikat her und führen Sie die`SHOW REPLICA STATUS` Anweisung aus.

  In der Ausgabe werden in den folgenden Feldern die Replikationsfilter für das Lesereplikat angezeigt:

  - `Replicate_Do_DB`
  - `Replicate_Ignore_DB`
  - `Replicate_Do_Table`
  - `Replicate_Ignore_Table`
  - `Replicate_Wild_Do_Table`
  - `Replicate_Wild_Ignore_Table`

  Weitere Informationen zu diesen Feldern finden Sie in der SQL Dokumentation Meine Dokumentation unter Überprüfen des Replikationsstatus.

> **Note**
>
> Frühere Versionen von MariaDB wurden`SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie vor 10.5 eine MariaDB-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

Konfigurieren der verzögerten Replikation mit MariaDB

Sie können die verzögerte Replikation als Strategie für die Notfallwiederherstellung einsetzen. Für die verzögerte Replikation geben Sie die Mindestanzahl von Sekunden an, um welche die Replikation von der Quelle in das Lesereplikat verzögert werden soll. Wenn es zu einem Notfall kommt, weil beispielsweise eine Tabelle versehentlich gelöscht wurde, können Sie das Problem mit den folgenden Schritten schnell beheben:

- Beenden Sie die Replikation im Lesereplikat, bevor die Änderung, die den Notfall verursacht hat, an das Lesereplikat gesendet wird.

  Verwenden Sie die gespeicherte Prozedur mysql.rds_stop_replication, um die Replikation zu stoppen.
- Stufen Sie das Lesereplikat zur neuen Quell-DB-Instance hoch, indem Sie der Anleitung unter folge Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

> **Note**
>
> - Die verzögerte Replikation wird für MariaDB 10.6 und höher unterstützt.
> - Verwenden Sie gespeicherte Prozeduren, um die verzögerte Replikation zu konfigurieren. Sie können die verzögerte Replikation nicht mit dem konfigurieren AWS Management Console, der AWS CLI, oder der Amazon RDSAPI.
> - Sie können die Replikation auf der Grundlage globaler Transaktions-Identifikatoren (GTIDs) in einer Konfiguration mit verzögerter Replikation verwenden.

Themen

- Konfigurieren der verzögerten Replikation während der Erstellung von Read Replicas

- Ändern der verzögerten Replikation einer vorhandenen Read Replica
- Hochstufen eines Lesereplikats

Konfigurieren der verzögerten Replikation während der Erstellung von Read Replicas

Um die verzögerte Replikation für zukünftig aus einer DB-Instance erstellte Lesereplikate zu konfigurieren, führen Sie die gespeicherte Prozedur mysql.rds_set_configuration mit dem Parameter `target delay` aus.

Konfigurieren Sie die verzögerte Replikation während der Lesereplikat-Erstellung wie folgt:

1. Stellen Sie als Master-Benutzer mit einem MariaDB-Client eine Verbindung zu der MariaDB-DB-Instance her, die als Quelle für Lesereplikate verwendet werden soll.
2. Führen Sie die gespeicherte Prozedur mysql.rds_set_configuration mit dem Parameter `target delay` aus.

   Führen Sie beispielsweise die folgende gespeicherte Prozedur aus, um anzugeben, dass die Replikation um mindestens eine Stunde (3.600 Sekunden) für jedes Lesereplikat verzögert werden soll, das aus der aktuellen DB-Instance erstellt wird.

   ```
   call mysql.rds_set_configuration('target delay', 3600);
   ```

   > **Note**
   >
   > Nachdem Sie diese gespeicherte Prozedur ausgeführt haben, wird jede Read Replica, die Sie mit dem AWS CLI oder Amazon RDS API ist so konfiguriert, dass die Replikation um die angegebene Anzahl von Sekunden verzögert wird.

Ändern der verzögerten Replikation einer vorhandenen Read Replica

Führen Sie die gespeicherte Prozedur mysql.rds_set_source_delay aus, um die verzögerte Replikation eines vorhandenen Lesereplikats zu ändern.

Ändern Sie die verzögerte Replikation eines existierenden Lesereplikats wie folgt:

1. Verwenden Sie einen MariaDB-Client, um als Master-Benutzer eine Verbindung zum Lesereplikat herzustellen.

2. Verwenden Sie die gespeicherte Prozedur mysql.rds_stop_replication, um die Replikation zu stoppen.
3. Führen Sie die gespeicherte Prozedur mysql.rds_set_source_delay aus.

   Führen Sie beispielsweise die folgende gespeicherte Prozedur aus, um anzugeben, dass die Replikation für das Lesereplikat um mindestens eine Stunde (3600 Sekunden) verzögert werden soll.

   ```
   call mysql.rds_set_source_delay(3600);
   ```
4. Verwenden Sie die gespeicherte Prozedur mysql.rds_start_replication, um die Replikation zu starten.

Hochstufen eines Lesereplikats

Nachdem die Replikation gestoppt wurde, können Sie in einem Szenario der Notfallwiederherstellung ein Lesereplikat zur neuen Quell-DB-Instance hochstufen. Weitere Informationen zum Hochstufen eines Lesereplikats finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Löschen von Lesereplikaten bei MariaDB

Lesereplikate wurden zur Unterstützung von Leseabfragen entwickelt. Jedoch könnten gelegentliche Updates notwendig sein. Sie könnten beispielsweise einen Index hinzufügen, um die spezifischen Abfragetypen beim Zugriff auf das Replica zu beschleunigen. Sie können Updates aktivieren, indem Sie den Parameter `read_only` in der DB-Parametergruppe für das Lesereplikat auf 0 setzen.

Arbeiten mit Bereitstellungen von Multi-AZ-Lesereplikaten mit MariaDB

Sie können ein Lesereplikat aus Single-AZ- oder Multi-AZ-DB-Instance-Bereitstellungen erstellen. Sie können Multi-AZ-Bereitstellungen verwenden, um die Haltbarkeit und Verfügbarkeit kritischer Daten zu verbessern, aber Sie können Multi-AZ nicht zweitranging noch für schreibgeschützte Abfragen verwenden. Stattdessen können Sie Lesereplikate aus Multi-AZ-DB-Instances mit hohem Datenverkehr erstellen, um schreibgeschützte Abfragen auslagern zu können. Wenn die Quell-Instance einer Multi-AZ-Bereitstellung ein Failover zur sekundären Instance durchführt, werden alle zugehörigen Lesereplikate automatisch auf die Verwendung der sekundären (jetzt primären) Instance als Replikationsquelle umgeschaltet. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Sie können ein Lesereplikat als Multi-AZ-DB-Instance erstellen. Amazon RDS erstellt eine Standby-Version Ihres Replikats in einer anderen Availability Zone zur Failover-Unterstützung für das Replikat. Das Erstellen Ihres Lesereplikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist.

Verwendung von kaskadierenden Read Replicas mit für MariaDB RDS

RDSfür MariaDB unterstützt kaskadierende Read Replicas. Mit kaskadierenden Read Replicas können Sie Lesevorgänge skalieren, ohne Ihrer Quelle RDS für die MariaDB-DB-Instance zusätzlichen Overhead hinzuzufügen.

Bei kaskadierenden Read Replicas sendet Ihre RDS for MariaDB-DB-Instance Daten an die erste Read Replica in der Kette. Dieses Lesereplikat sendet dann Daten an das zweite Replikat in der Kette usw. Das Endergebnis ist, dass alle Read Replicas in der Kette die Änderungen aus der MariaDB-DB-Instance enthalten, jedoch ohne den Overhead ausschließlich RDS für die Quell-DB-Instance.

Sie können eine Reihe von bis zu drei Read Replicas in einer Kette aus einer Quelle RDS für die MariaDB-DB-Instance erstellen. Nehmen wir zum Beispiel an, Sie haben eine RDS für MariaDB DB-Instance,. `mariadb-main` Sie haben die folgenden Möglichkeiten:

- Beginnend mit `mariadb-main` erstellen Sie das erste Lesereplikat in der Kette `read-replica-1`.
- Als Nächstes erstellen Sie ab `read-replica-1` das nächste Lesereplikat in der Kette `read-replica-2`.
- Schließlich erstellen Sie ab `read-replica-2` das nächste Lesereplikat in der Kette `read-replica-3`.

Sie können kein weiteres Lesereplikat über dieses dritte kaskadierende Lesereplikat hinaus in der Reihe für `mariadb-main` erstellen. Eine vollständige Reihe von Instances von einer Quell-DB-Instance RDS für MariaDB bis zum Ende einer Reihe von kaskadierenden Read Replicas kann aus höchstens vier DB-Instances bestehen.

Damit kaskadierende Read Replicas funktionieren, müssen für jede Quelle RDS für die MariaDB-DB-Instance automatische Backups aktiviert sein. Erstellen Sie zuerst das Lesereplikat und ändern Sie es dann, um automatische Backups für das Lesereplikat für zu aktivieren. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats.

Wie bei jedem Lesereplikat können Sie ein Lesereplikat, das Teil einer Kaskade ist, hochstufen. Wenn Sie ein Lesereplikat aus einer Kette von Lesereplikaten hochstufen, wird dieses Replikat aus der Kette entfernt. Angenommen, Sie möchten einen Teil der Workload von Ihrer `mariadb-main`-DB-Instance zu einer neuen Instance verschieben, die nur von der Buchhaltung verwendet wird. Ausgehend von der Kette von drei Lesereplikaten aus dem Beispiel entscheiden Sie sich, `read-replica-2` hochzustufen. Die Kette ist wie folgt betroffen:

- Durch Hochstufen von `read-replica-2` wird es aus der Replikationskette entfernt.
  - Es ist jetzt eine vollständige DB-Instance mit Lese-/Schreibzugriff.
  - Die Replizierung auf `read-replica-3` wird fortgesetzt wie vor der Hochstufung.
- Ihre `mariadb-main` setzt die Replizierung auf `read-replica-1` fort.

Weitere Informationen über das Hochstufen von Lesereplikaten finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Überwachung von MariaDB Read Replicas

Für MariaDB-Read Replicas können Sie die Replikationsverzögerung in Amazon überwachen, CloudWatch indem Sie sich die Amazon-Metrik ansehen. RDS `ReplicaLag` Die Kennzahl `ReplicaLag` meldet den Wert des Feldes `Seconds_Behind_Master` des Befehls `SHOW REPLICA STATUS`.

> **Note**
>
> Frühere Versionen von MariaDB wurden`SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie vor 10.5 eine MariaDB-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

Häufige Ursachen für Replikationsverzögerung in MariaDB:

- Ein Netzwerkausfall.
- Schreibvorgänge auf Tabellen mit Indizes auf einem Lesereplikat. Wenn der Parameter `read_only` im Lesereplikat nicht auf 0 gesetzt ist, kann es die Replikation unterbrechen.
- Verwenden Sie eine nicht transaktionale Speicher-Engine wie My. ISAM Die Replikation wird nur für die InnoDB-Speicher-Engine für MariaDB unterstützt.

Wenn die Metrik `ReplicaLag` 0 erreicht, hat das Replica den Stand der Quell-DB-Instance erreicht. Wenn die Metrik `ReplicaLag` -1 zurückgibt, ist die Replikation aktuell nicht aktiv. `ReplicaLag` = -1 ist gleich `Seconds_Behind_Master` = `NULL`.

Starten und Stoppen der Replikation mit MariaDB Read Replicas

Sie können den Replikationsprozess auf einer RDS Amazon-DB-Instance beenden und neu starten, indem Sie die gespeicherten Systemprozeduren mysql.rds_stop_replication und aufrufenmysql.rds_start_replication. Sie können dies tun, wenn Sie zwischen zwei RDS Amazon-Instances für lang andauernde Operationen wie die Erstellung großer Indizes replizieren. Sie müssen Replikation auch anhalten und starten, wenn Sie Datenbanken importieren oder exportieren. Weitere Informationen erhalten Sie unter Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit und Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation.

Wenn die Replikation für mehr als 30 aufeinanderfolgende Tage unterbrochen wird, entweder manuell oder aufgrund eines Replikationsfehlers, RDS beendet Amazon die Replikation zwischen der Quell-DB-Instance und allen Read Replicas. um erhöhten Speicheranforderungen in der Quell-DB-Instance vorzubeugen und lange Failover-Zeiten zu vermeiden. Die Lesereplikat-DB-Instance ist weiterhin verfügbar. Die Replikation kann jedoch nicht fortgesetzt werden, da die vom Lesereplikat benötigten Binärprotokolle aus der Quell-DB-Instance nach Beendigung der Replikation gelöscht wurden. Sie können ein neues Lesereplikat für die Quell-DB-Instance erstellen, um die Replikation erneut aufzunehmen.

Fehlerbehebung bei Problemen mit einer MariaDB Read Replica

Die Replikationstechnologien für MariaDB arbeiten asynchron. Da sie asynchron sind, steigt gelegentlich `BinLogDiskUsage` in der Quell-DB-Instance an und `ReplicaLag` ist im Lesereplikat zu erwarten. Beispielsweise kann auf der Quell-DB-Instance eine große Anzahl von Schreibvorgängen gleichzeitig ausgeführt werden. Dagegen werden Schreibvorgänge zum Lesereplikat über einen einzigen I/O-Thread seriell abgearbeitet. Dies kann zu einer Verzögerung zwischen der Quell-DB-Instance und dem Lesereplikat führen. Weitere Informationen über schreibgeschützte Replikate in der MariaDB-Dokumentation finden Sie unter Replikation - Übersicht.

Sie können folgende Dinge tun, um die Verzögerungszeit zwischen Aktualisierungen einer Quell-DB-Instance und der nachfolgenden Aktualisierung des Lesereplikats zu reduzieren:

- Passen Sie die Speichergröße und die DB-Instance-Klasse eines Lesereplikats an die der Quell-DB-Instance an.

- Stellen Sie sicher, dass Parametereinstellungen in den DB-Parametergruppen, die von der Quell-DB-Instance und den Lesereplikaten verwendet werden, kompatibel sind. Weitere Informationen und ein Beispiel finden Sie in der Beschreibung des `max_allowed_packet`-Parameters weiter unten in diesem Abschnitt.

Amazon RDS überwacht den Replikationsstatus Ihrer Read Replicas und aktualisiert das `Replication State` Feld der Read Replica-Instance auf den `Error` Fall, dass die Replikation aus irgendeinem Grund beendet wird. Ein Beispiel könnte sein, wenn DML Abfragen, die auf Ihrer Read Replica ausgeführt werden, in Konflikt mit den Aktualisierungen stehen, die auf der Quell-DB-Instance vorgenommen wurden.

Sie können die Details des von der MariaDB-Engine zurückgegebenen Fehlers im Feld `Replication Error` überprüfen. Ereignisse, die den Status des Lesereplikats angeben, werden ebenfalls generiert, einschließlich RDS-EVENT-0045, RDS-EVENT-0046 und RDS-EVENT-0047. Weitere Informationen über Ereignisse und Abonnements zu Ereignissen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten. Wenn eine MariaDB-Fehlermeldung zurückgegeben wird, überprüfen Sie den Fehler in der MariaDB error message documentation.

Ein häufiges Problem, das Replikationsfehler verursachen kann, besteht, wenn der Wert für den `max_allowed_packet`-Parameter eines Lesereplikats niedriger ist als der Wert für den `max_allowed_packet`-Parameter der Quell-DB-Instance. Der `max_allowed_packet` Parameter ist ein benutzerdefinierter Parameter, den Sie in einer DB-Parametergruppe festlegen können. Er wird verwendet, um die maximale DML Codegröße anzugeben, die in der Datenbank ausgeführt werden kann. In einigen Fällen ist der Parameterwert `max_allowed_packet` in der mit einer Quell-DB-Instance verknüpften DB-Parametergruppe kleiner als der Parameterwert `max_allowed_packet` in der mit dem Lesereplikat der Quelle verknüpften DB-Parametergruppe. In diesen Fällen kann der Replikationsvorgang einen Fehler zurückgeben (Paket größer als "max_allowed_packet" Bytes) und die Replikation anhalten. Sie können den Fehler beheben, indem Sie die DB-Parametergruppe der Quelle und des Lesereplikats mit denselben Parameterwerten für `max_allowed_packet` verwenden.

Weitere Situationen, bei denen Replikationsfehler auftreten können, sind die Folgenden:

- Schreibvorgänge auf Tabellen in einem Lesereplikat. Wenn Sie Indizes für ein Lesereplikat erstellen, müssen Sie den `read_only`-Parameter auf 0 setzen, um die Indizes zu erstellen. Wenn Sie in dem Lesereplikat in Tabellen schreiben, kann die Replikation unterbrochen werden.

- Für die Verwendung einer nicht transaktionalen Speicher-Engine wie MyISAM. Read Replicas ist eine Transaktionsspeicher-Engine erforderlich. Die Replikation wird nur für die InnoDB-Speicher-Engine für MariaDB unterstützt.
- Verwenden von nicht-deterministischen Abfragen wie `SYSDATE()`. Weitere Informationen finden Sie unter Erkennen sicherer und nicht sicherer Anweisungen in der binären Protokollierung.

Wenn Sie entscheiden, dass Sie einen Fehler problemlos überspringen können, folgen Sie den Schritten in Überspringen des aktuellen Replikationsfehlers für RDS für My SQL. Andernfalls können Sie das Lesereplikat löschen und eine Instance mit derselben DB-Instance-Kennung erstellen, sodass der Endpunkt mit dem alten Lesereplikat übereinstimmt. Wird ein Replikationsfehler behoben, ändert sich das Feld `Replication State` zu Replicating.

Bei MariaDB-DB-Instances kann in manchen Fällen nicht auf sekundäre Lesereplikate umgestellt werden, wenn während des Fehlers nicht alle Binärprotokollereignisse (binlog) bereinigt wurden. In diesen Fällen müssen Sie die Lesereplikate manuell löschen und neu erstellen. Sie können das Risiko minimieren, indem Sie die folgenden Parameterwerte einstellen: `sync_binlog=1` und `innodb_flush_log_at_trx_commit=1`. Diese Einstellungen könnten die Leistung reduzieren, daher ist es ratsam, sie vor der Implementierung in einer Produktionsumgebung ausgiebig zu testen.

Konfiguration der GTID basierten Replikation mit einer externen Quellinstanz

Sie können die Replikation basierend auf globalen Transaktions-Identifikatoren (GTIDs) von einer externen MariaDB-Instance der Version 10.0.24 oder höher in eine RDS for MariaDB-DB-Instance einrichten. Folgen Sie diesen Richtlinien, wenn Sie eine externe Quell-Instance und ein Replikat bei Amazon RDS einrichten:

- Überwachen Sie Failover-Ereignisse für die RDS for MariaDB-DB-Instance, die Ihr Replikat ist. Tritt ein Failover auf, kann die DB-Instance, die Ihr Replikat ist, auf einem neuen Host mit einer anderen Netzwerkadresse wiederhergestellt werden. Weitere Informationen zum Überwachen von Failover-Ereignissen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.
- Behalten Sie die Binärprotokolle auf Ihrer Quell-Instance, bis sichergestellt ist, dass sie auf das Replikat angewendet wurden. Durch das Aufbewahren können Sie sicherstellen, dass Sie Ihre Quell-Instance im Fall eines Ausfalls wiederherstellen können.
- Aktivieren Sie automatische Backups auf Ihrer MariaDB-DB-Instance bei Amazon. RDS Das Einschalten automatischer Backups stellt sicher, dass Sie Ihr Replikat zu einem bestimmten Zeitpunkt wiederherstellen können, wenn Sie den Master und das Replikat neu synchronisieren

müssen. Informationen zu Backups und Point-In-Time Restore finden Sie unter. Daten sichern, wiederherstellen und exportieren

> **Note**
>
> Die zum Starten der Replikation auf einer MariaDB-DB-Instance erforderlichen Berechtigungen sind eingeschränkt und stehen Ihrem RDS Amazon-Master-Benutzer nicht zur Verfügung. Aus diesem Grund müssen Sie die Amazon RDS mysql.rds_set_external_master_gtid - und mysql.rds_start_replication -Befehle verwenden, um die Replikation zwischen Ihrer Live-Datenbank und Ihrer RDS für MariaDB-Datenbank einzurichten.

Gehen Sie wie folgt vor, um die Replikation zwischen einer externen Quell-Instance und einer MariaDB-DB-Instance auf Amazon RDS zu starten.

So starten Sie eine Replikation:

1. Legen Sie die Quell-MariaDB-Instance als schreibgeschützt fest:

   ```
   mysql> FLUSH TABLES WITH READ LOCK;
   mysql> SET GLOBAL read_only = ON;
   ```

2. Ruft den aktuellen Stand GTID der externen MariaDB-Instanz ab. Sie können dies mithilfe von `mysql` oder mithilfe des Query Editors Ihrer Wahl tun, um `SELECT @@gtid_current_pos;` auszuführen.

   Das GTID ist formatiert als. `<domain-id>-<server-id>-<sequence-id>` Ein typisches GTID sieht ungefähr so **`0-1234510749-1728`** aus. Weitere Informationen zu GTIDs und ihren Bestandteilen finden Sie unter Global Transaction ID in der MariaDB-Dokumentation.

3. Kopieren Sie die Datenbank mithilfe von von der externen MariaDB-Instance in die MariaDB-DB-Instanc `mysqldump`. Für große Datenbanken empfiehlt es sich, die Prozedur in zu verwende Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   mysqldump \
       --databases database_name \
       --single-transaction \
   ```

```
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump ^
    --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Zwischen der Option `-p` und dem eingegebenen Passwort darf kein Leerzeichen vorhanden sein.
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Verwenden Sie die Optionen `--host`, `--user (-u)`, `--port` und `-p` im Befehl `mysql`, um Hostnamen, Benutzernamen, Port und Passwort für die Verbindung mit Ihrer MariaDB-DB-Instance festzulegen. Der Hostname ist beispielsweise der DNS Name des MariaDB-DB-Instance-Endpunkts. `myinstance.123456789012.us-east-1.rds.amazonaws.com` Sie finden den Endpunktwert in den Instance-Details in der Amazon RDS Management Console.

4. Legen Sie die Quell-MariaDB-Instance als wieder beschreibbar fest.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

5. Fügen Sie in der Amazon RDS Management Console die IP-Adresse des Servers, der die externe MariaDB-Datenbank hostet, zur VPC Sicherheitsgruppe für die MariaDB-DB-Instance hinzu. Weitere Informationen zum Ändern einer VPC Sicherheitsgruppe finden Sie unter Sicherheitsgruppen für Sie VPC im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

   Die IP-Adresse kann sich ändern, wenn die folgenden Bedingungen erfüllt sind:

   - Sie verwenden eine öffentliche IP-Adresse für die Kommunikation zwischen der externen Quell-Instance und der DB-Instance.
   - Die externe Quell-Instance wurde gestoppt und neu gestartet.

   Wenn diese Bedingungen erfüllt sind, überprüfen Sie die IP-Adresse, bevor Sie sie hinzufügen.

   Es könnte sein, dass Sie Ihr lokales Netzwerk so konfigurieren müssen, dass es Verbindungen von der IP-Adresse Ihrer MariaDB-DB-Instance zulässt, damit es mit Ihrer externen MariaDB-Instance kommunizieren kann. Verwenden Sie den Befehl `host`, um die IP-Adresse Ihrer MariaDB-DB-Instance zu herauszufinden.

   ```
   host db_instance_endpoint
   ```

   Der Hostname ist der DNS Name des MariaDB-DB-Instance-Endpunkts.

6. Verbinden Sie sich mithilfe eines Clients Ihrer Wahl mit der externen MariaDB-Instance und erstellen Sie einen MariaDB-Benutzer, der für die Replikation verwendet werden soll. Dieses Konto wird ausschließlich für die Replikation verwendet und muss auf Ihre Domäne beschränkt sein, um die Sicherheit zu erhöhen. Im Folgenden wird ein Beispiel gezeigt.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   > **Note**
   >
   > Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

7. Erteilen Sie für die externe MariaDB-Instance die Berechtigungen `REPLICATION CLIENT` und `REPLICATION SLAVE` für Ihren Replikationsbenutzer. Erteilen Sie beispielsweise die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` in allen Datenbank für den '`repl_user`'-Benutzer für Ihre Domäne, mit dem folgenden Befehl.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Machen Sie die MariaDB-DB-Instance zum Replikat. Verbinden Sie sich als Masterbenutzer mit der MariaDB-DB-Instance und bestimmen Sie die externe MariaDB-Datenbank mithilfe des Befehls mysql.rds_set_external_master_gtid als Quell-Instance. Verwenden Sie denGTID, den Sie in Schritt 2 festgelegt haben. Im Folgenden wird ein Beispiel gezeigt.

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'GTID', 1);
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

9. Verwenden Sie auf der MariaDB-DB-Instance den Befehl mysql.rds_start_replication, um die Replikation zu starten:

```
CALL mysql.rds_start_replication;
```

Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Sie können die Replikation zwischen einer RDS for My SQL - oder MariaDB-DB-Instance und einer My- SQL oder MariaDB-Instance, die sich außerhalb von Amazon befindet, RDS mithilfe der binären Protokolldateireplikation einrichten.

Themen

- Bevor Sie beginnen
- Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Bevor Sie beginnen

Sie konfigurieren die Replikation mithilfe der Position der Binärprotokolldatei von replizierten Transaktionen.

Die zum Starten der Replikation auf einer Amazon RDS DB-Instance erforderlichen Berechtigungen sind eingeschränkt und stehen Ihrem RDS Amazon-Master-Benutzer nicht zur Verfügung. Stellen Sie daher sicher, dass Sie die mysql.rds_start_replication Befehle Amazon RDS oder, und verwenden, um die Replikation zwischen Ihrer Live-Datenbank und Ihrer RDS Amazon-Datenbank einzurichten.

Um das binäre Protokollierungsformat für eine My SQL - oder MariaDB-Datenbank festzulegen, aktualisieren Sie den `binlog_format` Parameter. Wenn Ihre DB-Instance die standardmäßige DB-Instance-Parametergruppe verwendet, erstellen Sie eine neue DB-Parametergruppe, um den `binlog_format` Parameter zu ändern. In MariaDB und My SQL 8.0 und niedrigeren Versionen ist `binlog_format` standardmäßig. `MIXED` Sie können `binlog_format` jedoch auch auf `ROW` oder `STATEMENT` einstellen, wenn Sie ein spezifisches Format des Binärprotokolls (binlog) benötigen. Starten Sie Ihre DB-Instance neu, damit die Änderungen übernommen werden. In My SQL 8.4 und höheren Versionen ist standardmäßig. `binlog_format` `ROW`

Informationen zum Einstellen des Parameters `binlog_format` erhalten Sie unter Konfiguration von RDS für MySQL-Binärprotokollierung für Single-AZ-Datenbanken. Informationen zu den Auswirkungen der verschiedenen Typen von Meine SQL Replikation finden Sie unter Vor- und Nachteile der anweisungsbasierten und zeilenbasierten Replikation in der Dokumentation zu My. SQL

Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Folgen Sie diesen Richtlinien, wenn Sie eine externe Quell-Instance und ein Replikat bei Amazon RDS einrichten:

- Überwachen Sie Failover-Ereignisse für die Amazon RDS DB-Instance, die Ihr Replikat ist. Tritt ein Failover auf, kann die DB-Instance, die Ihr Replikat ist, auf einem neuen Host mit einer anderen Netzwerkadresse wiederhergestellt werden. Weitere Informationen zum Überwachen von Failover-Ereignissen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.
- Bewahren Sie die Binärprotokolle auf Ihrer Quell-Instance auf, bis Sie sichergestellt haben, dass sie auf das Replikat angewendet wurden. Durch das Aufbewahren können Sie sicherstellen, dass Sie Ihre Quell-Instance im Fall eines Ausfalls wiederherstellen können.
- Aktivieren Sie automatische Backups auf Ihrer Amazon RDS DB-Instance. Das Einschalten automatischer Sicherungen stellt sicher, dass Sie Ihr Replikat zu einem bestimmten Zeitpunkt wiederherstellen können, wenn Sie die Quell-Instance und das Replikat neu synchronisieren müssen. Informationen zu point-in-time Backups und Wiederherstellungen finden Sie unterDaten sichern, wiederherstellen und exportieren.

Konfigurieren Sie die Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance wie folgt:

1. Machen Sie die My SQL - oder MariaDB-Quellinstanz schreibgeschützt.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Führen Sie den `SHOW MASTER STATUS` Befehl auf der My SQL - oder MariaDB-Quellinstanz aus, um den Binlog-Speicherort zu ermitteln.

   Sie erhalten eine Ausgabe, ähnlich der im folgenden Beispiel.

```
File                          Position
-----------------------------------
 mysql-bin-changelog.000031      107
-----------------------------------
```

3. Kopieren Sie die Datenbank von der externen Instance in die Amazon RDS DB-Instance mit`mysqldump`. Für große Datenbanken empfiehlt es sich, die Prozedur in zu verwende Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
```

```
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Zwischen der Option `-p` und dem eingegebenen Passwort darf kein Leerzeichen vorhanden sein.

Um den Hostnamen, den Benutzernamen, den Port und das Passwort für die Verbindung mit Ihrer RDS Amazon-DB-Instance anzugeben `--host--user (-u)`, verwenden Sie die `-p` Optionen, `--port` und im `mysql` Befehl. Der Hostname ist beispielsweise der Name des Domain Name Service (DNS) vom RDS Amazon-DB-Instance-Endpunkt`myinstance.123456789012.us-east-1.rds.amazonaws.com`. Sie finden den Endpunktwert in den Instance-Details in der AWS Management Console.

4. Machen Sie die My SQL - oder MariaDB-Quellinstanz wieder beschreibbar.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Weitere Informationen zum Erstellen von Backups zur Verwendung bei der Replikation finden Sie in der Dokumentation Meine. SQL

5. Fügen Sie im die IP-Adresse des Servers AWS Management Console, der die externe Datenbank hostet, zur Sicherheitsgruppe Virtual Private Cloud (VPC) für die RDS Amazon-DB-Instance hinzu. Weitere Informationen zum Ändern einer VPC Sicherheitsgruppe finden Sie unter Sicherheitsgruppen für Sie VPC im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

   Die IP-Adresse kann sich ändern, wenn die folgenden Bedingungen erfüllt sind:

   - Sie verwenden eine öffentliche IP-Adresse für die Kommunikation zwischen der externen Quell-Instance und der DB-Instance.
   - Die externe Quell-Instance wurde gestoppt und neu gestartet.

Wenn diese Bedingungen erfüllt sind, überprüfen Sie die IP-Adresse, bevor Sie sie hinzufügen.

Möglicherweise müssen Sie auch Ihr lokales Netzwerk so konfigurieren, dass Verbindungen von der IP-Adresse Ihrer RDS Amazon-DB-Instance aus zugelassen werden. Sie tun dies, damit Ihr lokales Netzwerk mit Ihrer externen My SQL - oder MariaDB-Instance kommunizieren kann. Verwenden Sie den `host` Befehl, um die IP-Adresse der RDS Amazon-DB-Instance zu ermitteln.

```
host db_instance_endpoint
```

Der Hostname ist der DNS Name des RDS Amazon-DB-Instance-Endpunkts.

6. Stellen Sie mit einem Client Ihrer Wahl eine Verbindung zur Quell-Instance her und erstellen Sie den für die Replikation zu verwendenden Benutzer. Verwenden Sie dieses Konto ausschließlich für die Replikation und beschränken Sie es auf Ihre Domäne, um die Sicherheit zu erhöhen. Im Folgenden wird ein Beispiel gezeigt.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

7. Erteilen Sie für die externe Instance die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` für Ihren Replikationsbenutzer. Erteilen Sie beispielsweise die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` in allen Datenbank für den '`repl_user`'-Benutzer für Ihre Domäne, mit dem folgenden Befehl.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Machen Sie die Amazon RDS DB-Instance zum Replikat. Stellen Sie dazu zunächst als Master-Benutzer eine Verbindung zur Amazon RDS DB-Instance her. Identifizieren Sie dann die externe My SQL - oder MariaDB-Datenbank als Quellinstanz, indem Sie den Befehl or verwenden. Verwenden Sie den Namen und die Position der Protokolldatei, die Sie in Schritt 2 festgelegt haben. Die folgenden Befehle sind Beispiele.

   Mein SQL 8.4

```
CALL mysql.rds_set_external_source ('mysourceserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

MariaDB und My SQL 8.0 und 5.7

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> RDSBei My können Sie wählenSQL, ob Sie die verzögerte Replikation verwenden möchten, indem Sie stattdessen die mysql.rds_set_external_source_with_delay (für meine Hauptversionen 8.4 und höher) RDS SQL oder die mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL gespeicherte Prozedur ausführen. Ein Grund RDS für SQL die Verwendung der verzögerten Replikation ist die Aktivierung der Notfallwiederherstellung mit der gespeicherten Prozedur. Derzeit RDS unterstützt MariaDB verzögerte Replikation, unterstützt das Verfahren jedoch nicht. `mysql.rds_start_replication_until`

9. Geben Sie auf der Amazon RDS DB-Instance den mysql.rds_start_replication Befehl ein, um die Replikation zu starten.

```
CALL mysql.rds_start_replication;
```

Optionen für MariaDB-Datenbank-Engine

Im Folgenden finden Sie eine Beschreibung der Optionen oder zusätzlichen Funktionen, die für Amazon-RDS-Instances verfügbar sind, auf denen die MariaDB-DB-Engine ausgeführt wird. Zum Aktivieren dieser Optionen können Sie diese einer benutzerdefinierten Optionsgruppe hinzufügen und anschließend der Optionsgruppe für Ihre DB-Instance zuordnen. Weitere Informationen über das Arbeiten mit Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Amazon RDS unterstützt die folgenden Optionen für MariaDB:

Options-ID	Engine-Versionen
`MARIADB_AUDIT_PLUGIN`	MariaDB 10.3 und höher

MariaDB Audit-Plugin-Support

Amazon RDS unterstützt die Verwendung des MariaDB Steuerungs-Plugins in MariaDB-Datenbank-Instances. Das MariaDB-Audit-Plugin zeichnet Datenbank-Aktivitäten auf, z. B. Benutzer, die sich bei der Datenbank anmelden, Abfragen in der Datenbank ausführen und vieles mehr. Der Datensatz der Datenbankaktivität wird in einer Protokolldatei gespeichert.

Audit-Plugin-Optionseinstellungen

Amazon RDS unterstützt die folgenden Einstellungen für die MariaDB-Audit-Plugin-Option.

> **Note**
>
> Wenn Sie in der RDS-Konsole keine Optionseinstellung konfigurieren, verwendet RDS die Standardeinstellung.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	Der Speicherort der Protokolldatei. Die Protokolldatei beinhaltet den Datensatz der Aktivitäten die in festgelegt wurde `SERVER_AU`

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
			`DIT_EVENTS` . Weitere Informationen erhalten Sie unter Anzeigen und Auflisten von Datenbank-Protokolldateien und MariaDB-Datenbank-Protokolldateien.
`SERVER_AUDIT_FILE_ROTATE_SIZE`	1 – 1000000000	1000000	Die Größe in Bytes, die bei Erreichen der Datei dazu führt, dass die Datei rotiert. Weitere Informationen finden Sie unter Protokollrotation und Aufbewahrung für MariaDB.
`SERVER_AUDIT_FILE_ROTATIONS`	0 – 100	9	Die Anzahl der zu speichernden Protokollrotationen, wenn `server_audit_output_type=file` . Wenn der Wert auf 0 festgelegt ist, rotiert die Protokolldatei niemals. Weitere Informationen erhalten Sie unter Protokollrotation und Aufbewahrung für MariaDB und Herunterladen einer Datenbank-Protokolldatei.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_EVENTS	CONNECT, QUERY, TABLE, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	Die Arten von Aktivitäten, die im Protokoll aufgezeichnet werden sollen. Die Installation des MariaDB Audit Plugins ist selbst protokolliert. • CONNECT: Protokollieren Sie erfolgreiche und nicht erfolgreiche Verbindungen zur Datenbank und trennen Sie die Verbindung zur Datenbank. • QUERY: Protokollieren Sie den Text aller Abfragen, die für die Datenbank ausgeführt werden. • TABLE: Protokolltabellen, die von Abfragen betroffen sind, wenn die Abfragen für die Datenbank ausgeführt werden. • QUERY_DDL : Ähnlich dem QUERY-Ereignis, aber gibt nur Data Definition Language (DDL)-Abfragen zurück (CREATE, ALTER usw.). • QUERY_DML : Ähnlich dem QUERY-Ereignis, aber gibt nur Data Manipulation Language (DML)-Abfragen zurück (INSERT, UPDATE usw. und auch SELECT). • QUERY_DML_NO_SELECT : : Ähnlich dem QUERY_DML -Ereignis, protokolliert jedoch keine SELECT-Abfragen. • QUERY_DCL : Ähnlich dem QUERY-Ereignis, aber gibt nur Data Control Language (DCL)-Abfragen zurück (GRANT, REVOKE usw.).

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_INCL_USERS	Mehrere kommaseparierte Werte	Keine	Füge nur Aktivität von den angegebenen Benutzern ein. Standardmäßig werden Aktivitäten für alle Benutzer aufgezeichnet. SERVER_AUDIT_INCL_USERS und schließen SERVER_AUDIT_EXCL_USERS sich gegenseitig aus. Wenn Sie Werte hinzufügen SERVER_AUDIT_INCL_USERS , stellen Sie sicher, dass keine Werte hinzugefügt werden SERVER_AUDIT_EXCL_USERS .

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_EXCL_USERS	Mehrere kommaseparierte Werte	Keine	Aktivität von den angegebenen Benutzern ausschließen. Standardmäßig werden Aktivitäten für alle Benutzer aufgezeichnet. SERVER_AUDIT_INCL_USERS und schließen SERVER_AUDIT_EXCL_USERS sich gegenseitig aus. Wenn Sie Werte hinzufügen SERVER_AUDIT_EXCL_USERS , stellen Sie sicher, dass keine Werte hinzugefügt werden SERVER_AUDIT_INCL_USERS . Der rdsadmin-Benutzer fragt die Datenbank jede Sekunde ab, um den Zustand der Datenbank zu überprüfen. Abhängig von Ihren anderen Einstellungen kann diese Aktivität dazu führen, dass die Größe Ihrer Protokolldatei sehr schnell sehr groß wird. Wenn Sie diese Aktivität nicht aufzeichnen müssen, fügen Sie den Benutzer rdsadmin zur Liste SERVER_AUDIT_EXCL_USERS hinzu. **Note** CONNECTDie Aktivität wird stets für alle Benutzer erfasst, auch wenn ein Benutzer für diese Optionseinstellung angegeben ist.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_LOGGING	ON	ON	Die Protokollierung ist aktiv. Der einzige gültige Wert ist ON. Amazon RDS unterstützt die Deaktivierung der Protokollierung nicht. Wenn Sie die Protokollierung deaktivieren möchten, entfernen Sie das MariaDB-Audit-Plugin. Weitere Informationen finden Sie unter Entfernen des MariaDB-Audit-Plugins.
SERVER_AUDIT_QUERY_LOG_LIMIT	0 – 2147483647	1024	Die Längenbegrenzung der Abfragezeichenfolge in einem Datensatz.

Hinzufüge des MariaDB-Audit-Plugins

Der allgemeine Vorgang zum Hinzufügen des MariaDB-Audit-Plugins zu einer DB-Instance ist wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie das MariaDB-Audit-Plugin hinzugefügt haben, müssen Sie Ihre DB-Instance nicht neu starten. Sobald die Optionsgruppe aktiv ist, beginnt sofort die Überwachung.

So fügen Sie das MariaDB-Audit-Plugin hinzu

1. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe. Wählen Sie zuerst mariadb für Engine und dann 10.3 oder höher für Major engine version (Engine-Hauptversion) aus. Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.

2. Fügen Sie der Optionsgruppe die Option MARIADB_AUDIT_PLUGIN hinzu und konfigurieren Sie die Optionseinstellungen. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Audit-Plugin-Optionseinstellungen.
3. Wenden Sie die Optionsgruppe auf eine neue oder vorhandene DB-Instance an.

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die DB-Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Anzeigen und Herunterladen des MariaDB-Audit-Plugin-Protokolls

Nachdem Sie das MariaDB Audit-Plugin aktiviert haben, greifen Sie auf die Ergebnisse in den Protokolldateien genauso zu, wie auf andere textbasierte Protokolldateien. Die Auditprotokolldateien werden unter gespeicher `/rdsdbdata/log/audit/`. Weitere Informationen zum Anzeigen einer Protokolldatei in der Konsole finden Sie unter Anzeigen und Auflisten von Datenbank-Protokolldateien. Weitere Informationen zum Herunterladen der Protokolldatei finden Sie unter Herunterladen einer Datenbank-Protokolldatei.

Ändern der Einstellungen für das MariaDB-Audit-Plugin

Nachdem Sie das MariaDB-Audit-Plugin aktiviert haben, können Sie die Einstellungen für das Plugin ändern. Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Audit-Plugin-Optionseinstellungen.

Entfernen des MariaDB-Audit-Plugins

Amazon RDS unterstützt das Deaktivieren der Protokollierung im MariaDB-Audit-Plugin nicht. Sie können das Plugin jedoch aus einer DB-Instance entfernen. Wenn Sie das MariaDB-Audit-Plugin entfernen, wird die DB-Instance automatisch erneut gestartet, um die Überwachung zu beenden.

Führen Sie einen der folgenden Schritte aus, um das MariaDB-Audit-Plugin aus einer DB-Instance zu entfernen:

- Entfernen Sie das MariaDB-Audit-Plugin aus ihrer zugehörigen Optionsgruppe wie folgt: Diese Änderung wirkt sich auf alle DB-Instances aus, die die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe
- Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, die im Plugin nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Parameter für MariaDB

Standardmäßig verwendet eine MariaDB DB-Instance eine für eine MariaDB-Datenbank spezifische DB-Parametergruppe. Diese Parametergruppe enthält einige, aber nicht alle in Amazon RDS-DB-Parametergruppen für die MySQL-Datenbank-Engine enthaltenen Parameter. Sie enthält auch eine Reihe neuer, MariaDB-spezifischer Parameter. Weitere Informationen zum Arbeiten mit Parametergruppen und zum Festlegen von Parametern finden Sie unter Parametergruppen für Amazon RDS.

MariaDB-Parameter anzeigen

RDS for MariaDB-Parameter werden auf die Standardwerte der von Ihnen ausgewählten Speicher-Engine gesetzt. Weitere Informationen zu MariaDB-Parametern finden Sie in der MariaDB-Dokumentation. Weitere Informationen über MariaDB-Speicher-Engines finden Sie unter Unterstützte Speicher-Engines für MariaDB bei Amazon RDS.

Sie können die für eine bestimmte RDS-for-MariaDB-Version verfügbaren Parameter mithilfe der RDS-Konsole oder des AWS CLI anzeigen. Weitere Informationen zum Anzeigen von Parametern in einer MariaDB-Parametergruppe in der RDS-Konsole finden Sie unter Parameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen.

Mit dem AWS CLI können Sie die Parameter für eine Version von RDS for MariaDB anzeigen, indem Sie den Befehl `describe-engine-default-parameters` ausführen. Geben Sie einen der folgenden Werte für die Option `--db-parameter-group-family` an:

- `mariadb10.11`
- `mariadb10.6`
- `mariadb10.5`
- `mariadb10.4`
- `mariadb10.3`

Um beispielsweise die Parameter für RDS for MariaDB Version 10.6 anzuzeigen, führen Sie den folgenden Befehl aus.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6
```

Die Ausgabe sieht folgendermaßen oder ähnlich aus.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "alter_algorithm",
                "Description": "Specify the alter table algorithm.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "string",
                "AllowedValues": "DEFAULT,COPY,INPLACE,NOCOPY,INSTANT",
                "IsModifiable": true
            },
            {
                "ParameterName": "analyze_sample_percentage",
                "Description": "Percentage of rows from the table ANALYZE TABLE will
 sample to collect table statistics.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "float",
                "AllowedValues": "0-100",
                "IsModifiable": true
            },
            {
                "ParameterName": "aria_block_size",
                "Description": "Block size to be used for Aria index pages.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1024-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "aria_checkpoint_interval",
                "Description": "Interval in seconds between automatic checkpoints.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "0-4294967295",
                "IsModifiable": true
            },
        ...
```

Führen Sie den folgenden Befehl aus, um nur die änderbaren Parameter für RDS for MariaDB Version 10.6 aufzulisten.

Für Linux, macOSoder Unix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

MySQL-Parameter, die nicht verfügbar sind

Die folgenden MySQL-Parameter sind in MariaDB-spezifischen DB-Parametergruppen nicht verfügbar:

- bind_address
- binlog_error_action
- binlog_gtid_simple_recovery
- binlog_max_flush_queue_time
- binlog_order_commits
- binlog_row_image
- binlog_rows_query_log_events
- binlogging_impossible_mode
- block_encryption_mode
- core_file
- default_tmp_storage_engine
- div_precision_increment
- end_markers_in_json
- enforce_gtid_consistency
- eq_range_index_dive_limit
- explicit_defaults_for_timestamp
- gtid_executed

- gtid-mode
- gtid_next
- gtid_owned
- gtid_purged
- log_bin_basename
- log_bin_index
- log_bin_use_v1_row_events
- log_slow_admin_statements
- log_slow_slave_statements
- log_throttle_queries_not_using_indexes
- master-info-repository
- optimizer_trace
- optimizer_trace_features
- optimizer_trace_limit
- optimizer_trace_max_mem_size
- optimizer_trace_offset
- relay_log_info_repository
- rpl_stop_slave_timeout
- slave_parallel_workers
- slave_pending_jobs_size_max
- slave_rows_search_algorithms
- storage_engine
- table_open_cache_instances
- timed_mutexes
- transaction_allow_batching
- validate-password
- validate_password_dictionary_file
- validate_password_length
- validate_password_mixed_case_count
- validate_password_number_count

- validate_password_policy
- validate_password_special_char_count

Weitere Informationen zu MySQL-Parametern finden Sie in der MySQL-Dokumentation.

Daten von einem My SQL DB-Snapshot zu einer MariaDB-DB-Instance migrieren

Sie können einen RDS for My SQL DB-Snapshot auf eine neue DB-Instance migrieren, auf der MariaDB ausgeführt wird, indem Sie The AWS Management Console AWS CLI, The oder Amazon verwenden. RDS API Sie müssen einen DB-Snapshot verwenden, der von einer RDS Amazon-DB-Instance erstellt wurde, auf der My SQL 5.6 oder 5.7 ausgeführt wird. Informationen zum Erstellen eines Snapshots RDS für My SQL DB finden Sie unterErstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Die Migration des Snapshots wirkt sich nicht auf die ursprüngliche DB-Instance aus, von der der Snapshot entnommen wurde. Sie können die neue DB-Instance testen und validieren, bevor Sie den Datenverkehr als Ersatz für die ursprüngliche DB-Instance darauf umleiten.

Nach der Migration von My SQL zu MariaDB wird die MariaDB-DB-Instance der Standard-DB-Parametergruppe und Optionsgruppe zugeordnet. Nach dem Wiederherstellen des DB-Snapshots können Sie eine benutzerdefinierte DB-Parametergruppe mit der neuen DB-Instance zuordnen. Eine MariaDB-Parametergruppe hat jedoch einen anderen Satz konfigurierbarer Systemvariablen. Informationen zu den Unterschieden zwischen den Systemvariablen My SQL und MariaDB finden Sie unter Systemvariablen-Unterschiede zwischen MariaDB und My. SQL Informationen über DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS. Informationen über Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Durchführen der Migration

Sie können einen RDS for My SQL DB-Snapshot mit dem, dem oder dem auf eine neue MariaDB-DB-Instance migrieren. AWS Management Console AWS CLI RDS API

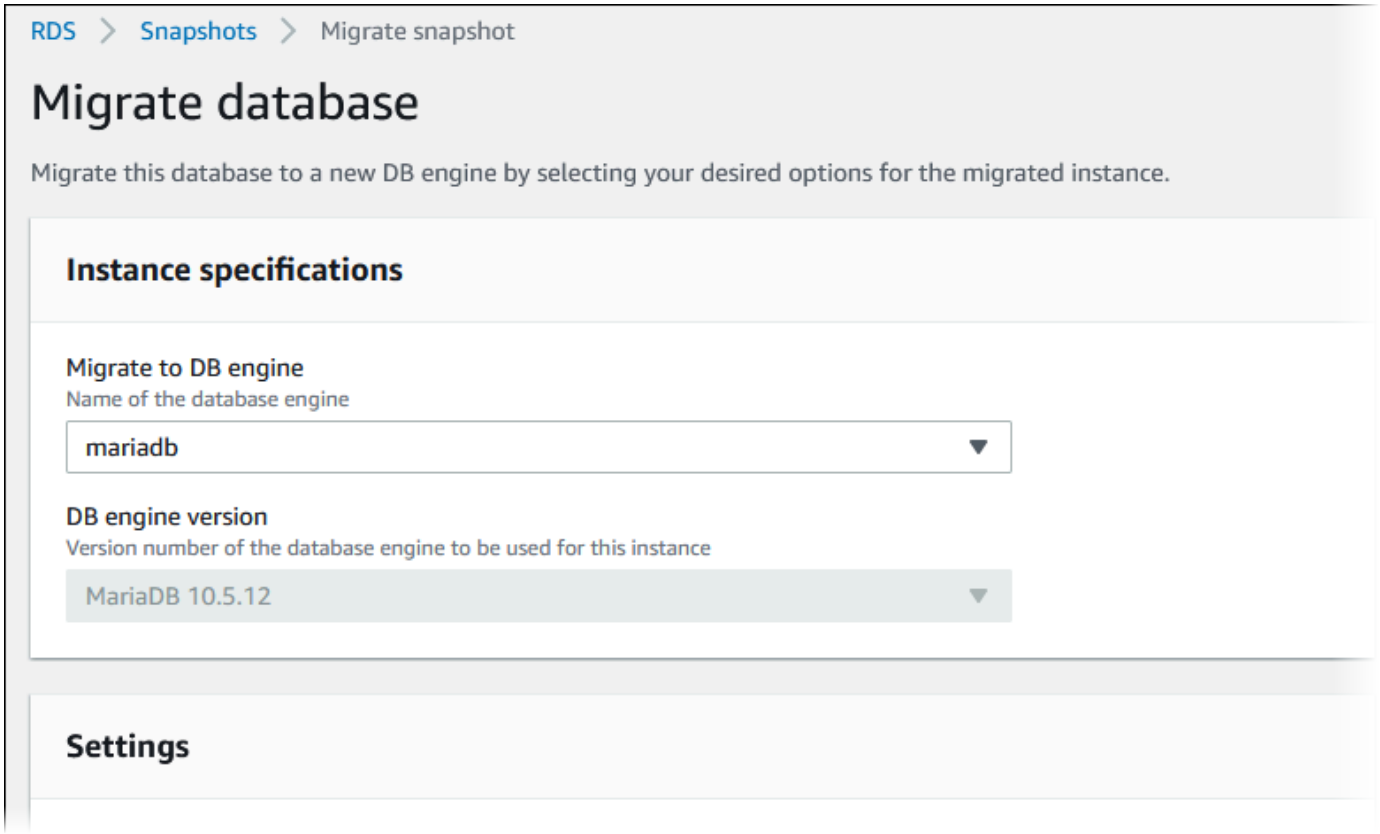
Konsole

Um einen My SQL DB-Snapshot auf eine MariaDB-DB-Instance zu migrieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Snapshots und dann den My SQL DB-Snapshot aus, den Sie migrieren möchten.
3. Wählen Sie unter Aktionen die Option Migrate Snapshot (Snapshot migrieren). Die Seite Datenbank migrieren wird angezeigt.

4. Wählen Sie unter In DB-Engine migrieren den Eintrag mariadb aus.

   Amazon RDS wählt die DB-Engine-Version automatisch aus. Sie können die Version der DB-Engine nicht ändern.



5. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die DB-Instance an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.
6. Wählen Sie Migrate (Migrieren).

AWS CLI

Um Daten von einem My SQL DB-Snapshot zu einer MariaDB-DB-Instance zu migrieren, führen Sie den AWS CLI `restore-db-instance-from-db-snapshot`Befehl mit den folgenden Optionen aus:

- -- db-instance-identifier — Name der DB-Instance, die aus dem DB-Snapshot erstellt werden soll.
- -- db-snapshot-identifier — Der Bezeichner für den DB-Snapshot, aus dem wiederhergestellt werden soll.
- --engine – Datenbank-Engine, die für die neue Instance verwendet werden soll.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier newmariadbinstance \
    --db-snapshot-identifier mysqlsnapshot \
    --engine mariadb
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier newmariadbinstance ^
    --db-snapshot-identifier mysqlsnapshot ^
    --engine mariadb
```

API

Um Daten von einem My SQL DB-Snapshot zu einer MariaDB-DB-Instance zu migrieren, rufen Sie den RDS API Amazon-Vorgang auf. `RestoreDBInstanceFromDBSnapshot`

Inkompatibilitäten zwischen MariaDB und My SQL

Zu den Inkompatibilitäten zwischen My SQL und MariaDB gehören die folgenden:

- Sie können einen mit My SQL 8.0 erstellten DB-Snapshot nicht nach MariaDB migrieren.
- Wenn die Quelle Meine SQL Datenbank einen SHA256 Passwort-Hash verwendet, stellen Sie sicher, dass Sie Benutzerkennwörter zurücksetzen, die SHA256 gehasht wurden, bevor Sie eine Verbindung zur MariaDB-Datenbank herstellen. Der folgende Code zeigt, wie Sie ein Passwort zurücksetzen, das gehasht wurde. SHA256

  ```
  SET old_passwords = 0;
  UPDATE mysql.user SET plugin = 'mysql_native_password',
  Password = PASSWORD('new_password')
  WHERE (User, Host) = ('master_user_name', %);
  FLUSH PRIVILEGES;
  ```

- Wenn Ihr RDS Hauptbenutzerkonto den Kennwort-Hash SHA -256 verwendet, stellen Sie sicher, dass Sie das Passwort mit der Operation AWS Management Console, dem `modify-`

`db-instance` AWS CLI Befehl oder der odifyDBInstanceRDSAPIM-Operation zurücksetzen. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

- MariaDB unterstützt das Memcached-Plug-In nicht. Die vom Memcached-Plug-In verwendeten Daten werden jedoch als InnoDB-Tabellen gespeichert. Nachdem Sie einen My SQL DB-Snapshot migriert haben, können Sie mithilfe von. auf die vom Memcached-Plug-In verwendeten Daten zugreifen. SQL Weitere Informationen über die innodb_memcache-Datenbank finden Sie unter InnoDB memcached Plugin Internals.

MariaDB auf Amazon RDS – SQL-Referenz

Im Folgenden werden gespeicherte Prozeduren beschrieben, die für Amazon RDS-Instances verfügbar sind, auf denen die MariaDB-DB-Engine ausgeführt wird.

Sie können die im System gespeicherten Prozeduren verwenden, die für MySQL-DB-Instances und für MariaDB-DB-Instances verfügbar sind. Die gespeicherten Prozeduren sind in dokumentier RDSals Referenz für meine SQL gespeicherte Prozedur. MariaDB-DB-Instances unterstützen alle der gespeicherten Prozeduren, mit Ausnahme von `mysql.rds_start_replication_until` und `mysql.rds_start_replication_until_gtid`.

Darüber hinaus werden die folgenden gespeicherten Systemprozeduren nur für Amazon RDS-DB-Instances unterstützt, auf denen die MariaDB ausgeführt wird:

- mysql.rds_replica_status
- mysql.rds_set_external_master_gtid
- mysql.rds_kill_query_id

mysql.rds_replica_status

Zeigt den Replikationsstatus einer MariaDB-Read Replica an.

Rufen Sie dieses Verfahren im Read Replica auf, um Statusinformationen zu wesentlichen Parametern der Replikat-Threads anzuzeigen

Syntax

```
CALL mysql.rds_replica_status;
```

Nutzungshinweise

Dieses Verfahren wird nur für MariaDB-DB-Instances unterstützt, auf denen MariaDB Version 10.5 und höher ausgeführt wird.

Dieses Verfahren entspricht dem Befehl `SHOW REPLICA STATUS`. Dieser Befehl wird für DB-Instances der MariaDB-Version 10.5 und höher nicht unterstützt.

In früheren Versionen von MariaDB benötigte der äquivalente Befehl `SHOW SLAVE STATUS` die Berechtigung `REPLICATION SLAVE`. In MariaDB Version 10.5 und höher benötigt er die

Berechtigung `REPLICATION REPLICA ADMIN`. Um das RDS-Management von MariaDB 10.5 und höher DB-Instances zu schützen, wird dieses neue Privileg dem RDS-Master-Benutzer nicht gewährt.

Beispiele

Das folgende Beispiel zeigt den Status einer MariaDB Read Replica:

```
call mysql.rds_replica_status;
```

Die Antwort ähnelt dem folgenden Beispiel:

```
*************************** 1. row ***************************
               Replica_IO_State: Waiting for master to send event
                    Source_Host: XX.XX.XX.XXX
                    Source_User: rdsrepladmin
                    Source_Port: 3306
                  Connect_Retry: 60
                Source_Log_File: mysql-bin-changelog.003988
            Read_Source_Log_Pos: 405
                 Relay_Log_File: relaylog.011024
                  Relay_Log_Pos: 657
          Relay_Source_Log_File: mysql-bin-changelog.003988
             Replica_IO_Running: Yes
            Replica_SQL_Running: Yes
                Replicate_Do_DB:
            Replicate_Ignore_DB:
             Replicate_Do_Table:
         Replicate_Ignore_Table:
mysql.rds_sysinfo,mysql.rds_history,mysql.rds_replication_status
        Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
                     Last_Errno: 0
                     Last_Error:
                   Skip_Counter: 0
            Exec_Source_Log_Pos: 405
                Relay_Log_Space: 1016
                Until_Condition: None
                 Until_Log_File:
                  Until_Log_Pos: 0
             Source_SSL_Allowed: No
             Source_SSL_CA_File:
             Source_SSL_CA_Path:
                Source_SSL_Cert:
```

```
               Source_SSL_Cipher:
                  Source_SSL_Key:
           Seconds_Behind_Master: 0
   Source_SSL_Verify_Server_Cert: No
                   Last_IO_Errno: 0
                   Last_IO_Error:
                  Last_SQL_Errno: 0
                  Last_SQL_Error:
     Replicate_Ignore_Server_Ids:
                Source_Server_Id: 807509301
                  Source_SSL_Crl:
              Source_SSL_Crlpath:
                      Using_Gtid: Slave_Pos
                     Gtid_IO_Pos: 0-807509301-3980
         Replicate_Do_Domain_Ids:
     Replicate_Ignore_Domain_Ids:
                   Parallel_Mode: optimistic
                       SQL_Delay: 0
             SQL_Remaining_Delay: NULL
       Replica_SQL_Running_State: Reading event from the relay log
              Replica_DDL_Groups: 15
Replica_Non_Transactional_Groups: 0
    Replica_Transactional_Groups: 3658
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.000 sec)
```

mysql.rds_set_external_master_gtid

Konfiguriert die GTID-basierte Replikation einer MariaDB-Instance, die außerhalb von Amazon RDS ausgeführt wird, zu einer MariaDB-DB-Instance. Diese gespeicherte Prozedur wird nur unterstützt, wenn die externe MariaDB-Instance Version 10.0.24 oder höher ist. Wenn Sie die Replikation einrichten, bei der eine oder beide Instances keine globalen Transaktionskennungen (GTIDs) von MariaDB unterstützen, verwenden Sie .

Durch die Verwendung von GTIDs für die Replikation werden Absturzsicherheitsfunktionen bereitgestellt, die von der binären Protokollreplikation nicht angeboten werden. Daher empfehlen wir es in den Fällen, in denen die Replikations-Instances dies unterstützen.

Syntax

```
CALL mysql.rds_set_external_master_gtid(
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , gtid
  , ssl_encryption
);
```

Parameter

host_name

Zeichenfolge. Der Hostname bzw. die IP-Adresse der außerhalb von Amazon RDS ausgeführten MariaDB-Instance, die als Quell-Instance festgelegt werden soll.

host_port

Ganzzahl. Der Port, der von der außerhalb von Amazon RDS ausgeführten MariaDB-Instance verwendet wird, die als Quell-Instance konfiguriert werden soll. Wenn Ihre Netzwerkkonfiguration die Replikation von SSH-Ports einschließt, welche die Portnummer konvertiert, geben Sie für diesen Parameter die von SSH offengelegte Portnummer an.

replication_user_name

Zeichenfolge. Die Kennung eines Benutzers mit `REPLICATION SLAVE`-Berechtigungen in der MariaDB-DB-Instance, die als Lesereplikat konfiguriert werden soll.

replication_user_password

Zeichenfolge. Das zu der für den Parameter angegebenen Benutzer-Kennung gehörige Passwor `replication_user_name`.

gtid

Zeichenfolge. Die globale Transaktionskennung für die Quell-Instance, von der aus die Replikation gestartet werden soll.

Sie können `@@gtid_current_pos` verwenden, um die aktuelle GTID zu erhalten, wenn die Quell-Instance gesperrt wurde, während Sie die Replikation konfigurieren. Das Binärprotokoll ändert sich also nicht zwischen den Punkten, wenn Sie die GTID erhalten und wenn die Replikation startet.

Andernfalls können Sie, wenn Sie `mysqldump`-Version 10.0.13 oder neuer für das Befüllen der Replikat-Instance vor der Replikation verwenden, die GTID-Position in der Ausgabe erhalten, indem Sie die Optionen `--master-data` oder `--dump-slave` verwenden. Wenn Sie `mysqldump`-Version 10.0.13 oder neuer verwenden, können Sie `SHOW MASTER STATUS` ausführen oder dieselben `mysqldump`-Optionen verwenden, um den Namen und die Position der Binärprotokolldatei zu erhalten und diese anschließend in eine GTID konvertieren, indem Sie `BINLOG_GTID_POS` auf der externen MariaDB-Instance ausführen:

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

Weitere Informationen über die Implementierung von GTIDs in MariaDB finden Sie unter Global Transaction ID in der MariaDB-Dokumentation.

ssl_encryption

Ein Wert, der angibt, ob die SSL-Verschlüsselung (Secure Socket Layer) für die Replikationsverbindung verwendet wird. 1 = SSL-Verschlüsselung, 0 = keine Verschlüsselung. Der Standardwert ist 0.

> **Note**
>
> Die Option `MASTER_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_master_gtid` muss vom Hauptbenutzer ausgeführt werden. Sie muss auf der MariaDB-DB-Instance ausgeführt werden, die Sie als Replikat einer MariaDB-Instance konfigurieren, die extern zu Amazon RDS ausgeführt wird. Bevor Sie `mysql.rds_set_external_master_gtid` ausführen, müssen Sie die Instance von MariaDB, die außerhalb von Amazon RDS ausgeführt wird, als Quell-Instance konfiguriert haben. Weitere Informationen finden Sie unter Daten in eine Amazon RDS for MariaDB-DB-Instance importieren.

> **Warning**
>
> Verwenden Sie nicht `mysql.rds_set_external_master_gtid`, um Replikation zwischen zwei Amazon RDS-DB-Instances zu verwalten. Verwenden Sie diese Option nur, wenn Sie

mit einer MariaDB-Instance replizieren, die extern zu RDS ausgeführt wird. Informationen zur Verwaltung der Replikation zwischen Amazon RDS-DB-Instances finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Nachdem Sie `mysql.rds_set_external_master_gtid` aufgerufen haben, um eine Amazon RDS-DB-Instance zum Verwenden als Lesereplikat zu konfigurieren, starten Sie durch Aufrufen von mysql.rds_start_replication den Replikationsvorgang für das Replikat. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Beim Aufrufen von `mysql.rds_set_external_master_gtid` werden Uhrzeit, Benutzer und eine „set master“-Aktion von Amazon RDS in den Tabellen `mysql.rds_history` und `mysql.rds_replication_status` protokolliert.

Beispiele

Bei der Ausführung auf einer MariaDB-DB-Instance wird das folgende Beispiel als Replikat einer Instance von MariaDB konfiguriert, die extern zu Amazon RDS ausgeführt wird.

```
call mysql.rds_set_external_master_gtid
 ('Sourcedb.some.com',3306,'ReplicationUser','SomePassW0rd','0-123-456',0);
```

mysql.rds_kill_query_id

Beendet eine Abfrage, die gegen den MariaDB-Server ausgeführt wird, um lang andauernde oder problematische Abfragen zu beenden. Sie können die Abfrage-ID identifizieren und eine bestimmte Abfrage effektiv beenden, um Leistungsprobleme zu beheben und einen optimalen Datenbankbetrieb aufrechtzuerhalten.

Syntax

```
CALL mysql.rds_kill_query_id(queryID);
```

Parameter

queryID

Ganzzahl. Die Identität der zu beendenden Abfrage.

Nutzungshinweise

Um eine an den MariaDB-Server übermittelte Abfrage zu beenden, verwenden Sie die Prozedur `mysql.rds_kill_query_id` und übergeben ihr als Parameter die Kennung der Abfrage. Um die Abfrage-ID zu erhalten, fragen Sie die PROCESSLISTMariaDB-Informationsschematabelle ab, wie folgt:

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM
               INFORMATION_SCHEMA.PROCESSLIST WHERE USER = '<user name>';
```

Die Verbindung zum MariaDB-Server bleibt bestehen.

Beispiele

Im folgenden Beispiel wird eine Abfrage mit der Abfrage-Kennung 230040 beendet:

```
call mysql.rds_kill_query_id(230040);
```

Lokale Zeitzone für MariaDB DB-Instances

Standardmäßig ist die Zeitzone für eine MariaDB-DB-Instance Universal Time Coordinated ()UTC. Sie können die Zeitzone für Ihre DB-Instance auf die lokale Zeitzone für Ihre Anwendung einstellen.

Setzen Sie den Parameter `time_zone` in der Parametergruppe für Ihre DB-Instance auf einen der unterstützten Werte, die weiter unten in diesem Abschnitt gelistet sind. Wenn Sie den Parameter `time_zone` für eine Parametergruppe setzen, wird bei allen DB-Instances und Lesereplikaten, die diese Parametergruppe verwenden, die neue lokale Zeitzone eingestellt. Weitere Informationen zum Einstellen von Parametern in einer Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

Nachdem Sie die lokale Zeitzone eingestellt haben, werden alle neuen Verbindungen zur Datenbank die Änderung reflektieren. Wenn Sie keine offenen Verbindungen zu Ihrer Datenbank haben, wenn Sie die lokale Zeitzone ändern, sehen Sie die lokale Zeitzonenaktualisierung nicht, nachdem Sie die Verbindung schließen und eine neue Verbindung öffnen.

Sie können eine andere lokale Zeitzone für eine DB-Instance sowie ein oder mehrere ihrer Lesereplikate einstellen. Verwenden Sie eine andere Parametergruppe für die DB-Instance und das/die Replica/s und stellen Sie den Parameter `time_zone` in jeder Parametergruppe auf eine andere lokale Zeit ein.

Wenn Sie über die Datenbank replizieren AWS-Regionen, verwenden die Quell-DB-Instance und die Read Replica unterschiedliche Parametergruppen (Parametergruppen gelten nur für eine). AWS-Region Sie müssen den Parameter `time_zone` in der Parametergruppe der Instance und des Lesereplikats einstellen, um dieselbe lokale Zeitzone für jede Instance zu verwenden.

Wenn Sie eine DB-Instance aus einem DB-Snapshot wiederherstellen, ist die lokale Zeitzone auf eingestellt. UTC Sie können die Zeitzone auf Ihre lokale Zeitzone einstellen, nachdem die Wiederherstellung abgeschlossen ist. Wenn Sie die DB-Instance auf einen Zeitpunkt wiederherstellen, ist die lokale Zeitzone für die wiederhergestellte DB-Instance die Zeitzoneneinstellung von der Parametergruppe für die wiederhergestellte DB-Instance.

Die Internet Assigned Numbers Authority (IANA) veröffentlicht mehrmals im Jahr neue Zeitzonen unter https://www.iana.org/time-zones. Jedes Mal, wenn eine neue kleinere Wartungsversion von MariaDB RDS veröffentlicht wird, wird sie mit den neuesten Zeitzonendaten zum Zeitpunkt der Veröffentlichung ausgeliefert. Wenn Sie die neueste Version RDS für MariaDB verwenden, haben Sie aktuelle Zeitzonendaten von. RDS Wenn Sie sichergehen möchten, dass Ihre DB-Instance über aktuelle Zeitzonendaten verfügt, empfehlen wir ein Upgrade auf eine höhere DB-Engine-Version.

Alternativ können Sie die Zeitzonentabellen in MariaDB-DB-Instances manuell ändern. Dazu können Sie SQL Befehle verwenden oder das Tool mysql_tzinfo_to_sql in einem Client ausführen. SQL Nachdem Sie die Zeitzonendaten manuell aktualisiert haben, starten Sie Ihre DB-Instance neu, damit die Änderungen wirksam werden. RDSändert oder setzt die Zeitzonendaten laufender DB-Instances nicht zurück. Neue Zeitzonendaten werden nur installiert, wenn Sie ein Upgrade der Datenbank-Engine-Version durchführen.

Sie können Ihre lokale Zeitzone auf die folgenden Werte einstellen.

Bereich	Zeitzone
Afrika	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
Amerika	America/Araguaina, America/Asuncion, America/Bogota, America/Buenos_Aires, America/Caracas, America/Chihuahua, America/Cuiaba, America/Denver, America/Fortaleza, America/Guatemala, America/Halifax, America/Manaus, America/Matamoros, America/Monterrey, America/Montevideo, America/Phoenix, America/Santiago, America/Tijuana
Asien	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damascus, Asia/Dhaka, Asia/Irkutsk, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Krasnoyarsk, Asia/Magadan, Asia/Muscat, Asia/Novosibirsk, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantik/Azoren
Australien	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Hobart, Australia/Perth, Australia/Sydney
Brasilien	Brazil/DeNoronha, Brazil/East
Kanada	Canada/Newfoundland, Canada/Saskatchewan, Canada/Yukon
Europa	Europe/Amsterdam, Europe/Athens, Europe/Dublin, Europe/Helsinki, Europe/Istanbul, Europe/Kaliningrad Europe/Moscow, Europe/Paris, Europe/Prague, Europe/Sarajevo

Bereich	Zeitzone
Pazifik	Pacific/Auckland, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Samoa
US	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific
UTC	UTC

Bekannte Probleme und Einschränkungen für RDS für MariaDB

Die folgenden Punkte sind bekannte Probleme und Einschränkungen bei der Verwendung von RDS für MariaDB.

> **Note**
>
> Diese Liste ist nicht umfassend.

Themen

- MariaDB-Dateigrößenlimits in Amazon RDS
- Reserviertes Wort InnoDB
- Benutzerdefinierte Ports
- Performance Insights

MariaDB-Dateigrößenlimits in Amazon RDS

Bei MariaDB-DB-Instances beträgt die Maximalgröße einer Tabelle 16 TB, wenn InnoDB-Datei-pro-Tabelle-Tablespaces verwendet werden. Dieses Limit beschränkt auch den Tabellenraum des Systems auf maximal 16 TB. InnoDB-Datei-pro-Tabelle-Tablespaces (mit Tabellen in jeweils einem eigenen Tablespace) werden für MariaDB-DB-Instances standardmäßig festgelegt. Dieses Limit hängt nicht mit dem maximalen Speicherlimit für MariaDB-DB-Instances zusammen. Weitere Informationen über das Speicherlimit finden Sie unter Amazon RDS-DB-Instance-Speicher.

Die Option der InnoDB-Tabellenräumen (innodb_file_per_table) bietet abhängig von der Anwendung sowohl Vor- als auch Nachteile. Um den besten Ansatz für Ihre Anwendung zu bestimmen, lesen Sie File-per-table tablespaces in der MySQL-Dokumentation.

Es wird nicht empfohlen, die Tabellen bis zur maximal möglichen Größe anwachsen zu lassen. Generell hat es sich bewährt, Daten in kleinere Tabellen zu partitionieren, wodurch sich die Leistung und die Wiederherstellungszeiten verbessern.

Eine Möglichkeit, mit der Sie eine große Tabelle in kleinere Tabellen aufteilen können, ist die Partitionierung. Die Partitionierung verteilt Teile Ihrer großen Tabelle in separate Dateien auf der Basis von Regeln, die Sie angeben. Wenn Sie beispielsweise Transaktionen nach Datum speichern, können Sie Partitionierungsregeln erstellen, mit denen ältere Transaktionen in separate Dateien

partitioniert werden. Anschließend können Sie regelmäßig die historischen Transaktionsdaten archivieren, die für Ihre Anwendung nicht ständig verfügbar sein müssen. Weitere Informationen finden Sie unter Partitionierung in der MySQL-Dokumentation.

So ermitteln Sie die Größe aller InnoDB-Tabellenräume

- Verwenden Sie den folgenden SQL-Befehl, um zu bestimmen, ob eine Ihrer Tabellen zu groß ist und evtl. partitioniert werden sollte.

> **Note**
>
> Für MariaDB 10.6 und höher gibt diese Abfrage auch die Größe des Tabellenraums des InnoDB-Systems zurück.
> Bei MariaDB-Versionen vor 10.6 können Sie die Größe des Tabellenraums des InnoDB-Systems nicht ermitteln, indem Sie die Systemtabellen abfragen. Es wird empfohlen, ein Upgrade auf eine neuere Version durchzuführen.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

So ermitteln Sie die Größe von Nicht-InnoDB-Benutzertabellen

- Verwenden Sie den folgenden SQL-Befehl, um zu bestimmen, ob eine Ihrer Nicht-InnoDB-Benutzertabellen zu groß ist.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

So aktivieren Sie InnoDB-Datei-pro-Tabelle-Tabellenräume

- Setzen Sie den `innodb_file_per_table`-Parameter in der Parametergruppe für die DB-Instance auf 1.

So deaktivieren Sie InnoDB-Datei-pro-Tabelle-Tabellenräume

- Setzen Sie den `innodb_file_per_table`-Parameter in der Parametergruppe für die DB-Instance auf `0`.

Weitere Informationen über das Updaten von Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Wenn Sie InnoDB-Datei-pro-Tabelle-Tabellenräume aktiviert oder deaktiviert haben, können Sie den Befehl `ALTER TABLE` ausführen. Sie können diesen Befehl verwenden, um eine Tabelle vom globalen Tablespace in einen eigenen Tablespace zu verschieben. Oder Sie können eine Tabelle aus ihrem eigenen Tablespace in den globalen Tablespace verschieben. Im Folgenden sehen Sie ein Beispiel.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

Reserviertes Wort InnoDB

`InnoDB` ist ein reserviertes Wort für RDS for MariaDB. Sie können diesen Namen für eine MariaDB-Datenbank nicht verwenden.

Benutzerdefinierte Ports

Amazon RDS blockiert Verbindungen zum benutzerdefinierten Port 33060 für die MariaDB-Engine. Wählen Sie einen anderen Port für Ihre MariaDB-Engine.

Performance Insights

InnoDB-Zähler sind in Performance Insights for RDS für MariaDB Version 10.11 nicht sichtbar, da sie von der MariaDB-Community nicht mehr unterstützt werden.

Amazon RDS for Microsoft SQL Server

Amazon RDS unterstützt mehrere Versionen und Editionen von Microsoft SQL Server. Die folgende Tabelle enthält die neueste unterstützte Version der jeweiligen Hauptversion. Eine vollständige Liste der unterstützten Versionen, Editionen und RDS-Engine-Versionen finden Sie unter Microsoft SQL Server-Versionen auf Amazon RDS.

Hauptversion	Service Pack/ GDR	Kumulatives Update	Unterversion	Knowledge-Base-Artikel	Veröffentlichungsdatum
SQL Server 2022	Nicht zutreffend	CU20	16.0.4205.1	KB5059390	10. Juli 2025
SQL Server 2019	GDR	CU32 DDR	15.0.4435,7	KB5058722	8. Juli 2025
SQL Server 2017	Nicht zutreffend	CU31 DDR	14.0.3495,9	KB5058714	8. Juli 2025
SQL Server 2016	SP3 DDR	CU14	13.0.7055,9	KB5058717	8. Juli 2025

Weitere Informationen zur Lizenzierung für SQL Server finden Sie unter Lizenzierung Microsoft SQL Server auf Amazon RDS. Informationen zu SQL Server-Builds finden Sie in diesem Microsoft-Supportartikel unter Wo finde ich Informationen zu den neuesten SQL Server-Builds.

Mit Amazon RDS können Sie DB-Instances und DB-Snapshots, point-in-time Wiederherstellungen und automatisierte oder manuelle Backups erstellen. DB-Instance in SQL Server können innerhalb einer VPC verwendet werden. Sie können auch mithilfe von Secure Sockets Layer (SSL) eine Verbindung zu einer DB-Instance herstellen, die SQL Server ausführt. Und Sie können mithilfe von Transparent Data Encryption (TDE) Data-at-Rest verschlüsseln. Amazon RDS unterstützt derzeit Multi-AZ-Bereitstellungen für SQL Server mithilfe von SQL Server Database Mirroring (DBM) oder Always On Availability Groups (AGs) als Failover-Lösung mit hoher Verfügbarkeit.

Um eine verwaltete Service-Erfahrung zu bieten, stellt Amazon RDS keinen Shell-Zugriff auf DB-Instances bereit, und beschränkt den Zugriff auf bestimmte Systemprozeduren und -tabellen, die

erweiterte Sonderrechte erfordern. Amazon RDS unterstützt den Zugriff auf Datenbanken auf einer DB-Instance mit jeder Standard-SQL-Client-Anwendung wie Microsoft SQL Server Management Studio. Amazon RDS erlaubt keinen direkten Hostzugriff auf eine DB-Instance über Telnet, Secure Shell (SSH) oder Windows Remote Desktop Connection. Wenn Sie eine DB-Instance erstellen, wird der Master-Benutzer der Rolle db_owner für alle Benutzerdatenbanken auf dieser Instance zugewiesen und hat alle Berechtigungen auf Datenbankebene, mit Ausnahme derjenigen, die für Backups verwendet werden. Amazon RDS verwaltet Backups für Sie.

Bevor Sie Ihre erste DB-Instance erstellen, sollten Sie die Schritte für die Einrichtung in diesem Leitfaden durchführen. Weitere Informationen finden Sie unter Einrichtung Ihrer Amazon RDS-Umgebung.

Themen

- Häufige Verwaltungsaufgaben für Microsoft SQL Server in Amazon RDS
- Einschränkungen für Microsoft SQL Server-DB-Instances
- Unterstützung für DB-Instance-Klassen für Microsoft SQL Server
- Sicherheit für Microsoft SQL Server
- Unterstützung zu Compliance-Programmen für Microsoft SQL Server-DB-Instances
- Microsoft SQL Server-Versionen auf Amazon RDS
- Funktionen von Microsoft SQL Server bei Amazon RDS
- Multi-AZ-Bereitstellungen, die die Microsoft SQL Server-Datenbankspiegelung oder AlwaysOn-Verfügbarkeitsgruppen verwenden
- Verwenden von Transparent Data Encryption zur Verschlüsselung ruhender Daten
- Funktionen und gespeicherte Prozeduren für Amazon RDS for Microsoft SQL Server
- Lokale Zeitzone für Microsoft SQL Server-DB-Instances
- Lizenzierung Microsoft SQL Server auf Amazon RDS
- Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen
- Arbeiten mit Active Directory mit RDS für SQL Server
- Upgrades der Microsoft SQL Server-DB-Engine
- Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung
- Arbeiten mit Read Replicas für Microsoft SQL Server in Amazon RDS

- Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL
- Zusätzliche Funktionen für Microsoft SQL Server auf Amazon RDS
- Optionen für die Microsoft SQL Server-Datenbank-Engine
- Allgemeine DBA-Aufgaben für Amazon RDS for Microsoft SQL Server

Häufige Verwaltungsaufgaben für Microsoft SQL Server in Amazon RDS

Im Folgenden werden die Verwaltungsaufgaben veranschaulicht, die Sie mit einer Amazon RDS for SQL Server-DB-Instance am häufigsten durchführen. Bei jeder Aufgabe sind Links zu relevanter Dokumentation enthalten.

Aufgabenbereich	Beschreibung	Relevante Dokumentation
Instance-Klassen, Speicher und PIOPS	Wenn Sie eine DB-Instance zu Produktionszwecken erstellen, müssen Sie wissen, wie Instance-Klassen, Speichertypen und bereitgestellte IOPS in Amazon RDS funktionieren.	Unterstützung für DB-Instance-Klassen für Microsoft SQL Server Amazon RDS-Speichertypen
Multi-AZ-Bereitstellungen	Bei einer DB-Instance für die Produktion sollten Multi-AZ-Bereitstellungen eingesetzt werden. Multi-AZ-Bereitstellungen bieten eine erhöhte Verfügbarkeit, eine längere Lebensdauer von Daten sowie eine höhere Fehlertoleranz für DB-Instances. Multi-AZ-Bereitstellungen für SQL Server werden mithilfe der systemeigenen DBM- oder Technologie von SQL Server implementiert. AGs	Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS Multi-AZ-Bereitstellungen, die die Microsoft SQL Server-Datenbankspiegelung oder AlwaysOn-Verfügbarkeitsgruppen verwenden

Aufgabenbereich	Beschreibung	Relevante Dokumentation
Amazon Virtual Private Cloud (VPC)	Wenn Ihr AWS Konto über eine Standard-VPC verfügt, wird Ihre DB-Instance automatisch in der Standard-VPC erstellt. Wenn Ihr Konto über keine Standard-VPC verfügt und Sie die DB-Instance in einer VPC erstellen möchten, müssen Sie zunächst die VPC und Subnetz-Gruppen erstellen.	Arbeiten mit einer DB-Instance in einer VPC
Sicherheitsgruppen	DB-Instances werden standardmäßig mit einer Firewall erstellt, die den Zugriff auf die Instance verhindert. Daher müssen Sie eine Sicherheitsgruppe mit den korrekten IP-Adressen und Netzwerkkonfigurationen erstellen, um auf die DB-Instance zuzugreifen.	Zugriffskontrolle mit Sicherheitsgruppen
Parametergruppen	Wenn Ihre DB-Instance spezifische Datenbankparameter erfordert, sollten Sie vor der DB-Instance eine Parametergruppe erstellen.	Parametergruppen für Amazon RDS
Optionsgruppen	Wenn Ihre DB-Instance spezifische Datenbankoptionen erfordert, sollten Sie vor der DB-Instance eine Optionsgruppe erstellen.	Optionen für die Microsoft SQL Server-Datenbank-Engine

Aufgabenbereich	Beschreibung	Relevante Dokumentation
Herstellen einer Verbindung mit einer DB-Instance	Nachdem Sie eine Sicherheitsgruppe erstellt und einer DB-Instance zugeordnet haben, können Sie über eine Standard-SQL-Client-Anwendung wie Microsoft SQL Server Management Studio eine Verbindung zu dieser DB-Instance aufbauen.	Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen
Backup und Wiederherstellung	Beim Erstellen Ihrer DB-Instance können Sie automatisierte Backups konfigurieren. Sie können Ihre Datenbanken auch mithilfe von vollständigen Sicherungsdateien (.bak-Dateien) manuell sichern oder wiederherstellen.	Einführung in Backups Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung
Überwachung	Sie können Ihre SQL Server-DB-Instance mithilfe von CloudWatch Amazon RDS-Metriken, Ereignissen und erweiterter Überwachung überwachen.	Metriken in der RDS Amazon-Konsole anzeigen RDSAmazon-Ereignisse anzeigen
Protokolldateien	Sie können auf die Protokolldateien für Ihre SQL Server-DB-Instance zugreifen.	Überwachung von Amazon RDS Amazon Datenbankprotokolldateien von Amazon RDS für Microsoft SQL Server

Es gibt auch erweiterte administrative Aufgaben für die Arbeit mit SQL Server-DB-Instances. Weitere Informationen finden Sie in der folgenden Dokumentation:

- Allgemeine DBA-Aufgaben für Amazon RDS for Microsoft SQL Server.
- Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server
- Zugriff auf die Datenbank tempdb

Einschränkungen für Microsoft SQL Server-DB-Instances

Für die Amazon RDS-Implementierung von Microsoft SQL Server in einer DB-Instance gelten einige Beschränkungen, die Sie kennen sollten:

- Die maximale Anzahl von Datenbanken, die auf einer DB-Instance unterstützt werden, hängt vom Instance-Klassentyp und dem Verfügbarkeitsmodus ab — Single-AZ, Multi-AZ Database Mirroring (DBM) oder Multi-AZ Availability Groups (). AGs Die Microsoft SQL Server-Systemdatenbanken werden bei der Feststellung der Anzahl nicht mit berücksichtigt.

  Die folgende Tabelle zeigt die maximale Anzahl unterstützter Datenbanken für die einzelnen Instance-Klassentypen und Verfügbarkeitsmodi. Durch diese Tabelle können Sie einfacher entscheiden, ob Sie den Instance-Klassentyp oder den Verfügbarkeitsmodus wechseln können. Wenn Ihre Quell-DB-Instance mehr Datenbanken hat als der Typ der Ziel-Instance-Klasse oder mehr als der Verfügbarkeitsmodus unterstützen kann, schlägt das Ändern der DB-Instance fehl. Sie können den Status Ihrer Anforderung im Bereich Events (Ereignisse) anzeigen.

Typ der Instance-Klasse	Single-AZ	Multi-AZ mit Datenbankspiegelung	Multi-AZ mit Always On AGs
db.*.micro to db.*.medium	30	–	–
db.*.large	30	30	30
db.*.xlarge to db.*.16xlarge	100	50	75
db.*.24xlarge zu db.*.32xlarge	100	50	100

  * stellt die unterschiedlichen Typen von Instance-Klassen dar.

Angenommen, Ihre DB-Instance wird auf einer db.*.16xlarge mit Single-AZ ausgeführt und verfügt über 76 Datenbanken. Sie ändern die DB-Instance, um ein Upgrade auf Multi-AZ Always On durchzuführen. AGs Dieses Upgrade schlägt fehl, da Ihre DB-Instance mehr Datenbanken enthält, als Ihre Ziel-Konfiguration unterstützen kann. Wenn Sie den Typ Ihrer Instance-Klasse stattdessen auf db.*.24xlarge upgraden, ist die Änderung erfolgreich.

Schlägt das Upgrade fehl, werden Ihnen Ereignisse und Nachrichten ähnlich der folgenden angezeigt:

- Die Klasse der Datenbank-Instance kann nicht geändert werden. Die Instance verfügt über 76 Datenbanken, aber nach der Konvertierung würde sie nur 75 unterstützen 75.
- Konvertierung der DB-Instance zu Multi-AZ nicht möglich: Die Instance verfügt über 76 Datenbanken, nach einer Konvertierung würden aber nur 75 unterstützt werden.

Wenn die point-in-time Wiederherstellung oder Snapshot-Wiederherstellung fehlschlägt, werden Ereignisse und Meldungen angezeigt, die den folgenden ähneln:

- Datenbank-Instance auf incompatible-restore festgelegt. Die Instance verfügt über 76 Datenbanken, aber nach der Konvertierung würde sie nur 75 unterstützen 75.

- Die folgenden Ports sind für Amazon RDS reserviert und Sie können sie nicht beim Erstellen der DB-Instance verwenden: `1234, 1434, 3260, 3343, 3389, 47001,` und `49152-49156`.
- Client-Verbindungen von IP-Adressen im Bereich 169.254.0.0/16 sind nicht erlaubt. Dies ist der APIPA-Bereich (Automatic Private IP Addressing), der für die Local-Link-Adressierung verwendet wird.
- SQL Server Standard Edition verwendet nur einen teil der verfügbaren Prozessoren, wenn die DB-Instance mehr Prozessoren als die Softwarelimits hat (24 Kerne, 4 Sockets und 128 GB RAM). Beispiele dafür sind die Instance-Klassen db.m5.24xlarge und db.r5.24xlarge.

  Weitere Informationen finden Sie in der Tabelle der Maßstabsgrenzen unter Editionen und unterstützte Funktionen von SQL Server 2019 (15.x) in der Microsoft-Dokumentation.
- Amazon RDS for SQL Server unterstützt nicht das Importieren von Daten in die msdb-Datenbank.
- Sie können eine Datenbank in einer DB-Instance in einer SQL Server-Multi-AZ-Bereitstellung nicht umbenennen.
- Stellen Sie sicher, dass Sie diese Richtlinien verwenden, wenn Sie die folgenden DB-Parameter auf RDS for SQL Server festlegen:
  - `max server memory (mb)` >= 256 MB
  - `max worker threads`>= (Anzahl der logischen Zeichen x CPUs 7)

Weitere Informationen zum Einstellen von DB-Parametern finden Sie unter Parametergruppen für Amazon RDS.

- Die maximale Speicherplatzgröße für SQL-Server-DB-Instances beträgt:
  - Allzweckspeicher (SSD): 16 TiB für alle Editionen
  - Bereitgestellter IOPS-Speicher — 64 TiB für alle Editionen
  - Magnetspeicher: 1 TiB für alle Editionen

  In einem Szenario mit größerem Speicherbedarf ist es möglich, diese Beschränkung durch Sharding über mehrere DB-Instances zu umgehen. Diese Methode verlangt eine datenabhängige Routing-Logik in Anwendungen, die eine Verbindung zum Sharding-System aufbauen. Sie können ein bestehendes Sharding-Framework nutzen oder Sharding durch benutzerdefinierten Code einrichten. Wenn Sie ein vorhandenes Framework nutzen, kann dieses keine Komponenten auf dem Server installieren, auf dem sich die DB-Instance befindet.
- Die Mindestspeicherplatzgröße für SQL-Server-DB-Instances beträgt:
  - General Purpose (SSD)-Speicher – 20 GiB für Enterprise, Standard, Web und Express Editionen
  - Bereitgestellter IOPS-Speicher – 20 GiB für Enterprise, Standard, Web und Express Editionen
  - Magnetspeicher-Speicher – 20 GiB für Enterprise, Standard, Web und Express Editionen
- Amazon RDS unterstützt nicht das Ausführen dieser Services auf dem selben Server wie Ihre RDS-DB-Instance:
  - Data Quality Services
  - Master Data Services

  Um diese Funktionen nutzen zu können, empfehlen wir, SQL Server auf einer EC2 Amazon-Instance zu installieren oder eine lokale SQL Server-Instance zu verwenden. In diesen Fällen fungiert die EC2 oder SQL Server-Instance als Master Data Services-Server für Ihre SQL Server-DB-Instance auf Amazon RDS. Sie können SQL Server gemäß den Microsoft-Lizenzrichtlinien auf einer EC2 Amazon-Instance mit Amazon EBS-Speicher installieren.
- Aufgrund von Beschränkungen in Microsoft SQL Server reflektiert eine zeitpunktbezogene Wiederherstellung vor der erfolgreichen Ausführung von `DROP DATABASE` möglicherweise nicht den Zustand dieser Datenbank zu diesem Zeitpunkt. Beispiel: Der Zustand der verworfenen Datenbank wird in der Regel 5 Minuten vor Ausgabe des `DROP DATABASE`-Befehls wiederhergestellt. Diese Art der Wiederherstellung bedeutet, dass Sie die Transaktionen, die während dieser paar Minuten durchgeführt wurden, nicht auf der verworfenen Datenbank wiederherstellen können. Um dieses Problem zu umgehen, können Sie den Befehl `DROP`

`DATABASE` erneut nach Abschluss der Wiederherstellungsoperation ausgeben. Durch Löschen einer Datenbank per Drop werden die Transaktionsprotokolle für diese Datenbank entfernt.

- In SQL Server erstellen Sie die Datenbanken nach der DB-Instance. Die Datenbanknamen folgen den üblichen SQL-Server-Namensregeln mit den folgenden Unterschieden:
  - Datenbanknamen dürfen nicht mit beginne `rdsadmin`.
  - Sie können nicht mit einem Leerzeichen oder einem Tabulator beginnen oder enden.
  - Sie dürfen keine Zeichen enthalten, die eine neue Zeile erzeugen.
  - Sie dürfen kein einzelnes Anführungszeichen (`'`) enthalten.
- Mit SQL Server Web Edition können Sie nur die Dev/Test-Vorlage verwenden, wenn Sie eine neue RDS für SQL Server-DB-Instance erstellen.

Unterstützung für DB-Instance-Klassen für Microsoft SQL Server

Die Rechen- und Speicherkapazität von DB-Instances wird über deren Klasse festgelegt. Die benötigte DB-Instance-Klasse richtet sich nach Ihren Rechen- und Speicheranforderungen. Weitere Informationen finden Sie unter .

Die folgende Liste der DB-Instance-Klassen, die für Microsoft SQL Server unterstützt werden, wird hier zur besseren Übersicht bereitgestellt. Die aktuelle Liste finden Sie in der RDS Konsole: https://console.aws.amazon.com/rds/.

Nicht alle DB-Instance-Klassen sind auf allen unterstützten SQL Server-Nebenversionen verfügbar. Zum Beispiel sind einige neuere DB-Instance-Klassen wie db.r6i für ältere Nebenversionen nicht verfügbar. Sie können den AWS CLI Befehl describe-orderable-db-instance-options verwenden, um herauszufinden, welche DB-Instance-Klassen für Ihre SQL Server-Edition und -Version verfügbar sind.

SQLServ - Edition	Unterstützungsbereich 2022	Supportbereich 2019	Supportbereich 2017 und 2016

Enterpris e Edition	`db.t3.xla rge –db.t3.2xlarge`
	
	`db.r5.xla rge –db.r5.24xlarge`
		`db.t3.xla rge –db.t3.2xlarge`
		
		`db.r5.xla rge –db.r5.24xlarge`
			`db.t3.xla rge –db.t3.2xlarge`
			
			`db.r5.xla rge –db.r5.24xlarge`

SQLServ - Edition	Unterstützungsbereich 2022	Supportbereich 2019	Supportbereich 2017 und 2016
	db.r5b.xl arge –db.r5b.24 xlarge	db.r5b.xl arge –db.r5b.24 xlarge	db.r5b.xl arge –db.r5b.24 xlarge
	db.r5d.xl arge –db.r5d.24 xlarge	db.r5d.xl arge –db.r5d.24 xlarge	db.r5d.xl arge –db.r5d.24 xlarge
	db.r6i.xl arge –db.r6i.32 xlarge	db.r6i.xl arge –db.r6i.32 xlarge	db.r6i.xl arge –db.r6i.32 xlarge
	db.m5.xla rge –db.m5.24xlarge	db.m5.xla rge –db.m5.24xlarge	db.m5.xla rge –db.m5.24xlarge
	db.m5d.xl arge –db.m5d.24 xlarge	db.m5d.xl arge –db.m5d.24 xlarge	db.m5d.xl arge –db.m5d.24 xlarge
	db.m6i.xl arge –db.m6i.32 xlarge	db.m6i.xl arge –db.m6i.32 xlarge	db.m6i.xl arge –db.m6i.32 xlarge
	db.x2iedn .xlarge –db.x2iedn .32xlarge	db.x1.16x large –db.x1.32x large	db.x1.16x large –db.x1.32x large
	db.z1d.xl arge –db.z1d.12 xlarge	db.x1e.xl arge –db.x1e.32 xlarge	db.x1e.xl arge –db.x1e.32 xlarge
		db.x2iedn .xlarge –db.x2iedn .32xlarge	db.x2iedn .xlarge –db.x2iedn .32xlarge

SQLServ - Edition	Unterstützungsbereich 2022	Supportbereich 2019	Supportbereich 2017 und 2016
		db.z1d.xl arge –db.z1d.12 xlarge	db.z1d.xl arge –db.z1d.12 xlarge

SQLServ - Edition	Unterstützungsbereich 2022	Supportbereich 2019	Supportbereich 2017 und 2016
Standarc Edition	db.t3.xla rge –db.t3.2xlarge db.r5.lar ge –db.r5.24xlarge db.r5b.la rge –db.r5b.8xlarge db.r5d.la rge –db.r5d.24 xlarge db.r6i.la rge –db.r6i.8xlarge db.m5.lar ge –db.m5.24xlarge db.m5d.la rge –db.m5d.24 xlarge db.m6i.la rge –db.m6i.8xlarge db.x2iedn .xlarge –db.x2iedn .8xlarge db.z1d.la rge –db.z1d.12 xlarge	db.t3.xla rge –db.t3.2xlarge db.r5.lar ge –db.r5.24xlarge db.r5b.la rge –db.r5b.24 xlarge db.r5d.la rge –db.r5d.24 xlarge db.r6i.la rge –db.r6i.8xlarge db.m5.lar ge –db.m5.24xlarge db.m5d.la rge –db.m5d.24 xlarge db.m6i.la rge –db.m6i.8xlarge db.x1.16x large –db.x1.32x large db.x1e.xl arge –db.x1e.32 xlarge	db.t3.xla rge –db.t3.2xlarge db.r5.lar ge –db.r5.24xlarge db.r5b.la rge –db.r5b.24 xlarge db.r5d.la rge –db.r5d.24 xlarge db.r6i.la rge –db.r6i.8xlarge db.m5.lar ge –db.m5.24xlarge db.m5d.la rge –db.m5d.24 xlarge db.m6i.la rge –db.m6i.8xlarge db.x1.16x large –db.x1.32x large db.x1e.xl arge –db.x1e.32 xlarge

SQLServ - Edition	Unterstützungsbereich 2022	Supportbereich 2019	Supportbereich 2017 und 2016
		db.x2iedn .xlarge –db.x2iedn .32xlarge db.z1d.la rge –db.z1d.12 xlarge	db.x2iedn .xlarge –db.x2iedn .32xlarge db.z1d.la rge –db.z1d.12 xlarge
Web Edition	db.t3.sma ll –db.t3.xlarge db.r5.lar ge –db.r5.4xlarge db.r5b.la rge –db.r5b.4xlarge db.r5d.la rge –db.r5d.4xlarge db.r6i.la rge –db.r6i.4xlarge db.m5.lar ge –db.m5.4xlarge db.m5d.la rge –db.m5d.4xlarge db.m6i.la rge –db.m6i.4xlarge db.z1d.la rge –db.z1d.13 xlarge	db.t3.sma ll –db.t3.2xlarge db.r5.lar ge –db.r5.4xlarge db.r5b.la rge –db.r5b.4xlarge db.r5d.la rge –db.r5d.4xlarge db.r6i.la rge –db.r6i.4xlarge db.m5.lar ge –db.m5.4xlarge db.m5d.la rge –db.m5d.4xlarge db.m6i.la rge –db.m6i.4xlarge db.z1d.la rge –db.z1d.3xlarge	db.t3.sma ll –db.t3.2xlarge db.r5.lar ge –db.r5.4xlarge db.r5b.la rge –db.r5b.4xlarge db.r5d.la rge –db.r5d.4xlarge db.r6i.la rge –db.r6i.4xlarge db.m5.lar ge –db.m5.4xlarge db.m5d.la rge –db.m5d.4xlarge db.m6i.la rge –db.m6i.4xlarge db.z1d.la rge –db.z1d.3xlarge

SQLServ - Edition	Unterstützungsbereich 2022	Supportbereich 2019	Supportbereich 2017 und 2016
Express Edition	db.t3.mic ro –db.t3.xlarge	db.t3.mic ro –db.t3.xlarge	db.t3.mic ro –db.t3.xlarge

Sicherheit für Microsoft SQL Server

Die Microsoft SQL Server-Datenbank-Engine verwendet rollenbasierte Sicherheit. Der Hauptbenutzername, den Sie beim Erstellen einer DB-Instance angeben, ist ein Anmeldename für die SQL Serverauthentifizierung, der zu den `setupadmin` festen `public` Serverrollen`processadmin`, und gehört.

Jeder Benutzer, der eine Datenbank erstellt, wird der db_owner-Rolle für diese Datenbank zugewiesen und er erhält die Berechtigungen für alle Datenbankebenen, außer für diejenigen, die für Datensicherungen verwendet werden. Amazon RDS verwaltet Backups für Sie.

Die folgenden Rollen auf Serverebene sind in Amazon RDS for SQL Server nicht verfügbar:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

Die folgenden Berechtigungen auf Serverebene sind RDS für SQL Server-DB-Instances nicht verfügbar:

- ALTER ANY DATABASE
- ALTER ANY EVENT NOTIFICATION
- ALTER RESOURCES
- ALTERSETTINGS(Sie können die API DB-Parametergruppen-Operationen verwenden, um Parameter zu ändern. Weitere Informationen finden Sie unterParametergruppen für Amazon RDS)

- AUTHENTICATE SERVER
- CONTROL_SERVER
- CREATE DDL EVENT NOTIFICATION
- CREATE ENDPOINT
- CREATE SERVER ROLE
- CREATE TRACE EVENT NOTIFICATION
- DROP ANY DATABASE
- EXTERNAL ACCESS ASSEMBLY
- SHUTDOWN(Sie können stattdessen die RDS Reboot-Option verwenden)
- UNSAFE ASSEMBLY
- ALTER ANY AVAILABILITY GROUP
- CREATE ANY AVAILABILITY GROUP

SSLUnterstützung für Microsoft SQL Server-DB-Instances

Sie können SSL damit Verbindungen zwischen Ihren Anwendungen und Ihren RDS Amazon-DB-Instances verschlüsseln, auf denen Microsoft SQL Server ausgeführt wird. Sie können auch die Verwendung SSL aller Verbindungen zu Ihrer DB-Instance erzwingen. Wenn Sie die Verwendung von Verbindungen erzwingenSSL, geschieht dies für den Client transparent, und der Client muss keine Arbeit für die Verwendung SSL ausführen.

SSLwird in allen AWS Regionen und für alle unterstützten SQL Server-Editionen unterstützt. Weitere Informationen finden Sie unter Verwendung SSL mit einer Microsoft SQL Server-DB-Instance.

Verwendung SSL mit einer Microsoft SQL Server-DB-Instance

Sie können Secure Sockets Layer (SSL) verwenden, um Verbindungen zwischen Ihren Client-Anwendungen und Ihren RDS Amazon-DB-Instances zu verschlüsseln, auf denen Microsoft SQL Server ausgeführt wird. SSLSupport ist in allen AWS Regionen für alle unterstützten SQL Server-Editionen verfügbar.

Wenn Sie eine SQL Server-DB-Instance erstellen, RDS erstellt Amazon ein SSL Zertifikat dafür. Das SSL Zertifikat enthält den DB-Instance-Endpunkt als Common Name (CN) für das SSL Zertifikat zum Schutz vor Spoofing-Angriffen.

Es gibt zwei Möglichkeiten, eine Verbindung SSL zu Ihrer SQL Server-DB-Instance herzustellen:

- SSLFür alle Verbindungen erzwingen — dies geschieht für den Client transparent, und der Client muss für die Nutzung SSL keinerlei Arbeit verrichten.

  > **Note**
  >
  > Wenn Sie SSMS Version `rds.force_ssl` 19.3, 20.0 1 und 20.2 auswählen und verwenden, überprüfen Sie Folgendes:
  >
  > - Aktivieren Sie das Trust Server-Zertifikat in. SSMS
  > - Importieren Sie das Zertifikat in Ihr System.

- Bestimmte Verbindungen verschlüsseln — dadurch wird eine SSL Verbindung von einem bestimmten Client-Computer aus eingerichtet, und Sie müssen am Client arbeiten, um Verbindungen zu verschlüsseln.

Informationen zur Unterstützung von Transport Layer Security (TLS) für SQL Server finden Sie unter TLS1.2 Unterstützung für Microsoft SQL Server.

Erzwingen Sie die Verwendung von Verbindungen zu Ihrer DB-Instance SSL

Sie können die Verwendung aller Verbindungen zu Ihrer DB-Instance erzwingenSSL. Wenn Sie die Verwendung von Verbindungen erzwingenSSL, geschieht dies für den Client transparent, und der Client muss keine Arbeit für die Verwendung SSL ausführen.

Wenn Sie erzwingen möchtenSSL, verwenden Sie den `rds.force_ssl` Parameter. Standardmäßig ist der `rds.force_ssl`-Parameter auf `0 (off)` festgelegt. Stellen Sie den `rds.force_ssl`

Parameter auf ein, `1 (on)` um die Verwendung von Verbindungen zu erzwingenSSL. Der `rds.force_ssl`-Parameter ist statisch, daher müssen Sie nach dem Ändern des Werts Ihre DB-Instance neu starten, damit die Änderung übernommen wird.

Um die Verwendung aller Verbindungen zu Ihrer DB-Instance zu erzwingen SSL

1. Bestimmen Sie die Parametergruppe, die an Ihre DB-Instance angefügt ist:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

   b. Wählen Sie in der oberen rechten Ecke der RDS Amazon-Konsole die AWS Region Ihrer DB-Instance aus.

   c. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann den Namen Ihrer DB-Instance aus, um deren Details anzuzeigen.

   d. Wählen Sie die Registerkarte Konfiguration aus. Suchen Sie die Parametergruppe im Abschnitt.

2. Falls erforderlich, erstellen Sie eine neue Parametergruppe. Falls Ihre DB-Instance die standardmäßige Parametergruppe verwendet, müssen Sie eine neue Parametergruppe erstellen. Falls Ihre DB-Instance eine nicht standardmäßige Parametergruppe verwendet, können Sie die vorhandene Parametergruppe bearbeiten oder eine neue Parametergruppe erstellen. Falls Sie eine vorhandene Parametergruppe bearbeiten, wirkt sich die Änderung auf alle DB-Instances aus, die diese Parametergruppe verwenden.

   Befolgen Sie die Anweisungen in , um eine neue Parametergruppe zu erstelle Erstellen einer DB-Parametergruppe in Amazon RDS.

3. Bearbeiten Sie Ihre neue oder vorhandene Parametergruppe, um den `rds.force_ssl`-Parameter auf `true` zu setzen. Befolgen Sie die Anweisungen in , um die Parametergruppe zu bearbeite Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

4. Falls Sie eine neue Parametergruppe erstellt haben, ändern Sie Ihre DB-Instance, um die neue Parametergruppe anzufügen. Ändern Sie die Einstellung DB-Parametergruppe der DB-Instance. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

5. Starten Sie Ihre DB-Instance neu. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Verschlüsseln spezifischer Verbindungen

Sie können die Verwendung aller Verbindungen zu Ihrer DB-Instance SSL erzwingen oder nur Verbindungen von bestimmten Client-Computern verschlüsseln. Für die Verwendung SSL von einem bestimmten Client aus müssen Sie Zertifikate für den Client-Computer abrufen, Zertifikate auf den Client-Computer importieren und dann die Verbindungen vom Client-Computer verschlüsseln.

> **Note**
>
> Alle SQL Serverinstanzen, die nach dem 5. August 2014 erstellt wurden, verwenden den DB-Instance-Endpunkt im Feld Common Name (CN) des SSL Zertifikats. Vor dem 5. August 2014 war die SSL Zertifikatsüberprüfung für VPC basierte SQL Serverinstanzen nicht verfügbar. Wenn Sie über eine VPC basierte SQL Server-DB-Instance verfügen, die vor dem 5. August 2014 erstellt wurde, und Sie die SSL Zertifikatsüberprüfung verwenden und sicherstellen möchten, dass der Instanzendpunkt als CN für das SSL Zertifikat für diese DB-Instance enthalten ist, benennen Sie die Instanz um. Wenn Sie eine DB-Instance umbenennen, wird ein neues Zertifikat bereitgestellt und die Instance neu gestartet, um das neue Zertifikat zu aktivieren.

Herunterladen von Zertifikaten für Client-Computer

Um Verbindungen von einem Client-Computer zu einer RDS Amazon-DB-Instance zu verschlüsseln, auf der Microsoft SQL Server ausgeführt wird, benötigen Sie ein Zertifikat auf Ihrem Client-Computer.

Laden Sie das Zertifikat auf Ihren Client-Computer herunter. Sie können ein Stammzertifikat herunterladen, das für alle Regionen funktioniert. Sie können auch ein Zertifikatpaket herunterladen, das sowohl das alte als auch das neue Stammzertifikat enthält. Zusätzlich können Sie regionsspezifische Zwischenzertifikate herunterladen. Weitere Informationen zum Herunterladen von Zertifikaten finden Sie unter .

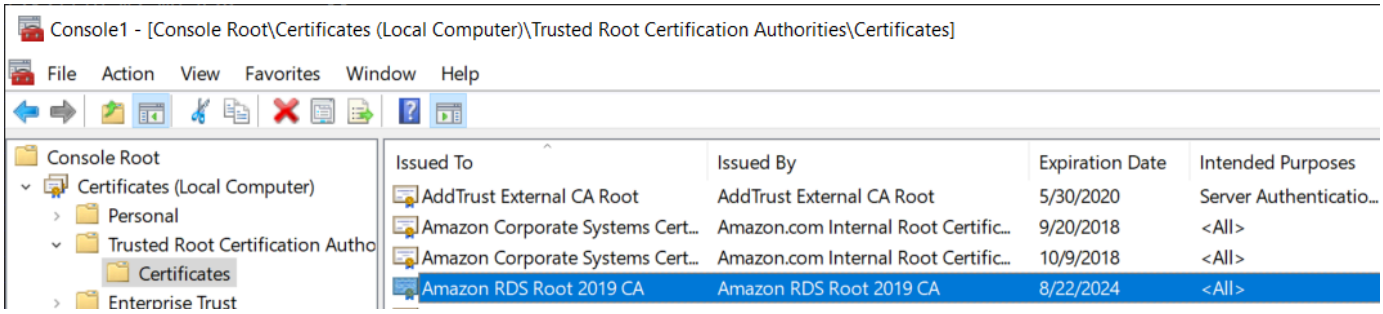
Nach dem Herunterladen des entsprechenden Zertifikats importieren Sie es anhand der Vorgehensweise im folgenden Abschnitt in Ihr Microsoft Windows-Betriebssystem.

Importieren von Zertifikaten auf Client-Computer

Sie können die folgende Vorgehensweise verwenden, um Ihr Zertifikat in das Microsoft Windows-Betriebssystem auf Ihrem Client-Computer zu importieren.

So importieren Sie das Zertifikat in Ihr Windows-Betriebssystem:

1. Geben Sie im Menü Start den Befehl **Run** in das Suchfeld ein und drücken Sie die Eingabetaste.
2. Geben Sie in das Feld Öffnen **MMC** ein und wählen Sie dann OK.
3. Wählen Sie in der MMC Konsole im Menü Datei die Option Snap-In hinzufügen/entfernen.
4. Wählen Sie im Dialogfeld Add or Remove Snap-ins (Snap-ins hinzufügen oder entfernen) für Available snap-ins (Verfügbare Snap-ins) die Option **Certificates** und dann Hinzufügen aus.
5. Wählen Sie im Dialogfeld Certificates snap-in (Snap-In-Zertifikate) die Option Computer account (Computerkonto) aus und klicken Sie anschließend auf Weiter.
6. Wählen Sie im Dialogfeld Select Computer (Computer auswählen) die Option Beenden aus.
7. Wählen Sie im Dialogfeld Add or Remove Snap-ins (Snap-ins hinzufügen oder entfernen) OK aus.
8. Erweitern Sie in der MMC Konsole Zertifikate, öffnen Sie das Kontextmenü (mit der rechten Maustaste) für Vertrauenswürdige Stammzertifizierungsstellen, wählen Sie Alle Aufgaben und dann Importieren aus.
9. Wählen Sie auf der ersten Seite des Assistenten zum Importieren von Zertifikaten Weiter.
10. Wählen Sie auf der zweiten Seite des Assistenten zum Importieren von Zertifikaten Durchsuchen. Ändern Sie im Suchfenster den Dateityp auf All files (*.*) (Alle Dateien (*.*)), da PEM keine standardmäßige Zertifikatserweiterung ist. Suchen Sie die PEM-Datei, die Sie zuvor heruntergeladen haben.
11. Wählen Sie Öffnen, um die Zertifikatdatei auszuwählen und wählen Sie dann Weiter.
12. Wählen Sie auf der dritten Seite des Assistenten zum Importieren von Zertifikaten Weiter.
13. Wählen Sie auf der vierten Seite des Assistenten zum Importieren von Zertifikaten Beenden. Ein Dialogfeld zeigt an, dass der Import erfolgreich war.
14. Erweitern Sie in der MMC Konsole Zertifikate, erweitern Sie Vertrauenswürdige Stammzertifizierungsstellen und wählen Sie dann Zertifikate aus. Suchen Sie das Zertifikat, um zu bestätigen, dass es existiert, wie hier gezeigt.



Verschlüsseln von Verbindungen zu einer RDS Amazon-DB-Instance, auf der Microsoft SQL Server ausgeführt wird

Nachdem Sie ein Zertifikat in Ihren Client-Computer importiert haben, können Sie Verbindungen vom Client-Computer zu einer RDS Amazon-DB-Instance verschlüsseln, auf der Microsoft SQL Server ausgeführt wird.

Verwenden Sie für SQL Server Management Studio das folgende Verfahren. Weitere Informationen zu SQL Server Management Studio finden Sie unter Verwenden von SQL Server Management Studio.

Um Verbindungen von SQL Server Management Studio aus zu verschlüsseln

1. Starten Sie SQL Server Management Studio.
2. Geben Sie für Connect to server (Mit Server verbinden) die Serverinformationen, den Anmeldebenutzernamen und das Passwort ein.
3. Wählen Sie Optionen aus.
4. Wählen Sie Encrypt connection (Verbindungen verschlüsseln) aus.
5. Wählen Sie Connect (Verbinden) aus.
6. Prüfen Sie, ob Ihre Verbindung verschlüsselt ist, indem Sie die folgende Abfrage ausführen. Die Abfrage muss `true` für `encrypt_option` zurückgeben.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Verwenden Sie für jeden anderen SQL Client das folgende Verfahren.

Um Verbindungen von anderen SQL Clients zu verschlüsseln

1. Fügen Sie `encrypt=true` an Ihre Verbindungszeichenfolge an. Diese Zeichenfolge ist möglicherweise als Option oder als Eigenschaft auf der Verbindungsseite in GUI Tools verfügbar.

> **Note**
>
> Um die SSL Verschlüsselung für Clients zu aktivieren, die eine Verbindung herstellenJDBC, müssen Sie möglicherweise das RDS SQL Amazon-Zertifikat zum Java CA-Zertifikatsspeicher (cacerts) hinzufügen. Dies erledigen Sie mithilfe des Dienstprogramms Keytool.

2. Prüfen Sie, ob Ihre Verbindung verschlüsselt ist, indem Sie die folgende Abfrage ausführen. Die Abfrage muss `true` für `encrypt_option` zurückgeben.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Konfiguration von SQL Serversicherheitsprotokollen und Chiffren

Sie können bestimmte Sicherheitsprotokolle und Verschlüsselungen mithilfe von DB-Parametern ein- und ausschalten. Die Sicherheitsparameter, die Sie konfigurieren können (mit Ausnahme von TLS Version 1.2), sind in der folgenden Tabelle aufgeführt.

DB-Parameter	Zulässige Werte (Standard wert in Fettdruck)	Beschreibung
rds.tls10	Standard, aktiviert, deaktiviert	TLS1.0.
rds.tls11	Standard, aktiviert, deaktiviert	TLS1.1.
rds.tls12	default	TLS1.2. Dieser Wert kann nicht verändert werden.
rds.fips	0, 1	Wenn Sie den Parameter auf 1 setzen, wird die Verwendung von Modulen RDS erzwungen , die dem Standard 140-2 des Federal Information Processin g Standard (FIPS) entsprech en. Weitere Informationen finden Sie in der Microsoft-Dokument ation unter Verwenden von SQL Server 2016 im FIPS 140-2-kompatiblen Modus.
rds.rc4	Standard, aktiviert, deaktiviert	RC4Stream-Chiffre.
rds.diffie-hellman	Standard, aktiviert, deaktiviert	Diffie-Hellman-Schlüsselaus tausch-Verschlüsselung.
rds. diffie-hellman-min-key-Bit-Länge	Standard, 1024, 2048, 3072, 4096	Minimale Bitlänge für Diffie-He llman-Schlüssel.

DB-Parameter	Zulässige Werte (Standard wert in Fettdruck)	Beschreibung
rds.curve25519	Standard, aktiviert, deaktiviert	Curve25519 Verschlüs selungsverfahren mit elliptisc her Kurve. Dieser Parameter wird nicht für alle Engine-Ve rsionen unterstützt.
rds.3des168	Standard, aktiviert, deaktiviert	Verschlüsselungschiffre nach dem Triple Data Encryptio n Standard (DES) mit einer Schlüssellänge von 168 Bit.

> **Note**
>
> Für kleinere Engine-Versionen nach 16.00.4120.1, 15.00.4365.2, 14.00.3465.1, 13.00.6435.1 und 12.00.6449.1 ist die Standardeinstellung für die DB-Parameter,,, und deaktiviert. *`rds.tls10` `rds.tls11` `rds.rc4` `rds.curve25519` `rds.3des168`* Andernfalls ist die Standardeinstellung aktiviert.
> Für kleinere Engine-Versionen nach 16.00.4120.1, 15.00.4365.2, 14.00.3465.1, 13.00.6435.1 und 12.00.6449.1 ist die Standardeinstellung für 3072. `rds.diffie-hellman-min-key-bit-length` Andernfalls ist die Standardeinstellung 2048.

Gehen Sie wie folgt vor, um die Sicherheitsprotokolle und Verschlüsselungen zu konfigurieren:

1. Erstellen Sie eine benutzerdefinierte DB-Parametergruppe.
2. Ändern Sie die Parameter in der Parametergruppe.
3. Ordnen Sie die neue DB-Parametergruppe der DB-Instance zu.

Weitere Informationen zu DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Erstellen der sicherheitsbezogenen Parametergruppe

Erstellen Sie eine Parametergruppe für Ihre sicherheitsrelevanten Parameter, die der SQL Serveredition und Version Ihrer DB-Instance entspricht.

Konsole

Das folgende Verfahren erstellt eine Parametergruppe für SQL Server Standard Edition 2016.

So erstellen Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie Create parameter group (Parametergruppe erstellen).
4. Führen Sie im Bereich Parametergruppe erstellen die folgenden Schritte aus:
   a. Wählen Sie für Parametergruppenfamilie die Option sqlserver-se-13.0 aus.
   b. Geben Sie unter Gruppenname einen Bezeichner für die Parametergruppe ein, z. B. **`sqlserver-ciphers-se-13`**.
   c. Geben Sie für Beschreibung den Text **`Parameter group for security protocols and ciphers`** ein.
5. Wählen Sie Create (Erstellen) aus.

CLI

Mit dem folgenden Verfahren wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
```

```
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Parameter group for security protocols and ciphers"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Parameter group for security protocols and ciphers"
```

Ändern von sicherheitsbezogenen Parametern

Ändern Sie die sicherheitsrelevanten Parameter in der Parametergruppe, die der SQL Serveredition und Version Ihrer DB-Instance entspricht.

Konsole

Das folgende Verfahren ändert die Parametergruppe, die Sie für SQL Server Standard Edition 2016 erstellt haben. In diesem Beispiel wird TLS Version 1.0 deaktiviert.

So ändern Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie die Parametergruppe aus, z. B. sqlserver-ciphers-se-13.
4. Filtern Sie unter Parameter die Parameterliste nach **rds**.
5. Wählen Sie Parameter bearbeiten aus.
6. Wählen Sie rds.tls10 aus.
7. Wählen Sie unter Werte die Option deaktiviert aus.
8. Wählen Sie Änderungen speichern.

CLI

Mit dem folgenden Verfahren wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben. In diesem Beispiel wird TLS Version 1.0 deaktiviert.

So ändern Sie die Parametergruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --parameters
 "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --parameters
 "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

Zuordnen der sicherheitsbezogenen Parametergruppe zu Ihrer DB-Instance

Um die Parametergruppe mit Ihrer DB-Instance zu verknüpfen, verwenden Sie den AWS Management Console oder den AWS CLI.

Konsole

Sie können die Parametergruppe einer neuen oder vorhandenen DB-Instance zuordnen:

- Bei einer neuen DB-Instance ordnen Sie diese zu, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Bei einer vorhandenen DB-Instance ordnen Sie diese zu, indem Sie die Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

CLI

Sie können die Parametergruppe einer neuen oder einer vorhandenen DB-Instance zuordnen.

So erstellen Sie eine DB-Instance mit der Parametergruppe

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Parametergruppe verwendet haben.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --master-user-password secret123 \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --db-parameter-group-name sqlserver-ciphers-se-13
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --master-user-password secret123 ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name sqlserver-ciphers-se-13
```

  > **Note**
  >
  > Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

So ändern Sie eine DB-Instance und ordnen die Parametergruppe zu

- Führen Sie einen der folgenden Befehle aus.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --apply-immediately
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --apply-immediately
```

Aktualisierung von Anwendungen für die Verbindung mit Microsoft SQL Server-DB-Instances mithilfe neuer SSL TLS /-Zertifikate

Seit dem 13. Januar 2023 RDS hat Amazon neue Certificate Authority (CA) -Zertifikate für die Verbindung zu Ihren RDS DB-Instances über Secure Socket Layer oder Transport Layer Security (SSL/TLS) veröffentlicht. Im Folgenden finden Sie Informationen dazu, wie Sie Ihre Anwendungen aktualisieren, um die neuen Zertifikate verwenden zu können.

Anhand dieses Themas können Sie feststellen, ob Client-AnwendungenSSL/verwenden, TLS um eine Verbindung zu Ihren DB-Instances herzustellen. Wenn dies der Fall ist, können Sie weiter überprüfen, ob diese Anwendungen zur Herstellung von Verbindungen Zertifikatverifizierungen erfordern.

> **Note**
>
> Einige Anwendungen sind so konfiguriert, dass sie nur dann eine Verbindung zu SQL Server-DB-Instances herstellen, wenn sie das Zertifikat auf dem Server erfolgreich verifizieren können.
> Für solche Anwendungen müssen Sie die Trust Stores Ihrer Client-Anwendung aktualisieren, damit diese die neuen CA-Zertifikate enthalten.

Nach der Aktualisierung der CA-Zertifikate in den Trust Stores Ihrer Client-Anwendung können Sie die Zertifikate auf Ihren DB-Instances rotieren. Es wird nachdrücklich empfohlen, diese Verfahren vor der Implementierung in Produktionsumgebungen in einer Entwicklungs- oder Testumgebung zu testen.

Weitere Informationen zur Zertifikatrotation finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate. Weitere Informationen zum Herunterladen von Zertifikaten finden Sie unter . Hinweise zur Verwendung vonSSL/TLSmit Microsoft SQL Server-DB-Instances finden Sie unterVerwendung SSL mit einer Microsoft SQL Server-DB-Instance.

Themen

- Ermitteln, ob Anwendungen eine Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen mithilfe SSL
- Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert
- Aktualisieren des Trust Stores Ihrer Anwendung

Ermitteln, ob Anwendungen eine Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen mithilfe SSL

Prüfen Sie die DB-Instance-Konfiguration auf den Wert des Parameters `rds.force_ssl`. Standardmäßig ist der Parameter `rds.force_ssl` auf 0 (aus) festgelegt. Wenn der `rds.force_ssl` Parameter auf 1 (on) gesetzt ist, müssen ClientsSSL/TLSfür Verbindungen verwenden. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Führen Sie die folgende Abfrage durch, um die aktuelle Verschlüsselungsoption für alle offenen Verbindungen zu einer DB-Instance zu erhalten. Die Spalte `ENCRYPT_OPTION` zeigt `TRUE`, wenn die Verbindung verschlüsselt ist.

```
select SESSION_ID,
    ENCRYPT_OPTION,
    NET_TRANSPORT,
    AUTH_SCHEME
    from SYS.DM_EXEC_CONNECTIONS
```

Diese Abfrage zeigt nur die aktuellen Verbindungen. Es wird nicht angezeigt, ob Anwendungen, die in der Vergangenheit eine Verbindung hergestellt oder getrennt haben, diese verwendet SSL haben.

Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert

Sie können überprüfen, ob verschiedene Arten von Clients zum Herstellen von Verbindungen Zertifikatverifizierungen erfordern.

> **Note**
>
> Wenn Sie andere als die aufgeführten Konnektoren verwenden, konsultieren Sie deren Dokumentation für Informationen zur Durchsetzung verschlüsselter Verbindungen. Weitere Informationen finden Sie unter Verbindungsmodule für SQL Microsoft-Datenbanken in der Microsoft SQL Server-Dokumentation.

SQLServer Management Studio

Überprüfen Sie, ob die Verschlüsselung für SQL Server Management Studio-Verbindungen erzwungen wird:

1. Starten Sie SQL Server Management Studio.
2. Geben Sie für Connect to server (Mit Server verbinden) die Serverinformationen, den Anmeldebenutzernamen und das Passwort ein.
3. Wählen Sie Optionen aus.
4. Prüfen Sie, ob auf der Verbindungsseite Encrypt connection (Verbindung verschlüsseln) ausgewählt ist.

Weitere Informationen zu SQL Server Management Studio finden Sie unter Verwenden von SQL Server Management Studio.

Sqlcmd

Das folgende Beispiel mit dem `sqlcmd` Client zeigt, wie die SQL Serververbindung eines Skripts überprüft wird, um festzustellen, ob erfolgreiche Verbindungen ein gültiges Zertifikat erfordern. Weitere Informationen finden Sie unter Connecting with sqlcmd in der Microsoft SQL Server-Dokumentation.

Bei der Verwendung muss eine SSL Verbindung anhand des Serverzertifikats überprüft werden`sqlcmd`, wenn Sie das `-N` Befehlsargument verwenden, um Verbindungen zu verschlüsseln, wie im folgenden Beispiel gezeigt.

```
$ sqlcmd -N -S dbinstance.rds.amazon.com -d ExampleDB
```

> **Note**
>
> Wenn `sqlcmd` mit der Option `-C` aufgerufen wird, wird dem Serverzertifikat vertraut, auch wenn dieses nicht mit dem clientseitigen Trust Store übereinstimmt.

ADO.NET

Im folgenden Beispiel stellt die Anwendung mithilfe des Serverzertifikats eine Verbindung herSSL, und das Serverzertifikat muss verifiziert werden.

```
using SQLC = Microsoft.Data.SqlClient;

...

    static public void Main()
    {
        using (var connection = new SQLC.SqlConnection(
            "Server=tcp:dbinstance.rds.amazon.com;" +
            "Database=ExampleDB;User ID=LOGIN_NAME;" +
            "Password=YOUR_PASSWORD;" +
            "Encrypt=True;TrustServerCertificate=False;"
            ))
        {
            connection.Open();
            ...
        }
```

Java

Im folgenden Beispiel stellt die Anwendung SSL eine Verbindung her und das Serverzertifikat muss verifiziert werden.

```
String connectionUrl =
    "jdbc:sqlserver://dbinstance.rds.amazon.com;" +
    "databaseName=ExampleDB;integratedSecurity=true;" +
    "encrypt=true;trustServerCertificate=false";
```

Um die SSL Verschlüsselung für Clients zu aktivierenJDBC, die eine Verbindung herstellen, müssen Sie möglicherweise das RDS Amazon-Zertifikat zum Java CA-Zertifikatsspeicher hinzufügen. Anweisungen finden Sie in der Microsoft SQL Server-Dokumentation unter Konfiguration des Clients für die Verschlüsselung. Sie können den Namen der Datei für das vertrauenswürdige CA-Zertifikat auch direkt bereitstellen, indem Sie der Verbindungszeichenfolge `trustStore=`*`path-to-certificate-trust-store-file`* anfügen.

> **Note**
>
> Wenn Sie in der Verbindungszeichenfolge `TrustServerCertificate=true` (oder entsprechend) verwenden, wird im Verbindungsvorgang die Vertrauenskettenvalidierung übersprungen. In diesem Fall stellt die Anwendung Verbindungen her, auch wenn das Zertifikat nicht verifiziert werden kann. Die Verwendung von `TrustServerCertificate=false` erzwingt die Zertifikatvalidierung und stellt eine bewährte Methode dar.

Aktualisieren des Trust Stores Ihrer Anwendung

Sie können den Trust Store für Anwendungen aktualisieren, die Microsoft SQL Server verwenden. Detaillierte Anweisungen finden Sie unter Verschlüsseln spezifischer Verbindungen. Weitere Informationen finden Sie in der Microsoft SQL Server-Dokumentation unter Konfiguration des Clients für die Verschlüsselung.

Wenn Sie ein anderes Betriebssystem als Microsoft Windows verwenden, finden Sie in der Softwareverteilungsdokumentation fürSSL/TLSimplementation Informationen zum Hinzufügen eines neuen Stammzertifizierungsstellenzertifikats. Beispielsweise TLS sind Open SSL und Gnu beliebte Optionen. Verwenden Sie die Implementierungsmethode, um dem RDS Root-CA-Zertifikat Vertrauen zu verleihen. Microsoft stellt Anweisungen zum Konfigurieren von Zertifikaten auf bestimmten Systemen bereit.

Informationen zum Herunterladen des Stammverzeichnisses finden Sie unter .

Beispiele für Skripte, die Zertifikate importieren, finden Sie unter Beispielskript für den Import von Zertifikaten in Ihren Trust Store.

> **Note**
>
> Wenn Sie den Trust Store aktualisieren, können Sie ältere Zertifikate beibehalten und die neuen Zertifikate einfach hinzufügen.

Unterstützung zu Compliance-Programmen für Microsoft SQL Server-DB-Instances

AWS Die im Leistungsumfang enthaltenen Dienstleistungen wurden von einem externen Prüfer umfassend bewertet und führen zu einer Zertifizierung, Konformitätsbescheinigung oder Authority to Operate (ATO). Weitere Informationen finden Sie unterAWS -Services im Rahmen des Compliance-Programms.

HIPAA-Unterstützung für Microsoft SQL Server-DB-Instances

Sie können Amazon RDS for Microsoft SQL Server-Datenbanken für die Erstellung von HIPAA-kompatiblen Anwendungen verwenden. Mit können Sie Gesundheitsdaten, darunter geschützte patientenbezogene Daten (protected health information, PHI), im Rahmen eines Geschäftspartnervertrags (Business Associate Agreement, BAA) speicher AWS. Weitere Informationen finden Sie unter HIPAA-Compliance.

Amazon RDS for SQL Server unterstützt HIPAA für folgende Versionen und Editions:

- SQL Server 2022 Enterprise, Standard und Web Editionen
- SQL Server 2019 Enterprise-, Standard- und Web-Editionen
- SQL Server 2017 Enterprise-, Standard- und Web-Editionen
- SQL Server 2016 Enterprise-, Standard- und Web-Editionen

Um die HIPAA-Unterstützung auf Ihrer DB-Instance zu aktivieren, richten Sie folgende drei Komponenten ein.

Komponente	Details
Prüfung	Um eine Prüfung einzurichten, legen Sie den Parameter `rds.sqlserver_audit` auf den Wert `fedramp_hipaa` fest. Wenn Ihre DB-Instance noch keine benutzerdefinierte DB-Parametergruppe verwendet, müssen Sie eine benutzerdefinierte Parametergruppe erstellen und diese an Ihre DB-Instance anfügen, bevor Sie den Parameter `rds.sqlserver_audit` bearbeiten. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

Komponente	Details
Transport verschlüsselung	Um eine Transportverschlüsselung einzurichten, müssen alle Verbindungen mit Ihrer DB-Instance über Secure Sockets Layer (SSL) erfolgen. Weitere Informationen finden Sie unter Erzwingen Sie die Verwendung von Verbindungen zu Ihrer DB-Instance SSL.
Verschlüsselung im Ruhezustand	Es gibt zwei Möglichkeiten, die Verschlüsselung im Ruhezustand einzurichten: 1. Wenn Sie SQL Server 2016—2022 Enterprise Edition oder 2022 Standard Edition ausführen, können Sie Transparent Data Encryption (TDE) verwenden, um eine Verschlüsselung im Ruhezustand zu erreichen . Weitere Informationen finden Sie unter Support für transparente Datenverschlüsselung im SQL Server. 2. Sie können die Verschlüsselung im Ruhezustand mithilfe von AWS Key Management Service (AWS KMS) Verschlüsselungsschlüsseln einrichten. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.

Microsoft SQL Server-Versionen auf Amazon RDS

Sie können eine beliebige aktuell unterstützte Microsoft SQL Server-Version festlegen, wenn Sie eine DB-Instance erstellen. Sie können die Microsoft SQL Server-Hauptversionen (wie z. B. Microsoft SQL Server 14.00) sowie eine beliebige Unterversion für die angegebene Hauptversion festlegen. Wenn keine Version angegeben wird, verwendet Amazon RDS standardmäßig eine unterstützte Version - in der Regel die aktuelle Version. Wenn die Hauptversion, jedoch nicht die Unterversion, festgelegt ist, verwendet Amazon RDS standardmäßig den letzten Release der Hauptversion, die Sie festgelegt haben.

Die folgende Tabelle zeigt die unterstützten SQL Server-Versionen für alle Editionen und alle AWS Regionen, sofern nicht anders angegeben.

> **Note**
>
> Sie können den describe-db-engine-versions AWS CLI Befehl auch verwenden, um eine Liste der unterstützten Versionen sowie die Standardeinstellungen für neu erstellte DB-Instances anzuzeigen. Sie können die Hauptversionen Ihrer SQL Server-Datenbanken anzeigen, indem Sie den AWS CLI Befehl describe-db-major-engine-versions ausführen oder den API-Vorgang Describe DBMajor EngineVersions RDS verwenden.

Hauptversion	Unterversion	RDS API **`EngineVersion`** und CLI **`engine-version`**
SQL Server 2022	16.00.4205.1 (0) CU2	`16.00.4205.1.v1`
	16,00,4195,2 CU19 ()	`16.00.4195.2.v1`
	16,00,4185,3 () CU18	`16.00.4185.3.v1`
	16,00,4175,1 () CU17	`16.00.4175.1.v1`
	16,00,4165,4 () CU16	`16.00.4165.4.v1`
	16,00,4150,1 () CU15	`16.00.4150.1.v1`
	16,00.4140,3 (DDRCU14)	`16.00.4140.3.v1`
	16,00.4135,4 (CU14)	`16.00.4135.4.v1`
	16,00,4131,2 () CU13	`16.00.4131.2.v1`
	16,00,4125,3 () CU13	`16.00.4125.3.v1`
	16,00.4120.1 (DDRCU12)	`16.00.4120.1.v1`
	16,00.4115,5 (CU12)	`16.00.4115.5.v1`
	16,00,4105,2 () CU11	`16.00.4105.2.v1`
	16,00,4095,4 (0) CU1	`16.00.4095.4.v1`
	16,00,4085,2 (CU9)	

Hauptversion	Unterversion	RDS API **EngineVersion** und CLI **engine-version**
		16.00.4085.2.v1

Hauptversion	Unterversion	RDS API **EngineVersion** und CLI **engine-version**
SQL Server 2019	15,00,4435,7 () CU32	15.00.4435.7.v1
	15,00,4430,1 () CU32	15.00.4430.1.v1
	15,00,4420,2 () CU31	15.00.4420.2.v1
	15,00,4415,2 (0CU3)	15.00.4415.2.v1
	15,00.4410,1 (CU29 DDR)	15.00.4410.1.v1
	15,00.4395,2 CU28 ()	15.00.4395.2.v1
	15,00,4390,2 () CU28	15.00.4390.2.v1
	15,00,4385,2 () CU28	15.00.4385.2.v1
	15,00,4382,1 () CU27	15.00.4382.1.v1
	15,00,4375,4 () CU27	15.00.4375.4.v1
	15,00,4365,2 () CU26	15.00.4365.2.v1
	15,00,4355,3 () CU25	15.00.4355.3.v1
	15,00,4345,5 () CU24	15.00.4345.5.v1
	15,00,4335,1 () CU23	15.00.4335.1.v1
	15,00,4322,2 () CU22	15.00.4322.2.v1
	15,00,4316,3 () CU21	15.00.4316.3.v1
	15,00,4312,2 (0CU2)	15.00.4312.2.v1
	15,00,4236,7 (CU16)	15.00.4236.7.v1
	15,00,4198,2 () CU15	15.00.4198.2.v1
	15,00,4153,1 () CU12	15.00.4153.1.v1

Hauptversion	Unterversion	RDS API **EngineVersion** und CLI **engine-version**
	15,00,4073,23 () CU8	15.00.4073.23.v1
	15,00,4043,16 () CU5	15.00.4043.16.v1
SQL Server 2017	14.00.3495,9 (DDRCU31)	14.00.3495.9.v1
	14,00.3485,1 (CU31 DDR)	14.00.3485.1.v1
	14,00.3480,1 (CU31)	14.00.3480.1.v1
	14,00.3475,1 () CU31	14.00.3475.1.v1
	14,00.3471,2 () CU31	14.00.3471.2.v1
	14,00.3465,1 () CU31	14.00.3465.1.v1
	14,00,3460,9 () CU31	14.00.3460.9.v1
	14,00,3451,2 (0CU3)	14.00.3451.2.v1
	CU2714,00.3421,10 ()	14.00.3421.10.v1
	14,00.3401,7 () CU25	14.00.3401.7.v1
	14,00,3381,3 (CU23)	14.00.3381.3.v1
	14,00.3356,20 () CU22	14.00.3356.20.v1
	14,00,3294,2 (0CU2)	14.00.3294.2.v1
	14,00,3281,6 (CU19)	14.00.3281.6.v1

Hauptversion	Unterversion	RDS API **`EngineVersion`** und CLI **`engine-version`**
SQL Server 2016	13,00.7055,9 (DDR)	`13.00.7055.9.v1`
	13.00.6455.2 (DDR)	`13.00.6455.2.v1`
	13.00.6450.1 (DDR)	`13.00.6450.1.v1`
	13,00.6445,1 (DDR)	`13.00.6445.1.v1`
	13,00.6441,1 (DDR)	`13.00.6441.1.v1`
	13.00.6435.1 (GDR)	`13.00.6435.1.v1`
	13.00.6430.49 (GDR)	`13.00.6430.49.v1`
	13.00.6419.1 (+ Hotfix) SP3	`13.00.6419.1.v1`
	13.00.6300.2 SP3 ()	`13.00.6300.2.v1`

Versionsverwaltung in Amazon RDS

Amazon RDS beinhaltet flexibles Versionsmanagement, mit dem Sie steuern können, wann und wie die Ihre DB-Instance gepatcht oder aktualisiert wird. Dadurch können Sie Folgendes für Ihre DB-Engine durchführen:

- Aufrechterhalten der Kompatibilität mit Datenbank-Engine-Patch-Versionen.
- Testen neuer Patch-Versionen, um zu prüfen, ob Sie mit Ihrer Anwendung funktionieren, bevor Sie sie in der Produktion bereitstellen.
- Planen und Ausführen von Versions-Upgrades, um Ihre Service Level Agreements und zeitlichen Anforderungen zu erfüllen.

Microsoft SQL Server-Engine-Patching in Amazon RDS

Amazon RDS aggregiert regelmäßig offizielle Microsoft SQL Server-Datenbank-Patches in einer Version der DB-Instance-Engine, die für Amazon RDS spezifisch ist. Weitere Informationen

über Microsoft SQL Server-Patches in jeder Version der Engine finden Sie unter Versions- und Funktionsunterstützung in Amazon RDS.

Derzeit können Sie alle Engine-Upgrades auf Ihrer DB-Instance manuell durchführen. Weitere Informationen finden Sie unter Upgrades der Microsoft SQL Server-DB-Engine.

Einstellungszeitplan für Engine-Hauptversionen von Microsoft SQL Server auf Amazon RDS

Die folgende Tabelle zeigt den geplanten Einstellungszeitplan für Engine-Hauptversionen von Microsoft SQL Server an.

Datum	Informationen
9. Juli 2024	Microsoft stellt kritische Patch-Updates für SQL Server 2014 ein. Weitere Information Microsoft SQL Server 2014 in der Microsoft-Dokumentation.
1. Juni 2024	Amazon RDS plant, die Unterstützung von Microsoft SQL Server 2014 in RDS für SQ Zu diesem Zeitpunkt wird geplant, dass alle verbleibenden Instances zu SQL Server (neueste verfügbare Unterversion). Weitere Informationen finden Sie unter Ankündigu zung für die Hauptversionen von Amazon RDS für SQL Server 2014. Um ein automatisches Upgrade von Microsoft SQL Server 2014 zu vermeiden, könne zu einem für Sie geeigneten Zeitpunkt durchführen. Weitere Informationen finden Sie Engine-Version für eine DB-Instance.
12. Juli 2022	Microsoft stellt kritische Patch-Updates für SQL Server 2012 ein. Weitere Information Microsoft SQL Server 2012 in der Microsoft-Dokumentation.
1. Juni 2022	Amazon RDS plant, die Unterstützung von Microsoft SQL Server 2012 auf RDS for S Zu diesem Zeitpunkt wird geplant, dass alle verbleibenden Instances auf SQL Server (neueste verfügbare Unterversion). Weitere Informationen finden Sie unter Ankündigu zung für die Hauptversionen von Amazon RDS for SQL Server 2012. Um ein automatisches Upgrade von Microsoft SQL Server 2012 zu vermeiden, könne geeigneten Zeitpunkt upgraden. Weitere Informationen finden Sie unter Upgrade der DB-Instance.

Datum	Informationen
1. September 2021	Amazon RDS beginnt, die Erstellung neuer RDS for SQL Server DB-Instances mithilf Server 2012 zu deaktivieren. Weitere Informationen finden Sie unter Ankündigung: E für die Hauptversionen von Amazon RDS for SQL Server 2012.
12. Juli 2019	Das Amazon RDS-Team hat die Unterstützung für Microsoft SQL Server 2008 R2 im Verbleibende Instances mit Microsoft SQL Server 2008 R2 migrieren auf SQL Server Version verfügbar). Um ein automatisches Upgrade von Microsoft SQL Server 2008 R2 zu vermeiden, kö Sie geeigneten Zeitpunkt upgraden. Weitere Informationen finden Sie unter Upgrade eine DB-Instance.
25. April 2019	Vor Ende April 2019 können Sie keine neuen Amazon RDS for SQL Server-Datenbar erstellen, die SQL Server 2008 R2 verwenden.

Funktionen von Microsoft SQL Server bei Amazon RDS

Die unterstützten SQL Serverversionen bei Amazon RDS beinhalten die folgenden Funktionen. Im Allgemeinen enthält eine Version auch Funktionen aus den vorherigen Versionen, sofern in der Microsoft-Dokumentation nichts anderes angegeben ist.

Themen

- Funktionen von Microsoft SQL Server 2022
- Funktionen von Microsoft SQL Server 2019
- Funktionen von Microsoft SQL Server 2017
- Funktionen von Microsoft SQL Server 2016
- Ende des Supports für Microsoft SQL Server 2014 bei Amazon RDS
- Ende des Supports für Microsoft SQL Server 2012 bei Amazon RDS
- Ende des Supports für Microsoft SQL Server 2008 R2 bei Amazon RDS
- Unterstützung für Change Data Capture für Microsoft SQL Server-DB-Instances
- Nicht unterstützte Funktionen und Funktionen mit beschränkter Unterstützung

Funktionen von Microsoft SQL Server 2022

SQLServer 2022 enthält viele neue Funktionen, wie zum Beispiel die folgenden:

- Parameterabhängige Planoptimierung — ermöglicht mehrere zwischengespeicherte Pläne für eine einzelne parametrisierte Anweisung, wodurch möglicherweise Probleme beim Parameter-Sniffing reduziert werden.
- SQLServer Ledger — bietet die Möglichkeit, kryptografisch nachzuweisen, dass Ihre Daten nicht unbefugt verändert wurden.
- Sofortige Dateiinitialisierung bei Wachstumsereignissen in Transaktionsprotokolldateien — führt zu einer schnelleren Ausführung von Protokollwachstumsereignissen bis zu 64 MB, auch für Datenbanken mit aktivierter Option. TDE
- Verbesserungen bei der Parallelität von Systemseitenverriegelungen — reduziert Konflikte beim Zuweisen und Aufheben der Zuweisung von Datenseiten und Datenausdehnungen und sorgt so für erhebliche Leistungsverbesserungen bei hohen Workloads. `tempdb`

Die vollständige Liste der SQL Server 2022-Features finden Sie in der Microsoft-Dokumentation unter Was ist neu in SQL Server 2022 (16.x).

Eine Liste der nicht unterstützten Funktionen finden Sie unter Nicht unterstützte Funktionen und Funktionen mit beschränkter Unterstützung.

Funktionen von Microsoft SQL Server 2019

SQLServer 2019 enthält viele neue Funktionen, darunter die folgenden:

- Beschleunigte Datenbankwiederherstellung (ADR) — Verkürzt die Zeit für die Wiederherstellung nach einem Absturz nach einem Neustart oder einem lang andauernden Transaktions-Rollback.
- Intelligente Abfrageverarbeitung ()IQP:
  - Berechtigungs-Feedback im Zeilenmodus – korrigiert übermäßige Berechtigungen, die andernfalls zu verschwendetem Speicher und reduzierter Parallelität führen würden, automatisch.
  - Stapelmodus im Zeilenspeicher – aktiviert die Ausführung des Stapelmodus für analytische Workloads ohne Spaltenspeicher-Indizes.
  - Durch Tabellenvariable verzögerte Kompilierung – verbessert die Planqualität und die Gesamtleistung für Abfragen, bei denen Tabellenvariablen verwiesen werden.

- Intelligente Leistung:
  - `OPTIMIZE_FOR_SEQUENTIAL_KEY` Indexoption – verbessert den Durchsatz für Einfügungen mit hoher Parallelität in Indizes.
  - Verbesserte Skalierbarkeit von indirekten Checkpoints — Hilft Datenbanken mit hoher DML Arbeitslast.
  - Gleichzeitige Updates von Page Free Space (PFS) — Ermöglicht die Verarbeitung als gemeinsam genutzter Latch und nicht als exklusiver Latch.
- Überwachen von Verbesserungen:
  - `WAIT_ON_SYNC_STATISTICS_REFRESH` Wait-Typ – zeigt die kumulierte Zeit auf Instance-Ebene an, die mit synchrone Statistikaktualisierungsvorgänge benötigt wird.
  - Konfigurationen des Datenbankbereichs – umfassen `LIGHTWEIGHT_QUERY_PROFILING` und `LAST_QUERY_PLAN_STATS`.
  - Dynamische Verwaltungsfunktionen (DMFs) — Einschließlich und. `sys.dm_exec_query_plan_stats` `sys.dm_db_page_info`
- Ausführliche Kürzungswarnungen – die Fehlermeldung bei Datenkürzung enthält standardmäßig Tabellen- und Spaltennamen sowie den gekürzten Wert.
- Wiederaufnahmefähige Online-Indexerstellung — In SQL Server 2017 wird nur die wiederaufnehmbare Online-Indexerstellung unterstützt.

Die vollständige Liste der SQL Server 2019-Funktionen finden Sie unter Was ist neu in SQL Server 2019 (15.x) in der Microsoft-Dokumentation.

Eine Liste der nicht unterstützten Funktionen finden Sie unter Nicht unterstützte Funktionen und Funktionen mit beschränkter Unterstützung.

Funktionen von Microsoft SQL Server 2017

SQLServer 2017 enthält viele neue Funktionen, wie z. B. die folgenden:

- Adaptive Abfrageverarbeitung
- Automatische Plankorrektur (eine automatische Optimierungsfunktion)
- GraphDB
- Fortsetzbare Index-Umbauten

Die vollständige Liste der SQL Server 2017-Funktionen finden Sie in der Microsoft-Dokumentation unter Was ist neu in SQL Server 2017.

Eine Liste der nicht unterstützten Funktionen finden Sie unter Nicht unterstützte Funktionen und Funktionen mit beschränkter Unterstützung.

Funktionen von Microsoft SQL Server 2016

Amazon RDS unterstützt die folgenden Funktionen von SQL Server 2016:

- „Always Encrypted“
- JSONSupport
- Betriebsanalysen
- Abfragespeicher
- Temporäre Tabellen

Die vollständige Liste der SQL Server 2016-Funktionen finden Sie in der Microsoft-Dokumentation unter Was ist neu in SQL Server 2016.

Ende des Supports für Microsoft SQL Server 2014 bei Amazon RDS

SQLServer 2014 hat das Ende des Supports bei Amazon erreichtRDS.

RDSaktualisiert alle vorhandenen DB-Instances, die noch SQL Server 2014 verwenden, auf die neueste Nebenversion von SQL Server 2016. Weitere Informationen finden Sie unter Versionsverwaltung in Amazon RDS.

Ende des Supports für Microsoft SQL Server 2012 bei Amazon RDS

SQLServer 2012 hat das Ende des Supports bei Amazon erreichtRDS.

RDSaktualisiert alle vorhandenen DB-Instances, die noch SQL Server 2012 verwenden, auf die neueste Nebenversion von SQL Server 2016. Weitere Informationen finden Sie unter Versionsverwaltung in Amazon RDS.

Ende des Supports für Microsoft SQL Server 2008 R2 bei Amazon RDS

SQLServer 2008 R2 hat das Ende des Supports bei Amazon erreichtRDS.

RDSaktualisiert alle vorhandenen DB-Instances, die noch SQL Server 2008 R2 verwenden, auf die neueste Nebenversion von SQL Server 2012. Weitere Informationen finden Sie unter Versionsverwaltung in Amazon RDS.

Unterstützung für Change Data Capture für Microsoft SQL Server-DB-Instances

Amazon RDS unterstützt Change Data Capture (CDC) für Ihre DB-Instances, auf denen Microsoft SQL Server ausgeführt wird. CDCerfasst Änderungen, die an den Daten in Ihren Tabellen vorgenommen wurden, und speichert Metadaten zu jeder Änderung, auf die Sie später zugreifen können. Weitere Informationen finden Sie unter Change Data Capture in der Microsoft-Dokumentation.

Amazon RDS unterstützt CDC die folgenden SQL Server-Editionen und -versionen:

- Microsoft SQL Server Enterprise Edition (Alle Versionen)
- Microsoft SQL Server Standard Edition:
  - 2022
  - 2019
  - 2017
  - Version 2016 SP1 CU2 13.00.4422.0 und später

Für die Verwendung CDC mit Ihren RDS Amazon-DB-Instances müssen Sie die Aktivierung oder Deaktivierung CDC zunächst auf RDS Datenbankebene mithilfe von bereitgestellten gespeicherten Prozeduren vornehmen. Danach kann jeder Benutzer, der die `db_owner` Rolle für diese Datenbank innehat, die systemeigenen gespeicherten Prozeduren von Microsoft CDC zur Steuerung dieser Datenbank verwenden. Weitere Informationen finden Sie unter Verwenden von Change Data Capture für Amazon RDS for SQL Server.

Sie können die fortlaufende Replikation von SQL Server-DB-Instances aus verwenden CDC und AWS Database Migration Service aktivieren.

Nicht unterstützte Funktionen und Funktionen mit beschränkter Unterstützung

Die folgenden Microsoft SQL Server-Funktionen werden auf Amazon nicht unterstütztRDS:

- Sichern in Microsoft Azure Blob Storage
- Erweiterung des Puffer-Pools
- Richtlinien für benutzerdefinierte Kennwörter
- Data Quality Services
- Datenbank-Protokollversand
- Datenbank-Snapshots (Amazon RDS unterstützt nur DB-Instance-Snapshots)
- Erweiterte gespeicherte Prozeduren, einschließlich xp_cmdshell
- FILESTREAM-Support
- Datenbanktabellen
- Machine Learning and R Services (erfordert für die Installation Betriebssystemzugriff)
- Wartungspläne
- Performance-Datenauflister
- Richtlinienbasierte Verwaltung
- PolyBase
- Replikation
- Ressourcenkontrolle

  In einer Umgebung mit mehreren Mandanten empfehlen wir, sich mit den Leistungsanforderungen und Überlegungen vertraut zu machen, um Probleme aufgrund des Konkurrenzkampfs der Workloads um Ressourcen zu minimieren.
- Auslöser auf Serverebene
- Service Broker-Endpunkte
- Stretch-Datenbank
- TRUSTWORTHYDatenbankeigenschaft (erfordert die Rolle Sysadmin)
- SQLT-Endpunkte (alle verwendeten CREATE ENDPOINT Operationen sind nicht verfügbar)
- WCFDatendienste

Die folgenden Microsoft SQL Server-Funktionen werden bei Amazon nur eingeschränkt unterstütztRDS:

- Verteilte Abfragen/Verbindungsserver. Weitere Informationen finden Sie unter Implementieren von Verbindungsservern mit Amazon RDS für Microsoft SQL Server.

- Allgemeine Laufzeitsprache (CLR). On RDS für SQL Server 2016 und niedrigere Versionen, CLR wird nur im SAFE Modus und unter Verwendung von Assembly-Bits unterstützt. CLRwird auf RDS SQL Server 2017 und höheren Versionen nicht unterstützt. Weitere Informationen finden Sie unterIntegration von Common Runtime Languagein der Microsoft-Dokumentation.
- Verknüpfen Sie Server mit Oracle OLEDB in Amazon RDS for SQL Server. Weitere Informationen finden Sie unter Support für mit Oracle OLEDB verknüpfte Server in Amazon RDS für SQL Server.

Die folgenden Funktionen werden auf Amazon RDS mit SQL Server 2022 nicht unterstützt:

- Datenbank für Snapshot sperren
- Externe Datenquelle
- Backup und Wiederherstellung auf S3-kompatiblem Objektspeicher
- Integration des Objektspeichers
- TLS1.3 und TDS MS-8.0
- Offloading der Backup-Komprimierung mit QAT
- SQLServeranalysedienste () SSAS
- Datenbankspiegelung mit Multi-AZ-Bereitstellungen. SQL Server Always On ist die einzige unterstützte Methode bei Multi-AZ-Bereitstellungen.

Multi-AZ-Bereitstellungen, die die Microsoft SQL Server-Datenbankspiegelung oder AlwaysOn-Verfügbarkeitsgruppen verwenden

Amazon RDS unterstützt Multi-AZ-Bereitstellungen für DB-Instances, auf denen Microsoft SQL Server ausgeführt wird, mithilfe von SQL Server Database Mirroring (DBM) oder Always On Availability Groups (). AGs Multi-AZ-Bereitstellungen bieten eine erhöhte Verfügbarkeit, eine längere Lebensdauer von Daten sowie eine höhere Fehlertoleranz für DB-Instances. Im Falle einer geplanten Datenbankwartung oder einer ungeplanten Serviceunterbrechung schaltet Amazon RDS automatisch auf das up-to-date sekundäre Replikat um, sodass der Datenbankbetrieb schnell und ohne manuelles Eingreifen wieder aufgenommen werden kann. Die primären und sekundären Instances verwenden denselben Endpunkt, dessen physische Netzwerkadresse als Teil des Failoverprozesses an das passive sekundäre Replikat übergeben wird. Sie müssen Ihre Anwendung nicht neu konfigurieren, wenn ein Failover auftritt.

Amazon RDS verwaltet das Failover durch aktive Überwachung Ihrer Multi-AZ-Bereitstellung und Initiierung eines Failovers, wenn ein Problem mit Ihrer Primär-Instance auftritt. Failover treten nicht auf, wenn Standby- und Primär-Instance vollständig synchron sind. Amazon RDS verwaltet Ihre Multi-AZ-Bereitstellung aktiv durch automatisches Reparieren instabiler DB-Instances und erneutes Einrichten einer synchronen Replikation. Sie müssen nichts verwalten. Amazon RDS wickelt die Primär-Instance, den Zeugen und die Standby-Instance für Sie ab. Wenn Sie SQL Server Multi-AZ einrichten, konfiguriert RDS passive sekundäre Instances für alle Datenbanken auf der Instance.

Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.

Verwenden von Transparent Data Encryption zur Verschlüsselung ruhender Daten

Amazon RDS unterstützt Microsoft SQL Server Transparent Data Encryption (TDE), die gespeicherte Daten transparent verschlüsselt. Amazon RDS verwendet Optionsgruppen, um diese Funktionen zu aktivieren und zu konfigurieren. Weitere Informationen zur Option TDE finden Sie unter Support für transparente Datenverschlüsselung im SQL Server.

Funktionen und gespeicherte Prozeduren für Amazon RDS for Microsoft SQL Server

Die folgenden Liste enthält Amazon-RDS-Funktionen und gespeicherte Prozeduren, mit denen SQL-Server-Aufgaben automatisiert werden können.

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche
Administrative Aufgaben	`rds_drop_database`	Löschen einer Datenbank in einer Amazon RDS for Microsoft SQL Server Server-DB-Instance
	`rds_failover_time`	Ermitteln der letzten Failover-Zeit für Amazon RDS for Server SQL
	`rds_manage_view_db`	Anzeige von Datenbanknamen für Amazon RDS for SQL Server verweigern oder zulassen

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche
	`_permission`	
	`rds_modify_db_name`	Umbenennen einer Amazon RDS for Microsoft SQL Server Server-Datenbank in einer Multi-AZ-Bereitstellung
	`rds_read_error_log`	Anzeigen von Fehler- und Agent-Protokollen
	`rds_set_configuration`	Diese Operation wird verwendet, um verschiedene DB-Instance-Konfigurationen einzustellen: • Change Data Capture für Multi-AZ-Instances • Festlegen des Aufbewahrungszeitraums für Trace- und Dump-Dateien • Komprimieren von Sicherungsdateien
	`rds_set_database_online`	Umstellung einer Amazon RDS for SQL Server-Datenbank von zu OFFLINE ONLINE

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche

	`rds_set_system_database_sync_objects`
	
	`rds_fn_get_system_database_sync_objects`
	
	`rds_fn_server_object_last_sync_time`
		Aktivieren der Auftragsreplikation von SQL Server Agent

	`rds_show_configuration`	Informationen zu den Werten, die mit `rds_set_configuration` festgelegt werden, finden Sie in den folgenden Themen: • Change Data Capture für Multi-AZ-Instances • Festlegen des Aufbewahrungszeitraums für Trace- und Dump-Dateien
	`rds_shrink_tempdbfile`	Verkleinern der Datenbank tempdb
Erfassung von Datenänderungen (Change Data Capture, CDC)	`rds_cdc_disable_db`	Deaktivieren von CDC

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche
	`rds_cdc_enable_db`	Akitivieren von CDC
Datenbank-E-Mail	`rds_fn_sysmail_allitems`	Anzeigen von Nachrichten, Protokollen und Anhängen
	`rds_fn_sysmail_event_log`	Anzeigen von Nachrichten, Protokollen und Anhängen
	`rds_fn_sysmail_mailattachments`	Anzeigen von Nachrichten, Protokollen und Anhängen
	`rds_sysmail_control`	Diese Operation wird zum Starten und Stoppen der Mail-Warteschlange verwendet: • Starten der Mail-Warteschlange • Stoppen der Mail-Warteschlange
	`rds_sysmail_delete_mailitems_sp`	Löschen von Nachrichten
Native Backups und Wiederherstellungen	`rds_backup_database`	Sichern einer Datenbank
	`rds_cancel_task`	Abbrechen einer Aufgabe

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche
	`rds_finish_restore`	Abschluss einer Datenbankwiederherstellung
	`rds_restore_database`	Wiederherstellen einer Datenbank
	`rds_restore_log`	Wiederherstellen eines Protokolls
Amazon S3-Dateiübertragung	`rds_delete_from_filesystem`	Löschen von Dateien auf der RDS DB-Instance
	`rds_download_from_s3`	Dateien von einem Amazon S3 S3-Bucket auf eine SQL Server-DB-Instance herunterladen
	`rds_gather_file_details`	Dateien auf der RDS DB-Instance auflisten
	`rds_upload_to_s3`	Dateien von einer SQL Server-DB-Instance in einen Amazon S3 S3-Bucket hochladen
Microsoft Distributed Transaction Coordinator (MSDTC)	`rds_msdtc_transaction_tracing`	Verwenden der Transaktionsnachverfolgung
SQL Server Audit	`rds_fn_get_audit_file`	Anzeigen von Audit-Protokollen

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche

Transparente Datenverschlüsselung in	`rds_backup_tde_certificate`
	
	`rds_drop_tde_certificate`
	
	`rds_restore_tde_certificate`
	
	`rds_fn_list_user_tde_certificates`
		Support für transparente Datenverschlüsselung im SQL Server

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche
Microsoft Business Intelligence (MSBI)	`rds_msbi_task`	Diese Operation wird mit SQL Server Analysis Services (SSAS) verwendet: • SSASProjekte auf Amazon bereitstellen RDS • Hinzufügen eines Domänenbenutzers als Datenbankadministrator • Eine SSAS Datenbank sichern • Eine SSAS Datenbank wiederherstellen Diese Operation wird auch mit SQL Server Integration Services (SSIS) verwendet: • Administratorberechtigungen für SSISDB • Bereitstellen eines SSIS-Projekts Diese Operation wird auch mit SQL Server Reporting Services (SSIS) verwendet: • Gewähren des Zugriffs für Domänenbenutzer • Widerrufen von Berechtigungen auf Systemebene
	`rds_fn_task_status`	Dieser Vorgang zeigt den Status von MSBI-Aufgaben an: • SSAS: Überwachen des Status einer Bereitstellungsaufgabe • SSIS: Überwachen des Status einer Bereitstellungsaufgabe • SSRS: Überwachung des Status einer Aufgabe

Aufgabentyp	Prozedur oder Funktion	Verwendungsbereiche
SSIS	`rds_drop_ssis_database`	Löschen der SSISDB Datenbank
	`rds_sqlagent_proxy`	Einen Proxy erstellen SSIS
SSRS	`rds_drop_ssrs_databases`	Löschen der SSRS Datenbanken

Lokale Zeitzone für Microsoft SQL Server-DB-Instances

Die Zeitzone einer RDS Amazon-DB-Instance, auf der Microsoft SQL Server ausgeführt wird, ist standardmäßig festgelegt. Die aktuelle Standardeinstellung ist Coordinated Universal Time (UTC). Sie können die Zeitzone für Ihre DB-Instance stattdessen auf eine lokale Zeitzone einstellen, damit sie mit der Zeitzone Ihrer Anwendungen übereinstimmt.

Sie legen die Zeitzone bei der Erstellung Ihrer DB-Instance fest. Sie können Ihre DB-Instance erstellen AWS Management Console, indem Sie die Amazon RDS API reateDBInstanceC-Aktion oder den AWS CLI create-db-instanceBefehl verwenden.

Wenn Ihre DB-Instance Teil einer Multi-AZ-Bereitstellung ist (mit SQL Server DBM oderAGs), bleibt Ihre Zeitzone beim Failover die von Ihnen festgelegte lokale Zeitzone. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen, die die Microsoft SQL Server-Datenbankspiegelung oder AlwaysOn-Verfügbarkeitsgruppen verwenden.

Wenn Sie eine point-in-time Wiederherstellung anfordern, geben Sie die Uhrzeit für die Wiederherstellung an. Die Uhrzeit wird in Ihrer lokalen Zeitzone angezeigt. Weitere Informationen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Folgende Beschränkungen gelten beim Festlegen der lokalen Zeitzone für Ihre DB-Instance:

- Sie können die Zeitzone einer vorhandenen SQL Server-DB-Instance nicht ändern.
- Sie können einen Snapshot aus einer DB-Instance in einer Zeitzone nicht in eine DB-Instance in einer anderen Zeitzone wiederherstellen.
- Es wird dringend davon abgeraten, eine Sicherungsdatei aus einer Zeitzone für eine andere Zeitzone wiederherzustellen. Wenn Sie eine Sicherungsdatei aus einer Zeitzone in einer anderen Zeitzone wiederherstellen, müssen Sie Ihre Abfragen und Anwendungen auf Auswirkungen durch die Zeitzonenänderung überprüfen. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Unterstützte Zeitzonen

Sie können Ihre lokale Zeitzone auf einen der in der folgenden Tabelle gelisteten Werte einstellen.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Afghanistan Standardzeit	(UTC+ 04:30)	Kabul	Diese Zeitzone berücksichtigt keine Sommerzeit.
Alaska Standardzeit	(UTC— 09:00)	Alaska	
Aleuten Normalzeit	(UTC— 10:00)	Aleuten-Inseln	
Altai Normalzeit	(UTC+ 07:00)	Barnaul, Gorno-Altaisk	
Arabische Normalzeit	(UTC+ 03:00)	Kuwait, Riad	Diese Zeitzone berücksichtigt keine Sommerzeit.
Arabische Standardzeit	(UTC+ 04:00)	Abu Dhabi, Muscat	
Arabische Normalzeit	(UTC+ 03:00)	Bagdad	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Argentinien Normalzeit	(UTC— 03:00)	Buenos Aires Stadt	Diese Zeitzone berücksichtigt keine Sommerzeit.
Astrachan Normalzeit	(UTC+ 04:00)	Astrachan, Uljanowsk	
Atlantik Standardzeit	(UTC— 04:00)	Atlantic Time (Kanada)	
AUSZentrale Standardzeit	(UTC+ 09:30)	Darwin	Diese Zeitzone berücksichtigt keine Sommerzeit.
Zentralaustralische Normalzeit	(UTC+ 08:45)	Eucla	
AUSÖstliche Normalzeit	(UTC+ 10:00)	Canberra, Melbourne, Sydney	
Aserbaidschan Normalzeit	(UTC+ 04:00)	Baku	
Azoren Normalzeit	(UTC— 01:00)	Azoren	
Bahia Normalzeit	(UTC— 03:00)	Salvador	
Bangladesch Normalzeit	(UTC+ 06:00)	Dhaka	Diese Zeitzone berücksichtigt keine Sommerzeit.
Belarus Standardzeit	(UTC+ 03:00)	Minsk	Diese Zeitzone berücksichtigt keine Sommerzeit.
Bougainville Normalzeit	(UTC+ 11:00)	Bougainville-Insel	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Canada Central Standard Time	(UTC— 06:00)	Saskatchewan	Diese Zeitzone berücksichtigt keine Sommerzeit.
Kap Verde Standardzeit	(UTC— 01:00)	Kapverdische Inseln	Diese Zeitzone berücksichtigt keine Sommerzeit.
Kaukasus Normalzeit	(UTC+ 04:00)	Eriwan	
Cen. Australia Standard Time	(UTC+ 09:30)	Adelaide	
Mittelamerikanische Standardzeit	(UTC— 06:00)	Mittelamerika	Diese Zeitzone berücksichtigt keine Sommerzeit.
Zentralasiatische Standardzeit	(UTC+ 06:00)	Astana	Diese Zeitzone berücksichtigt keine Sommerzeit.
Central Brazilian Standard Time	(UTC— 04:00)	Cuiaba	
Mitteleuropäische Standardzeit	(UTC+ 01:00)	Belgrad, Bratislava, Budapest, Ljubljana, Prag	
Mitteleuropäische Standardzeit	(UTC+ 01:00)	Sarajevo, Skopje, Warschau, Zagreb	
Zentralpazifische Standardzeit	(UTC+ 11:00)	Salomon-Inseln, Neukaledonien	Diese Zeitzone berücksichtigt keine Sommerzeit.
Central Standard Time	(UTC— 06:00)	Central Time (USA und Kanada)	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Central Standard Time (Mexiko)	(UTC— 06:00)	Guadalajara, Mexiko-Stadt, Monterrey	
Chatham-Inseln Normalzeit	(UTC+ 12:45)	Chatham-Inseln	
China Standardzeit	(UTC+ 08:00)	Beijing, Chongqing, Hongkong, Urumqi	Diese Zeitzone berücksichtigt keine Sommerzeit.
Kuba Normalzeit	(UTC— 05:00)	Havanna	
Datumsgrenze, Normalzeit	(UTC— 12:00)	Internationale Datumsgrenze West	Diese Zeitzone berücksichtigt keine Sommerzeit.
O. Afrikanische Standardzeit	(UTC+ 03:00)	Nairobi	Diese Zeitzone berücksichtigt keine Sommerzeit.
O. Australia Standard Time	(UTC+ 10:00)	Brisbane	Diese Zeitzone berücksichtigt keine Sommerzeit.
O. Europäische Standardzeit	(UTC+ 02:00)	Chisinau	
O. Südamerikanische Standardzeit	(UTC— 03:00)	Brasilia	
Osterinsel Normalzeit	(UTC— 06:00)	Osterinsel	
Ost Standardzeit	(UTC— 05:00)	Ostküstenzeit (USA und Kanada)	
Östliche Normalzeit (Mexiko)	(UTC— 05:00)	Chetumal	
Ägypten Normalzeit	(UTC+ 02:00)	Kairo	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Jekaterinburg Normalzeit	(UTC+ 05:00)	Jekaterinburg	
Fidschi Normalzeit	(UTC+ 12:00)	Fidschi	
FLEStandardzeit	(UTC+ 02:00)	Helsinki, Kiew, Riga, Sofia, Tallinn, Wilna	
Georgien Standardzeit	(UTC+ 04:00)	Tiflis	Diese Zeitzone berücksichtigt keine Sommerzeit.
GMTStandardzeit	(UTC)	Dublin, Edinburgh, Lissabon, London	Diese Zeitzone ist nicht dieselbe wie Greenwich Mean Time. Diese Zeitzone berücksichtigt die Sommerzeit.
Grönland Standardzeit	(UTC— 03:00)	Grönland	
Greenwich Standardzeit	(UTC)	Monrovia, Reykjavik	Diese Zeitzone berücksichtigt keine Sommerzeit.
GTBStandardzeit	(UTC+ 02:00)	Athen, Bukarest	
Haiti Normalzeit	(UTC— 05:00)	Haiti	
Hawaii Standardzeit	(UTC— 10:00)	Hawaii	
Indien Standardzeit	(UTC+ 05:30)	Chennai, Kolkata, Mumbai, Neu-Delhi	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Iran Normalzeit	(UTC+ 03:30)	Teheran	
Israel Normalzeit	(UTC+ 02:00)	Jerusalem	
Jordanien Standardzeit	(UTC+ 02:00)	Amman	
Kaliningrad Normalzeit	(UTC+ 02:00)	Kaliningrad	
Kamtschatka Normalzeit	(UTC+ 12:00)	Petropawlowsk-Kamtschatski – Alt	
Korea Standardzeit	(UTC+ 09:00)	Seoul	Diese Zeitzone berücksichtigt keine Sommerzeit.
Libyen Normalzeit	(UTC+ 02:00)	Tripolis	
Linieninseln Normalzeit	(UTC+ 14:00)	Kiritimati-Insel	
Lord Howe Normalzeit	(UTC+ 10:30)	Lord-Howe-Insel	
Magadan Normalzeit	(UTC+ 11:00)	Magadan	Diese Zeitzone berücksichtigt keine Sommerzeit.
Magallan Normalzeit	(UTC— 03:00)	Punta Arenas	
Marquesas Normalzeit	(UTC— 09:30)	Marquesas-Inseln	
Mauritius Normalzeit	(UTC+ 04:00)	Port Louis	Diese Zeitzone berücksichtigt keine Sommerzeit.
Mittlerer Osten Standardzeit	(UTC+ 02:00)	Beirut	
Montevideo Normalzeit	(UTC— 03:00)	Montevideo	
Marokko Normalzeit	(UTC+ 01:00)	Casablanca	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Mountain Standard Time	(UTC— 07:00)	Mountain Time (USA und Kanada)	
Mountain Standard Time (Mexiko)	(UTC— 07:00)	Chihuahua, La Paz, Mazatlan	
Myanmar Normalzeit	(UTC+ 06:30)	Yangon (Rangun)	Diese Zeitzone berücksichtigt keine Sommerzeit.
N. Zentralasiatische Standardzeit	(UTC+ 07:00)	Nowosibirsk	
Namibia Normalzeit	(UTC+ 02:00)	Windhuk	
Nepal Normalzeit	(UTC+ 05:45)	Kathmandu	Diese Zeitzone berücksichtigt keine Sommerzeit.
Neuseeland Standardzeit	(UTC+ 12:00)	Auckland, Wellington	
Neufundland Standardzeit	(UTC— 03:30)	Neufundland	
Norfolk Normalzeit	(UTC+ 11:00)	Norfolkinsel	
Ost-Nordasiatische Normalzeit	(UTC+ 08:00)	Irkutsk	
Nordasien Normalzeit	(UTC+ 07:00)	Krasnojarsk	
Nordkorea Normalzeit	(UTC+ 09:00)	Pjöngjang	
Omsk Normalzeit	(UTC+ 06:00)	Omsk	
Pacific SA Standard Time	(UTC— 03:00)	Santiago	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Pacific Standard Time	(UTC— 08:00)	Pacific Time (USA und Kanada)	
Pacific Standard Time (Mexiko)	(UTC— 08:00)	Baja California	
Pakistan Normalzeit	(UTC+ 05:00)	Islamabad, Karatschi	Diese Zeitzone berücksichtigt keine Sommerzeit.
Paraguay Normalzeit	(UTC— 04:00)	Asunción	
Romanische Normalzeit	(UTC+ 01:00)	Brüssel, Kopenhagen, Madrid, Paris	
Russland Zeitzone 10	(UTC+ 11:00)	Tschokurdach	
Russland Zeitzone 11	(UTC+ 12:00)	Anadyr, Petropawlowsk-Kamtschatski	
Russland Zeitzone 3	(UTC+ 04:00)	Ischewsk, Samara	
Russische Standardzeit	(UTC+ 03:00)	Moskau, St. Petersburg, Wolgograd	Diese Zeitzone berücksichtigt keine Sommerzeit.
Östl. Südamerika Normalzeit	(UTC— 03:00)	Cayenne, Fortaleza	Diese Zeitzone berücksichtigt keine Sommerzeit.
SA Pacific Standard Time	(UTC— 05:00)	Bogota, Lima, Quito, Rio Branco	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Mittl. Südamerika Normalzeit	(UTC— 04:00)	Georgetown, La Paz, Manaus, San Juan	Diese Zeitzone berücksichtigt keine Sommerzeit.
Saint Pierre Normalzeit	(UTC— 03:00)	St. Pierre und Miquelon	
Sachalin Normalzeit	(UTC+ 11:00)	Sachalin	
Samoa Normalzeit	(UTC+ 13:00)	Samoa	
São Tomé Normalzeit	(UTC+ 01:00)	São Tomé	
Saratow Normalzeit	(UTC+ 04:00)	Saratow	
Südostasiatische Standardzeit	(UTC+ 07:00)	Bangkok, Hanoi, Jakarta	Diese Zeitzone berücksichtigt keine Sommerzeit.
Singapur Standardzeit	(UTC+ 08:00)	Kuala Lumpur, Singapur	Diese Zeitzone berücksichtigt keine Sommerzeit.
Südafrika Normalzeit	(UTC+ 02:00)	Harare, Pretoria	Diese Zeitzone berücksichtigt keine Sommerzeit.
Sri Lanka Normalzeit	(UTC+ 05:30)	Sri Jayawarde nepura	Diese Zeitzone berücksichtigt keine Sommerzeit.
Sudan Normalzeit	(UTC+ 02:00)	Khartum	
Syrien Normalzeit	(UTC+ 02:00)	Damaskus	

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
Taipei Normalzeit	(UTC+ 08:00)	Taipeh	Diese Zeitzone berücksichtigt keine Sommerzeit.
Tasmanien Normalzeit	(UTC+ 10:00)	Hobart	
Tocantins Normalzeit	(UTC— 03:00)	Araguaina	
Japanische Standardzeit	(UTC+ 09:00)	Osaka, Sapporo, Tokio	Diese Zeitzone berücksichtigt keine Sommerzeit.
Tomsk Normalzeit	(UTC+ 07:00)	Tomsk	
Tonga Normalzeit	(UTC+ 13:00)	Nuku'alofa	Diese Zeitzone berücksichtigt keine Sommerzeit.
Transbaikal Normalzeit	(UTC+ 09:00)	Tschita	
Türkei Normalzeit	(UTC+ 03:00)	Istanbul	
Turks- und Caicosinseln Normalzeit	(UTC— 05:00)	Turks- und Caicosinseln	
Ulan-Bator Normalzeit	(UTC+ 08:00)	Ulan-Bator	Diese Zeitzone berücksichtigt keine Sommerzeit.
US Eastern Standard Time	(UTC— 05:00)	Indiana (Osten)	
US Mountain Standard Time	(UTC— 07:00)	Arizona	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
UTC	UTC	Coordinated Universal Time	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC—02	(UTC— 02:00)	Coordinated Universal Time–02	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC—08	(UTC— 08:00 Uhr)	Coordinated Universal Time–08	
UTC—09	(UTC— 09:00 Uhr)	Coordinated Universal Time–09	
UTC—11	(UTC— 11:00 Uhr)	Coordinated Universal Time–11	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC+12	(UTC+ 12:00)	Coordinated Universal Time+12	Diese Zeitzone berücksichtigt keine Sommerzeit.
UTC+13	(UTC+ 13:00)	Koordinierte Weltzeit+13	
Venezuela Normalzeit	(UTC— 04:00)	Caracas	Diese Zeitzone berücksichtigt keine Sommerzeit.
Wladiwostok Normalzeit	(UTC+ 10:00)	Wladiwostok	
Wolgograd Normalzeit	(UTC+ 04:00)	Wolgograd	
W. Australia Standard Time	(UTC+ 08:00)	Perth	Diese Zeitzone berücksichtigt keine Sommerzeit.

Zeitzone	Standardzeit-Versatz	Beschreibung	Hinweise
W. Zentralafrikanische Standardzeit	(UTC+ 01:00)	West-Zentralafrika	Diese Zeitzone berücksichtigt keine Sommerzeit.
W. Europäische Standardzeit	(UTC+ 01:00)	Amsterdam, Berlin, Bern, Rom, Stockholm, Wien	
W. Mongolei Normalzeit	(UTC+ 07:00)	Hovd	
Westasien Normalzeit	(UTC+ 05:00)	Aschgabat, Taschkent	Diese Zeitzone berücksichtigt keine Sommerzeit.
Westjordanland Normalzeit	(UTC+ 02:00)	Gaza, Hebron	
Westpazifische Normalzeit	(UTC+ 10:00)	Guam, Port Moresby	Diese Zeitzone berücksichtigt keine Sommerzeit.
Jakutsk Normalzeit	(UTC+ 09:00)	Jakutsk	

Lizenzierung Microsoft SQL Server auf Amazon RDS

Beim Einrichten einer Amazon RDS-DB-Instance für Microsoft SQL Server ist die Software-Lizenz enthalten.

Das bedeutet, dass Sie keine separaten SQL-Server-Lizenzen erwerben müssen. AWS hat die Lizenz für die SQL-Server-Datenbanksoftware. Amazon-RDS-Preise beinhalten die Softwarelizenz, die unterliegenden Hardware-Ressourcen sowie die Amazon-RDS-Verwaltungsfunktionen.

Amazon RDS unterstützt die folgenden Microsoft SQL Server-Editionen:

- Enterprise
- Standard
- Web
- Express

> **Note**
>
> Die Lizenzierung für SQL Server Web Edition unterstützt nur öffentliche und über das Internet abrufbare Webseiten, Websites, Webanwendungen und Webservices. Dieser Support-Umfang ist erforderlich, um den Nutzungsrechten von Microsoft zu entsprechen. Weitere Informationen finden Sie unter AWSServicebedingungen.

Amazon RDS unterstützt Multi-AZ-Bereitstellungen für DB-Instances, die Microsoft SQL Server mit SQL Server-Datenbankspiegelung oder -AlwaysOn-Verfügbarkeitsgruppen ausführen. Es bestehen keine weiteren Lizenzanforderungen für Multi-AZ-Bereitstellungen. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.

Wiederherstellen von DB-Instances, die aus Lizenzgründen beendet wurden

Amazon RDS erstellt Snapshots von DB-Instances, die aus Lizenzgründen beendet wurden. Wenn Ihre Instance aus Lizenzgründen beendet wurde, können Sie sie vom Snapshot in einer neuen DB-Instance wiederherstellen. Neue DB-Instances enthalten eine Lizenz.

Weitere Informationen finden Sie unter Wiederherstellung von DB-Instances mit gekündigter Lizenz für Amazon for Server RDS SQL.

Entwicklung und Test

Aufgrund von Lizenzierungsanforderungen ist die SQL Server Developer-Edition in Amazon RDS nicht verfügbar. Für viele Entwicklungs- und Testaufgaben außerhalb der Produktion können Sie die Express-Edition verwenden. Wenn Sie jedoch alle Funktionen einer Installation von SQL Server auf Unternehmensebene für die Entwicklung benötigen, können Sie SQL Server Developer Edition auf RDS Custom für SQL Server mit einer CEV mit BYOM herunterladen und installieren. Weitere Informationen finden Sie unter Vorbereitung einer CEV mit Bring Your Own Media (BYOM). Für die Developer-Edition ist keine dedizierte Infrastruktur erforderlich. Durch die Verwendung eines eigenen Hosts erhalten Sie auch Zugriff auf andere programmierbare Funktionen, die in Amazon RDS nicht verfügbar sind. Weitere Informationen zum Unterschied zwischen SQL Server-Editionen finden Sie unter Editionen und unterstützte Funktionen von SQL Server 2019 in der Microsoft-Dokumentation.

Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen

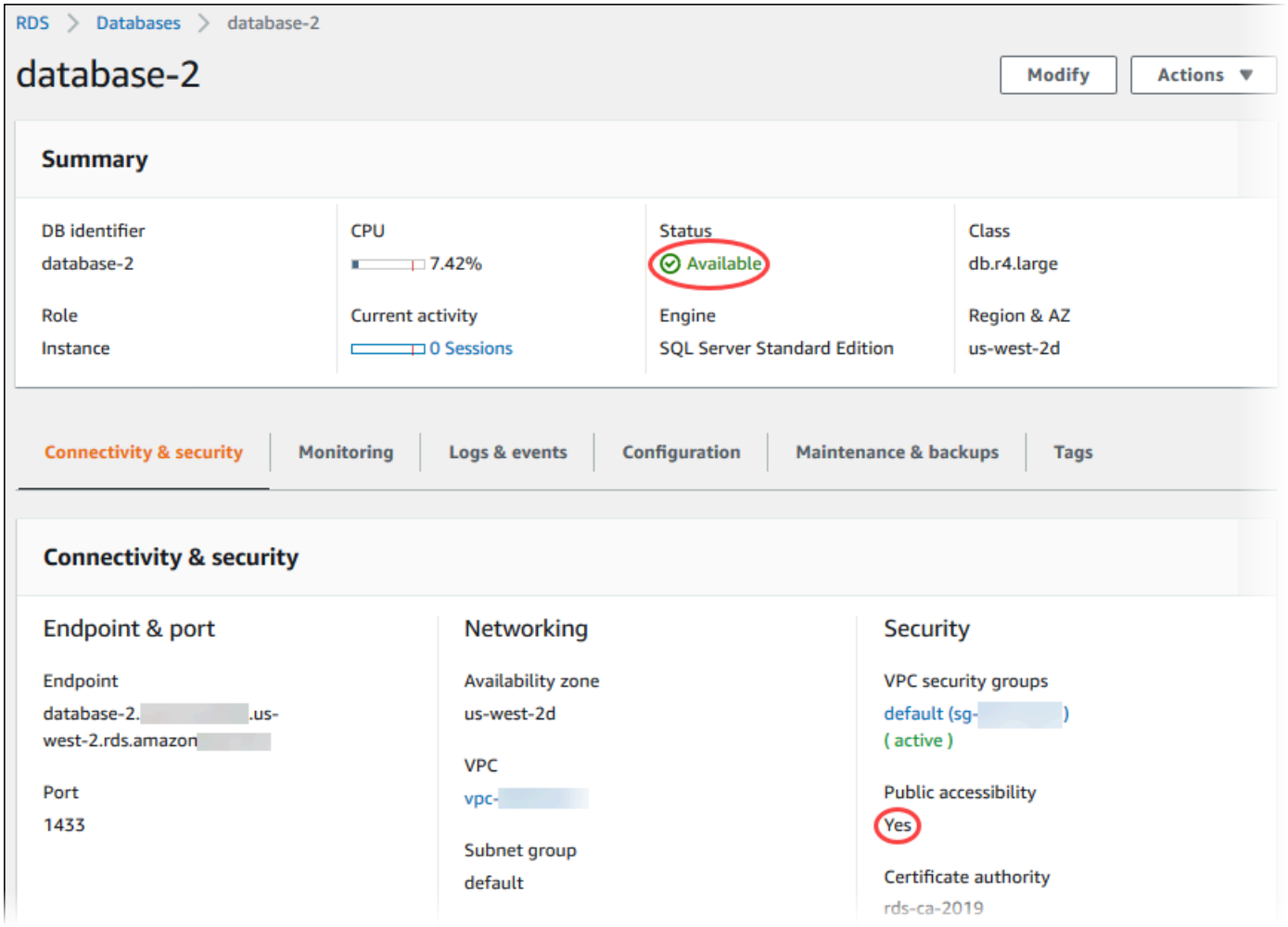
Nachdem Amazon RDS Ihre DB-Instance bereitgestellt hat, können Sie die Verbindung zu dieser über eine beliebige Standard-SQL-Clientanwendung herstellen. In diesem Thema wird beschrieben, wie Sie mithilfe von Microsoft SQL Server Management Studio (SSMS) oder SQL Workbench/J die Verbindung zur DB-Instance herstellen.

Ein Beispiel mit einer Anleitung zum Erstellen und Verbinden für eine Beispiel-DB-Instance finden Sie unter Erstellen einer DB-Instance von Microsoft SQL Server und Herstellen einer Verbindung.

Bevor Sie sich verbinden

Bevor Sie eine Verbindung zu Ihrer DB-Instance herstellen können, muss diese verfügbar und zugänglich sein.

1. Stellen Sie sicher, dass der Status laute `available`. Sie können dies auf der Detailseite für Ihre Instance im AWS Management Console oder mithilfe des describe-db-instances AWS CLI Befehls überprüfen.



2. Stellen Sie sicher, dass sie für Ihre Quelle zugänglich ist. Abhängig von Ihrem Szenario muss sie möglicherweise nicht öffentlich zugänglich sein. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.
3. Stellen Sie sicher, dass die eingehenden Regeln Ihrer VPC-Sicherheitsgruppe den Zugriff auf Ihre DB-Instance ermöglichen. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Finden des Endpunkts und der Portnummer der DB-Instance

Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.

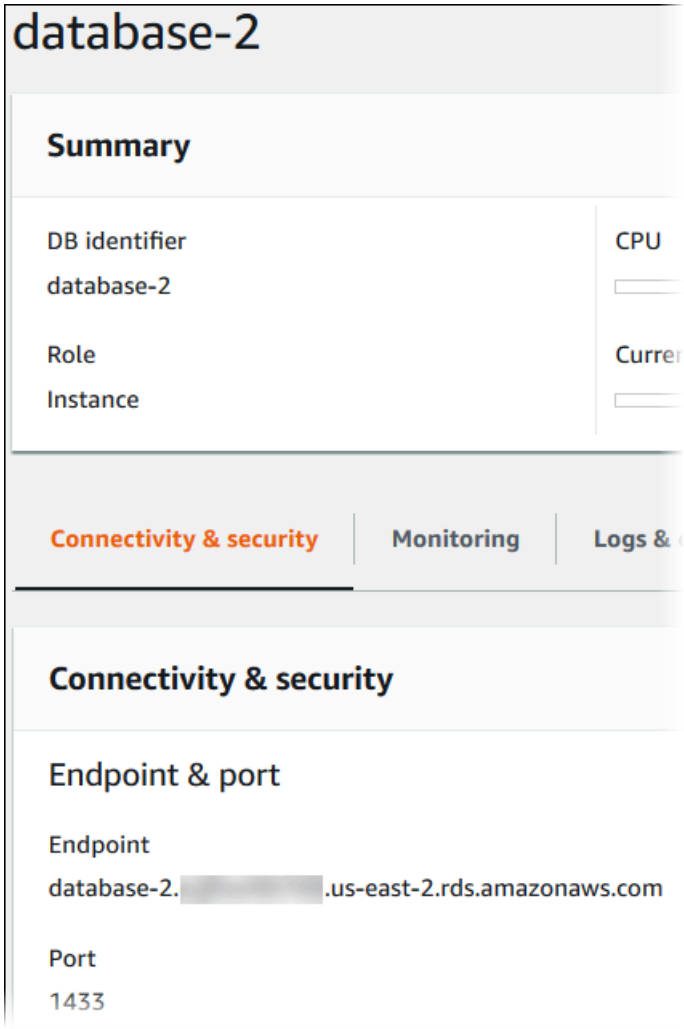
So finden Sie den Endpunkt und den Port

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die AWS Region Ihrer DB-Instance aus.
3. Suchen Sie den DNS-Namen (Domain Name System) (Endpunkt) und die Portnummer für Ihre DB-Instance:

   a. Öffnen Sie die RDS-Konsole und wählen Sie Databases (Datenbanken) aus, um eine Liste Ihrer DB-Instances anzeigen zu lassen.

   b. Wählen Sie den Namen der SQL Server-DB-Instance, um deren Details anzuzeigen.

   c. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt.



d. Notieren Sie sich auch die Portnummer.

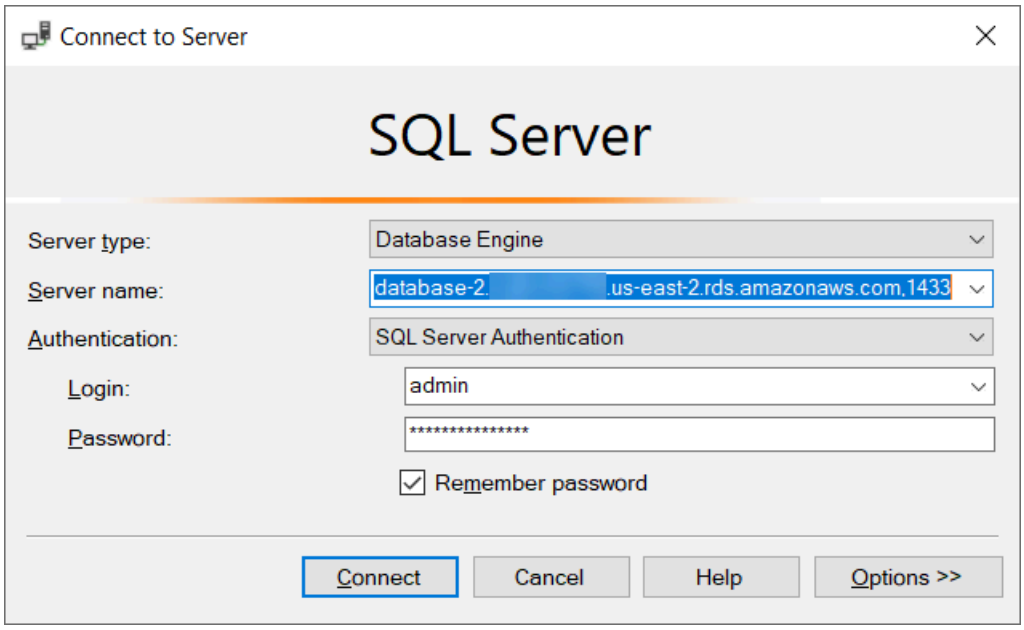
Mit Microsoft SQL Server Management Studio eine Verbindung zu Ihrer DB-Instance herstellen

In diesem Verfahren stellen Sie mithilfe von Microsoft SQL Server Management Studio (SSMS) eine Verbindung zu Ihrer Beispiel-DB-Instance her. Informationen zum Herunterladen einer eigenständigen Version dieses Dienstprogramms finden Sie unter SQL Server Management Studio herunterladen (SSMS) in der Microsoft-Dokumentation.

Um eine Verbindung zu einer DB-Instance herzustellen, verwenden Sie SSMS

1. Starten Sie SQL Server Management Studio.

   Das Dialogfeld Connect to Server (Mit Server verbinden) erscheint.



2. Geben Sie die Informationen für Ihre DB-Instance an:

   a. Wählen Sie für Servertyp die Option Datenbank-Engine aus.

   b. Geben Sie als Servername den DNS Namen (Endpunkt) und die Portnummer Ihrer DB-Instance ein, getrennt durch ein Komma.

> **Important**
>
> Ändern Sie den Doppelpunkt zwischen dem Endpunkt und der Portnummer in ein Komma.

Ihr Servername sollte wie im folgenden Beispiel aussehen.

```
database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433
```

c. Wählen Sie für Authentifizierung die Option SQLServerauthentifizierung aus.

d. Geben Sie unter Login (Anmeldename) den Hauptbenutzernamen für Ihre DB-Instance ein.

e. Geben Sie unter Password (Passwort) das Passwort für Ihre DB-Instance ein.

3. Wählen Sie Connect (Verbinden) aus.

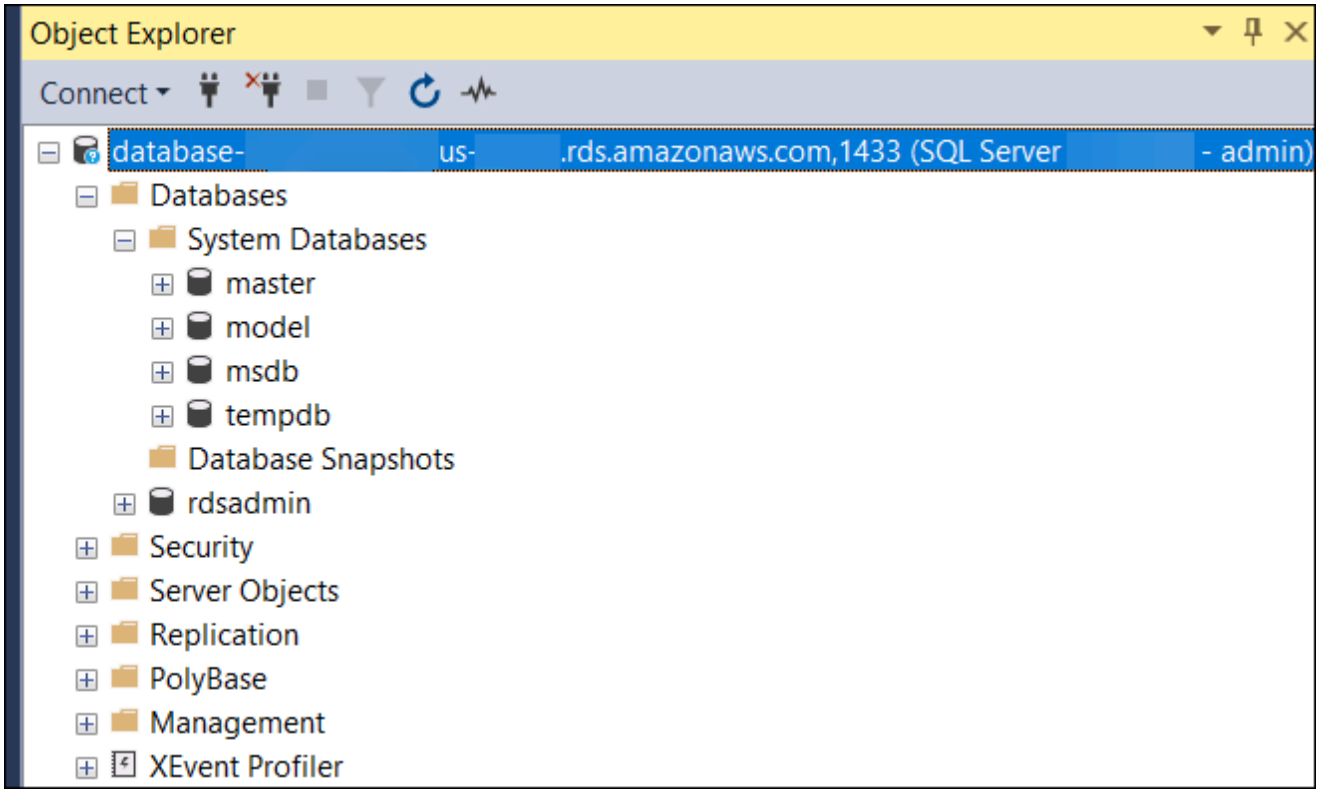
   Stellt nach einigen Augenblicken eine SSMS Verbindung zu Ihrer DB-Instance her.

   Weitere Informationen für den Fall, dass keine Verbindung zur DB-Instance hergestellt werden kann, finden Sie unter Überlegungen zu Sicherheitsgruppen und Problembehandlung bei Verbindungen zu Ihrer SQL Server-DB-Instance.

4. Ihre SQL Server-DB-Instance verfügt über die integrierten Standardsystemdatenbanken von SQL Server (`master`,`model`,`msdb`, und`tempdb`). Führen Sie folgende Schritte aus, um die Systemdatenbanken zu durchforschen:

   a. Wählen Sie im SSMS Menü Ansicht die Option Objekt-Explorer aus.

   b. Erweitern Sie die DB-Instance und dann Datenbanken und Systemdatenbanken.



5. Ihre SQL Server-DB-Instance enthält auch eine Datenbank mit dem Namen`rdsadmin`. Amazon RDS verwendet diese Datenbank, um die Objekte zu speichern, die es zur Verwaltung Ihrer Datenbank verwendet. Die Datenbank `rdsadmin` beinhaltet auch gespeicherte Prozeduren, die Sie ausführen können, um erweiterte Aufgaben durchzuführen. Weitere Informationen finden Sie unter Allgemeine DBA-Aufgaben für Amazon RDS for Microsoft SQL Server.

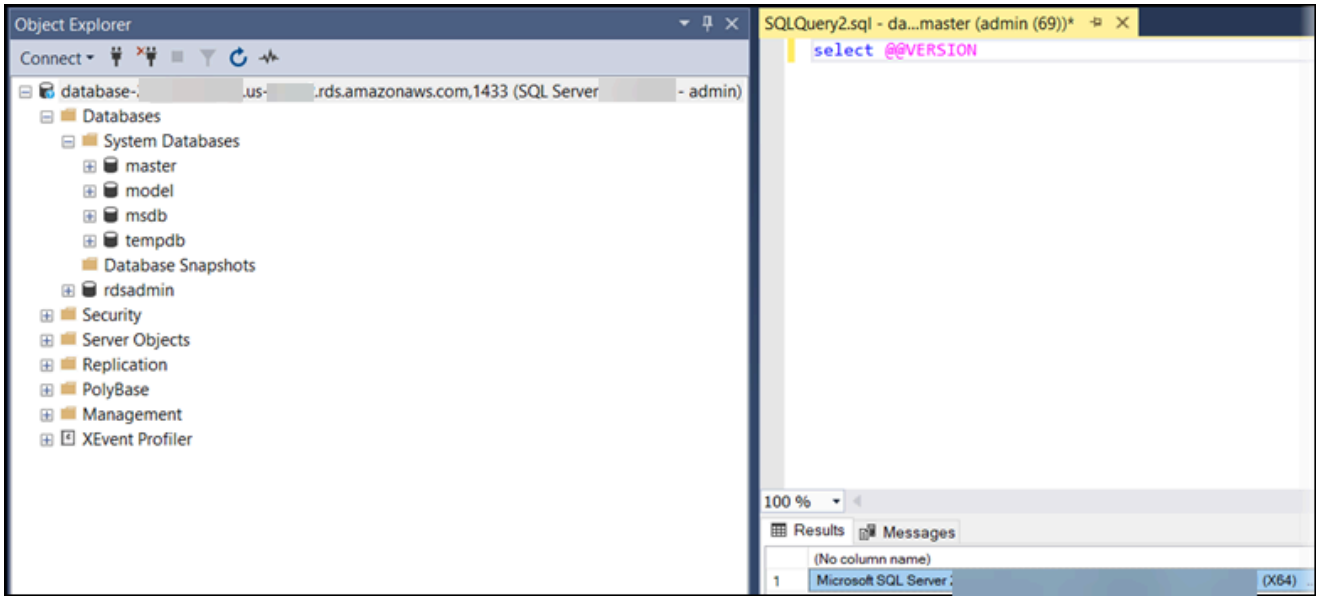
6. Sie können nun wie üblich beginnen Ihre eigenen Datenbanken zu erstellen und Abfragen gegen Ihre DB-Instance und Datenbanken auszuführen. Gehen Sie wie folgt vor, um eine Testabfrage für die DB-Instance auszuführen:

   a. Zeigen Sie in im SSMS Menü Datei auf Neu und wählen Sie dann Abfrage mit aktueller Verbindung aus.

   b. Geben Sie die folgende SQL Abfrage ein.

   ```
   select @@VERSION
   ```

   c. Führen Sie die Abfrage aus. SSMSgibt die SQL Serverversion Ihrer Amazon RDS DB-Instance zurück.



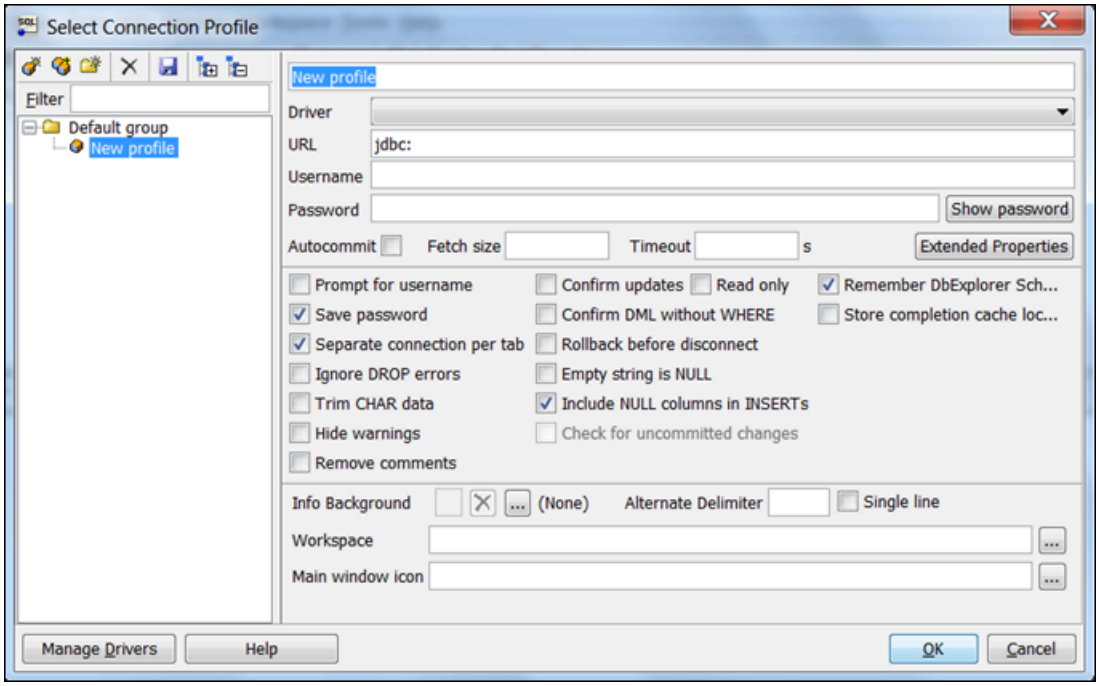
Herstellen einer Verbindung zu Ihrer DB-Instance mit SQL Workbench/J

Dieses Beispiel zeigt, wie Sie mithilfe von SQL eine Verbindung zu einer DB-Instance herstellen, auf der die Microsoft SQL Server-Datenbank-Engine ausgeführt wirdWorkbench/J database tool. To download SQL Workbench/J, siehe SQL Workbench/J.

SQL Workbench/J verwendet JDBC, um eine Verbindung zur DB-Instance herzustellen. Sie benötigen auch den JDBC-Treiber für SQL Server. Informationen zum Herunterladen dieses Treibers finden Sie unter Microsoft JDBC-Treiber für SQL Server herunterladen.

So stellen Sie eine Verbindung zu einer DB-Instance mithilfe von SQL Workbench/J her

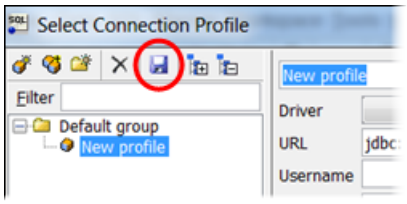
1. Open-SQL-Workbench/J. Das Dialogfeld Select Connection Profile (Verbindungsprofil auswählen) wird wie folgt angezeigt.



2. Geben Sie im ersten Feld oben im Dialogfeld einen Namen für das Profil an.
3. Wählen Sie bei Driver (Treiber) die Option **`SQL JDBC 4.0`** aus.
4. Geben Sie unter URL **`jdbc:sqlserver://`** gefolgt vom Endpunkt Ihrer DB-Instance ein. Der URL-Wert könnte beispielsweise wie folgt lauten.

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

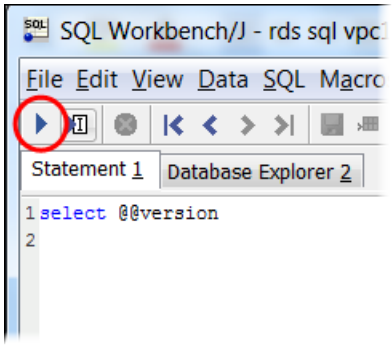
5. Geben Sie unter Username (Benutzername) den Hauptbenutzernamen der DB-Instance ein.
6. Geben Sie unter Password (Passwort) das Passwort für den Hauptbenutzer ein.
7. Wählen Sie in der Symbolleiste des Dialogfelds wie folgt das Speichersymbol aus.



8. Klicken Sie auf OK. Nach wenigen Augenblicken stellt SQL Workbench/J die Verbindung zur DB-Instance her. Weitere Informationen für den Fall, dass keine Verbindung zur DB-Instance hergestellt werden kann, finden Sie unter Überlegungen zu Sicherheitsgruppen und Problembehandlung bei Verbindungen zu Ihrer SQL Server-DB-Instance.
9. Geben Sie im Abfragebereich die folgende SQL-Abfrage ein.

```
select @@VERSION
```

10. Wählen Sie in der Symbolleiste wie folgt das Ausführungssymbol `Execute` aus.



Die Abfrage gibt wie nachfolgend angezeigt die Versionsinformation für Ihre DB-Instance zurück.

```
Microsoft SQL Server 2017 (RTM-CU22) (KB4577467) - 14.0.3356.20 (X64)
```

Überlegungen zu Sicherheitsgruppen

Um eine Verbindung zu Ihrer DB-Instance herzustellen, muss Ihre DB-Instance einer Sicherheitsgruppe zugeordnet sein. Diese Sicherheitsgruppe enthält die IP-Adressen und die Netzwerkkonfiguration, die Sie für den Zugriff auf die DB-Instance verwenden. Sie haben der DB-Instance möglicherweise bereits direkt beim Erstellen eine geeignete Sicherheitsgruppe zugeordnet. Falls Sie beim Erstellen Ihrer DB-Instance eine standardmäßige, nicht konfigurierte Sicherheitsgruppe zugeordnet haben, verhindert die Firewall Ihrer DB-Instance jeglichen Verbindungsversuch.

In einigen Fällen müssen Sie möglicherweise eine neue Sicherheitsgruppe erstellen, um den Zugriff zu ermöglichen. Anweisungen zum Erstellen einer neuen Sicherheitsgruppe finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen. Ein Thema, das Sie Schritt für Schritt durch die Einrichtung von Regeln für Ihre VPC Sicherheitsgruppe führt, finden Sie unterTutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur).

Nachdem Sie die neue Sicherheitsgruppe erstellt haben, ändern Sie Ihre DB-Instance, um ihr die Sicherheitsgruppe zuzuordnen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sie können die Sicherheit erhöhenSSL, indem Sie Verbindungen zu Ihrer DB-Instance verschlüsseln. Weitere Informationen finden Sie unter Verwendung SSL mit einer Microsoft SQL Server-DB-Instance.

Problembehandlung bei Verbindungen zu Ihrer SQL Server-DB-Instance

Die folgende Tabelle enthält Fehlermeldungen, die möglicherweise auftreten, wenn Sie versuchen, eine Verbindung zu Ihrer SQL Server-DB-Instance herzustellen.

Problem	Vorschläge für die Fehlerbehebung
Verbindung zum SQL Server konnte nicht geöffnet werden — Microsoft SQL Server, Fehler: 53	Stellen Sie sicher, dass Sie den Servernamen korrekt angegeben haben. Geben Sie als Servername den DNS Namen und die Portnummer Ihrer Beispiel-DB-Instance ein, getrennt durch ein Komma. **Important** Wenn zwischen dem DNS Namen und der Portnummer ein Doppelpunkt steht, ändern Sie den Doppelpunkt in ein Komma. Ihr Servername sollte wie im folgenden Beispiel aussehen. `sample-instance.cg034itsfake.us-east-1.rds.amazonaws.com,1433`
Es konnte keine Verbindung hergestellt werden, da der Zielcomputer sie aktiv abgelehnt hat — Microsoft SQL Server, Fehler: 10061	Die DB-Instance ist zwar erreichbar, jedoch wurde der Verbindungsversuch abgelehnt. Dieses Problem wird normalerweise durch falsche Angabe des Benutzernamens oder Passworts verursacht. Überprüfen Sie den Benutzernamen und das Passwort und versuchen Sie es erneut.
Beim Herstellen einer Verbindung zum Server ist ein netzwerkbezogener oder instanzspezifischer Fehler aufgetreten. SQL Der Server wurde nicht gefunden oder war nicht zugänglich... Das Zeitlimit	Die von Ihrer lokalen Firewall erzwungenen Zugriffsregeln und die für den Zugriff auf Ihre DB-Instance autorisierten IP-Adressen stimmen möglicherweise nicht überein. Das Problem sind höchstwahrscheinlich die Regeln für eingehenden Datenverkehr in Ihrer Sicherheitsgruppe. Weitere Informationen finden Sie unter Sicherheit in Amazon RDS.

Problem	Vorschläge für die Fehlerbehebung
für den Wartevorgang wurde überschritten — Microsoft SQL Server, Fehler: 258	Ihre Datenbank-Instance muss öffentlich zugänglich sein. Um von außerhalb der eine Verbindung zu ihr herzustellenVPC, muss der Instanz eine öffentliche IP-Adresse zugewiesen sein.

> **Note**
>
> Weitere Informationen zu Verbindungsproblemen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Arbeiten mit Active Directory mit RDS für SQL Server

Sie können eine RDS-für-SQL-Server-DB-Instance einer Microsoft Active Directory (AD)-Domain hinzufügen. Ihre AD-Domain kann in AWS Managed AD innerhalb von AWS oder in einem selbstverwalteten AD an einem Standort Ihrer Wahl gehostet werden, einschließlich Ihrer Unternehmensrechenzentren, AWS EC2 oder anderer Cloud-Anbieter.

Sie können Domain-Benutzer bei einem selbstverwaltetem Active Directory mithilfe der NTLM-Authentifizierung authentifizieren. Sie können die Kerberos- und NTLM-Authentifizierung mit AWS Managed Active Directory verwenden.

In den folgenden Abschnitten finden Sie Informationen zum Arbeiten mit einem selbstverwalteten Active Directory und mit AWS Managed Active Directory für Microsoft SQL Server auf Amazon RDS.

Themen

- Arbeiten mit Self Managed Active Directory mit einer Amazon RDS for SQL Server-DB-Instance
- Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server

Arbeiten mit Self Managed Active Directory mit einer Amazon RDS for SQL Server-DB-Instance

Sie können Ihre RDS vier SQL Server-DB-Instances direkt mit Ihrer selbstverwalteten Active Directory (AD) -Domäne verbinden, unabhängig davon, wo Ihr AD gehostet wird: in Unternehmensrechenzentren AWS EC2, auf oder bei anderen Cloud-Anbietern. Mit selbstverwaltetem AD verwenden Sie die NTLM Authentifizierung, um die Authentifizierung von Benutzern und Diensten auf Ihren RDS vier SQL Server-DB-Instances direkt zu steuern, ohne zwischengeschaltete Domänen und Forest Trusts zu verwenden. Wenn sich Benutzer mit einer RDS for SQL Server-DB-Instance authentifizieren, die zu Ihrer selbstverwalteten AD-Domäne gehört, werden Authentifizierungsanforderungen an eine von Ihnen angegebene selbstverwaltete AD-Domäne weitergeleitet.

Themen

- Verfügbarkeit von Regionen und Versionen
- Voraussetzungen
- Einschränkungen
- Übersicht über die Einrichtung eines selbstverwalteten Active Directory
- Einrichten eines selbstverwalteten Active Directory
- Verwalten einer DB-Instance in einer selbstverwalteten Active-Directory-Domain
- Grundlegendes zur Mitgliedschaft in einer selbstverwalteten Active-Directory-Domain
- Fehlerbehebung für selbstverwaltetes Active Directory
- Wiederherstellung einer SQL Server-DB-Instance und anschließendes Hinzufügen zu einer selbstverwalteten Active Directory-Domäne

Verfügbarkeit von Regionen und Versionen

Amazon RDS unterstützt insgesamt die Verwendung NTLM von Self Managed AD for SQL Server AWS-Regionen.

Voraussetzungen

Stellen Sie sicher, dass Sie die folgenden Anforderungen erfüllen, bevor Sie eine RDS for SQL Server-DB-Instance zu Ihrer selbstverwalteten AD-Domäne hinzufügen.

Themen

- Konfigurieren Ihres On-Premises-AD
- Konfigurieren Ihrer Netzwerkkonnektivität
- Konfigurieren Ihres AD-Domain-Servicekontos

Konfigurieren Ihres On-Premises-AD

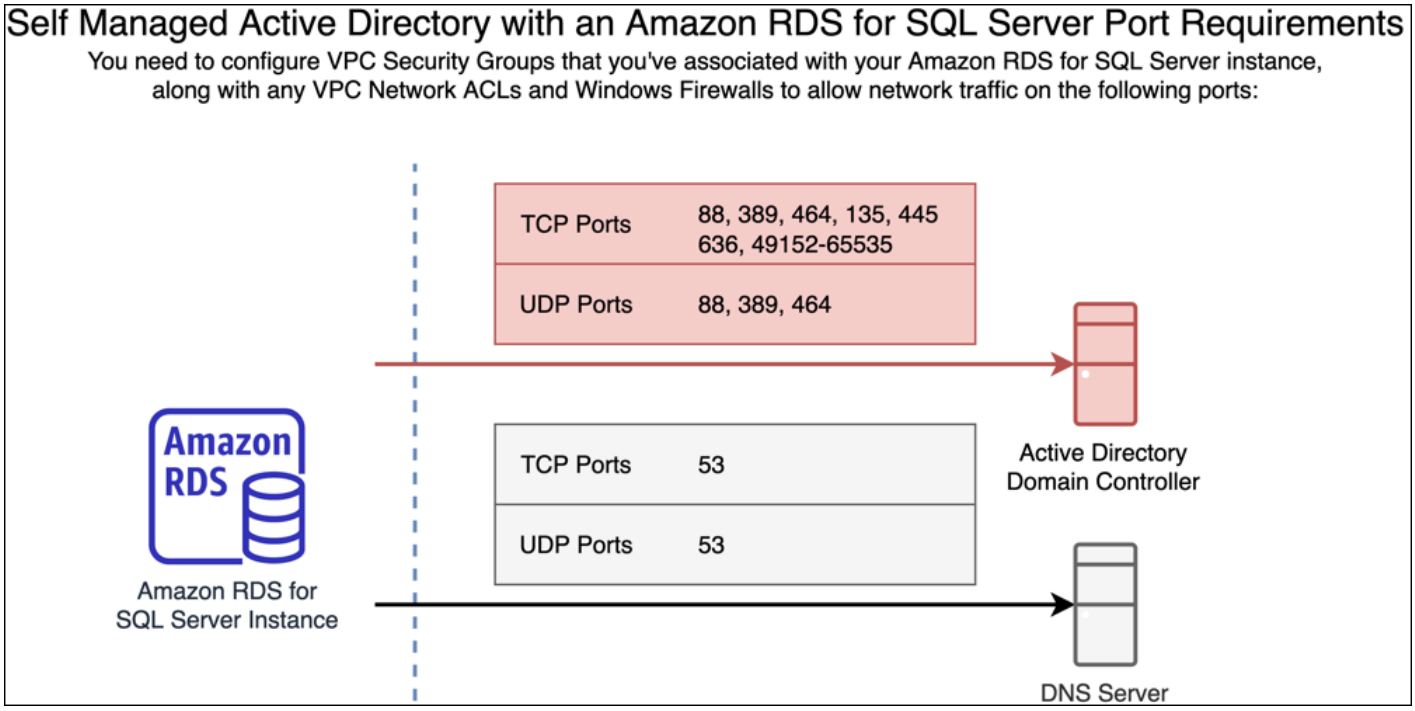
Stellen Sie sicher, dass Sie über ein lokales oder ein anderes selbstverwaltetes Microsoft AD verfügen, dem Sie der Amazon RDS for SQL Server-Instance beitreten können. Ihr On-Premises-AD sollte folgende Konfiguration aufweisen:

- Wenn Sie Active Directory-Standorte definiert haben, stellen Sie sicher, dass die Subnetze in der mit Ihrer SQL Server-DB-Instance VPC verknüpften Subnetze an Ihrem RDS Active Directory-Standort definiert sind. Stellen Sie sicher, dass keine Konflikte zwischen den Subnetzen an Ihren VPC und den Subnetzen an Ihren anderen AD-Standorten bestehen.
- Ihr AD-Domain-Controller weist die Domain-Funktionsebene Windows Server 2008 R2 oder höher auf.
- Ihr AD-Domainname darf nicht im Format Single Label Domain (SLD) vorliegen. RDSfor SQL Server unterstützt keine SLD Domänen.
- Der vollqualifizierte Domainname (FQDN) für Ihr AD darf 47 Zeichen nicht überschreiten.

Konfigurieren Ihrer Netzwerkkonnektivität

Stellen Sie sicher, dass die folgenden Netzwerkkonfigurationsanforderungen erfüllt sind:

- Die Konnektivität wurde zwischen dem AmazonVPC, auf dem Sie die RDS for SQL Server-DB-Instance erstellen möchten, und Ihrem selbstverwalteten Active Directory konfiguriert. Sie können Konnektivität mit AWS Direct Connect AWS VPN, VPC Peering oder AWS Transit Gateway einrichten.
- Bei VPC Sicherheitsgruppen wurde die Standardsicherheitsgruppe für Ihre VPC Standard-Amazon-Instance bereits zu Ihrer RDS SQL Server-DB-Instance in der Konsole hinzugefügt. Stellen Sie sicher, dass die Sicherheitsgruppe und das VPC Netzwerk ACLs für die Subnetze, in denen Sie Ihre RDS SQL Server-DB-Instance erstellen, Datenverkehr auf den Ports und in den Anweisungen zulassen, die in der folgenden Abbildung dargestellt sind.



In der folgenden Tabelle ist die Rolle der einzelnen Ports aufgeführt.

Protokoll	Ports	Rolle
TCP/UDP	53	Domänennamensystem () DNS
TCP/UDP	88	Kerberos-Authentifizierung
TCP/UDP	464	Passwort ändern/festlegen
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
TCP	135	Verteilte Computerumgebung/ End Point Mapper (DCE/ EPMAP)
TCP	445	SMBDateifreigabe durch Verzeichnisdienste

Protokoll	Ports	Rolle
TCP	636	Lightweight Directory Access Protocol überTLS/SSL(LDAPS)
TCP	49152–65535	Kurzlebige Ports für RPC

- Im Allgemeinen befinden sich die DNS Domänenserver in den AD-Domänencontrollern. Sie müssen den VPC DHCP Optionssatz nicht konfigurieren, um diese Funktion verwenden zu können. Weitere Informationen finden Sie unter DHCPOptionssätze im VPCAmazon-Benutzerhandbuch.

> **Important**
>
> Wenn Sie ein VPC Netzwerk verwendenACLs, müssen Sie auch ausgehenden Datenverkehr über dynamische Ports (49152-65535) von Ihrer Server-DB-Instance aus zulassen. RDS SQL Stellen Sie sicher, dass diese Verkehrsregeln auch auf den Firewalls widergespiegelt werden, die für die einzelnen AD-Domänencontroller, Server und Server-DB-Instances gelten. DNS RDS SQL
> Während VPC Sicherheitsgruppen verlangen, dass Ports nur in der Richtung geöffnet werden, in der der Netzwerkverkehr initiiert wird, ACLs erfordern die meisten Windows-Firewalls und VPC Netzwerke, dass Ports in beide Richtungen geöffnet sind.

Konfigurieren Ihres AD-Domain-Servicekontos

Stellen Sie sicher, dass die folgenden Anforderungen für ein AD-Domain-Servicekonto erfüllt sind:

- Stellen Sie sicher, dass Sie in Ihrer selbstverwalteten AD-Domain über ein Servicekonto mit delegierten Berechtigungen zum Hinzufügen von Computern zu der Domain verfügen. Ein Domain-Servicekonto ist ein Benutzerkonto in Ihrem selbstverwalteten AD, an das die Berechtigung zur Ausführung bestimmter Aufgaben delegiert wurde.
- Dem Domänendienstkonto müssen die folgenden Berechtigungen in der Organisationseinheit (OU), der Sie Ihre RDS for SQL Server-DB-Instance beitreten, delegiert werden:
  - Die Fähigkeit, in den DNS Hostnamen zu schreiben, wurde bestätigt
  - Überprüfte Fähigkeit zum Schreiben in den Prinzipalnamen des Service
  - Erstellen und Löschen von Computerobjekten

Dies sind die erforderlichen Mindestberechtigungen, um Computerobjekte zu Ihrem selbstverwalteten Active Directory hinzuzufügen. Weitere Informationen finden Sie unter Fehler beim Versuch, Computer einer Domain hinzuzufügen in der Dokumentation zu Microsoft Windows Server.

> **Important**
>
> Verschieben Sie keine Computerobjekte, die RDS for SQL Server nach der Erstellung Ihrer DB-Instance in der Organisationseinheit erstellt. Wenn Sie die zugehörigen Objekte verschieben, wird Ihre RDS for SQL Server-DB-Instance falsch konfiguriert. Wenn Sie die von Amazon erstellten Computerobjekte verschieben müssenRDS, verwenden Sie die odifyDBInstanceRDSAPIM-Operation, um die Domänenparameter mit dem gewünschten Speicherort der Computerobjekte zu ändern.

Einschränkungen

Die folgenden Einschränkungen gelten für Self Managed AD for SQL Server.

- NTLMist der einzige unterstützte Authentifizierungstyp. Die Kerberos-Authentifizierung wird nicht unterstützt. Wenn Sie die Kerberos-Authentifizierung verwenden müssen, können Sie AWS Managed AD anstelle von selbstverwaltetem AD verwenden.
- Der Dienst Microsoft Distributed Transaction Coordinator (MSDTC) wird nicht unterstützt, da er eine Kerberos-Authentifizierung erfordert.
- Ihre RDS vier SQL Server-DB-Instances verwenden nicht den Network Time Protocol (NTP) - Server Ihrer selbstverwalteten AD-Domäne. Sie verwenden stattdessen einen AWS NTP Dienst.
- SQLSerververknüpfte Server müssen die SQL Authentifizierung verwenden, um eine Verbindung zu anderen RDS SQL For-Server-DB-Instances herzustellen, die zu Ihrer selbstverwalteten AD-Domäne gehören.
- Einstellungen für Microsoft Group Policy Object (GPO) aus Ihrer selbstverwalteten AD-Domäne werden nicht auf RDS SQL Server-DB-Instances angewendet.

Übersicht über die Einrichtung eines selbstverwalteten Active Directory

Um selbstverwaltetes AD RDS für eine SQL Server-DB-Instance einzurichten, führen Sie die folgenden Schritte aus, die unter ausführlicher beschrieben werden: Einrichten eines selbstverwalteten Active Directory

In Ihrer AD-Domain:

- Erstellen Sie eine Organisationseinheit.
- Erstellen Sie einen AD-Domain-Benutzer.
- Delegieren Sie die Kontrolle an den AD-Domain-Benutzer.

Aus dem AWS Management Console oderAPI:

- Erstellen Sie einen AWS KMS Schlüssel.
- Erstellen Sie mit AWS Secrets Manager ein Geheimnis.
- Erstellen oder ändern Sie eine RDS for SQL Server-DB-Instance und fügen Sie sie Ihrer selbstverwalteten AD-Domäne hinzu.

Einrichten eines selbstverwalteten Active Directory

Gehen Sie wie folgt vor, um ein selbstverwaltetes AD einzurichten.

Themen

- Schritt 1: Erstellen einer Organisationseinheit in Ihrem AD
- Schritt 2: Erstellen eines AD-Domain-Benutzers in Ihrem AD
- Schritt 3: Delegieren der Kontrolle an den AD-Benutzer
- Schritt 4: Erstellen Sie einen AWS KMS Schlüssel
- Schritt 5: Erstellen Sie ein AWS Geheimnis
- Schritt 6: Erstellen oder Ändern einer SQL-Server-DB-Instance
- Schritt 7: Erstellen von SQL-Server-Anmeldungen für die Windows-Authentifizierung

Schritt 1: Erstellen einer Organisationseinheit in Ihrem AD

> **Important**
>
> Wir empfehlen, für jedes AWS Konto, das eine RDS for SQL Server-DB-Instance besitzt, die Ihrer selbstverwalteten AD-Domäne beigetreten ist, eine dedizierte Organisationseinheit und Dienstanmeldedaten zu erstellen, die auf diese Organisationseinheit beschränkt sind. Durch die Zuordnung einer Organisationseinheit und von Service-Anmeldeinformationen können Sie widersprüchliche Berechtigungen vermeiden und dem Prinzip der geringsten Berechtigung folgen.

So erstellen Sie eine Organisationseinheit in Ihrem AD

1. Stellen Sie als Domain-Administrator eine Verbindung zu Ihrer AD-Domain her.
2. Öffnen Sie Active Directory-Benutzer und -Computer und wählen Sie die Domain aus, in der Sie Ihre Organisationseinheit erstellen möchten.
3. Klicken Sie mit der rechten Maustaste auf die Domain und wählen Sie Neu und dann Organisationseinheit aus.
4. Geben Sie einen Namen für die Organisationseinheit ein.
5. Lassen Sie das Kontrollkästchen für Container vor versehentlichem Löschen schützen aktiviert.
6. Klicken Sie auf OK. Ihre neue Organisationseinheit wird unter Ihrer Domain angezeigt.

Schritt 2: Erstellen eines AD-Domain-Benutzers in Ihrem AD

Die Anmeldeinformationen des Domänenbenutzers werden für das Geheimnis in AWS Secrets Manager verwendet.

So erstellen Sie einen AD-Domain-Benutzer in Ihrem AD

1. Öffnen Sie Active-Directory-Benutzer und -Computer und wählen Sie die Domain und die Organisationseinheit aus, in der Sie Ihren Benutzer erstellen möchten.
2. Klicken Sie mit der rechten Maustaste auf das Objekt Benutzer und wählen Sie Neu und dann Benutzer aus.
3. Geben Sie einen Vornamen, Nachnamen und Anmeldenamen für den Benutzer ein. Klicken Sie auf Weiter.

4. Geben Sie ein Passwort für den Benutzer ein. Wählen Sie nicht Benutzer muss das Passwort bei der nächsten Anmeldung ändern aus. Wählen Sie nicht Konto ist deaktiviert aus. Klicken Sie auf Weiter.
5. Klicken Sie auf OK. Ihr neuer Benutzer wird unter Ihrer Domain angezeigt.

Schritt 3: Delegieren der Kontrolle an den AD-Benutzer

So delegieren Sie die Kontrolle an den AD-Domain-Benutzer in Ihrer Domain

1. Öffnen Sie das MMC-Snap-In Active-Directory-Benutzer und -Computer und wählen Sie die Domain aus, in der Sie Ihren Benutzer erstellen möchten.
2. Klicken Sie mit der rechten Maustaste auf die Organisationseinheit, die Sie zuvor erstellt haben, und wählen Sie Kontrolle delegieren aus.
3. Klicken Sie im Assistenten für die Delegation der Kontrolle auf Weiter.
4. Klicken Sie im Abschnitt Benutzer oder Gruppen auf Hinzufügen.
5. Geben Sie im Abschnitt Benutzer, Computer oder Gruppen auswählen den von Ihnen erstellten AD-Benutzer ein und klicken Sie auf Namen überprüfen. Wenn Ihre AD-Benutzerprüfung erfolgreich ist, klicken Sie auf OK.
6. Bestätigen Sie im Abschnitt Benutzer oder Gruppen, dass Ihr AD-Benutzer hinzugefügt wurde, und klicken Sie auf Weiter.
7. Wählen Sie im Abschnitt Zu delegierende Aufgaben die Option Eine zu delegierende benutzerdefinierte Aufgabe erstellen aus und klicken Sie auf Weiter.
8. Gehen Sie im Abschnitt Active-Directory-Objekttyp wie folgt vor:
    a. Wählen Sie Nur die folgenden Objekte in dem Ordner aus.
    b. Wählen Sie Computerobjekte aus.
    c. Wählen Sie Ausgewählte Objekte in diesem Ordner erstellen aus.
    d. Wählen Sie Ausgewählte Objekte in diesem Ordner löschen aus und klicken Sie auf Weiter.
9. Gehen Sie im Abschnitt Berechtigungen wie folgt vor:
    a. Behalten Sie die Auswahl von Allgemein bei.
    b. Wählen Sie Überprüfter Schreibvorgang in den DNS-Hostnamen aus.
    c. Wählen Sie Überprüfter Schreibvorgang in den Service-Prinzipalnamen aus und klicken Sie auf Weiter.

10. Überprüfen und bestätigen Sie Ihre Einstellungen unter Den Assistenten für die Delegation der Kontrolle abschließen und klicken Sie auf Fertig stellen.

Schritt 4: Erstellen Sie einen AWS KMS Schlüssel

Der KMS-Schlüssel wird verwendet, um Ihr AWS Geheimnis zu verschlüsseln.

Um einen Schlüssel zu erstellen AWS KMS

> **Note**
>
> Verwenden Sie für den Verschlüsselungsschlüssel nicht den AWS Standard-KMS-Schlüssel. Stellen Sie sicher, dass Sie den AWS KMS Schlüssel in demselben AWS Konto erstellen, das die RDS for SQL Server-DB-Instance enthält, die Sie Ihrem selbstverwalteten AD hinzufügen möchten.

1. Wählen Sie in der AWS KMS Konsole Create Key aus.
2. Wählen Sie für Schlüsseltyp Symmetrisch aus.
3. Wählen Sie für Schlüsselnutzung die Option Verschlüsseln und Entschlüsseln aus.
4. Für Advanced options (Erweiterte Optionen):
   a. Wählen Sie unter Schlüsselmaterialursprung KMS aus.
   b. Wählen Sie für Regionalität Single-Region-Schlüssel aus und klicken Sie auf Weiter.
5. Geben Sie für Alias einen Namen für den KMS-Schlüssel an.
6. (Optional) Geben Sie unter Beschreibung eine Beschreibung des KMS-Schlüssels an.
7. (Optional) Geben Sie für Tags ein Tag für den KMS-Schlüssel an und klicken Sie auf Weiter.
8. Geben Sie für Schlüsseladministratoren den Namen eines IAM-Benutzers an und wählen Sie ihn aus.
9. Lassen Sie für Schlüssellöschung das Kontrollkästchen für Ermöglicht Schlüsseladministratoren das Löschen dieses Schlüssels aktiviert und klicken Sie auf Weiter.
10. Geben Sie für Schlüsselbenutzer den IAM-Benutzer aus dem vorherigen Schritt an und wählen Sie ihn aus. Klicken Sie auf Weiter.
11. Prüfen Sie die Konfiguration.
12. Fügen Sie für Schlüsselrichtlinie Folgendes zur Richtlinien-Anweisung hinzu:

```
{
    "Sid": "Allow use of the KMS key on behalf of RDS",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": "kms:Decrypt",
    "Resource": "*"
}
```

13. Klicken Sie auf Beenden.

Schritt 5: Erstellen Sie ein AWS Geheimnis

So erstellen Sie ein Secret

> **Note**
>
> Stellen Sie sicher, dass Sie das Geheimnis in demselben AWS Konto erstellen, das die RDS for SQL Server-DB-Instance enthält, die Sie Ihrem selbstverwalteten AD hinzufügen möchten.

1. Wählen Sie im AWS Secrets Manager die Option Neues Geheimnis speichern aus.
2. Als Secret-Typ wählen Sie Anderer Secret-Typ aus.
3. Fügen Sie für Schlüssel/Wert-Paare Ihre beiden Schlüssel hinzu:
    a. Geben Sie als ersten Schlüssel `SELF_MANAGED_ACTIVE_DIRECTORY_USERNAME` ein.
    b. Geben Sie für den Wert des ersten Schlüssels nur den Benutzernamen (ohne das Domainpräfix) des AD-Benutzers ein. Geben Sie den Domainnamen nicht an, da dies dazu führt, dass die Instanzerstellung fehlschlägt.
    c. Geben Sie als zweiten Schlüssel `SELF_MANAGED_ACTIVE_DIRECTORY_PASSWORD` ein.
    d. Geben Sie als Wert des zweiten Schlüssels das Passwort ein, das Sie für den AD-Benutzer in Ihrer Domain erstellt haben.
4. Geben Sie als Verschlüsselungsschlüssel den KMS-Schlüssel ein, den Sie in einem vorherigen Schritt erstellt haben, und klicken Sie auf Weiter.

5. Geben Sie als Secret-Name einen aussagekräftigen Namen ein, anhand dessen Sie das Secret später leichter finden können.
6. (Optional) Geben Sie im Feld Beschreibung eine Beschreibung für den Secret-Namen ein.
7. Klicken Sie unter Ressourcenberechtigung auf Bearbeiten.
8. Fügen Sie der Berechtigungsrichtlinie folgende Richtlinie hinzu:

> **Note**
>
> Wir empfehlen Ihnen, die `aws:sourceAccount` und `aws:sourceArn` Bedingungen in der Police zu verwenden, um das Problem des verwirrten Vertreters zu vermeiden. Verwenden Sie Ihre AWS-Konto für `aws:sourceAccount` und die RDS for SQL Server-DB-Instance ARN für`aws:sourceArn`. Weitere Informationen finden Sie unter Vermeidung des dienstübergreifenden Confused-Deputy-Problems.

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Principal":
            {
                "Service": "rds.amazonaws.com"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "aws:sourceAccount": "123456789012"
                },
                "ArnLike":
                {
                    "aws:sourceArn": "arn:aws:rds:us-
west-2:123456789012:db:*"
```

```
                }
            }
        }
    ]
}
```

9. Klicken Sie auf Speichern und dann auf Weiter.
10. Behalten Sie für Rotationseinstellungen konfigurieren die Standardwerte bei und wählen Sie Weiter aus.
11. Überprüfen Sie die Einstellungen für das Secret und klicken Sie auf Speichern.
12. Wählen Sie das von Ihnen erstellte Secret aus und kopieren Sie den Wert für den Secret-ARN. Dieser wird im nächsten Schritt bei der Einrichtung des selbstverwalteten Active Directory verwendet.

Schritt 6: Erstellen oder Ändern einer SQL-Server-DB-Instance

Sie können die Konsole, die CLI oder die RDS-API verwenden, um eine RDS-für- SQL-Server-DB-Instance mit einer selbstverwalteten AD-Domain zu verknüpfen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Erstellen Sie mit der Konsole, dem create-db-instanceCLI-Befehl oder dem Vorgang Create DBInstance RDS API eine neue SQL Server-DB-Instance.

  Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine vorhandene SQL Server-DB-Instance mithilfe der Konsole, des modify-db-instanceCLI-Befehls oder der Operation Modify DBInstance RDS API.

  Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie mithilfe der Konsole, des CLI-Befehls -db-snapshot oder des API-Vorgangs Restore From DBSnapshot RDS eine SQL restore-db-instance-fromServer-DB-Instance DBInstance aus einem DB-Snapshot wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.
- Stellen Sie eine SQL Server-DB-Instance point-in-time mithilfe der Konsole, des Befehls restore-db-instance-to- point-in-time CLI oder der Operation Restore DBInstance ToPointInTime RDS API in einer wieder her.

Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Wenn Sie den verwenden AWS CLI, sind die folgenden Parameter erforderlich, damit die DB-Instance die von Ihnen erstellte selbstverwaltete Active Directory-Domäne verwenden kann:

- Verwenden Sie für den Parameter `--domain-fqdn` den vollständig qualifizierten Domain-Namen (FQDN) Ihres selbstverwalteten Active Directory.
- Verwenden Sie für den Parameter `--domain-ou` die Organisationseinheit, die Sie in Ihrem selbstverwalteten AD erstellt haben.
- Verwenden Sie für den Parameter `--domain-auth-secret-arn` den Wert des Secret-ARN, den Sie in einem vorherigen Schritt erstellt haben.
- Verwenden Sie für den `--domain-dns-ips` Parameter die primären und sekundären IPv4 Adressen der DNS-Server für Ihr selbstverwaltetes AD. Wenn Sie keine sekundäre DNS-Server-IP-Adresse haben, geben Sie die primäre IP-Adresse zweimal ein.

Die folgenden CLI-Beispielbefehle zeigen, wie Sie eine RDS-für-SQL-Server-DB-Instance mit einer selbstverwalteten AD-Domain erstellen, ändern und entfernen.

> **Important**
>
> Wenn Sie eine DB-Instance ändern, um sie einer selbstverwalteten AD-Domain hinzuzufügen oder aus dieser zu entfernen, ist ein Neustart der DB-Instance erforderlich, damit die Änderung wirksam wird. Sie können wählen, ob Sie die Änderungen sofort übernehmen oder bis zum nächsten Wartungsfenster warten möchten. Wenn Sie die Option Sofort anwenden auswählen, führt dies bei einer Single-AZ-DB-Instance zu Ausfallzeiten. Eine Multi-AZ-DB-Instance führt ein Failover durch, bevor ein Neustart ausgeführt wird. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

Mit dem folgenden CLI-Befehl wird eine neue RDS-für-SQL-Server-DB-Instance erstellt und einer selbstverwalteten AD-Domain hinzugefügt.

Für LinuxmacOS, oderUnix:

```
aws rds create-db-instance \
```

```
    --db-instance-identifier my-DB-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 50 \
    --engine sqlserver-se \
    --engine-version 15.00.4043.16.v1 \
    --license-model license-included \
    --master-username my-master-username \
    --master-user-password my-master-password \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 50 ^
    --engine sqlserver-se ^
    --engine-version 15.00.4043.16.v1 ^
    --license-model license-included ^
    --master-username my-master-username ^
    --master-user-password my-master-password ^
    --domain-fqdn my-AD-test.my-AD.mydomain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \ ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Mit dem folgenden CLI-Befehl wird eine vorhandene RDS-für-SQL-Server-DB-Instance, so geändert, dass sie eine selbstverwaltete Active-Directory-Domain verwendet.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
```

```
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DBinstance ^
    --domain-fqdn my_AD_domain.my_AD.my_domain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Mit dem folgenden CLI-Befehl wird eine RDS-für-SQL-Server-DB-Instance aus einer selbstverwalteten Active-Directory-Domain entfernt.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --disable-domain
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --disable-domain
```

Schritt 7: Erstellen von SQL-Server-Anmeldungen für die Windows-Authentifizierung

Verwenden Sie die Anmeldeinformationen für den Amazon-RDS-Masterbenutzer, um eine Verbindung zur SQL-Server-DB-Instance herzustellen, wie Sie es bei jeder anderen DB-Instance tun würden. Da die DB-Instance der selbstverwalteten AD-Domain hinzugefügt wurde, können Sie SQL-Server-Anmeldungen und -Benutzer bereitstellen. Sie tun dies über das AD-Dienstprogramm für Benutzer und Gruppen in Ihrer selbstverwalteten AD-Domain. Datenbankberechtigungen werden über die Standard-SQL-Server-Berechtigungen verwaltet, die für die Windows-Anmeldungen gewährt und widerrufen werden.

Damit sich ein Benutzer des selbstverwalteten Active Directory mit SQL Server authentifizieren kann, muss eine SQL-Server-Windows-Anmeldung für den Benutzer des selbstverwalteten AD oder für eine Gruppe des selbstverwalteten AD, der der Benutzer angehört, vorliegen. Eine differenzierte

Zugriffskontrolle wird durch das Gewähren und Widerrufen von Berechtigungen für diese SQL Server-Anmeldungen gewährleistet. Ein Benutzer des selbstverwalteten AD, der über keine SQL-Server-Anmeldung verfügt oder zu keiner Gruppe des selbstverwalteten AD mit einer solchen Anmeldung gehört, kann nicht auf die SQL-Server-DB-Instance zugreifen.

Die Berechtigung ALTER ANY LOGIN ist erforderlich, um eine SQL-Server-Anmeldung für das selbstverwaltete AD zu erstellen. Wenn Sie mit dieser Berechtigung noch keine Anmeldungen erstellt haben, verbinden Sie sich mithilfe der SQL-Server-Authentifizierung als Masterbenutzer der DB-Instance und erstellen Sie Ihre SQL-Server-Anmeldungen für das selbstverwaltete AD im Kontext des Masterbenutzers.

Sie können einen Data Definition Language (DDL)-Befehl wie im Folgenden dargestellt ausführen, um eine SQL-Server-Anmeldung für einen Benutzer des selbstverwalteten Active Directory oder eine entsprechende Gruppe zu erstellen.

> **Note**
>
> Geben Sie Benutzer und Gruppen unter Verwendung des Anmeldenamens von Windows 2000 im Format a *my_AD_domain**my_AD_domain_user*. Sie können keinen User Principle Name (UPN) im Format verwenden *my_AD_domain_user*@*my_AD_domain*.

```
USE [master]
GO
CREATE LOGIN [my_AD_domain\my_AD_domain_user] FROM WINDOWS WITH DEFAULT_DATABASE =
 [master], DEFAULT_LANGUAGE = [us_english];
GO
```

Weitere Informationen finden Sie unter ANMELDENAME ERSTELLEN (Transact-SQL) in der Microsoft Developer Network-Dokumentation.

Benutzer (sowohl menschliche Benutzer als auch Anwendungen) aus Ihrer Domain können sich nun von einem einer selbstverwalteten AD-Domain hinzugefügten Client-Computer mithilfe der Windows-Authentifizierung mit der RDS-für-SQL-Server-Instance verbinden.

Verwalten einer DB-Instance in einer selbstverwalteten Active-Directory-Domain

Sie können die Konsole oder Amazon verwenden AWS CLI, RDS API um Ihre DB-Instance und ihre Beziehung zu Ihrer selbstverwalteten AD-Domain zu verwalten. Beispielsweise können Sie die DB-Instance in Domänen, aus Domänen oder zwischen Domänen verschieben.

Wenn Sie Amazon verwenden RDSAPI, können Sie beispielsweise Folgendes tun:

- Um erneut zu versuchen, aufgrund einer fehlgeschlagenen Mitgliedschaft einen selbstverwalteten Domain-Join durchzuführen, verwenden Sie die odifyDBInstanceAPIM-Operation und geben Sie denselben Parametersatz an:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Um eine DB-Instance aus einer selbstverwalteten Domäne zu entfernen, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie `--disable-domain` für den Domänenparameter Folgendes an.
- Um eine DB-Instance von einer selbstverwalteten Domain in eine andere zu verschieben, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie die Domänenparameter für die neue Domain an:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Verwenden Sie die Operation D escribeDBInstancesAPI, um die selbstverwaltete AD-Domänenmitgliedschaft für jede DB-Instance aufzulisten.

Grundlegendes zur Mitgliedschaft in einer selbstverwalteten Active-Directory-Domain

Nachdem Sie Ihre DB-Instance erstellt oder geändert haben, wird die Instance ein Mitglied der selbstverwalteten AD-Domain. Die AWS Konsole zeigt den Status der selbstverwalteten Active Directory-Domänenmitgliedschaft für die DB-Instance an. Der Status der DB-Instance kann einer der folgenden sein:

- joined – Die Instance ist Mitglied der AD-Domain.
- joining – Die Instance ist gerade dabei, Mitglied der AD-Domain zu werden.
- pending-join – Die Mitgliedschaft der Instance steht noch aus.
- pending-maintenance-join— versucht, AWS die Instance während des nächsten geplanten Wartungsfensters zu einem Mitglied der AD-Domäne zu machen.

- pending-removal – Das Entfernen der Instance von der AD-Domain steht noch aus.
- pending-maintenance-removal— versucht, AWS die Instanz während des nächsten geplanten Wartungsfensters aus der AD-Domäne zu entfernen.
- failed – Ein Konfigurationsproblem hat verhindert, dass die Instance der Domain beitreten konnte. Überprüfen und korrigieren Sie Ihre Konfiguration, bevor Sie den Befehl zu Änderung der Instance erneut ausführen.
- removing – Die Instance wird gerade von der selbstverwalteten AD-Domain entfernt.

Eine Anfrage, Mitglied einer selbstverwalteten AD-Domain zu werden, kann wegen eines Netzwerkverbindungsproblems fehlschlagen. Es könnte beispielsweise sein, dass Sie eine DB-Instance erstellen oder eine vorhandene Instance ändern und der Versuch, die DB-Instance zu einem Mitglied einer selbstverwalteten AD-Domain zu machen, fehlschlägt. Geben Sie in diesem Fall entweder den Befehl zum Erstellen oder Ändern der DB-Instance neu aus oder ändern Sie die neu erstellte Instance, damit sie der selbstverwalteten AD-Domain beitreten kann.

Fehlerbehebung für selbstverwaltetes Active Directory

Die folgenden Probleme können auftreten, wenn Sie ein selbstverwaltetes AD einrichten oder ändern.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 2 / 0x2	Die angegebene Datei wurde nicht gefunden.	Das Format oder der Speicherort für die Organisationseinheit, das/der mit dem Parameter –`domain-ou` angegeben wurde, ist ungültig. Das über AWS Secrets Manager angegebene Domänendienstkonto verfügt nicht über die erforderlichen Berechtigungen, um der Organisationseinheit beizutreten.	Überprüfen Sie den Parameter –`domain-ou` . Stellen Sie sicher, dass das Domain-Servicekonto über die richtigen Berechtigungen für die Organisationseinheit verfügt. Weitere Informationen finden Sie unter Konfigurieren Ihres AD-Domain-Servicekontos.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 5 / 0x5	Zugriff verweigert.	Falsch konfigurierte Berechtigungen für das Domain-Servicekonto oder das Computerkonto ist bereits in der Domain vorhanden.	Überprüfen Sie die Berechtigungen für das Domänendienstkonto in der Domäne und stellen Sie sicher, dass das RDS Computerkonto in der Domäne nicht doppelt vorhanden ist. Sie können den Namen des RDS Computerkontos überprüfen, indem Sie es `SELECT @@SERVERNAME` auf Ihrer RDS for SQL Server-DB-Instance ausführen. Wenn Sie Multi-AZ verwenden, versuchen Sie einen Neustart mit Failover und überprüfen Sie dann erneut, ob das RDS Computerkonto vorhanden ist. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 87 / 0x57	Der Parameter ist falsch.	Das über AWS Secrets Manager angegebene Domänendienstkonto verfügt nicht über die richtigen Berechtigungen. Auch das Benutzerprofil ist möglicherweise beschädigt.	Prüfen Sie die Anforderungen für das Domain-Servicekonto. Weitere Informationen finden Sie unter Konfigurieren Ihres AD-Domain-Servicekontos.
Fehler 234 / 0xEA	Die angegebene Organisationseinheit ist nicht vorhanden.	Die mit dem Parameter `–domain-ou` angegebene Organisationseinheit ist in Ihrem selbstverwalteten AD nicht vorhanden.	Überprüfen Sie den Parameter `–domain-ou` und stellen Sie sicher, dass die angegebene Organisationseinheit in Ihrem selbstverwalteten AD vorhanden ist.
Fehler 1326 / 0x52E	Der Benutzername oder das Passwort ist falsch.	Die in AWS Secrets Manager angegebenen Anmeldeinformationen für das Domänendienstkonto enthalten einen unbekannten Benutzernamen oder ein falsches Passwort. Möglicherweise ist das Domain-Konto auch in Ihrem selbstverwalteten AD deaktiviert.	Stellen Sie sicher, dass die in AWS Secrets Manager angegebenen Anmeldeinformationen korrekt sind und das Domänenkonto in Ihrem selbstverwalteten Active Directory aktiviert ist.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 1355 / 0x54B	Die angegebene Domain ist nicht vorhanden oder konnte nicht kontaktiert werden.	Die Domäne ist ausgefallen, die angegebenen Gruppen DNS IPs sind nicht erreichbar, oder die angegebene Domäne FQDN ist nicht erreichbar.	Stellen Sie sicher, dass die Parameter `–domain-dns-ips` und `–domain-fqdn` korrekt sind. Überprüfen Sie die Netzwerkkonfiguration Ihrer RDS for SQL Server-DB-Instance und stellen Sie sicher, dass Ihr selbstverwaltetes AD erreichbar ist. Weitere Informationen finden Sie unter Konfigurieren Ihrer Netzwerkkonnektivität.
Fehler 1722/0x6BA	Der RPC Server ist nicht verfügbar.	Beim Erreichen des RPC Dienstes Ihrer AD-Domain ist ein Problem aufgetreten. Dies könnte auf einen Service- oder Netzwerkfehler zurückzuführen sein.	Stellen Sie sicher, dass der RPC Dienst auf Ihren Domänencontrollern ausgeführt wird und dass die TCP Ports `135` und in Ihrer Domain von Ihrer RDS for SQL Server-DB-Instance aus erreichbar `49152-65535` sind.
Fehler 2224 / 0x8B0	Das Benutzerkonto besteht bereits.	Das Computerkonto, das zu Ihrem selbstverwalteten AD hinzugefügt werden soll, ist bereits vorhanden.	Identifizieren Sie das Computerkonto, indem Sie es `SELECT @@SERVERNAME` auf Ihrer RDS for SQL Server-DB-Instance ausführen, und entfernen Sie es dann vorsichtig aus Ihrem selbstverwalteten AD.

Fehlercode	Beschreibung	Häufige Ursachen	Vorschläge für die Fehlerbehebung
Fehler 2242 / 0x8c2	Das Passwort dieses Benutzers ist abgelaufen.	Das Passwort für das über AWS Secrets Manager angegebene Domänendienstkonto ist abgelaufen.	Aktualisieren Sie das Passwort für das Domänendienstkonto, das verwendet wurde, um Ihre RDS for SQL Server-DB-Instance mit Ihrem selbstverwalteten AD zu verbinden.

Wiederherstellung einer SQL Server-DB-Instance und anschließendes Hinzufügen zu einer selbstverwalteten Active Directory-Domäne

Sie können einen DB-Snapshot wiederherstellen oder point-in-time recovery (PITR) für eine SQL Server-DB-Instance durchführen und sie dann zu einer selbstverwalteten Active Directory-Domäne hinzufügen. Wenn die DB-Instance wiederhergestellt wurde, ändern Sie die Instance, indem Sie den unter Schritt 6: Erstellen oder Ändern einer SQL-Server-DB-Instance beschriebenen Prozess anwenden, um die DB-Instance einer selbstverwalteten AD-Domain hinzuzufügen.

Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server

Sie können AWS Managed Microsoft AD es verwenden, um Benutzer mit der Windows-Authentifizierung zu authentifizieren, wenn sie eine Verbindung zu Ihrer RDS for SQL Server-DB-Instance herstellen. Die DB-Instance funktioniert mit AWS Directory Service for Microsoft Active Directory, wird auch genannt AWS Managed Microsoft AD, um die Windows-Authentifizierung zu aktivieren. Wenn Benutzer sich mit einer SQL Server-DB-Instance authentifizieren, die mit einer vertrauenswürdigen Domäne verbunden ist, werden die Authentifizierungsanfragen an das Domänenverzeichnis weitergeleitet, das Sie mit erstellt habe AWS Directory Service.

Verfügbarkeit von Regionen und Versionen

Amazon RDS unterstützt nur die Verwendung AWS Managed Microsoft AD für die Windows-Authentifizierung. RDS unterstützt die Verwendung nicht AD Connector. Weitere Informationen finden Sie unter:

- Richtlinie zur Anwendungskompatibilität für AWS Managed Microsoft AD
- Richtlinie zur Anwendungskompatibilität für AD-Connector

Informationen zur Versions- und Regionsverfügbarkeit finden Sie unter Kerberos-Authentifizierung mit RDS für SQL Server.

Übersicht zur Einrichtung einer Windows-Authentifizierung

Amazon RDS verwendet einen gemischten Modus für die Windows-Authentifizierung. Mit dieser Vorgehensweise verwendet der Hauptbenutzer (zum Erstellen der SQL Server-DB-Instance verwendeter Name und verwendetes Passwort) die SQL-Authentifizierung. Da das Masterbenutzerkonto Anmeldedaten mit besonderen Berechtigungen enthält, sollte der Zugriff auf dieses Konto beschränkt sein.

Um mithilfe eines lokalen oder selbst gehosteten Microsoft Active Directory Windows-Authentifizierung abzurufen, erstellen Sie eine Gesamtstruktur-Vertrauensstellung. Die Vertrauensstellung kann uni- oder bidirektional sein. Weitere Informationen zum Einrichten von Forest Trusts mithilfe von AWS Directory Service finden Sie unter Wann sollte eine Vertrauensstellung erstellt werden? im AWS Directory Service Administratorhandbuch.

Führen Sie die folgenden Schritte aus, um eine Windows-Authentifizierung für eine SQL Server-DB-Instance einzurichten. Diese Schritt werden unter noch genauer erklär Einrichten einer Windows-Authentifizierung für SQL Server-DB-Instances:

1. Verwenden Sie AWS Managed Microsoft AD entweder über die AWS Management Console oder die AWS Directory Service API, um ein AWS Managed Microsoft AD Verzeichnis zu erstellen.
2. Wenn Sie die AWS CLI oder Amazon RDS-API verwenden, um Ihre SQL Server-DB-Instance zu erstellen, erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle. Diese Rolle verwendet die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` und ermöglicht es Amazon RDS, Aufrufe an Ihr Verzeichnis durchzuführen. Wenn Sie die Konsole zum Erstellen der SQL Server-DB-Instance verwenden, erstellt AWS die IAM-Rolle für Sie.

   Damit die Rolle Zugriff gewährt, muss der Endpunkt AWS Security Token Service (AWS STS) in der AWS Region für Ihr AWS Konto aktiviert sein. AWS STS Endpunkte sind standardmäßig in allen AWS Regionen aktiv, und Sie können sie ohne weitere Aktionen verwenden. Weitere Informationen finden Sie unter Verwalten von AWS STS in einer AWS-Region im IAM-Benutzerhandbuch.
3. Erstellen und konfigurieren Sie Benutzer und Gruppen im AWS Managed Microsoft AD Verzeichnis mithilfe der Microsoft Active Directory-Tools. Weitere Informationen zum Erstellen von Benutzern und Gruppen in Ihrem Active Directory finden Sie unter Verwalten von Benutzern und Gruppen in AWS Managed Microsoft AD im AWS Directory Service Administrationshanbuch.
4. Wenn Sie planen, das Verzeichnis und die DB-Instance an unterschiedlichen Orten zu lokalisieren VPCs, aktivieren Sie den vPC-übergreifenden Verkehr.
5. Verwenden Sie Amazon RDS, um eine neue SQL Server-DB-Instance entweder über die Konsole oder über die Amazon RDS-API zu erstellen. AWS CLI In der Anforderung zum Erstellen geben Sie den Domänenbezeichner an ("`d-*`"-Bezeichner), der beim Erstellen des Verzeichnisses generiert wurde, sowie den Namen der von Ihnen erstellten Rolle. Sie können auch eine vorhandene SQL Server-DB-Instance für die Verwendung der Windows-Authentifizierung ändern, indem Sie die Domänen- und IAM-Rollenparameter für die DB-Instance festlegen.
6. Verwenden Sie die Anmeldeinformationen für den Amazon RDS-Hauptbenutzer, um eine Verbindung zur SQL Server-DB-Instance herzustellen, wie Sie es bei jeder anderen DB-Instance tun würden. Da die DB-Instance mit der AWS Managed Microsoft AD Domäne verbunden ist, können Sie SQL Server-Logins und Benutzer aus den Active Directory-Benutzern und -Gruppen in ihrer Domäne bereitstellen. (Diese werden als SQL Server "Windows"-Anmeldungen bezeichnet.) Datenbankberechtigungen werden über die Standard-SQL-Server-Berechtigungen verwaltet, die für die Windows-Anmeldungen gewährt und widerrufen werden.

Wenn Sie mithilfe der Amazon RDS-Konsole eine mit einer Domäne verbundene RDS for SQL Server-DB-Instance erstellen, AWS wird automatisch die `rds-directoryservice-access-role`

IAM-Rolle erstellt. Diese Rolle ist für die Verwaltung domänenverbundener Instances unerlässlich und für die folgenden Vorgänge erforderlich:

- Vornehmen von Konfigurationsänderungen an mit einer Domäne verbundenen SQL Server-Instanzen
- Verwaltung der Active Directory-Integrationseinstellungen
- Durchführung von Wartungsvorgängen an domänengebundenen Instanzen

> **Important**
>
> Wenn Sie die `rds-directoryservice-access-role` IAM-Rolle löschen, können Sie über die Amazon RDS-Konsole oder API keine Änderungen an Ihrer mit der Domäne verbundenen SQL Server-Instance vornehmen. Der Versuch, die Instance zu ändern, führt zu einer Fehlermeldung, die besagt: Sie haben keine Berechtigung für iam:. CreateRole Um Zugriff anzufordern, kopieren Sie den folgenden Text und senden Sie ihn an Ihren AWS Administrator.
> Dieser Fehler tritt auf, weil Amazon RDS die Rolle zur Verwaltung der Domänenverbindung neu erstellen muss, aber nicht über die erforderlichen Berechtigungen verfügt. Darüber hinaus ist dieser Fehler nicht angemeldet CloudTrail, was die Fehlerbehebung erschweren kann.

Wenn Sie den versehentlich löschen`rds-directoryservice-access-role`, müssen Sie über die `iam:CreateRole` erforderlichen Berechtigungen verfügen, um ihn erneut zu erstellen, bevor Sie Änderungen an Ihrer mit der Domäne verbundenen SQL Server-Instanz vornehmen können. Um die Rolle manuell neu zu erstellen, stellen Sie sicher, dass ihr die `AmazonRDSDirectoryServiceAccess` verwaltete Richtlinie und die entsprechende Vertrauensstellung zugeordnet sind, die es dem RDS-Dienst ermöglicht, die Rolle zu übernehmen.

Erstellen des Endpunkts für die Kerberos-Authentifizierung

Für die Kerberos-basierte Authentifizierung muss der Endpunkt der vom Kunden angegebene Hostname, ein Punkt und dann der vollqualifizierte Domänenname () sein. FQDN Im Folgenden sehen Sie ein Beispiel für einen Endpunkt, den Sie mit Kerberos-basierter Authentifizierung verwenden können. In diesem Beispiel lautet der Hostname der SQL Server-DB-Instance `ad-test` und der Domänenname lautet. `corp-ad.company.com`

```
ad-test.corp-ad.company.com
```

Wenn Sie überprüfen möchten, ob Ihre Verbindung Kerberos verwendet, führen Sie die folgende Abfrage aus:

```
SELECT net_transport, auth_scheme
  FROM sys.dm_exec_connections
 WHERE session_id = @@SPID;
```

Einrichten einer Windows-Authentifizierung für SQL Server-DB-Instances

Sie verwenden AWS Directory Service for Microsoft Active Directory, auch genannt AWS Managed Microsoft AD, um die Windows-Authentifizierung für eine SQL Server-DB-Instance einzurichten. Um die Windows-Authentifizierung einzurichten, führen Sie folgende Schritte aus.

Schritt 1: Erstellen Sie ein Verzeichnis mit dem AWS Directory Service for Microsoft Active Directory

AWS Directory Service erstellt ein vollständig verwaltetes Microsoft Active Directory in der AWS Cloud. Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service erstellt in Ihrem Namen zwei Domänencontroller und DNS-Server (Domain Name Service). Die Verzeichnis-Server werden in zwei Subnetzen in zwei verschiedenen Availability Zones in einer VPC erstellt. Diese Redundanz hilft sicherzustellen, dass Ihr Verzeichnis verfügbar bleibt, auch wenn ein Fehler auftritt.

Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service führt er in Ihrem Namen die folgenden Aufgaben aus:

- Richtet ein Microsoft Active Directory in der VPC ein.
- Erstellt ein Verzeichnisadministratorkonto mit dem Benutzernamen Admin und dem angegebenen Passwort. Mit diesem Konto verwalten Sie das Verzeichnis.
- Erstellt eine Sicherheitsgruppe für die Verzeichniscontroller.

Wenn Sie eine starten AWS Directory Service for Microsoft Active Directory, AWS erstellt eine Organisationseinheit (OU), die alle Objekte Ihres Verzeichnisses enthält. Diese OU erhält den NetBIOS-Namen, den Sie beim Erstellen des Verzeichnisses eingegeben haben, und befindet sich im Domainstamm. Der Domänenstamm gehört und wird von verwaltet AWS.

Das Admin-Konto, das zusammen mit dem Verzeichnis AWS Managed Microsoft AD erstellt wurde, verfügt über die Berechtigungen für die häufigsten administrativen Aufgaben in Bezug auf Ihre OU:

- Erstellen, Aktualisieren oder Löschen von Benutzern, Gruppen und Computern
- Hinzufügen von Ressourcen zu Ihrer Domäne, etwa Datei- oder Druckserver, und anschließendes Gewähren der zugehörigen Ressourcenberechtigungen für Benutzer und Gruppen in der OU.
- Erstellen Sie zusätzliche OUs Container.
- Delegieren von Befugnissen.
- Erstellen und Verknüpfen von Gruppenrichtlinien.
- Wiederherstellen von gelöschten Objekten aus dem Active Directory-Papierkorb.
- Führen Sie AD- und PowerShell DNS-Windows-Module im Active Directory-Webdienst aus.

Das Admin-Konto verfügt zudem über die Rechte zum Ausführen der folgenden domänenübergreifenden Aktivitäten:

- Verwalten von DNS-Konfigurationen (Hinzufügen, Entfernen oder Aktualisieren von Datensätzen, Zonen und Weiterleitungen).
- Aufrufen von DNS-Ereignisprotokollen.
- Anzeigen von Sicherheitsereignisprotokollen.

Um ein Verzeichnis zu erstellen mit AWS Managed Microsoft AD

1. Wählen Sie im Navigationsbereich der AWS Directory Service -Konsole die Option Directories (Verzeichnisse) und dann Set up directory (Verzeichnis einrichten) aus.
2. Wählen Sie AWS Managed Microsoft AD. Dies ist zurzeit die einzige für Amazon RDS unterstützte Option.
3. Wählen Sie Weiter aus.
4. Geben Sie auf der Seite Enter directory information (Verzeichnisinformationen eingeben) die folgenden Informationen ein:

   Edition

   Wählen Sie die Edition aus, die Ihre Anforderungen erfüllt.

   DNS-Name des Verzeichnisses

   Den vollständig qualifizierten Namen für das Verzeichnis, z. B. `corp.example.com`. Namen, die länger als 47 Zeichen sind, werden von SQL Server nicht unterstützt.

NetBIOS-Name des Verzeichnisses

Ein optionaler Kurzname für das Verzeichnis, z. B. `CORP`.

Verzeichnisbeschreibung

Eine optionale Beschreibung des Verzeichnisses.

Administratorpasswort

Das Passwort für den Verzeichnisadministrator. Während des Verzeichniserstellungsprozesses wird ein Administratorkonto mit dem Benutzernamen Admin und diesem Passwort angelegt.

Das Passwort für den Verzeichnisadministrator darf nicht das Wort „“ enthalte `admin`. Beachten Sie beim Passwort die Groß- und Kleinschreibung und es muss 8 bis 64 Zeichen lang sein. Zudem muss es mindestens ein Zeichen aus dreien der vier folgenden Kategorien enthalten:

- Kleinbuchstaben (a – z)
- Großbuchstaben (A – Z)
- Zahlen (0 – 9)
- Nicht-alphanumerische Zeichen (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Passwort bestätigen

Geben Sie das Administratorpasswort erneut ein.

5. Wählen Sie Weiter aus.
6. Geben Sie auf der Seite Choose VPC and subnets (VPC und Subnetze wählen) die folgenden Informationen an.

VPC

Wählen Sie die VPC für das Verzeichnis aus.

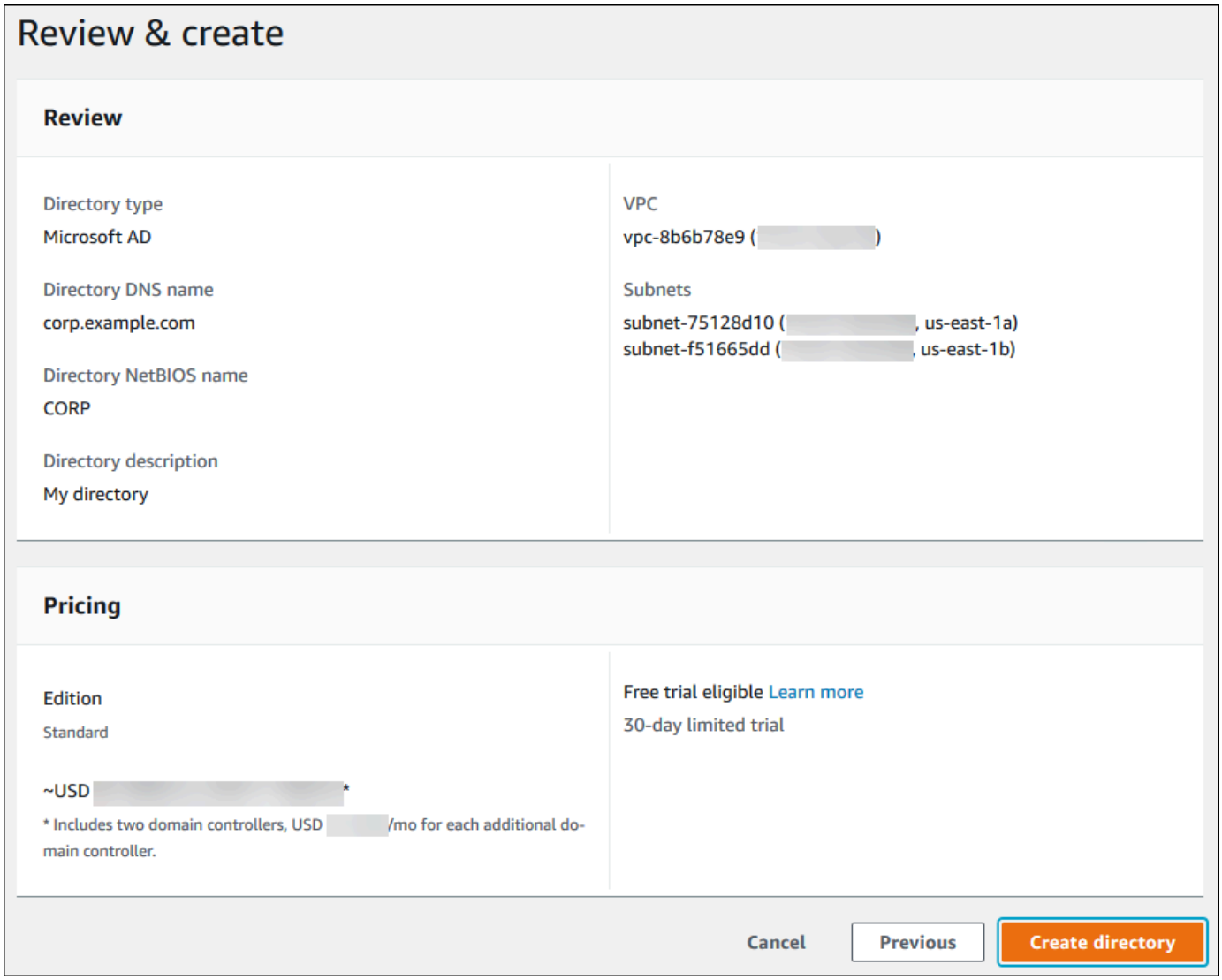
> **Note**
>
> Sie können das Verzeichnis und die DB-Instance an unterschiedlichen Orten lokalisieren. Wenn Sie dies tun VPCs, stellen Sie jedoch sicher, dass Sie den VPC-übergreifenden Verkehr aktivieren. Weitere Informationen finden Sie unter Schritt 4:

[Aktivieren des VPC-übergreifenden Datenverkehrs zwischen dem Verzeichnis und der DB-Instance](#).

Subnetze

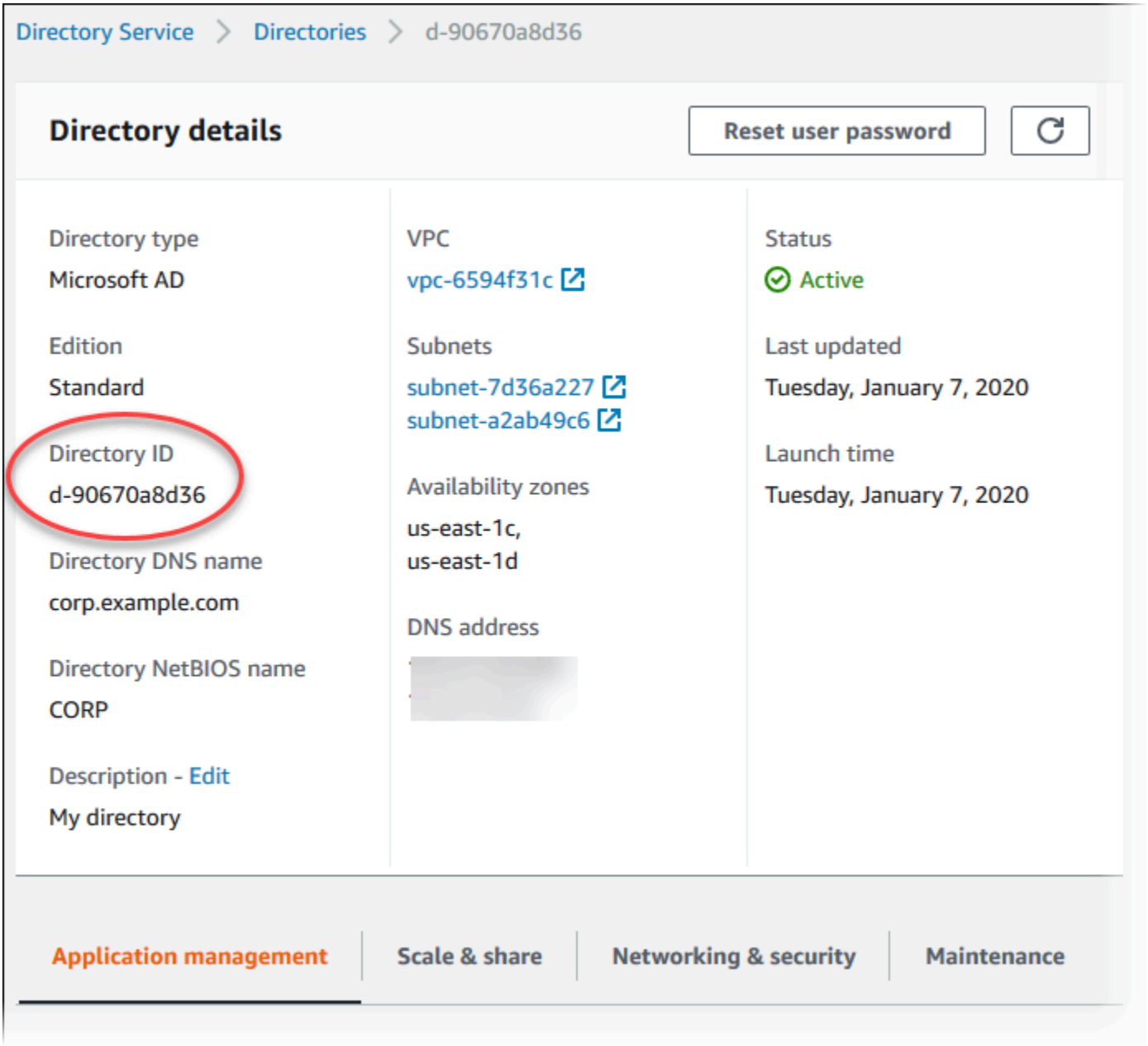
Wählen Sie Subnetze für die Verzeichnis-Server aus. Die beiden Subnetze müssen zu verschiedenen Availability-Zonen gehören.

7. Wählen Sie Weiter aus.
8. Überprüfen Sie die Verzeichnisinformationen. Falls Sie noch Änderungen vornehmen möchten, klicken Sie auf Previous (Zurück). Wenn die Informationen richtig sind, wählen Sie Create directory (Verzeichnis erstellen).



Es dauert einige Minuten, bis das Verzeichnis erstellt wurde. Wenn es erfolgreich erstellt wurde, ändert sich der Wert Status in Active (Aktiv).

Um Informationen über das Verzeichnis anzuzeigen, wählen Sie die Verzeichnis-ID in der Verzeichnisauflistung aus. Notieren Sie sich den Wert Directory ID (Verzeichnis-ID). Sie benötigen diesen Wert, wenn Sie die SQL Server-DB-Instance erstellen oder modifizieren.



Schritt 2: Erstellen der IAM-Rolle für die Verwendung durch Amazon RDS

Wenn Sie die Konsole zum Erstellen der SQL Server-DB-Instance verwenden, können Sie diesen Schritt überspringen. Wenn Sie zum Erstellen der SQL Server-DB-Instance die CLI oder

RDS-API verwenden, müssen Sie eine IAM-Rolle erstellen, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet. Diese Rolle ermöglicht es Amazon RDS, AWS Directory Service für Sie Anrufe an die zu tätigen.

Wenn Sie eine benutzerdefinierte Richtlinie für den Beitritt zu einer Domain verwenden, anstatt die AWS-verwaltete `AmazonRDSDirectoryServiceAccess` Richtlinie zu verwenden, stellen Sie sicher, dass Sie die `ds:GetAuthorizedApplicationDetails` Aktion zulassen. Diese Anforderung gilt aufgrund einer Änderung der AWS Directory Service API ab Juli 2019.

Die folgende IAM-Richtlinie (`AmazonRDSDirectoryServiceAccess`) ermöglicht den Zugriff auf AWS Directory Service.

Example IAM-Richtlinie für die Bereitstellung des Zugriffs auf AWS Directory Service

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
            "ds:DescribeDirectories",
            "ds:AuthorizeApplication",
            "ds:UnauthorizeApplication",
            "ds:GetAuthorizedApplicationDetails"
        ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Vertrauensbeziehungen, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Sie können beide globalen Bedingungskontextschlüssel verwenden und der Wert `aws:SourceArn` enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der Wert `aws:SourceAccount`

und das Konto im Wert `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Anweisung verwendet werden.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Stellen Sie in der Vertrauensbeziehung sicher, dass Sie den globalen Bedingungskontextschlüssel `aws:SourceArn` mit dem vollständigen Amazon-Ressourcennamen (ARN) der Ressourcen verwenden, die auf die Rolle zugreifen. Stellen Sie für die Windows-Authentifizierung sicher, dass Sie die DB-Instances einschließen, wie im folgenden Beispiel gezeigt.

Example Vertrauensbeziehung mit dem globalen Bedingungskontextschlüssel für die Windows-Authentifizierung

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [

 "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    ]
                }
            }
        }
    ]
}
```

Erstellen Sie eine IAM-Rolle mit dieser IAM-Richtlinie und Vertrauensbeziehung. Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter Erstellen von kundenverwalteten Richtlinien im IAM-Benutzerhandbuch.

Schritt 3: Erstellen und Konfigurieren von Benutzern und Gruppen

Sie können Benutzer und Gruppen mit dem Tool "Active Directory-Benutzer und -Computer" erstellen. Dieses Tool ist eines der "Active Directory Domain Services"- und "Active Directory Lightweight Directory Services"-Tools. „Benutzer“ sind Einzelpersonen oder Entitäten, die Zugriff auf Ihr Verzeichnis haben. Gruppen sind sehr nützlich, um Berechtigungen zu erteilen oder zu verweigern, anstatt diese Berechtigungen für jeden einzelnen Benutzer erstellen zu müssen.

Um Benutzer und Gruppen in einem AWS Directory Service Verzeichnis zu erstellen, müssen Sie mit einer EC2 Windows-Instanz verbunden sein, die Mitglied des AWS Directory Service Verzeichnisses ist. Außerdem müssen Sie als Benutzer angemeldet sein, der über Rechte zum Erstellen von Benutzern und Gruppen verfügt. Weitere Informationen finden Sie unter Hinzufügen von Benutzern und Gruppen (Simple AD und AWS Managed Microsoft AD) im AWS Directory Service Administratorhandbuch.

Schritt 4: Aktivieren des VPC-übergreifenden Datenverkehrs zwischen dem Verzeichnis und der DB-Instance

Wenn Sie beabsichtigen, das Verzeichnis und die DB-Instance in derselben VPC zu platzieren, überspringen Sie diesen Schritt und fahren Sie mit for Schritt 5: Erstellen oder Modifizieren einer SQL Server-DB-Instance.

Wenn Sie planen, das Verzeichnis und die DB-Instance unterschiedlich zu lokalisieren VPCs, konfigurieren Sie den VPC-übergreifenden Verkehr mithilfe von VPC-Peering oder Transit Gateway.AWS

Das folgende Verfahren ermöglicht den Datenverkehr zwischen VPCs mithilfe von VPC-Peering. Folgen Sie den Anweisungen unter Was ist VPC Peering? im Handbuch zu Amazon Virtual Private Cloud-Peering.

Aktivieren des VPC-übergreifenden Datenverkehrs mit VPC Peering

1. Richten Sie geeignete VPC-Routing-Regeln ein, um sicherzustellen, dass Netzwerk-Datenverkehr in beide Richtungen fließen kann.
2. Stellen Sie sicher, dass die Sicherheitsgruppe der DB-Instance eingehenden Datenverkehr von der Sicherheitsgruppe des Verzeichnisses empfangen kann.

3. Stellen Sie sicher, dass keine ACL-Regel (Network Access Control List) zum Blockieren des Datenverkehrs vorhanden ist.

Wenn ein anderes AWS Konto Eigentümer des Verzeichnisses ist, müssen Sie das Verzeichnis gemeinsam nutzen.

Um das Verzeichnis von mehreren AWS Konten gemeinsam zu nutzen

1. Beginnen Sie mit der gemeinsamen Nutzung des Verzeichnisses mit dem AWS Konto, unter dem die DB-Instance erstellt werden soll. Folgen Sie dazu den Anweisungen im AWS Directory Service Administratorhandbuch im Tutorial: AWS Managed Microsoft AD Verzeichnis teilen für einen nahtlosen EC2 Domänenbeitritt.
2. Melden Sie sich mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole an und stellen Sie sicher, dass die Domain den `SHARED` Status hat, bevor Sie fortfahren.
3. Notieren Sie sich den Wert der Verzeichnis-ID, während Sie mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole angemeldet sind. Sie verwenden diese Verzeichnis-ID, um die DB-Instance mit der Domäne zu verbinden.

Schritt 5: Erstellen oder Modifizieren einer SQL Server-DB-Instance

Erstellen oder ändern Sie eine SQL Server-DB-Instance für die Verwendung mit Ihrem Verzeichnis. Sie können die Konsole, CLI oder RDS-API verwenden, um eine DB-Instance einem Verzeichnis zuzuordnen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Erstellen Sie mit der Konsole, dem create-db-instanceCLI-Befehl oder dem Vorgang Create DBInstance RDS API eine neue SQL Server-DB-Instance.

  Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine vorhandene SQL Server-DB-Instance mithilfe der Konsole, des modify-db-instanceCLI-Befehls oder der Operation Modify DBInstance RDS API.

  Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie mithilfe der Konsole, des CLI-Befehls -db-snapshot oder des API-Vorgangs Restore From DBSnapshot RDS eine SQL restore-db-instance-fromServer-DB-Instance DBInstance aus einem DB-Snapshot wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.

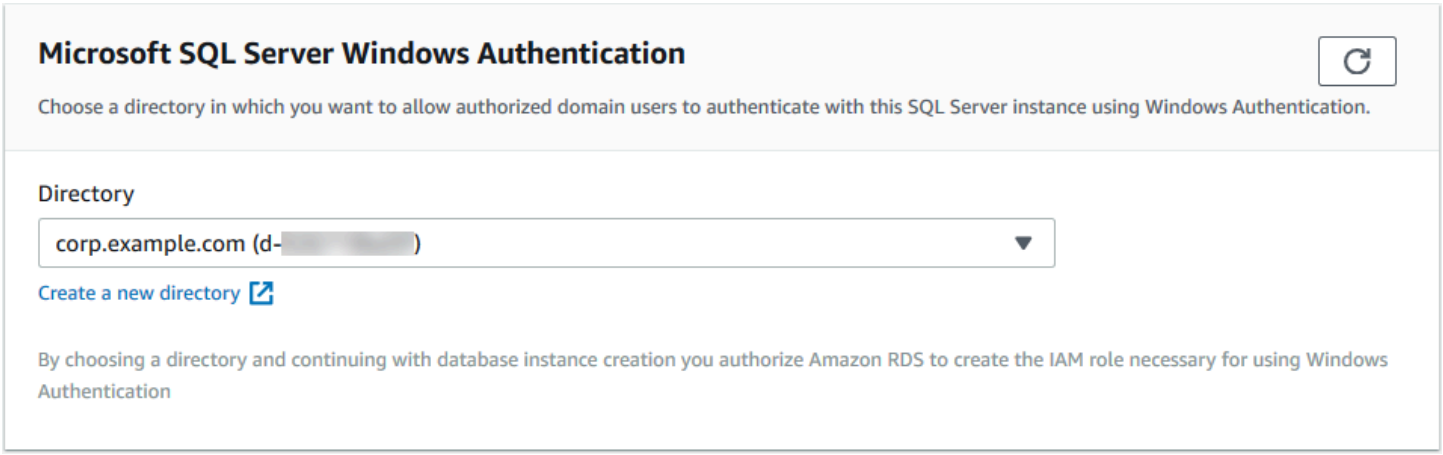
- Stellen Sie eine SQL Server-DB-Instance point-in-time mithilfe der Konsole, des Befehls restore-db-instance-to- point-in-time CLI oder der Operation Restore DBInstance ToPointInTime RDS API in einer wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Die Windows-Authentifizierung wird nur für SQL Server-DB-Instances in einer VPC unterstützt.

Damit die DB-Instance das von Ihnen erstellte Domänenverzeichnis verwenden kann, ist Folgendes erforderlich:

- Für Directory (Verzeichnis) müssen Sie den Domänenbezeichner (`d-`*`ID`*) auswählen, der beim Erstellen des Verzeichnisses generiert wurde.
- Stellen Sie sicher, dass die VPC-Sicherheitsgruppe über eine ausgehende Regel verfügt, mit der die DB-Instance mit dem Verzeichnis kommunizieren kann.



Wenn Sie den verwenden AWS CLI, sind die folgenden Parameter erforderlich, damit die DB-Instance das von Ihnen erstellte Verzeichnis verwenden kann:

- Für den `--domain`-Parameter verwenden Sie den Domänenbezeichner (`d-`*`ID`*), der beim Erstellen des Verzeichnisses generiert wurde.
- Verwenden Sie für den `--domain-iam-role-name`-Parameter die von Ihnen erstellte Rolle, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet.

Beispielsweise ändert der folgende CLI-Befehl eine DB-Instance zur Verwendung eines Verzeichnisses.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Wenn Sie eine DB-Instance zur Aktivierung der Kerberos-Authentifizierung ändern, starten Sie die DB-Instance neu, nachdem Sie die Änderung vorgenommen haben.

Schritt 6: Erstellen von SQL Server-Anmeldungen für die Windows-Authentifizierung

Verwenden Sie die Anmeldeinformationen für den Amazon RDS-Hauptbenutzer, um eine Verbindung zur SQL Server-DB-Instance herzustellen, wie Sie es bei jeder anderen DB-Instance tun würden. Da die DB-Instance mit der AWS Managed Microsoft AD Domäne verbunden ist, können Sie SQL Server-Logins und -Benutzer bereitstellen. Sie tun dies über die Active Directory-Benutzer und -Gruppen in Ihrer Domäne. Datenbankberechtigungen werden über die Standard-SQL-Server-Berechtigungen verwaltet, die für die Windows-Anmeldungen gewährt und widerrufen werden.

Damit ein Active Directory-Benutzer sich mit SQL Server authentifizieren kann, muss eine SQL Server-Windows-Anmeldung für den Benutzer oder für eine Gruppe, der der Benutzer angehört, vorliegen. Eine differenzierte Zugriffskontrolle wird durch das Gewähren und Widerrufen von Berechtigungen für diese SQL Server-Anmeldungen gewährleistet. Ein Benutzer, der keine SQL Server-Anmeldung hat oder zu keiner Gruppe mit einer solchen Anmeldung gehört, kann nicht auf die SQL Server-DB-Instance zugreifen.

Die Berechtigung ALTER ANY LOGIN ist erforderlich, um eine SQL Server-Anmeldung für das Active-Directory zu erstellen. Wenn Sie mit dieser Berechtigung noch keine Anmeldungen erstellt haben, verbinden Sie sich mithilfe der SQL Server-Authentifizierung als Masterbenutzer der DB-Instance.

Führen Sie den folgenden Data Definition Language (DDL)-Befehl aus, um eine SQL Server-Anmeldung für einen Active-Directory-Benutzer oder eine entsprechende Gruppe zu erstellen.

> **Note**
>
> Geben Sie Benutzer und Gruppen unter Verwendung des Anmeldenamens von Windows 2000 im Format a *domainName**login_name*. Sie können keinen User Principle Name (UPN) im Format verwenden *login_name*@*DomainName*.
> Sie können eine Windows-Authentifizierungsanmeldung auf einer RDS für SQL Server-Instanz nur mithilfe von T-SQL-Anweisungen erstellen. Sie können das SQL Server Management Studio nicht verwenden, um eine Windows-Authentifizierungs-Anmeldung zu erstellen.

```
USE [master]
GO
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
 DEFAULT_LANGUAGE = [us_english];
GO
```

Weitere Informationen finden Sie unter ANMELDENAME ERSTELLEN (Transact-SQL) in der Microsoft Developer Network-Dokumentation.

Benutzer (sowohl menschliche Benutzer als auch Anwendungen) von Ihrer Domäne können sich nun von einem über eine Domäne verbundenen Client-Computer mithilfe der Windows-Authentifizierung an der RDS-for-SQL-Server-Instance anmelden.

Verwalten einer DB-Instance in einer Domäne

Sie können die Konsole oder Amazon verwenden AWS CLI, RDS API um Ihre DB-Instance und ihre Beziehung zu Ihrer Domain zu verwalten. Beispielsweise können Sie die DB-Instance in Domänen, aus Domänen oder zwischen Domänen verschieben.

Wenn Sie Amazon verwenden RDSAPI, können Sie beispielsweise Folgendes tun:

- Um einen erneuten Domainbeitritt für eine fehlgeschlagene Mitgliedschaft zu versuchen, verwenden Sie die odifyDBInstanceAPIM-Operation und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft an.
- Um den IAM Rollennamen für die Mitgliedschaft zu aktualisieren, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft und die neue IAM Rolle an.
- Um eine DB-Instance aus einer Domäne zu entfernen, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie `none` als Domänenparameter an.
- Um eine DB-Instance von einer Domain in eine andere zu verschieben, verwenden Sie den `ModifyDBInstance` API Vorgang und geben Sie den Domain-Identifier der neuen Domain als Domänenparameter an.
- Verwenden Sie die escribeDBInstances API Operation D, um die Mitgliedschaft für jede DB-Instance aufzulisten.

Grundlegendes zur Domänenmitgliedschaft

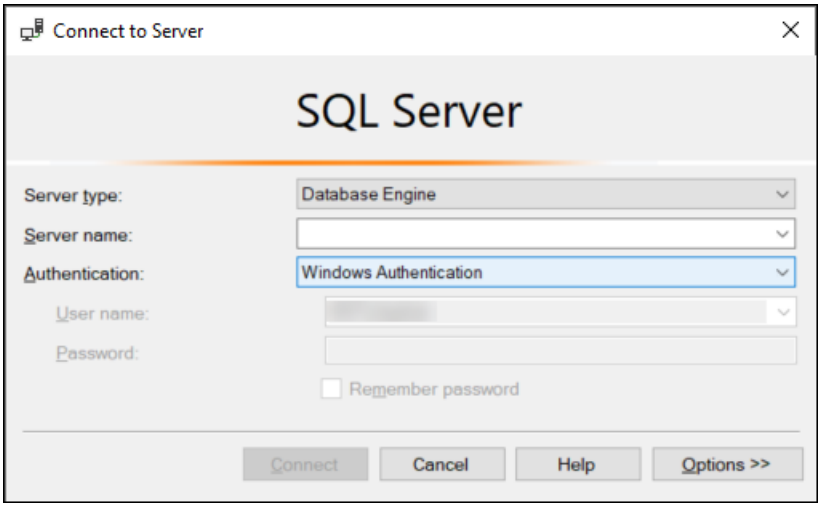
Nachdem Sie Ihre DB-Instance erstellt oder modifiziert haben, wird die Instance ein Mitglied der Domäne. Die AWS Konsole zeigt den Status der Domänenmitgliedschaft für die DB-Instance an. Der Status der DB-Instance kann einer der folgenden sein:

- joined – Die Instance ist Mitglied der Domäne.
- joining – Die Instance ist gerade im Prozess, Mitglied einer Domäne zu werden.
- pending-join – Die Mitgliedschaft der Instance steht noch aus.
- pending-maintenance-join— versucht, AWS die Instance während des nächsten geplanten Wartungsfensters zu einem Mitglied der Domain zu machen.
- pending-removal – Das Entfernen der Instance von der Domäne steht noch aus.
- pending-maintenance-removal— versucht, AWS die Instanz während des nächsten geplanten Wartungsfensters aus der Domain zu entfernen.
- failed – Ein Konfigurationsproblem hat verhindert, dass die Instance mit der Domäne verbunden werden konnte. Überprüfen und korrigieren Sie Ihre Konfiguration, bevor Sie den Befehl zu Änderung der Instance erneut ausführen.
- removing – Die Instance wird gerade von der Domäne entfernt.

Eine Anfrage, Mitglied einer Domäne zu werden, kann aufgrund eines Problems mit der Netzwerkverbindung oder einer falschen IAM Rolle fehlschlagen. Beispielsweise können Sie eine DB-Instance erstellen oder eine vorhandene Instance ändern und der Versuch, dass die DB-Instance Mitglied einer Domäne wird, wird fehlschlagen. Geben Sie in diesem Fall entweder den Befehl, um die DB-Instance zu erstellen oder zu ändern, neu aus oder ändern Sie die neu erstellte Instance, um der Domäne beizutreten.

Verbindung zum SQL Server mit Windows-Authentifizierung herstellen

Um eine Verbindung zum SQL Server mit Windows-Authentifizierung herzustellen, müssen Sie als Domänenbenutzer an einem Computer angemeldet sein, der einer Domäne angehört. Wählen Sie nach dem Start von SQL Server Management Studio die Windows-Authentifizierung als Authentifizierungstyp aus, wie im Folgenden dargestellt.



Wiederherstellen einer SQL Server DB-Instance und Hinzufügen zu einer Domäne

Sie können einen DB-Snapshot wiederherstellen oder eine point-in-time Wiederherstellung (PITR) für eine SQL Server-DB-Instance durchführen und sie dann einer Domäne hinzufügen. Wenn die DB-Instance wiederhergestellt wurde, modifizieren Sie die Instance, indem Sie den unter Schritt 5: Erstellen oder Modifizieren einer SQL Server-DB-Instance beschriebenen Prozess anwenden, um die DB-Instance einer Domäne hinzuzufügen.

Upgrades der Microsoft SQL Server-DB-Engine

Sofern Amazon RDS eine neue Version der Datenbank-Engine unterstützt, können Sie Ihre DB-Instances auf die neue Version aktualisieren. Es gibt zwei Arten von Upgrades für SQL Server DB-Instances: Hauptversionsupgrades und Unterversionsupgrades.

Hauptversions-Upgrades können Datenbankänderungen enthalten, die nicht mit vorhandenen Anwendungen rückwärts kompatibel sind. Daher müssen Sie Hauptversions-Upgrades Ihrer DB-Instances manuell durchführen. Sie können ein Hauptversions-Upgrade starten, indem Sie Ihre DB-Instance ändern. Bevor Sie jedoch ein Hauptversionsupgrade durchführen, empfehlen wir Ihnen, das Upgrade zu testen, indem Sie die in beschriebenen Schritte ausführe Testen und RDS für SQL Serverupgrade.

Upgrades kleinerer Versionen enthalten nur Änderungen, die mit vorhandenen Anwendungen abwärtskompatibel sind. Sie können die Nebenversion für Ihre DB-Instance auf zwei Arten aktualisieren:

- Manuell — Ändern Sie Ihre DB-Instance, um das Upgrade zu initiieren
- Automatisch — Aktivieren Sie automatische Upgrades für kleinere Versionen für Ihre DB-Instance

Wenn Sie automatische Upgrades für Nebenversionen aktivieren, aktualisiert RDS for SQL Server Ihre Datenbankinstanz automatisch während der geplanten Wartungsfenster, wenn wichtige Sicherheitsupdates in einer neueren Nebenversion verfügbar sind.

Für kleinere Engine-Versionen nach `16.00.4120.115.00.4365.2`, `14.00.3465.113.00.6435.1`,, sind die folgenden Sicherheitsprotokolle standardmäßig deaktiviert:

- `rds.tls10`(TLS 1.0-Protokoll)
- `rds.tls11`(TLS 1.1-Protokoll)
- `rds.rc4`(RC4 Chiffre)
- `rds.curve25519`(Curve25519-Verschlüsselung)
- `rds.3des168`(Triple-DES-Verschlüsselung)

Für frühere Engine-Versionen aktiviert Amazon RDS diese Sicherheitsprotokolle standardmäßig.

```
...
```

```
"ValidUpgradeTarget": [
    {
        "Engine": "sqlserver-se",
        "EngineVersion": "14.00.3281.6.v1",
        "Description": "SQL Server 2017 14.00.3281.6.v1",
        "AutoUpgrade": false,
        "IsMajorVersionUpgrade": false
    }
...
```

Weitere Informationen zur Durchführung von Upgrades finden Sie unter Aktualisieren einer SQL Server-DB-Instance. Weitere Informationen zu unterstützten SQL Server-Versionen in Amazon RDS finden Sie unter Amazon RDS for Microsoft SQL Server.

Themen

- Hauptversions-Upgrades für RDS für SQL Server
- Überlegungen zu SQL Server-Upgrades
- Testen und RDS für SQL Serverupgrade
- Aktualisieren einer SQL Server-DB-Instance
- Aktualisieren veralteter DB-Instances vor dem Ende des Supports

Hauptversions-Upgrades für RDS für SQL Server

Amazon unterstützt RDS derzeit die folgenden Hauptversions-Upgrades für eine Microsoft SQL Server-DB-Instance.

Sie können Ihre bestehende DB-Instance von jeder Version außer SQL Server 2008 auf SQL Server 2017 oder 2019 aktualisieren. Um ein Upgrade von SQL Server 2008 durchzuführen, führen Sie zunächst ein Upgrade auf eine der anderen Versionen durch.

Aktuelle Version	Unterstützte Upgrade-Versionen
SQLServer 2019	SQLServer 2022
SQLServer 2017	SQLServer 2022 SQLServer 2019
SQLServer 2016	SQLServer 2022

Aktuelle Version	Unterstützte Upgrade-Versionen
	SQLServer 2019 SQLServer 2017

Sie können eine AWS CLI Abfrage wie das folgende Beispiel verwenden, um die verfügbaren Upgrades für eine bestimmte Datenbank-Engine-Version zu finden.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
    --engine sqlserver-se \
    --engine-version 14.00.3281.6.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
    --engine sqlserver-se ^
    --engine-version 14.00.3281.6.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

Die Ausgabe zeigt, dass Sie Version 14.00.3281.6 auf die neuesten verfügbaren SQL Server 2017- oder 2019-Versionen aktualisieren können.

```
--------------------------
|DescribeDBEngineVersions|
+------------------------+
|      EngineVersion     |
+------------------------+
|  14.00.3294.2.v1       |
|  14.00.3356.20.v1      |
|  14.00.3381.3.v1       |
|  14.00.3401.7.v1       |
|  14.00.3421.10.v1      |
|  14.00.3451.2.v1       |
```

```
|  15.00.4043.16.v1      |
|  15.00.4073.23.v1      |
|  15.00.4153.1.v1       |
|  15.00.4198.2.v1       |
|  15.00.4236.7.v1       |
+------------------------+
```

Datenbank-Kompatibilitätsstufe

Sie können Microsoft SQL Server-Datenbankkompatibilitätsstufen verwenden, um einige Datenbankverhaltensweisen so anzupassen, dass sie frühere Versionen von SQL Server nachahmen. Weitere Informationen finden Sie unter Compatibility Level in der Microsoft-Dokumentation. Wenn Sie Ihre DB-Instance upgraden, behalten alle bestehenden Datenbanken ihren ursprünglichen Kompatibilitätsgrad.

Sie können den Kompatibilitätsgrad einer Datenbank mithilfe des ALTER DATABASE Befehls ändern. Um beispielsweise eine Datenbank mit dem Namen so `customeracct` zu ändern, dass sie mit SQL Server 2016 kompatibel ist, geben Sie den folgenden Befehl ein:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 130
```

Überlegungen zu SQL Server-Upgrades

Amazon RDS macht zwei DB-Snapshots während des Upgrades. Der erste DB-Snapshot gehört zur DB-Instance, bevor Änderungen am Upgrade vorgenommen wurden. Der zweite DB-Snapshot wird nach Abschluss des Upgrades übernommen.

> **Note**
>
> Amazon RDS nimmt nur DB-Snapshots auf, wenn Sie den Sicherungsaufbewahrungszeitraum für Ihre DB-Instance auf eine Zahl größer als 0 festgelegt haben. Informationen über das Ändern Ihres Aufbewahrungszeitraums für Backups finden Sie unter Ändern einer Amazon RDS DB-Instance.

Nachdem ein Upgrade abgeschlossen ist, können Sie nicht zur vorherigen Version der Datenbank-Engine zurückkehren. Wenn Sie zur vorherigen Version zurückkehren möchten, stellen Sie den DB-Snapshot wieder her, der vor dem Upgrade erstellt wurde, um eine neue DB-Instance zu erstellen.

Während eines Unterversion- oder Hauptversion-Upgrades von SQL Server zeigen die Metriken Freier Speicherplatz und Tiefe der Datenträgerwarteschlange den Wert -1 an. Nachdem das Upgrade abgeschlossen ist, kehren beide Metriken wieder in den Ausgangszustand zurück.

Lesen Sie die folgenden Informationen, bevor Sie Ihre SQL Server-Instanz aktualisieren.

Themen

- Bewährte Methoden vor dem Initiieren eines Upgrades
- Überlegungen zu Multi-AZ
- Überlegungen zu Lesereplikaten
- Überlegungen zu Optionsgruppen
- Überlegungen zu Parametergruppen

Bewährte Methoden vor dem Initiieren eines Upgrades

Bevor Sie mit dem Upgrade-Prozess beginnen, sollten Sie die folgenden vorbereitenden Schritte durchführen, um eine optimale Upgrade-Leistung zu gewährleisten und potenzielle Probleme zu minimieren:

Zeit- und Arbeitslastmanagement

- Planen Sie Upgrades in Zeiträumen mit geringem Transaktionsvolumen.
- Minimiere Schreibvorgänge während des Upgrade-Fensters.

  Dadurch kann Amazon RDS Upgrades schneller abschließen, indem die Anzahl der Transaktionsprotokoll-Backupdateien reduziert wird, die RDS beim secondary-to-primary Pairing wiederherstellen muss.

Verwaltung von Transaktionen

- Identifizieren und überwachen Sie lang andauernde Transaktionen.
- Stellen Sie sicher, dass alle kritischen Transaktionen bestätigt wurden, bevor Sie mit dem Upgrade beginnen.
- Vermeiden Sie lang andauernde Transaktionen während des Upgrade-Fensters.

Optimierung der Protokolldateien

Überprüfen und optimieren Sie die Transaktionsprotokolldateien:

- Verkleinern Sie übergroße Protokolldateien.
- Reduzieren Sie Muster mit hohem Protokollverbrauch.

- Virtuelle Protokolldateien verwalten (VLFs).
- Sorgen Sie für ausreichend freien Speicherplatz für den normalen Betrieb.

Überlegungen zu Multi-AZ

Amazon RDS unterstützt Multi-AZ-Bereitstellungen für DB-Instances, auf denen Microsoft SQL Server ausgeführt wird, mithilfe von SQL Server Database Mirroring (DBM) oder Always On Availability Groups (). AGs Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.

Wenn in einer Multi-AZ-Bereitstellung (Mirroring/AlwaysOn) ein Upgrade angefordert wird, folgt RDS einer fortlaufenden Upgrade-Strategie für die primären und sekundären Instances. Rollende Upgrades stellen sicher, dass mindestens eine Instanz für Transaktionen verfügbar ist, während die sekundäre Instanz aktualisiert wird. Der Ausfall wird voraussichtlich nur für die Dauer eines Failovers andauern.

Während des Upgrades entfernt RDS die sekundäre Instance aus der Multi-AZ-Konfiguration, führt ein Upgrade der sekundären Instance durch und stellt alle Transaktionsprotokoll-Backups von der primären Instance wieder her, die während der Verbindungsunterbrechung erstellt wurden. Nachdem alle Protokollsicherungen wiederhergestellt wurden, verbindet RDS die aktualisierte sekundäre mit der primären. Wenn sich alle Datenbanken in einem synchronisierten Zustand befinden, führt RDS einen Failover zur aktualisierten sekundären Instanz durch. Sobald der Failover abgeschlossen ist, fährt RDS mit dem Upgrade der alten primären Instance fort, stellt alle Backups der Transaktionsprotokolle wieder her und koppelt sie mit der neuen primären Instance.

Um diese Failover-Dauer zu minimieren, empfehlen wir, den Listener-Endpunkt für AlwaysOn AGs Verfügbarkeitsgruppen zu verwenden, wenn Sie Clientbibliotheken verwenden, die die `MultiSubnetFailover` Verbindungsoption in der Verbindungszeichenfolge unterstützen. Bei Verwendung des Listener-Endpunkts für Verfügbarkeitsgruppen betragen die Failover-Zeiten in der Regel weniger als 10 Sekunden. Diese Dauer beinhaltet jedoch keine zusätzliche Zeit für die Wiederherstellung nach einem Absturz.

Überlegungen zu Lesereplikaten

Während eines Datenbankversions-Upgrades aktualisiert Amazon RDS alle Ihre Lesereplikate zusammen mit der primären DB-Instance. Amazon RDS unterstützt keine separaten Datenbankversions-Upgrades für die Lesereplikate. Weitere Informationen über Lesereplikate finden Sie unter Arbeiten mit Read Replicas für Microsoft SQL Server in Amazon RDS.

Wenn Sie ein Datenbankversions-Upgrade der primären DB-Instance durchführen, werden alle entsprechenden Lesereplikate automatisch aktualisiert. Amazon RDS aktualisiert alle Lesereplikate gleichzeitig, bevor die primäre DB-Instance aktualisiert wird. Lesereplikate sind möglicherweise erst verfügbar, wenn das Upgrade der Datenbankversion auf der primären DB-Instance abgeschlossen ist.

Überlegungen zu Optionsgruppen

Wenn Ihre DB-Instance eine benutzerdefinierte DB-Optionsgruppe verwendet, kann Amazon RDS in einigen Fällen Ihre DB-Instance nicht automatisch einer neuen Optionsgruppe zuweisen. Wenn Sie zum Beispiel auf eine neue Hauptversion aktualisieren, müssen Sie eine neue Optionsgruppe angeben. Wir empfehlen, dass Sie eine neue Optionsgruppe erstellen und dieser dieselben Optionen hinzufügen, die Sie in Ihrer bestehenden benutzerdefinierten Optionsgruppe hatten.

Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe oder Kopieren einer Optionsgruppe.

Überlegungen zu Parametergruppen

Wenn Ihre DB-Instance eine benutzerdefinierte DB-Parametergruppe verwendet:

- Amazon RDS startet die DB-Instance nach einem Upgrade automatisch neu.
- In einigen Fällen kann RDS Ihrer DB-Instance nicht automatisch eine neue Parametergruppe zuweisen.

  Wenn Sie zum Beispiel auf eine neue Hauptversion aktualisieren, müssen Sie eine neue Parametergruppe angeben. Wir empfehlen, dass Sie eine neue Parametergruppe erstellen und die Parameter so konfigurieren wie in Ihrer bestehenden benutzerdefinierten Parametergruppe.

Weitere Informationen finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS oder Kopieren einer DB-Parametergruppe in Amazon RDS.

Testen und RDS für SQL Serverupgrade

Bevor Sie ein neues Hauptversions-Upgrade für Ihre DB-Instance durchführen, sollten Sie Ihre Datenbank und alle Anwendungen, die Zugriff auf die Datenbank haben, sorgfältig auf die Kompatibilität mit der neuen Version prüfen. Wir empfehlen Ihnen folgendes Vorgehen.

Um ein Hauptversions-Upgrade zu testen

1. Suchen Sie in der Microsoft-Dokumentation nach Upgrade SQL Server für die neue Version der Datenbank-Engine, um festzustellen, ob Kompatibilitätsprobleme vorliegen, die sich auf Ihre Datenbank oder Anwendungen auswirken könnten.
2. Wenn Ihre DB-Instance eine benutzerdefinierte Optionsgruppe verwendet, erstellen Sie eine neue Optionsgruppe, die kompatibel mit der neuen Version ist, auf die Sie upgraden. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.
3. Wenn Ihre DB-Instance eine benutzerdefinierte Parametergruppe verwendet, erstellen Sie eine neue Parametergruppe, die kompatibel mit der neuen Version ist, auf die Sie upgraden. Weitere Informationen finden Sie unter Überlegungen zu Parametergruppen.
4. Erstellen Sie einen DB-Snapshot der zu aktualisierenden DB-Instance. Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
5. Stellen Sie den DB-Snapshot wieder her, um eine neue Test-DB-Instance zu erstellen. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.
6. Ändern Sie diese neue Test-DB-Instance mit den folgenden Methoden, um sie auf die neue Version upzugraden:

   - Konsole
   - AWS CLI
   - RDS-API
7. Beurteilen Sie den Speicherplatz, den die upgegradete Instance verwendet, um zu bestimmen, ob das Upgrade zusätzlichen Speicherplatz benötigt.
8. Führen Sie so viele Qualitätssicherungstests mit der upgegradeten DB-Instance durch, wie nötig, um sicherzustellen, dass Ihre Datenbank und Anwendung mit der neuen Version korrekt ausgeführt werden. Führen Sie alle nötigen neuen Tests aus, um die Auswirkungen von Kompatibilitätsproblemen abzuwägen, die Sie in Schritt 1 bestimmt haben. Testen Sie alle gespeicherten Prozeduren und Funktionen. Leiten Sie Testversionen Ihrer Anwendungen an die aktualisierte DB-Instance weiter.
9. Wenn alle Tests erfolgreich sind, führen Sie das Upgrade für Ihre Produktions-DB-Instance durch. Wir empfehlen, dass Sie keine Schreiboperationen auf der DB-Instance zulassen, bis Sie bestätigen können, dass alles richtig ausgeführt wird.

Aktualisieren einer SQL Server-DB-Instance

Informationen über ein manuelles oder automatisches Upgrade einer SQL Server-DB-Instance finden Sie unter:

- Upgrade der Engine-Version für eine DB-Instance
- Best practices for upgrading SQL Server 2008 R2 to SQL Server 2016 on Amazon RDS for SQL Server

> **Important**
>
> Wenn Sie über Snapshots verfügen, die mit verschlüsselt sind, empfehlen wir Ihnen AWS KMS, ein Upgrade zu initiieren, bevor der Support endet.

Aktualisieren veralteter DB-Instances vor dem Ende des Supports

Wenn eine Hauptversion veraltet ist, können Sie diese nicht mehr auf neuen DB-Instances installieren. RDS versucht automatisch, ein Upgrade für alle vorhandenen DB-Instances durchzuführen.

Wenn Sie eine veraltete DB-Instance wiederherstellen müssen, können Sie eine point-in-time Wiederherstellung (PITR) durchführen oder einen Snapshot wiederherstellen. Dadurch erhalten Sie vorübergehend Zugriff auf eine DB-Instance, die die veraltete Version verwendet. Wenn eine Hauptversion jedoch komplett veraltet ist, werden diese DB-Instances automatisch auf eine unterstützte Version aktualisiert.

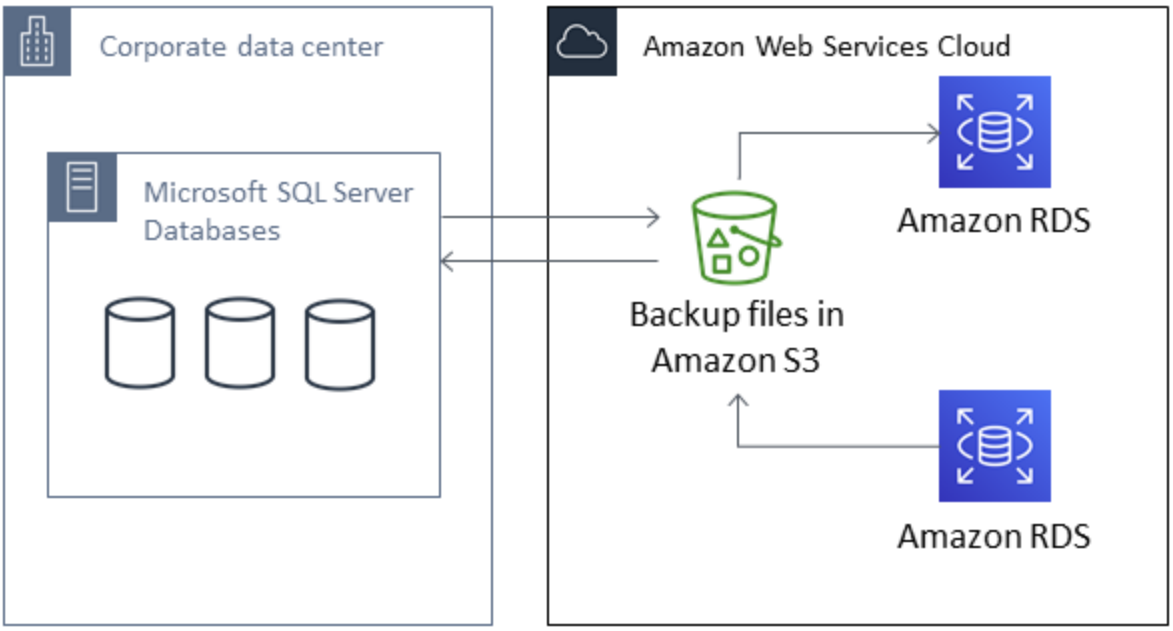
Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung

Amazon RDS unterstützt native Backups und Wiederherstellungen für Microsoft SQL Server-Datenbanken unter Verwendung von vollständigen Sicherungsdateien (.bak-Dateien). Bei RDS greifen Sie auf in Amazon S3 gespeicherte Daten zu, anstatt das lokale Datensystem des Datenbankservers zu nutzen.

Beispielsweise können Sie auf Ihrem lokalen Server ein vollständiges Backup erstellen, dieses in S3 speichern und anschließend in einer verfügbaren Amazon RDS-DB-Instance wiederherstellen. Ebenso können Sie über RDS Backups erstellen, in S3 speichern und dann dort wiederherstellen, wo Sie dies möchten.

Native Sicherung und Wiederherstellung sind in allen AWS Regionen für Single-AZ- und Multi-AZ-DB-Instances verfügbar, einschließlich Multi-AZ-DB-Instances mit Read Replicas. Native Backups und Wiederherstellungen sind für alle in Amazon RDS unterstützten Versionen von Microsoft SQL Server verfügbar.

Das folgende Diagramm veranschaulicht die unterstützten Szenarien.



Die Verwendung nativer .bak-Dateien ist i. d. R. die schnellste Methode zum Sichern und Wiederherstellen von Datenbanken. Es gibt viele zusätzliche Vorteile bei der Verwendung nativer Backups und Wiederherstellungen. Sie können z. B. Folgendes tun:

- Migrieren von Datenbanken zu oder aus Amazon RDS
- Verschieben von Datenbanken zwischen RDS for SQL Server DB-Instances
- Migrieren von Daten, Schemata, gespeicherten Prozeduren, Auslösern und anderem Datenbankcode in .bak-Dateien.
- Einzelne Datenbanken anstelle von ganzen DB-Instances sichern und wiederherstellen
- Erstellen Sie Kopien von Datenbanken zu Entwicklungs-, Test-, Trainings- und Demozwecken.
- Speichern und übertragen Sie Sicherungsdateien mit Amazon S3, um zusätzlichen Schutz für die Notfallwiederherstellung zu gewährleisten.
- Erstellen Sie native Backups von Datenbanken mit aktivierter TDE (Transparent Data Encryption) und stellen Sie diese Backups in lokalen Datenbanken wieder her. Weitere Informationen finden Sie unter Support für transparente Datenverschlüsselung im SQL Server.
- Stellen Sie native Backups von lokalen Datenbanken, deren TDE-Verschlüsselung aktiviert ist, auf DB-Instances von RDS for SQL Server wieder her. Weitere Informationen finden Sie unter Support für transparente Datenverschlüsselung im SQL Server.

Inhalt

- Einschränkungen und Empfehlungen
- Einrichtung für native Backups und Wiederherstellungen
  - Manuelles Erstellen einer IAM-Rolle für native Backups und Wiederherstellungen
- Verwenden nativer Backups und Wiederherstellungen
  - Sichern einer Datenbank
    - Verwendung
    - Beispiele
  - Wiederherstellen einer Datenbank
    - Verwendung
    - Beispiele
  - Wiederherstellen eines Protokolls
    - Verwendung
    - Beispiele
  - Abschluss einer Datenbankwiederherstellung
    - Verwendung
  - Arbeiten mit teilweise wiederhergestellten Datenbanken

        - Verwerfen einer teilweise wiederhergestellten Datenbank
        - Snapshot-Wiederherstellung und point-in-time Wiederherstellungsverhalten für teilweise wiederhergestellte Datenbanken
    - Abbrechen einer Aufgabe
        - Verwendung
    - Verfolgen des Status von Aufgaben
        - Verwendung
        - Beispiele
        - Antwort
- Komprimieren von Sicherungsdateien
- Fehlerbehebung
- Importieren und Exportieren von SQL Server-Daten mithilfe anderer Methoden
    - Importieren von Daten in RDS for SQL Server mithilfe eines Snapshots
        - Importieren der Daten
            - Assistent für das Generieren und Veröffentlichen von Skripts
            - Assistent für den Import und Export
            - Bulk-Kopie
    - Exportieren von Daten aus RDS for SQL Server
        - SQL Server-Assistent für Import und Export (SQL Server Import and Export Wizard)
        - SQL Server-Assistent für das Generieren und Veröffentlichen von Skripts und Hilfsprogramm bcp
- Verwenden des BCP-Dienstprogramms unter Linux zum Importieren und Exportieren von Daten
    - Voraussetzungen
    - Installation der SQL Server-Befehlszeilentools unter Linux
    - Exportieren von Daten aus RDS for SQL Server
        - Grundlegende Exportsyntax
        - Beispiel exportieren
    - Daten nach RDS für SQL Server importieren
        - Grundlegende Importsyntax
        - Beispiel importieren
    - Allgemeine BCP-Optionen

- Bewährte Methoden und Überlegungen
- Behebung häufig auftretender Probleme

Einschränkungen und Empfehlungen

Folgende Einschränkungen gelten bei der Verwendung nativer Backups und Wiederherstellungen:

- Sie können keine Backups in einem Amazon S3-Bucket in einer anderen AWS Region als Ihrer Amazon RDS-DB-Instance erstellen oder aus einem solchen Bucket wiederherstellen.
- Sie können eine Datenbank nicht mit dem gleichen Namen wie eine vorhandene Datenbank wiederherstellen. Die Namen der Datenbank sind eindeutig.
- Es wird dringend davon abgeraten, Backups aus einer Zeitzone für eine andere Zeitzone wiederherzustellen. Wenn Sie Backups aus einer bestimmten Zeitzone in einer anderen Zeitzone wiederherstellen, müssen Sie Ihre Abfragen und Anwendungen auf mögliche Auswirkungen der Zeitzonenänderung überprüfen.
- Amazon S3 hat eine Größenbeschränkung von 5 TB pro Datei. Für systemeigene Backups größerer Datenbanken können Sie Multidatei-Backups verwenden.
- Die maximale Datenbankgröße, die zu S3 gesichert werden kann, hängt vom verfügbaren Arbeitsspeicher, der CPU, der I/O-Leistung und den Netzwerkressourcen der DB-Instance ab. Je größer die Datenbank ist, umso mehr Speicher verbraucht der Sicherungsagent.
- Sie können gleichzeitig in nicht mehr als 10 Sicherungsdateien sichern oder aus ihnen wiederherstellen.
- Ein differentielles Backup basiert auf dem letzten vollständigen Backup. Damit differentielle Backups funktionieren, darf kein Snapshot zwischen dem letzten vollständigen Backup und dem differentiellen Backup aufgenommen werden. Wenn Sie ein differentielles Backup erstellen möchten, aber ein manueller oder automatisierter Snapshot vorhanden ist, erstellen Sie ein weiteres vollständiges Backup, bevor Sie mit dem differentiellen Backup fortfahren.
- Differenzielle und Protokollwiederherstellungen werden für Datenbanken mit Dateien, deren file_guid (eindeutige ID) auf eingestellt ist, nicht unterstütz `NULL`.
- Sie können bis zu zwei Backup- oder Wiederherstellungsaufgaben gleichzeitig ausführen.
- Sie können keine systemeigenen Protokollsicherungen von SQL Server auf Amazon RDS ausführen.
- RDS unterstützt native Wiederherstellungen von Datenbanken mit bis zu 64 TiB. Native Wiederherstellungen von Datenbanken auf SQL Server Express Edition sind auf 10 GB beschränkt.

- Ein natives Backup der Datenbank ist nicht möglich während des Wartungsfensters bzw. während Amazon RDS einen Snapshot der Datenbank aufnimmt. Wenn sich eine native Sicherungsaufgabe mit dem täglichen RDS-Sicherungsfenster überschneidet, wird die native Sicherungsaufgabe abgebrochen.
- Bei Multi-AZ-DB-Instances können Sie nur Datenbanken nativ wiederherstellen, die im vollständigen Wiederherstellungsmodell gesichert wurden.
- Die Wiederherstellung aus differentiellen Backups auf Multi-AZ-Instances wird nicht unterstützt.
- Das Aufrufen der RDS-Verfahren zu nativen Backups und Wiederherstellungen innerhalb einer Transaktion wird nicht unterstützt.
- Verwenden Sie eine symmetrische Verschlüsselung, um Ihre AWS KMS key Backups zu verschlüsseln. Amazon RDS unterstützt keine asymmetrischen KMS-Schlüssel. Weitere Informationen finden Sie unter Erstellen symmetrischer KMS-Verschlüsselungsschlüssel im AWS Key Management Service -Entwicklerhandbuch.
- Native Backup-Dateien werden mit dem angegebenen KMS-Schlüssel unter Verwendung des Verschlüsselungsmodus "Encryption-Only" verschlüsselt. Bei der Wiederherstellung verschlüsselter Sicherheitsdateien sollten Sie stets bedenken, dass diese Dateien mit dem Krypto-Modus "Nur Verschlüsselung" verschlüsselt wurden.
- Eine Datenbank, die eine FILESTREAM-Dateigruppe enthält, kann nicht wiederhergestellt werden.
- Die serverseitige Amazon S3 S3-Verschlüsselung mit AWS KMS (SSE-KMS) wird derzeit nicht unterstützt. Wenn Sie einen KMS-Schlüssel für eine gespeicherte Prozedur bereitstellen, werden alle systemeigenen Sicherungen und Wiederherstellungen auf der Clientseite mit dem KMS-Schlüssel ver- und entschlüsselt. AWS speichert die Backups im S3-Bucket mit SSE-S3.

Wenn Ihre Datenbank beim Anlegen, Kopieren und Wiederherstellen der Sicherungsdatei offline sein kann, empfehlen wir, Ihre Datenbank mithilfe nativer Backups und Wiederherstellungen zu RDS zu migrieren, Wenn Ihre lokale Datenbank nicht offline sein kann, empfehlen wir Ihnen, die zu verwenden, um Ihre Datenbank AWS Database Migration Service zu Amazon RDS zu migrieren. Weitere Informationen finden Sie unter Was ist AWS Database Migration Service?

Native Backups und Wiederherstellungen sind nicht als Ersatz für die Datenwiederherstellungsfunktionen der regionsübergreifenden Snapshot-Kopierfunktion gedacht. Wir empfehlen Ihnen, Snapshot Copy zu verwenden, um Ihren Datenbank-Snapshot für die regionsübergreifende Notfallwiederherstellung in Amazon RDS in eine andere AWS Region zu kopieren. Weitere Informationen finden Sie unter Kopieren eines DB-Snapshots für Amazon RDS.

Einrichtung für native Backups und Wiederherstellungen

Zum Einrichten nativer Backup und Wiederherstellungen benötigen Sie drei Komponenten:

1. Einen Amazon S3-Bucket zum Speichern Ihrer Sicherungsdateien

   Sie müssen für Ihre Sicherungsdateien einen S3-Bucket verwenden und dann die Backups hochladen, die Sie zu RDS migrieren möchten. Wenn Sie bereits über einen Amazon S3-Bucket verfügen, können Sie diesen verwenden. Andernfalls können Sie einen Bucket erstellen. Alternativ können Sie wählen, dass ein neuer Bucket für Sie erstellt wird, wenn Sie die Option `SQLSERVER_BACKUP_RESTORE` mithilfe der AWS Management Console hinzufügen.

   Weitere Informationen zur Verwendung von S3 finden Sie im Benutzerhandbuch für Amazon Simple Storage Service.
2. Eine AWS Identity and Access Management (IAM-) Rolle für den Zugriff auf den Bucket.

   Wenn Sie bereits über eine IAM-Rolle verfügen, können Sie diese verwenden. Alternativ können Sie wählen, dass eine neue IAM-Rolle für Sie erstellt wird, wenn Sie die Option `SQLSERVER_BACKUP_RESTORE` mithilfe der AWS Management Console hinzufügen. Alternativ können Sie manuell eine neue Rolle erstellen.

   Wenn Sie eine neue IAM-Rolle manuell erstellen möchten, verwenden Sie die Methode, die im nächsten Abschnitt besprochen wird. Führen Sie dasselbe aus, wenn Sie einer vorhandenen IAM-Rolle Vertrauensstellungen und Berechtigungsrichtlinien zuordnen möchten.
3. Die Option `SQLSERVER_BACKUP_RESTORE` in einer Optionsgruppe auf Ihrer DB-Instance

   Zur Aktivierung nativer Backups und Wiederherstellungen auf Ihrer DB-Instance fügen Sie einer Optionsgruppe auf Ihrer DB-Instance die Option `SQLSERVER_BACKUP_RESTORE` hinzu. Weitere Informationen und Anweisungen finden Sie unter Unterstützung für native Sicherung und Backup in SQL Server.

Manuelles Erstellen einer IAM-Rolle für native Backups und Wiederherstellungen

Wenn Sie manuell eine neue IAM-Rolle zur Verwendung mit nativen Backups und Wiederherstellungen erstellen möchten, ist dies möglich. In diesem Fall erstellen Sie eine Rolle für die Delegierung der Berechtigungen vom Amazon RDS-Service zu Ihrem Amazon S3-Bucket. Wenn Sie eine IAM-Rolle erstellen, fügen Sie eine Vertrauensstellung und eine Berechtigungsrichtlinie an. Die Vertrauensstellung ermöglicht es RDS, diese Rolle zu übernehmen. Die Berechtigungsrichtlinie

definiert die Aktionen, die über diese Rolle ausgeführt werden können. Weitere Informationen zum Erstellen der Rolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS -Service.

Für die Funktion zu nativen Backups und Wiederherstellungen verwenden Sie ähnliche Vertrauensbeziehungen und Berechtigungsrichtlinien wie in den in diesem Abschnitt gezeigten Beispielen. Im folgenden Beispiel wird der Dienstprinzipalname `rds.amazonaws.com` als Alias für alle Dienstkonten verwendet. In den anderen Beispielen wird durch Angabe eines Amazon-Ressourcennamens (ARN) einem anderen Konto, einem anderen Benutzer oder einer anderen Rolle der Zugriff auf die Vertrauensrichtlinie gewährt.

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Vertrauensbeziehungen, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Sie können beide globalen Bedingungskontextschlüssel verwenden und der Wert `aws:SourceArn` enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der Wert `aws:SourceAccount` und das Konto im Wert `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Anweisung verwendet werden.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Stellen Sie in der Vertrauensbeziehung sicher, dass Sie den globalen Bedingungskontextschlüssel `aws:SourceArn` mit dem vollständigen ARN der Ressourcen verwenden, die auf die Rolle zugreifen. Stellen Sie bei nativen Backups und Wiederherstellungen sicher, dass Sie sowohl die DB-Optionsgruppe als auch die DB-Instances einschließen, wie im folgenden Beispiel gezeigt.

Example Vertrauensverhältnis mit globalem Bedingungskontextschlüssel für systemeigene Backups und Wiederherstellungen

JSON

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [

 "arn:aws:rds:Region:0123456789:db:db_instance_identifier",
                        "arn:aws:rds:Region:0123456789:og:option_group_name",
                  "arn:aws:s3:::amzn-s3-demo-bucket"
                    ],
                    "aws:SourceAccount": "0123456789"
                }
            }
        }
    ]
}
```

Im folgenden Beispiel wird ein ARN zur Angabe einer Ressource verwendet. Weitere Informationen zur Verwendung ARNs finden Sie unter Amazon-Ressourcennamen (ARNs).

Example der Berechtigungsrichtlinie für systemeigenes Backup und Wiederherstellung ohne Verschlüsselungsunterstützung

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
```

```
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Example Berechtigungsrichtlinie für native Backups und Wiederherstellungen mit Verschlüsselungsunterstützung

Wenn Sie Ihre Sicherungsdateien verschlüsseln möchten, geben Sie in Ihrer Berechtigungsrichtlinie einen Verschlüsselungsschlüssel an. Weitere Informationen zu Verschlüsselungsschlüsseln finden Sie unter Erste Schritte im AWS Key Management Service -Entwicklerhandbuch.

> **Note**
>
> Sie müssen einen symmetrischen KMS-Verschlüsselungsschlüssel verwenden, um Ihre Backups zu verschlüsseln. Amazon RDS unterstützt keine asymmetrischen KMS-Schlüssel. Weitere Informationen finden Sie unter Erstellen symmetrischer KMS-Verschlüsselungsschlüssel im AWS Key Management Service -Entwicklerhandbuch.
> Die IAM-Rolle muss auch ein Schlüsselbenutzer und Schlüsseladministrator für den KMS-Schlüssel sein, d. h. sie muss in der Schlüsselrichtlinie angegeben werden. Weitere Informationen finden Sie unter Erstellen symmetrischer KMS-Verschlüsselungsschlüssel im AWS Key Management Service -Entwicklerhandbuch.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToKey",
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:Encrypt",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-east-1:123456789012:key/key-id"
    },
    {
      "Sid": "AllowAccessToS3",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
    },
    {
      "Sid": "GetS3Info",
      "Effect": "Allow",
      "Action": [
        "s3:GetObjectAttributes",
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
  ]
}
```

Verwenden nativer Backups und Wiederherstellungen

Sobald Sie native Backups und Wiederherstellungen konfiguriert haben, können Sie mit deren Verwendung beginnen. Bauen Sie zuerst eine Verbindung zu Ihrer Microsoft SQL Server-Datenbank auf, und rufen Sie dann eine gespeicherte Amazon RDS-Prozedur auf, um die Aufgabe zu erledigen. Anleitungen zum Herstellen einer Verbindung zu Ihrer Datenbank finden Sie unter Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen.

Einige der gespeicherten Prozeduren verlangen, dass Sie einen Amazon-Ressourcennamen (ARN) für Ihren Amazon S3-Bucket und die Datei angeben. Das Format für Ihren ARN lautet `arn:aws:s3:::`*`bucket_name`*`/`*`file_name.extension`*. Amazon S3 benötigt keine Kontonummer oder AWS Region in ARNs.

Wenn Sie auch einen optionalen KMS-Schlüssel angeben, lautet das Format für die ARN des Schlüssels `arn:aws:kms:`*`region`*`:`*`account-id`*`:key/`*`key-id`*. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) und AWS Service-Namespaces. Sie müssen einen symmetrischen KMS-Verschlüsselungsschlüssel verwenden, um Ihre Backups zu verschlüsseln. Amazon RDS unterstützt keine asymmetrischen KMS-Schlüssel. Weitere Informationen finden Sie unter Erstellen symmetrischer KMS-Verschlüsselungsschlüssel im AWS Key Management Service - Entwicklerhandbuch.

> **Note**
>
> Unabhängig davon, ob Sie einen KMS-Schlüssel verwenden oder nicht, aktivieren die nativen Sicherungs- und Wiederherstellungsaufgaben standardmäßig eine serverseitige 256-Bit-Verschlüsselung (Advanced Encryption Standard, AES) für Dateien, die in S3 hochgeladen werden.

Anleitungen zum Aufrufen der einzelnen gespeicherten Prozeduren finden Sie in den folgenden Themen:

- Sichern einer Datenbank
- Wiederherstellen einer Datenbank
- Wiederherstellen eines Protokolls
- Abschluss einer Datenbankwiederherstellung
- Arbeiten mit teilweise wiederhergestellten Datenbanken

- Abbrechen einer Aufgabe
- Verfolgen des Status von Aufgaben

Sichern einer Datenbank

Verwenden Sie zum Sichern Ihrer Datenbank die gespeicherte Prozedur `rds_backup_database`.

> **Note**
>
> Ein Backup der Datenbank ist nicht möglich während des Wartungsfensters, oder wenn Amazon RDS gerade einen Snapshot der Datenbank erfasst.

Verwendung

```
exec msdb.dbo.rds_backup_database
 @source_db_name='database_name',
 @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@overwrite_s3_backup_file=0|1],
 [@block_size=512|1024|2048|4096|8192|16384|32768|65536],
        [@max_transfer_size=n],
        [@buffer_count=n],
 [@type='DIFFERENTIAL|FULL'],
 [@number_of_files=n];
```

Die folgenden Parameter sind erforderlich:

- `@source_db_name` – Der Name der zu sichernden Datenbank.
- `@s3_arn_to_backup_to` – Der ARN, der den Amazon S3-Bucket für das Backup angibt, mit dem Namen der Sicherungsdatei.

  Die Datei kann eine beliebige Erweiterung haben, üblicherweise wird jedoch `.bak` verwendet.

Die folgenden Parameter sind optional:

- `@kms_master_key_arn` – Der ARN für den symmetrischen KMS-Verschlüsselungsschlüssel, der für die Verschlüsselung des Objekts verwendet werden soll.

  - Sie können den Standard-Verschlüsselungsschlüssel nicht verwenden. Wenn Sie den Standardschlüssel verwenden, wird die Datenbank nicht gesichert.
  - Wenn Sie keine KMS-Schlüsselkennung angeben, wird die Sicherungsdatei nicht verschlüsselt. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
  - Wenn Sie einen KMS-Schlüssel angeben, wird eine client-seitige Verschlüsselung verwendet.
  - Amazon RDS unterstützt keine asymmetrischen KMS-Schlüssel. Weitere Informationen finden Sie unter Erstellen symmetrischer KMS-Verschlüsselungsschlüssel im AWS Key Management Service -Entwicklerhandbuch.
- `@overwrite_s3_backup_file` – Ein Wert, der angibt, ob eine vorhandene Sicherungsdatei überschrieben werden soll.
  - `0` – Eine vorhandene Datei wird nicht überschrieben. Dieser Wert ist der Standard.

    Wenn Sie `@overwrite_s3_backup_file` auf 0 setzen, wird ein Fehler ausgegeben, wenn die Datei bereits vorhanden ist.
  - `1` – Eine vorhandene Datei mit dem angegebenen Namen wird überschrieben, auch wenn es sich nicht um eine Sicherungsdatei handelt.
- `@type` – Der Typ der Backup.
  - `DIFFERENTIAL` – Erstellt eine differentielle Backup.
  - `FULL` – Erstellt eine vollständige Backup. Dieser Wert ist der Standard.

  Eine differentielle Sicherung basiert auf der letzten vollständigen Sicherung. Damit differentielle Sicherungen funktionieren, darf kein Snapshot zwischen der letzten vollständigen Sicherung und der differentiellen Sicherung aufgenommen werden. Wenn Sie ein differentielles Backup erstellen wollen und ein Snapshot existiert, erstellen Sie ein weiteres vollständiges Backup, bevor Sie mit dem differentiellen Backup fortfahren.

  Sie können das letzte vollständige Backup oder den Snapshot mit der folgenden Beispiel-SQL-Abfrage suchen:

```
select top 1
database_name
,  backup_start_date
,  backup_finish_date
from    msdb.dbo.backupset
where   database_name='mydatabase'
and     type = 'D'
```

```
order by backup_start_date desc;
```

- `@number_of_files` – Die Anzahl der Dateien, in die das Backup aufgeteilt wird (aufgeschlüsselt). Die maximale Anzahl ist 10.
  - Das Backup mehrerer Dateien wird sowohl für vollständige als auch für differenzielle Backups unterstützt.
  - Wenn Sie den Wert 1 eingeben oder den Parameter weglassen, wird eine einzelne Sicherungsdatei erstellt.

  Geben Sie das Präfix an, das den Dateien gemeinsam ist, und danach das Suffix mit einem Sternchen (`*`). Das Sternchen kann sich an einer beliebigen *`file_name`* Stelle im S3-ARN ARN. Das Sternchen wird durch eine Reihe von alphanumerischen Zeichenfolgen in den generierten Dateien ersetzt, beginnend mit `1-of-`*`number_of_files`*.

  Wenn beispielsweise die Dateinamen im S3-ARN `backup*.bak` lauten und Sie `@number_of_files=4` festlegen, sind die generierten Sicherungsdateien `backup1-of-4.bak`, `backup2-of-4.bak`, `backup3-of-4.bak` und `backup4-of-4.bak`.
  - Wenn einer der Dateinamen bereits vorhanden und `@overwrite_s3_backup_file` 0 ist, wird ein Fehler ausgegeben.
  - Backups mit mehreren Dateien können nur ein Sternchen im *`file_name`* Teil des S3-ARN haben.
  - Einzeldatei-Backups können eine beliebige Anzahl von Sternchen im *`file_name`* Teil des S3-ARN haben. Sternchen werden nicht aus dem generierten Dateinamen entfernt.
- `@block_size`— Blockgröße (in Byte), die die physische Blockgröße für Backup-Operationen angibt. Gültige Werte sind 512, 1024, 2048, 4096, 8192, 16384, 32768 und 65536
- `@max_transfer_size`— Die maximale Übertragungsgröße gibt die Obergrenze des Datenvolumens (in Byte) an, das pro Vorgang während des Backup-Vorgangs übertragen wird. I/O Gültige Werte sind Vielfache von 65536 Byte (64 KB) bis zu 4194304 Byte (4 MB).
- `@buffer_count`— Gesamtzahl der I/O Puffer, die für den Backup-Prozess verwendet werden sollen.

Beispiele

Example für eine differenzielle Backup

```
exec msdb.dbo.rds_backup_database
```

```
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@overwrite_s3_backup_file=1,
@type='DIFFERENTIAL';
```

Example für ein vollständiges Backup mit Verschlüsselung

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@overwrite_s3_backup_file=1,
@type='FULL';
```

Example für ein Backup mehrerer Dateien

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4;
```

Example für ein differenzielles Backup mehrerer Dateien

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@type='DIFFERENTIAL',
@number_of_files=4;
```

Example für ein Backup mehrerer Dateien mit Verschlüsselung

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@number_of_files=4;
```

Example für ein Backup mehrerer Dateien mit S3-Überschreibung

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
```

```
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@overwrite_s3_backup_file=1,
@number_of_files=4;
```

Example des Backups mit Blockgröße

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@block_size=512;
```

Example von Backups mit mehreren Dateien mit und **`@max_transfer_size@buffer_count`**

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4,
@max_transfer_size=4194304,
@buffer_count=10;
```

Example für ein Backup einer einzelnen Datei mit dem Parameter @number_of_files

In diesem Beispiel wird eine Sicherungsdatei mit dem Namen generier `backup*.bak`.

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=1;
```

Wiederherstellen einer Datenbank

Rufen Sie zum Wiederherstellen Ihrer Datenbank die gespeicherte Prozedur `rds_restore_database` auf. Amazon RDS erstellt einen anfänglichen Snapshot der Datenbank nach Abschluss der Wiederherstellung und wenn die Datenbank offen ist.

Verwendung

```
exec msdb.dbo.rds_restore_database
 @restore_db_name='database_name',
 @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name.extension',
 @with_norecovery=0|1,
```

```
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
       [@block_size=512|1024|2048|4096|8192|16384|32768|65536],
       [@max_transfer_size=n],
       [@buffer_count=n],
[@type='DIFFERENTIAL|FULL'];
```

Die folgenden Parameter sind erforderlich:

- `@restore_db_name` – Der Name der wiederherzustellenden Datenbank. Die Namen der Datenbank sind eindeutig. Sie können eine Datenbank nicht mit dem gleichen Namen wie eine vorhandene Datenbank wiederherstellen.
- `@s3_arn_to_restore_from` – Der ARN, der das Amazon S3-Präfix und den Namen der Sicherungsdateien anzeigt, die zum Wiederherstellen der Datenbank verwendet werden.
  - Geben Sie für das Backup einer einzelnen Datei den gesamten Dateinamen an.
  - Geben Sie für das Backup mehrerer Dateien das Präfix an, das den Dateien gemeinsam ist, und danach das Suffix mit einem Sternchen (*).
  - Wenn `@s3_arn_to_restore_from` leer ist, wird die folgende Fehlermeldung ausgegeben: S3 ARN prefix cannot be empty (Präfix des S3-ARN kann nicht leer sein).

Der folgende Parameter ist für differentielle Wiederherstellungen erforderlich, für vollständige Wiederherstellungen jedoch optional.

- `@with_norecovery` – Die für die Wiederherstellungsoperation zu verwendende Wiederherstellungsklausel.
  - Setzen Sie sie für die Wiederherstellung mit RECOVERY auf `0`. In diesem Fall ist die Datenbank nach der Wiederherstellung online.
  - Setzen Sie sie für die Wiederherstellung mit NORECOVERY auf `1`. In diesem Fall verbleibt die Datenbank nach dem Abschluss der Wiederherstellungsaufgabe im Status RESTORING. Diese Vorgehensweise erlaubt spätere differentielle Wiederherstellungen.
  - Geben Sie für DIFFERENTIELLE Wiederherstellungen `0` oder `1` an.
  - Für FULL-Wiederherstellungen ist dieser Wert standardmäßig `0`.

Die folgenden Parameter sind optional:

- `@kms_master_key_arn` – Wenn Sie die Sicherungsdatei verschlüsselt haben, den KMS-Schlüssel, der zum Entschlüsseln der Datei verwendet werden soll.

Wenn Sie einen KMS-Schlüssel angeben, wird eine client-seitige Verschlüsselung verwendet.

- `@type` – Der Typ der Wiederherstellung. Gültige Typen sind `DIFFERENTIAL` und `FULL`. Der Standardwert ist `FULL`.
- `@block_size`— Blockgröße (in Byte), die die physische Blockgröße für Backup-Operationen angibt. Gültige Werte sind 512, 1024, 2048, 4096, 8192, 16384, 32768 und 65536
- `@max_transfer_size`— Die maximale Übertragungsgröße gibt die Obergrenze des Datenvolumens (in Byte) an, das pro Vorgang während des Backup-Vorgangs übertragen wird. I/O Gültige Werte sind Vielfache von 65536 Byte (64 KB) bis zu 4194304 Byte (4 MB).
- `@buffer_count`— Gesamtzahl der I/O Puffer, die für den Backup-Prozess verwendet werden sollen.

> **Note**
>
> Bei differentiellen Wiederherstellungen muss sich die Datenbank entweder im Status RESTORING befinden, oder es muss bereits eine Aufgabe existieren, die mit NORECOVERY wiederherstellt.
> Sie können differentielle Backups nicht später wiederherstellen, während die Datenbank online ist.
> Sie können keine Wiederherstellungsaufgabe für eine Datenbank absenden, für die bereits eine Wiederherstellungsaufgabe mit RECOVERY aussteht.
> Vollständige Wiederherstellungen mit NORECOVERY werden auf Multi-AZ-Instances oder Instances mit Read Replicas nicht unterstützt.
> Differenzielle Wiederherstellungen werden in keiner Multi-AZ-Bereitstellungskonfiguration unterstützt.
> Alle nativen Wiederherstellungen werden nicht auf Instances unterstützt, die Read Replicas haben, einschließlich regionsübergreifender Read Replicas.
> Bei unterstützten Konfigurationen ähnelt die Wiederherstellung einer Datenbank auf einer Multi-AZ-Instance mit Read Replicas der Wiederherstellung einer Datenbank auf einer Multi-AZ-Instance. Sie müssen keine zusätzlichen Aktionen ausführen, um eine Datenbank auf einem Replikat wiederherzustellen.

Beispiele

Example für die Wiederherstellung einer einzelnen Datei

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

Example für die Wiederherstellung mehrerer Dateien

Um Fehler beim Wiederherstellen mehrerer Dateien zu vermeiden, stellen Sie sicher, dass alle Sicherungsdateien dasselbe Präfix haben, und dass dieses Präfix von keinen anderen Dateien verwendet wird.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*';
```

Example für eine vollständige Datenbankwiederherstellung mit RECOVERY

Die folgenden drei Beispiele führen die gleiche Aufgabe durch, vollständige Wiederherstellung mit RECOVERY.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
[@type='DIFFERENTIAL|FULL'];
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=0;
```

Example für eine vollständige Datenbankwiederherstellung mit Verschlüsselung

```
exec msdb.dbo.rds_restore_database
```

```
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example der Wiederherstellung mit Blockgröße

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@block_size=512;
```

Example der Wiederherstellung mehrerer Dateien mit @max_transfer_size und @buffer_count

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*',
@max_transfer_size=4194304,
@buffer_count=10;
```

Example für eine vollständige Datenbankwiederherstellung mit NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=1;
```

Example für eine differenzielle Wiederherstellung mit NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=1;
```

Example für eine differenzielle Wiederherstellung mit RECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
```

```
@type='DIFFERENTIAL',
@with_norecovery=0;
```

Wiederherstellen eines Protokolls

Zum Wiederherstellen Ihres Protokolls rufen Sie die gespeicherte Prozedur `rds_restore_log` auf.

Verwendung

```
exec msdb.dbo.rds_restore_log
 @restore_db_name='database_name',
 @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/log_file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@with_norecovery=0|1],
 [@stopat='datetime'],
 [@block_size=512|1024|2048|4096|8192|16384|32768|65536],
        [@max_transfer_size=n],
        [@buffer_count=n];
```

Die folgenden Parameter sind erforderlich:

- `@restore_db_name` – Der Name der Datenbank, deren Protokoll wiederhergestellt werden soll.
- `@s3_arn_to_restore_from` – Der ARN mit dem Amazon S3-Präfix und dem Namen der Protokolldatei für die Wiederherstellung des Protokolls. Die Datei kann eine beliebige Erweiterung haben, üblicherweise wird jedoch `.trn` verwendet.

  Wenn `@s3_arn_to_restore_from` leer ist, wird die folgende Fehlermeldung ausgegeben: S3 ARN prefix cannot be empty (Präfix des S3-ARN kann nicht leer sein).

Die folgenden Parameter sind optional:

- `@kms_master_key_arn` - Wenn Sie das Protokoll verschlüsselt haben, den KMS-Schlüssel, der zur Entschlüsselung des Protokolls verwendet werden soll.
- `@with_norecovery` – Die für die Wiederherstellungsoperation zu verwendende Wiederherstellungsklausel. Dieser Wert ist standardmäßig `1`.
  - Setzen Sie sie für die Wiederherstellung mit RECOVERY auf `0`. In diesem Fall ist die Datenbank nach der Wiederherstellung online. Sie können differentielle Protokollsicherungen nicht später wiederherstellen, während die Datenbank online ist.

  - Setzen Sie sie für die Wiederherstellung mit NORECOVERY auf 1. In diesem Fall verbleibt die Datenbank nach dem Abschluss der Wiederherstellungsaufgabe im Status RESTORING. Diese Vorgehensweise erlaubt spätere Protokollwiederherstellungen.
- `@stopat` – Ein Wert, der angibt, dass die Datenbank zu ihrem Zustand am angegebenen Datum und zur angegebenen Uhrzeit (im Datetime-Format) wiederhergestellt wird. Nur vor diesem angegebenen Zeitpunkt geschriebene Transaktionsprotokolldatensätze werden auf die Datenbank angewendet.

  Wenn dieser Parameter nicht angegeben wird (NULL ist), wird das vollständige Protokoll wiederhergestellt.
- `@block_size`— Blockgröße (in Byte), die die physische Blockgröße für Backup-Operationen angibt. Gültige Werte sind 512, 1024, 2048, 4096, 8192, 16384, 32768 und 65536
- `@max_transfer_size`— Die maximale Übertragungsgröße gibt die Obergrenze des Datenvolumens (in Byte) an, das pro Vorgang während des Backup-Vorgangs übertragen wird. I/O Gültige Werte sind Vielfache von 65536 Byte (64 KB) bis zu 4194304 Byte (4 MB).
- `@buffer_count`— Gesamtzahl der I/O Puffer, die für den Backup-Prozess verwendet werden sollen.

> **Note**
>
> Bei Protokollwiederherstellungen muss sich die Datenbank entweder im Status RESTORING befinden, oder es muss bereits eine Aufgabe existieren, die mit NORECOVERY wiederherstellt.
> Sie können differentielle Protokollsicherungen nicht wiederherstellen, während die Datenbank online ist.
> Sie können keine Protokollwiederherstellungsaufgabe für eine Datenbank absenden, für die bereits eine Wiederherstellungsaufgabe mit RECOVERY aussteht.
> Protokollwiederherstellungen werden auf Multi-AZ-Instances nicht unterstützt.

Beispiele

Example für eine Protokollwiederherstellung

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
```

```
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example für eine Protokollwiederherstellung mit Verschlüsselung

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example für eine Protokollwiederherstellung mit NORECOVERY

Die folgenden zwei Beispiele führen die gleiche Aufgabe durch, die Protokollwiederherstellung mit NORECOVERY.

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=1;
```

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example der Wiederherstellung mit Blockgröße

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@block_size=512;
```

Example für eine Protokollwiederherstellung mit RECOVERY

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0;
```

Example für eine Protokollwiederherstellung mit STOPAT-Klausel

```
exec msdb.dbo.rds_restore_log
```

```
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0,
@stopat='2019-12-01 03:57:09';
```

Abschluss einer Datenbankwiederherstellung

Wenn die letzte Wiederherstellungsaufgabe auf der Datenbank mit `@with_norecovery=1` durchgeführt wurde, befindet sich die Datenbank jetzt im Status RESTORING. Öffnen Sie diese Datenbank für den normalen Betrieb mit der gespeicherten Prozedur `rds_finish_restore`.

Verwendung

```
exec msdb.dbo.rds_finish_restore @db_name='database_name';
```

> **Note**
>
> Für diese Vorgehensweise muss sich die Datenbank im Status RESTORING ohne ausstehende Wiederherstellungsaufgaben befinden.
> Die `rds_finish_restore`-Prozedur wird auf Multi-AZ-Instances nicht unterstützt.
> Verwenden Sie zum Abschluss der Wiederherstellung der Datenbank die Master-Anmeldeinformationen. Sie können auch die Benutzeranmeldung verwenden, die zuletzt die Datenbank oder das Protokoll mit NORECOVERY wiederhergestellt hat.

Arbeiten mit teilweise wiederhergestellten Datenbanken

Verwerfen einer teilweise wiederhergestellten Datenbank

Verwenden Sie zum Verwerfen einer teilweise wiederhergestellten Datenbank (die im Status RESTORING verbleibt) die gespeicherte Prozedur `rds_drop_database`.

```
exec msdb.dbo.rds_drop_database @db_name='database_name';
```

> **Note**
>
> Sie können keine DROP-Datenbankanfrage für eine Datenbank absenden, für die bereits eine ausstehende Wiederherstellungs- oder Wiederherstellungsabschlussaufgabe besteht.

Verwenden Sie zum Verwerfen der Datenbank die Master-Anmeldung. Sie können auch die Benutzeranmeldung verwenden, die zuletzt die Datenbank oder das Protokoll mit NORECOVERY wiederhergestellt hat.

Snapshot-Wiederherstellung und point-in-time Wiederherstellungsverhalten für teilweise wiederhergestellte Datenbanken

Teilweise wiederhergestellte Datenbanken in der Quellinstanz (im Status point-in-time RESTORING belassen) werden während der Snapshot-Wiederherstellung und -Wiederherstellung aus der Zielinstanz gelöscht.

Abbrechen einer Aufgabe

Zum Abbruch einer Sicherungs- oder Wiederherstellungsaufgabe rufen Sie die gespeicherte Prozedur `rds_cancel_task` auf.

> **Note**
>
> Eine FINISH_RESTORE-Aufgabe kann nicht abgebrochen werden.

Verwendung

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

Der folgende Parameter ist erforderlich:

- `@task_id` – ID der abzubrechenden Aufgabe Sie erhalten die ID der Aufgabe durch den Aufruf von `rds_task_status`.

Verfolgen des Status von Aufgaben

Um den Status Ihrer Sicherungs- oder Wiederherstellungsaufgaben zu verfolgen, rufen Sie die gespeicherte Prozedur `rds_task_status` auf. Wenn Sie keine Parameter angeben, gibt die gespeicherte Prozedur den Status aller Aufgaben zurück. Der Status für Aufgaben wird etwa alle zwei Minuten aktualisiert. Der Abfrageverlauf wird 36 Tage gespeichert.

Verwendung

```
exec msdb.dbo.rds_task_status
 [@db_name='database_name'],
 [@task_id=ID_number];
```

Die folgenden Parameter sind optional:

- `@db_name` – Name der Datenbank, für die der Aufgabenstatus angezeigt werden soll
- `@task_id` – ID der Aufgabe, für die der Aufgabenstatus angezeigt werden soll

Beispiele

Example für die Auflistung des Status für eine bestimmte Aufgabe

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example für die Auflistung des Status für eine bestimmte Datenbank und Aufgabe

```
exec msdb.dbo.rds_task_status
@db_name='my_database',
@task_id=5;
```

Example für die Auflistung aller Aufgaben und ihrer Status auf einer bestimmten Datenbank

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example für die Auflistung aller Aufgaben und ihrer Status auf der aktuellen Instance

```
exec msdb.dbo.rds_task_status;
```

Antwort

Die gespeicherte Prozedur `rds_task_status` gibt die folgenden Spalten zurück.

Spalte	Beschreibung
`task_id`	Die ID der Aufgabe.

Spalte	Beschreibung

`task_type`	Aufgabentyp je nach Eingabeparametern, wie folgt:
	• Für Sicherungsaufgaben:
	  • BACKUP_DB – Vollständige Datenbanksicherung
	  • BACKUP_DB_DIFFERENTIAL – Differentielle Datenbanksicherung
	• Für Wiederherstellungsaufgaben:
	  • RESTORE_DB – Vollständige Datenbankwiederherstellung mit RECOVERY
	  • RESTORE_DB_NORECOVERY – Vollständige Datenbankwiederherstellung mit NORECOVERY
	  • RESTORE_DB_DIFFERENTIAL – Differentielle Datenbankwiederherstellung mit RECOVERY
	  • RESTORE_DB_DIFFERENTIAL_NORECOVERY – Differentielle Datenbankwiederherstellung mit NORECOVERY
	  • RESTORE_DB_LOG – Protokollwiederherstellung mit RECOVERY
	  • RESTORE_DB_LOG_NORECOVERY – Protokollwiederherstellung mit NORECOVERY
	• Für Aufgaben, die eine Wiederherstellung abschließen:
	  • FINISH_RESTORE – Abschluss der Wiederherstellung und Öffnen der Datenbank

Spalte	Beschreibung
	Amazon RDS erstellt einen anfänglichen Snapshot der Datenbank nach ihrer Öffnung oder nach dem Abschluss der folgenden Wiederherstellungsaufgaben: • RESTORE_DB • RESTORE_DB_DIFFERENTIAL • RESTORE_DB_LOG • FINISH_RESTORE
`database_name`	Der Name der Datenbank, der die Aufgabe zugeordnet ist.
`% complete`	Der Fortschritt der Aufgabe als Prozentwert.
`duration (mins)`	Zeitdauer für die Ausführung der Aufgabe (in Minuten).

Spalte	Beschreibung
`lifecycle`	Der Status der Aufgabe. Die folgenden Status sind möglich: • `CREATED`: Sobald Sie `rds_backup_database` oder `rds_restore_database` aufrufen, wird eine Aufgabe angelegt und der Status auf `CREATED` gesetzt. • `IN_PROGRESS` – Nach dem Start einer Sicherungs- oder Wiederherstellungsaufgabe wird der Status auf gesetzt `IN_PROGRESS`. Es kann bis zu 5 Minuten dauern, bis sich der Status von `CREATED` zu `IN_PROGRESS` ändert. • `SUCCESS` – Nach dem Abschluss einer Sicherungs- oder Wiederherstellungsaufgabe wird der Status auf gesetzt `SUCCESS`. • `ERROR` – Wenn eine Sicherungs- oder Wiederherstellungsaufgabe fehlschlägt, wird der Status auf gesetzt `ERROR`. Weitere Informationen über den Fehler können Sie der Spalte `task_info` entnehmen. • `CANCEL_REQUESTED`: Sobald Sie `rds_cancel_task` aufrufen, wird der Status der Aufgabe auf `CANCEL_REQUESTED` gesetzt. • `CANCELLED` – Nachdem die Aufgabe abgebrochen wurde, wird der Status der Aufgabe auf gesetzt `CANCELLED`.
`task_info`	Zusätzliche Informationen über die Aufgabe. Wenn während des Sicherns oder Wiederherstellens einer Datenbank ein Fehler auftritt, enthält diese Spalte Informationen über den Fehler. Eine Liste möglicher Fehler und Abhilfemaßnahmen finden Sie unter Fehlerbehebung.
`last_updated`	Datum und Uhrzeit der letzten Aktualisierung des Aufgabenstatus. Der Status wird jeweils nach 5 Prozent Fortschritt aktualisiert.
`created_at`	Datum und Uhrzeit, an denen die Aufgabe angelegt wurde.

Spalte	Beschreibung
`S3_object_arn`	Der ARN mit dem Amazon S3-Präfix und dem Namen der Datei, die gesichert oder wiederhergestellt wird.
`overwrite_s3_backup_file`	Wert des Parameters `@overwrite_s3_backup_file`, der beim Aufruf einer Sicherungsaufgabe angegeben wurde. Weitere Informationen finden Sie unter Sichern einer Datenbank.
`KMS_master_key_arn`	Der ARN für den KMS-Schlüssel, der für die Verschlüsselung (für die Sicherung) und Entschlüsselung (für die Wiederherstellung) verwendet wird.
`filepath`	Gilt nicht für native Sicherungs- und Wiederherstellungsaufgaben
`overwrite_file`	Gilt nicht für native Sicherungs- und Wiederherstellungsaufgaben

Komprimieren von Sicherungsdateien

Zur Platzersparnis in Ihrem Amazon S3-Bucket können Sie Ihre Sicherungsdateien komprimieren. Weitere Informationen zum Komprimieren von Sicherungsdateien finden Sie unter Sicherungskomprimierung in der Microsoft-Dokumentation.

Die Komprimierung Ihrer Sicherungsdateien wird für die folgenden Datenbankversionen unterstützt:

- Microsoft SQL Server Enterprise Edition
- Microsoft SQL Server Standard Edition

Führen Sie den folgenden Code aus, um die Komprimierungsoption für Ihre Sicherungsdateien zu überprüfen:

```
exec rdsadmin.dbo.rds_show_configuration 'S3 backup compression';
```

Um die Komprimierung für Ihre Sicherungsdateien einzuschalten, führen Sie den folgenden Code aus:

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'true';
```

Um die Komprimierung für Ihre Sicherungsdateien auszuschalten, führen Sie den folgenden Code aus:

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'false';
```

Fehlerbehebung

Die folgenden Probleme können bei der Verwendung nativer Backups und Wiederherstellungen auftreten.

Problem	Vorschläge für die Fehlerbehebung
Die Option zur Backup/Wiederherstellung der Datenbank ist noch nicht aktiviert oder wird gerade aktiviert. Bitte versuchen Sie es später noch einmal.	Achten Sie darauf, die `SQLSERVER_BACKUP_RESTORE` -Option zur DB-Optionsgruppe hinzuzufügen, die mit Ihrer DB-Instance verbunden ist. Weitere Informationen finden Sie unter Hinzufügen der Option „Native Sicherung und Backup“.
Die EXECUTE-Berechtigung wurde für das Objekt '', die Datenbank *`rds_backup_database`* 'msdb', das Schema 'dbo' verweigert.	Stellen Sie sicher, dass Sie den Masterbenutzer verwenden, wenn Sie die gespeicherte Prozedur ausführen. Wenn dieser Fehler auftritt, obwohl Sie als Masterbenutzer angemeldet sind, kann dies daran liegen, dass die Administratorberechtigungen falsch ausgerichtet sind. Um den Masterbenutzer zurückzusetzen, verwenden Sie den AWS Management Console. Siehe Zurücksetzen der db_owner-Rollenmitgliedschaft für den Master-Benutzer für Amazon for Server RDS SQL.
Die EXECUTE-Berechtigung wurde für das Objekt '', die Datenbank *`rds_restore_database`* 'msdb', das Schema 'dbo' verweigert.	Stellen Sie sicher, dass Sie den Masterbenutzer verwenden, wenn Sie die gespeicherte Prozedur ausführen. Wenn dieser Fehler auftritt, obwohl Sie als Masterbenutzer angemeldet sind, kann dies daran liegen, dass die Administratorberechtigungen falsch ausgerichtet sind. Um den Masterbenutzer zurückzusetzen, verwenden Sie den AWS Management Console. Siehe Zurücksetzen der db_owner-Rollenmitgliedschaft für den Master-Benutzer für Amazon for Server RDS SQL.

Problem	Vorschläge für die Fehlerbehebung
Zugriff verweigert	Der Sicherungs- oder Wiederherstellungsprozess kann nicht auf die Sicherungsdatei zugreifen. Dies hat normalerweise Ursachen wie die folgenden: • Verweis auf einen falschen Bucket. Verweis auf einen Bucket unter Verwendung eines falschen Formats. Verweis auf einen Dateinamen ohne Verwendung des ARN. • Falsche Berechtigungen für die Bucket-Datei. Wenn die Datei beispielsweise von einem anderen Konto als dem erstellt wurde, mit dem jetzt darauf zugegriffen werden soll, müssen die richtigen Berechtigungen hinzugefügt werden. • Eine falsche oder unvollständige IAM-Richtlinie. Die IAM-Rolle muss alle erforderlichen Elemente enthalten also beispielsweise auch die richtige Version. Die Elemente werden unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung beschrieben.
BACKUP-DATENBANK MIT KOMPRESSION wird in Edition nicht unterstützt<edition_name>	Die Komprimierung Ihrer Sicherungsdateien wird nur für Microsoft SQL Server Enterprise Edition und Standard Edition unterstützt. . Weitere Informationen finden Sie unter Komprimieren von Sicherungsdateien.
Schlüssel existiert nicht<ARN>	Sie haben versucht, ein verschlüsseltes Backup wiederherzustellen , haben aber keinen Verschlüsselungsschlüssel angegeben. Prüfen Sie Ihren Verschlüsselungsschlüssel und versuchen Sie es erneut. Weitere Informationen finden Sie unter Wiederherstellen einer Datenbank.

Problem	Vorschläge für die Fehlerbehebung
Bitte geben Sie die Aufgabe mit dem richtigen Typ neu aus und überschreiben Sie die Eigenschaft	Wenn Sie versuchen, Ihre Datenbank zu sichern, und Sie den Namen einer bereits vorhandenen Datei angeben, aber die Eigenschaft "overwrite" auf "false" gesetzt haben, schlägt der Sicherungsvorgang fehl. Geben Sie zur Korrektur dieses Fehlers entweder den Namen einer noch nicht vorhandenen Datei an, oder setzen Sie die Eigenschaft "overwrite" auf "true". Weitere Informationen finden Sie unter Sichern einer Datenbank. Zudem ist es möglich, dass Sie Ihre Datenbank wiederherstellen wollten, aber versehentlich die gespeicherte Prozedur `rds_backup_database` aufgerufen haben. Rufen Sie in diesem Fall stattdessen die gespeicherte Prozedur `rds_restore_database` auf. Weitere Informationen finden Sie unter Wiederherstellen einer Datenbank. Wenn Sie Ihre Datenbank wiederherstellen wollten und die gespeicherte Prozedur `rds_restore_database` aufgerufen haben, vergewissern Sie sich, dass Sie den Namen einer gültigen Sicherungsdatei angegeben haben. Weitere Informationen finden Sie unter Verwenden nativer Backups und Wiederherstellungen.
Bitte geben Sie einen Bucket an, der sich in der gleichen Region wie die RDS-Instance befindet	Sie können keine Backups in einem Amazon S3-Bucket in einer anderen AWS Region als Ihrer Amazon RDS-DB-Instance erstellen oder aus einem solchen Bucket wiederherstellen. Sie können die Amazon S3 S3-Replikation verwenden, um die Sicherungsdatei in die richtige AWS Region zu kopieren. Weitere Informationen finden Sie unter Regionenübergreifende Replikation in der Amazon S3-Dokumentation.

Problem	Vorschläge für die Fehlerbehebung
Der angegebene Bucket existiert nicht	Stellen Sie sicher, dass Sie den korrekten ARN für Ihren Bucket und die Datei im richtigen Format angegeben haben. Weitere Informationen finden Sie unter Verwenden nativer Backups und Wiederherstellungen.
Der Benutzer ist nicht berechtigt, auf Ressourcen zu arbeiten<ARN> <kms action> <ARN>	Sie haben einen verschlüsselten Vorgang angefordert, aber nicht die richtigen AWS KMS Berechtigungen erteilt. Überprüfen Sie, ob Sie die korrekten Berechtigungen haben oder fügen Sie diese hinzu. Weitere Informationen finden Sie unter Einrichtung für native Backups und Wiederherstellungen.
Die Wiederherstellungsaufgabe kann nicht aus mehr als 10 Sicherungsdateien wiederhergestellt werden). Bitte reduzieren Sie die Anzahl der übereinstimmenden Dateien und versuchen Sie es erneut.	Reduzieren Sie die Anzahl der Dateien, aus denen Sie wiederherstellen möchten. Sie können jede einzelne Datei bei Bedarf größer machen.
Die Datenbank '*database_name* ' ist bereits vorhanden. Zwei Datenbanken, die sich nur nach Fall oder Akzent unterscheiden, sind nicht zulässig. Wählen Sie einen anderen Datenbanknamen aus.	Sie können eine Datenbank nicht mit dem gleichen Namen wie eine vorhandene Datenbank wiederherstellen. Die Namen der Datenbank sind eindeutig.

Importieren und Exportieren von SQL Server-Daten mithilfe anderer Methoden

Im Folgenden finden Sie Informationen zur Verwendung von Snapshots für den Import Ihrer Microsoft SQL Server-Daten zu Amazon RDS. Sie finden hier auch Informationen zur Verwendung von Snapshots für den Export Ihrer Daten von einer RDS DB-Instance mit SQL Server.

Wenn es Ihr Szenario unterstützt, ist es einfacher, Daten mithilfe der nativen Sicherungs- und Wiederherstellungsfunktionalität zu und aus Amazon RDS zu verschieben. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

> **Note**
>
> Amazon RDS für Microsoft SQL Server unterstützt nicht das Importieren von Daten in die `msdb`-Datenbank.

Importieren von Daten in RDS for SQL Server mithilfe eines Snapshots

So können Sie Daten in eine SQL Server-DB-Instance mithilfe eines Snapshots importieren

1. Erstellen Sie eine DB -Instance. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
2. Stoppen Sie den Zugriff von Anwendungen auf die gewünschte DB-Instance.

   Wenn Sie den Zugriff auf Ihre DB-Instance blockieren, während Sie Daten importieren, läuft die Datenübertragung schneller. Zudem müssen Sie sich keine Sorgen über Konflikte machen, während die Daten geladen werden, wenn andere Anwendungen nicht in die DB-Instance schreiben können. Falls ein Problem auftritt und Sie zum vorherigen Datenbank-Snapshot zurückkehren müssen, verlieren Sie lediglich die importierten Daten. Sie können diese Daten dann nach der Behebung des Problems erneut importieren.

   Weitere Informationen über die Zugriffskontrolle auf Ihre DB-Instance finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
3. Erstellen Sie einen Snapshot der Ziel-Datenbank.

Wenn die Zieldatenbank bereits Daten enthält, empfehlen wir Ihnen einen Snapshot dieser Datenbank zu machen, bevor Sie Daten importieren. Wenn etwas beim Datenimport schief geht oder Sie die Änderungen verwerfen möchten, können Sie den vorherigen Zustand der Datenbank mithilfe des Snapshots wiederherstellen. Weitere Informationen zu Datenbank-Snapshots finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

> **Note**
>
> Wenn Sie einen Datenbanksnapshot erstellen, werden I/O Operationen an der Datenbank für einen Moment (Millisekunden) unterbrochen, während die Sicherung läuft.

4. Deaktivieren Sie automatische Backups für die Ziel-Datenbank.

   Das Deaktivieren der automatischen Backups auf der DB-Ziel-Instance verbessert die Leistung beim Importieren Ihrer Daten, da Amazon RDS die Transaktionen nicht protokolliert, wenn automatische Backups deaktiviert sind. Jedoch müssen einige Dinge beachtet werden. Automatisierte Backups sind erforderlich, um eine Wiederherstellung durchzuführen. point-in-time Sie können daher nicht eine Datenbank zu einem bestimmten Zeitpunkt wiederherstellen, während Sie Daten importieren. Darüber hinaus werden alle automatisierten Backups, die für die DB-Instance erstellt wurden, gelöscht - es sei denn, Sie möchten sie behalten.

   Wenn Sie sich für die Aufbewahrung der automatisierten Backups entscheiden, können Sie sich vor versehentlichem Löschen von Daten schützen. Amazon RDS speichert außerdem die Eigenschaften der Datenbank-Instance zusammen mit jedem automatisierten Backup, um die Wiederherstellung zu erleichtern. Mit dieser Option können Sie eine gelöschte Datenbankinstanz auch nach dem Löschen zu einem bestimmten Zeitpunkt innerhalb der Aufbewahrungsfrist für Backups wiederherstellen. Automatische Backups werden am Ende des angegebenen Backup-Zeitraums automatisch gelöscht, genau wie bei einer aktiven Datenbank-Instance.

   Sie können auch frühere Snapshots verwenden, um die Datenbank wiederherzustellen. Snapshots, die Sie erstellt haben, bleiben verfügbar. Weitere Informationen zu automatischen Backups finden Sie unter Einführung in Backups.

5. Deaktivieren Sie auswärtige Schlüsselbeschränkungen, wenn möglich.

   Wenn Sie auswärtige Schlüsselbeschränkungen deaktivieren müssen, können Sie das folgende Skript verwenden.

```
--Disable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' NOCHECK CONSTRAINT
 ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;

    GO
```

6. Verwerfen Sie Indizes, wenn möglich.
7. Deaktivieren Sie Auslöser, wenn möglich.

   Wenn Sie Auslöser deaktivieren müssen, können Sie das folgende Skript verwenden.

```
--Disable triggers on all tables
    DECLARE @enable BIT = 0;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
     table_object.name table_name
    FROM sysobjects trigger_object
    JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
    WHERE trigger_object.type = 'TR';

    OPEN trigger_cursor;
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

    WHILE @@FETCH_STATUS = 0 BEGIN
      IF @enable = 1
         SET @cmd = 'ENABLE ';
```

```
      ELSE
         SET @cmd = 'DISABLE ';

      SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
      EXEC (@cmd);
      FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
   END

   CLOSE trigger_cursor;
   DEALLOCATE trigger_cursor;

   GO
```

8. Suchen Sie in der Quell-SQL Server-Instance nach Anmeldedaten, die Sie in die gewünschte DB-Instance importieren möchten.

   SQL Server speichert Anmeldedaten und Passwörter in der `master`-Datenbank. Da Amazon RDS keinen Zugriff auf die `master`-Datenbank gewährt, können Sie Anmeldedaten und Passwörter nicht direkt in Ihre DB-Instance importieren. Stattdessen müssen Sie die `master`-Datenbank auf der SQL Server-Quell-Instance abfragen, um eine DDL (Data Definition Language)-Datei zu generieren. Diese Datei sollte alle Anmeldedaten und Passwörter enthalten, die Sie der DB-Ziel-Instance hinzufügen möchten. Diese Datei sollte auch die Rollenmitgliedschaften und Berechtigungen enthalten, die Sie übertragen möchten.

   Informationen zum Abfragen der `master` Datenbank finden Sie in der Microsoft Knowledge Base unter Übertragen von Anmeldungen und Kennwörtern zwischen Instanzen von SQL Server.

   Die Ausgabe des Skripts ist ein anderes Skript, das Sie in der gewünschten DB-Instance ausführen können. Das Skript im Knowledge Base-Artikel hat den folgenden Code:

   ```
   p.type IN
   ```

   Jedes Mal wenn `p.type` erscheint, verwenden Sie stattdessen den folgenden Code:

   ```
   p.type = 'S'
   ```

9. Importieren Sie Daten mithilfe der Methode in Importieren der Daten.
10. Gewähren Sie Anwendungen Zugriff auf die Ziel-DB-Instance.

Wenn Ihr Datenimport abgeschlossen ist, können Sie den Anwendungen, die Sie während des Imports blockiert haben, den Zugang zur DB-Instance gewähren. Weitere Informationen über die Zugriffskontrolle auf Ihre DB-Instance finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

11. Aktivieren Sie automatische Backups für die Ziel-DB-Instance.

    Weitere Informationen zu automatischen Backups finden Sie unter Einführung in Backups.

12. Aktivieren Sie auswärtige Schlüsselbeschränkungen.

    Wenn Sie auswärtige Schlüsselbeschränkungen vorher deaktiviert haben, können Sie diese jetzt mit dem folgenden Skript aktivieren.

```
--Enable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' CHECK CONSTRAINT ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;
```

13. Aktivieren Sie Indizes, wenn möglich.
14. Aktivieren Sie Auslöser, wenn möglich.

    Wenn Sie Auslöser vorher deaktiviert haben, können Sie diese jetzt mit dem folgenden Skript aktivieren.

```
--Enable triggers on all tables
    DECLARE @enable BIT = 1;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
```

```
    table_object.name table_name
  FROM sysobjects trigger_object
  JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
  WHERE trigger_object.type = 'TR';

  OPEN trigger_cursor;
  FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

  WHILE @@FETCH_STATUS = 0 BEGIN
    IF @enable = 1
       SET @cmd = 'ENABLE ';
    ELSE
       SET @cmd = 'DISABLE ';

    SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
    EXEC (@cmd);
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
  END

  CLOSE trigger_cursor;
  DEALLOCATE trigger_cursor;
```

Importieren der Daten

Microsoft SQL Server Management Studio ist ein grafischer SQL Server-Client, der in allen Microsoft SQL Server Editionen enthalten ist, außer in der Express Edition. SQL Server Management Studio Express von Microsoft ist als kostenloser Download verfügbar. Sie finden diesen Download auf der Microsoft-Website.

> **Note**
>
> SQL Server Management Studio ist nur als Windows-basierte Anwendung verfügbar.

SQL Server Management Studio beinhaltet die folgenden Tools, die nützlich für den Import von Daten in eine SQL Server DB-Instance sind:

- Assistent für das Generieren und Veröffentlichen von Skripts
- Assistent für den Import und Export

- Bulk-Kopie

Assistent für das Generieren und Veröffentlichen von Skripts

Der Assistent für das Generieren und Veröffentlichen von Skripts erstellt ein Skript, das das Schema einer Datenbank, die Daten selbst, oder beides enthält. Sie können ein Skript für eine Datenbank in Ihrer lokalen SQL Server-Bereitstellung generieren. Sie können dann das Skript ausführen, um die darin enthaltenen Informationen zu einer Amazon RDS-DB-Instance zu übertragen.

> **Note**
>
> Bei Datenbanken mit einer Größe von 1 GiB oder mehr ist es effizienter, nur für das Datenbankschema ein Skript zu verwenden. Anschließend verwenden Sie den Assistenten für Import und Export oder die Massenkopierfunktion von SQL Server, um die Daten zu übertragen.

Weitere detaillierte Informationen über den Assistent für das Generieren und Veröffentlichen von Skripts finden Sie in der Microsoft SQL Server-Dokumentation.

Beachten Sie im Assistenten besonders die erweiterten Optionen auf der Seite Set Scripting Options (Scripting-Optionen einstellen), um sicherzustellen, dass alle Inhalte, die im Skript enthalten sein sollen, ausgewählt wurden. Beispielsweise sind standardmäßig keine Datenbank-Auslöser im Skript enthalten.

Wenn das Skript generiert und gespeichert ist, können Sie SQL Server Management Studio verwenden, um sich mit Ihrer DB-Instance zu verbinden und das Skript auszuführen.

Assistent für den Import und Export

Der Import-Export-Assistent erstellt ein spezielles Paket mit Integrationsdiensten, die Sie verwenden können, um Daten aus Ihrer lokalen SQL Server-Datenbank in die gewünschte DB-Instance zu kopieren. Der Assistent kann filtern, welche Tabellen und sogar welche Tupel innerhalb von Tabellen in die bestimmte DB-Instance kopiert werden sollen.

> **Note**
>
> Der Import-Export-Assistent funktioniert gut mit großen Datensätzen, aber es ist nicht unbedingt der schnellste Weg, um Daten remote aus Ihrer lokalen Einrichtung zu exportieren. Eine schnellere Methode ist die SQL Server-Bulk-Kopie-Funktion.

Weitere detaillierte Informationen über den Import-Export-Assistenten finden Sie in der Microsoft SQL Server-Dokumentation.

Führen Sie im Assistenten auf der Seite Choose a Destination (Ziel auswählen) folgende Schritte aus:

- Geben Sie im Feld Servername den Namen des Endpunkts für Ihre DB-Instance ein.
- Wählen Sie als Server-Authentifizierungsmodus Use SQL Server Authentication (SQL Server-Authentifizierung verwenden) aus.
- Geben Sie im Feld Benutzername und Passwort die Anmeldeinformationen für den Hauptbenutzer ein, den Sie für die DB-Instance erstellt haben.

Bulk-Kopie

Die SQL Server-Bulk-Kopie-Funktion ist eine effiziente Methode zum Kopieren von Daten aus einer Quelldatenbank auf Ihre DB-Instance. Bulk-Kopie schreibt die Daten, die Sie für die Datei festlegen, wie z. B. eine ASCII-Datei. Sie können die Bulk-Kopie nochmal ausführen, um die Inhalte der Datei in die gewünschte DB-Instance zu schreiben.

In diesem Abschnitt wird das Hilfsprogramm bcp verwendet, das in allen Editionen von SQL Server enthalten ist. Weitere detaillierte Informationen zu Bulk-Import und -Export-Operationen finden Sie in der Microsoft SQL Server-Dokumentation.

> **Note**
>
> Bevor Sie eine Bulk-Kopie verwenden, müssen Sie zuerst Ihr Datenbankschema in die gewünschte DB-Instance importieren. Der Assistent für das Generieren und Veröffentlichen von Skripts, der vorher in diesem Thema beschrieben wurde, ist ein ausgezeichnetes Tool für diesen Zweck.

Der folgende Befehl stellt eine Verbindung zur lokalen SQL Server-Instance her. Er generiert eine tabulatorgetrennte Datei für eine angegebene Tabelle im C:\-Stammverzeichnis Ihrer bestehenden SQL Server-Bereitstellung. Die Tabelle wird durch ihren vollständigen, gültigen Namen angegeben und die Textdatei hat denselben Namen wie die kopierte Tabelle.

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -
P password -b 10000
```

Der vorherige Code beinhaltet die folgenden Optionen:

- `-n` gibt an, dass eine Massenkopie die nativen Datentypen der Daten verwendet, die zu kopieren sind.
- `-S` gibt die SQL Server-Instance an, mit der sich das Hilfsprogramm bcp verbindet.
- `-U` gibt den Benutzernamen des Kontos an, das in der SQL Server-Instance angemeldet wird.
- `-P` gibt das Passwort für den Benutzer an `-U`.
- `-b` gibt die Anzahl der Zeilen pro Batch importierter Daten an.

> **Note**
>
> Es könnte noch andere Parameter geben, die für Ihren Importvorgang wichtig sind. Beispielsweise könnten Sie den Parameter `-E` benötigen, der sich auf die Identitätswerte bezieht. Weitere Informationen finden Sie in der vollständigen Beschreibung der Befehlszeilensyntax für das Hilfsprogramm bcp in der Microsoft SQL Server-Dokumentation.

Angenommen, eine Datenbank mit dem Namen `store`, die das Standardschema dbo verwendet, enthält eine Tabelle mit dem Namen `customers`. Das Benutzerkonto `admin` mit dem Passwort `insecure` kopiert 10.000 Zeilen aus der Tabelle `customers` in eine Datei mit dem Namen `customers.txt`.

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b
 10000
```

Nachdem Sie die Datendatei erstellt haben, können Sie die Daten mit einem ähnlichen Befehl zu Ihrer DB-Instance hochladen. Erstellen Sie vorher die Datenbank und das Schema auf der DB-Ziel-Instance. Verwenden Sie dann das Argument `in`, um die Input-Datei anzugeben, anstelle von `out`,

um die Output-Datei anzugeben. Anstelle der Verwendung von localhost zur Angabe der lokalen SQL Server-Instance, geben Sie den Endpunkt Ihrer DB-Instance an. Wenn Sie einen anderen Port als 1433 verwenden, geben Sie auch diesen an. Der Benutzername und das Passwort entsprechen den Angaben für den Master-Benutzer Ihrer DB-Instance. Die Syntax ist wie folgt.

```
bcp dbname.schema_name.table_name
     in C:\table_name.txt -n -S endpoint,port -U master_user_name -
P master_user_password -b 10000
```

Um mit dem vorherigen Beispiel fortzufahren, nehmen wir an, dass der Benutzername `admin` und das Passwort `insecure` lauten. Der Endpunkt für diese DB-Instance lautet `rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com` und verwendet Port 4080. Der Befehl lautet wie folgt.

```
bcp store.dbo.customers in C:\customers.txt -n -S rds.ckz2kqd4qsn1.us-
east-1.rds.amazonaws.com,4080 -U admin -P insecure -b 10000
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Exportieren von Daten aus RDS for SQL Server

Sie können eine der folgenden Optionen auswählen, um Daten aus einer RDS for SQL-Server-DB-Instance zu exportieren:

- Native Datenbanksicherung mithilfe einer vollständigen Sicherungsdatei (BAK): Die Verwendung von BAK-Dateien, um Datenbanken zu sichern, ist hochoptimiert und im Normalfall die schnellste Methode, um Daten zu exportieren. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.
- SQL Server Import and Export Wizard – weitere Informationen finden Sie unter SQL Server-Assistent für Import und Export (SQL Server Import and Export Wizard).
- SQL Server Generate and Publish Scripts Wizard und bcp-Hilfsprogramm: Weitere Informationen finden Sie unter SQL Server-Assistent für das Generieren und Veröffentlichen von Skripts und Hilfsprogramm bcp.

SQL Server-Assistent für Import und Export (SQL Server Import and Export Wizard)

Sie können den SQL Server-Assistenten für Import und Export verwenden, um eine oder mehrere Tabellen, Ansichten oder Abfragen aus Ihrer RDS for SQL-Server-DB-Instance in einen anderen Datenspeicher zu kopieren. Dies ist die beste Wahl, wenn sich der Ziel-Datenspeicher nicht in SQL Server befindet. Weitere Informationen finden Sie unter SQL Server Import and Export Wizard in der SQL Server-Dokumentation.

Der SQL Server-Assistent für Import und Export ist als Teil von Microsoft SQL Server Management Studio verfügbar. Dieser grafische SQL Server-Client ist in allen Microsoft SQL Server-Editionen außer in der Express Edition enthalten. SQL Server Management Studio ist nur als Windows-basierte Anwendung verfügbar. SQL Server Management Studio Express von Microsoft ist als kostenloser Download verfügbar. Sie finden diesen Download auf der Microsoft-Website.

So verwenden Sie SQL Server-Assistent für Import und Export, um Daten zu exportieren

1. Stellen Sie in SQL Server Management Studio eine Verbindung mit Ihrer RDS for SQL-Server-DB-Instance her. Weitere detaillierte Informationen hierzu finden Sie unter Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen.
2. Erweitern Sie im Object Explorer (Objektexplorer) den Abschnitt Datenbanken, öffnen Sie das Kontextmenü (rechter Mausklick) für die Quelldatenbank, wählen Sie Aufgaben und anschließend Export Data (Daten exportieren) aus. Der Assistent wird angezeigt.
3. Öffnen Sie die Seite Choose a Data Source (Datenquelle auswählen) und führen Sie folgende Schritte durch:

    a. Wählen Sie für Datenquelle **`SQL Server Native Client 11.0`** aus.

    b. Stellen Sie sicher, dass das Feld Server name (Servername) den Endpunkt Ihrer RDS for SQL-Server-DB-Instance anzeigt.

    c. Wählen Sie Use SQL Server Authentication (SQL Server-Authentifizierung verwenden) aus. Geben Sie für Benutzername und Passwort den Hauptbenutzernamen und das Hauptpasswort Ihrer DB-Instance ein.

    d. Stellen Sie sicher, dass das Feld Datenbank die Datenbank anzeigt, von der Sie Daten exportieren möchten.

    e. Wählen Sie Weiter aus.
4. Führen Sie auf der Seite Choose a Destination (Ziel auswählen) folgende Schritte durch:

    a. Geben Sie für Ziel die Zeichenfolge **`SQL Server Native Client 11.0`** an.

> **Note**
>
> Es sind weitere Zieldatenquellen verfügbar. Dazu gehören: .NET Framework-Datenanbieter, OLE DB-Anbieter, SQL Server Native Client-Anbieter, ADO.NET-Anbieter, Microsoft Office Excel, Microsoft Office Access und die Flat File-Quelle. Wenn Sie eine dieser Datenquellen auswählen, überspringen Sie den Rest von Schritt 4. Einzelheiten zu den Verbindungsinformationen, die als Nächstes anzugeben sind, finden Sie unter Auswählen eines Ziels in der SQL Server-Dokumentation.

b. Geben Sie für Servername den Servernamen der SQL Server-DB-Ziel-Instance an.

c. Wählen Sie den angemessenen Authentifizierungstyp aus. Geben Sie einen Benutzernamen und Passwort ein, wenn nötig.

d. Wählen Sie für Datenbank den Namen der Zieldatenbank aus oder wählen Sie Neu aus, um eine neue Datenbank zu erstellen, in die die exportierten Daten gespeichert werden.

   Wenn Sie New (Neu) auswählen, finden Sie unter Datenbank erstellen in der SQL Server-Dokumentation weitere Einzelheiten über bereitzustellende Datenbankinformationen.

e. Wählen Sie Weiter aus.

5. Wählen Sie auf der Seite Table Copy or Query (Tabellenkopie oder -abfrage) die Option Copy data from one or more tables or views (Daten von einer oder mehreren Tabellen oder Ansichten kopieren) oder Write a query to specify the data to transfer (Anfrage zur Spezifizierung der Transferdaten schreiben) aus. Wählen Sie Weiter aus.

6. Wenn Sie Write a query to specify the data to transfer (Anfrage zur Spezifizierung der Transferdaten schreiben) auswählen, wird die Seite Provide a Source Query (Quellabfrage bereitstellen) angezeigt. Kopieren oder tippen Sie eine SQL-Abfrage ein und wählen Sie anschließend Parse (Parsen) aus, um sie zu überprüfen. Wenn die Abfrage bestätigt ist, klicken Sie auf Weiter.

7. Führen Sie auf der Seite Select Source Tables and Views (Quelltabellen oder -ansichten auswählen) die folgenden Schritte durch:

   a. Wählen Sie die Tabellen und Ansichten, die Sie exportieren möchten, aus oder überprüfen Sie, dass die Abfrage, die sie getätigt haben, ausgewählt ist.

b. Wählen Sie Edit Mappings (Zuweisungen bearbeiten) aus und geben Sie die Datenbank- und Spaltenzuweisungsinformationen an. Weitere Informationen finden Sie unter Spaltenzuordnungen in der SQL Server-Dokumentation.

c. (Optional) Wählen Sie die Tabelle, Ansicht oder Abfrage aus und klicken Sie anschließend auf Preview (Vorversion), um eine Vorversion der zu exportierenden Daten anzuzeigen.

d. Wählen Sie Weiter aus.

8. Auf der Seite Run Package (Paket ausführen) muss Run immediately (Sofort ausführen) ausgewählt sein. Wählen Sie Weiter.
9. Überprüfen Sie auf der Seite Complete the Wizard (Assistenten abschließen), dass die Details zum Datenexport wie erwartet sind. Wählen Sie Finish (Abschließen).
10. Wählen Sie auf der Seite The execution was successful (Ausführung erfolgreich) die Option Schließen aus.

SQL Server-Assistent für das Generieren und Veröffentlichen von Skripts und Hilfsprogramm bcp

Sie können den SQL Server-Assistenten für das Generieren und Veröffentlichen von Skripts verwenden, um Skripts für eine gesamte Datenbank oder nur für ausgewählte Objekte zu erstellen. Sie können diese Skripts auf einer Ziel-SQL Server-DB-Instance ausführen, um die geskripteten Objekte neu zu erstellen. Sie können anschließend das Hilfsprogramm bcp verwenden, um einen Bulk-Export der Daten für die ausgewählten Objekte in die Ziel-DB-Instance durchzuführen. Diese Methode ist die beste Wahl, wenn Sie eine gesamte Datenbank (einschließlich Objekten, die keine Tabellen sind) oder große Datenmengen zwischen zwei SQL Server-DB-Instances verschieben möchten. Eine vollständige Beschreibung der bcp-Befehlszeilensyntax finden Sie unter bcp-Dienstprogramm in der Microsoft SQL Server-Dokumentation.

Der SQL Server-Assistent zum Generieren und Veröffentlichen von Skripts ist als Teil von Microsoft SQL Server Management Studio verfügbar. Dieser grafische SQL Server-Client ist in allen Microsoft SQL Server-Editionen außer in der Express Edition enthalten. SQL Server Management Studio ist nur als Windows-basierte Anwendung verfügbar. SQL Server Management Studio Express von Microsoft ist als kostenloser Download verfügbar.

So können Sie den SQL Server-Assistenten für das Generieren und Veröffentlichen von Skripts und das Hilfsprogramm bcp für einen Datenexport verwenden

1. Stellen Sie in SQL Server Management Studio eine Verbindung mit Ihrer RDS für SQL-DB-Instance her. Weitere detaillierte Informationen hierzu finden Sie unter Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen.

2. Erweitern Sie in Object Explorer (Objektexplorer) den Knotenpunkt Datenbanken und wählen Sie die Datenbank aus, die Sie skripten möchten.
3. Folgen Sie den Anweisungen in Assistent zum Generieren und Veröffentlichen von Skripts in der SQL Server-Dokumentation, um eine Skriptdatei zu erstellen.
4. Verbinden Sie sich in SQL Server Management Studio mit Ihrer Ziel-SQL Server-DB-Instance.
5. Mit ausgewählter SQL Server-DB-Ziel-Instance im Object Explorer (Objektexplorer) wählen Sie im Menü File (Datei) die Option Open (Öffnen) aus, wählen Sie File (Datei) und öffnen Sie die Skriptdatei.
6. Wenn Sie für die gesamte Datenbank ein Skript erstellt haben, prüfen Sie die Anweisung CREATE DATABASE im Skript. Stellen Sie sicher, dass die Datenbank an dem von Ihnen gewünschten Speicherort und mit den korrekten Parametern erstellt wird. Weitere Informationen finden Sie unter CREATE DATABASE in der SQL Server-Dokumentation.
7. Wenn Sie im Skript Benutzer für die Datenbank erstellen, überprüfen Sie, ob bereits Serveranmeldedaten in der Ziel-DB-Instance für diese Benutzer vorhanden sind. Wenn dies nicht der Fall ist, erstellen Sie Anmeldedaten für diese Benutzer. Andernfalls werden die geskripteten Befehle für das Erstellen der Datenbankbenutzer fehlschlagen. Weitere Informationen finden Sie unter Erstellen eines Anmeldenamens in der SQL Server-Dokumentation.
8. Wählen Sie im SQL-Editor-Menü !Execute aus, um die Skriptdatei auszuführen und die Datenbankobjekte zu erstellen. Sobald das Skript abgeschlossen ist, überprüfen Sie, dass alle Datenbankobjekte wie erwartet vorhanden sind.
9. Verwenden Sie das Hilfsprogramm bcp, um Daten aus der RDS for SQL-Server-DB-Instance in Dateien zu exportieren. Öffnen Sie eine Eingabeaufforderung und geben Sie den folgenden Befehl ein.

   ```
   bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -
   U username -P password
   ```

   Der vorherige Code beinhaltet die folgenden Optionen:

   - table_name ist der Name einer der Tabellen, die Sie in der Zieldatenbank neu erstellt haben und die jetzt mit Daten versehen wird.
   - data_file ist der vollständige Pfad und Name der Datei, die erstellt werden soll.
   - `-n` gibt an, dass eine Massenkopie die nativen Datentypen der Daten verwendet, die zu kopieren sind.
   - `-S` gibt die SQL Server-DB-Instance an, aus der exportiert werden soll.

- `-U` gibt den zu verwendenden Benutzernamen an, wenn eine Verbindung mit der SQL Server-DB-Instance hergestellt wird.
- `-P` gibt das Passwort für den Benutzer an `-U`.

Im Folgenden wird ein Beispielbefehl gezeigt.

```
bcp world.dbo.city out C:\Users\JohnDoe\city.dat -n -S sql-jdoe.1234abcd.us-
west-2.rds.amazonaws.com,1433 -U JohnDoe -P ClearTextPassword
```

Wiederholen Sie diesen Schritt, bis Sie Dateien für alle Tabellen haben, die Sie exportieren möchten.

10. Bereiten Sie Ihre DB-Ziel-Instance für den Massenimport von Daten mithilfe der Anweisungen unter Vorbereiten des Massenimports von Daten in der SQL Server-Dokumentation vor.
11. Entscheiden Sie sich für eine Methode zum Massenimport, nachdem Sie die Leistung und andere wichtige Faktoren eingeschätzt haben, die unter Informationen zu Massenimport- und Massenexportvorgängen in der SQL Server-Dokumentation aufgeführt werden.
12. Führen Sie einen Massenimport der Daten aus den mit dem bcp-Hilfsprogramm erstellten Datendateien aus. Befolgen Sie hierzu die Anweisungen unter Im- und Exportieren von Massendaten mithilfe des Dienstprogramms bcp oder Importieren von Massendaten mithilfe von BULK INSERT oder OPENROWSET(BULK...) in der SQL Server-Dokumentation, abhängig von Ihrer Entscheidung in Schritt 11.

Verwenden des BCP-Dienstprogramms unter Linux zum Importieren und Exportieren von Daten

Das BCP-Hilfsprogramm (Bulk Copy Program) bietet eine effiziente Möglichkeit, große Datenmengen zwischen Ihrer RDS for SQL Server-DB-Instance und Datendateien zu übertragen. Sie können BCP von Linux-Umgebungen aus verwenden, um Massendatenoperationen durchzuführen, was es für Datenmigrationen, ETL-Prozesse und regelmäßige Datenübertragungen nützlich macht.

BCP unterstützt sowohl das Importieren von Daten aus Dateien in SQL Server-Tabellen als auch das Exportieren von Daten aus SQL Server-Tabellen in Dateien. Dies ist besonders effektiv für die Übertragung strukturierter Daten in verschiedenen Formaten, einschließlich Textdateien mit Trennzeichen.

Voraussetzungen

Bevor Sie BCP mit Ihrer RDS for SQL Server-DB-Instance von Linux aus verwenden, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Eine Linux-Umgebung mit Netzwerkkonnektivität zu Ihrer RDS for SQL Server-DB-Instance
- Auf Ihrem Linux-System installierte Microsoft SQL Server-Befehlszeilentools, darunter:
  - sqlcmd — SQL Server-Befehlszeilen-Abfragetool
  - bcp — Programm zum Massenkopieren
- Gültige Anmeldeinformationen für Ihre RDS for SQL Server-DB-Instance
- Netzwerkzugriff, der über Sicherheitsgruppen konfiguriert wird, um Verbindungen über den SQL Server-Port zuzulassen (normalerweise 1433)
- Geeignete Datenbankberechtigungen für die Operationen, die Sie ausführen möchten

Installation der SQL Server-Befehlszeilentools unter Linux

Um BCP von Linux aus zu verwenden, müssen Sie die Microsoft SQL Server-Befehlszeilentools installieren. Detaillierte Installationsanweisungen für Ihre spezifische Linux-Distribution finden Sie in der folgenden Microsoft-Dokumentation:

- Installieren Sie die SQL Server-Befehlszeilentools sqlcmd und bcp unter Linux
- BCP-Hilfsprogramm — Vollständige Referenz für das BCP-Hilfsprogramm

Stellen Sie nach der Installation sicher, dass die Tools in Ihrem PATH verfügbar sind, indem Sie Folgendes ausführen:

```
bcp -v
sqlcmd -?
```

Exportieren von Daten aus RDS for SQL Server

Sie können BCP verwenden, um Daten aus Ihrer RDS for SQL Server-DB-Instance in Dateien auf Ihrem Linux-System zu exportieren. Dies ist nützlich, um Backups zu erstellen, Daten zu analysieren oder Daten für die Migration vorzubereiten.

Grundlegende Exportsyntax

Die grundlegende Syntax für den Export von Daten mit BCP lautet:

```
bcp database.schema.table out output_file -S server_name -U username -P password
 [options]
```

Wobei gilt:

- `database.schema.table`- Der vollqualifizierte Tabellenname
- `output_file`- Der Pfad und der Name der Ausgabedatei
- `server_name`- Ihr RDS für SQL Server-Endpunkt
- `username`- Ihr Datenbank-Benutzername
- `password`- Ihr Datenbank-Passwort

Beispiel exportieren

Das folgende Beispiel exportiert Daten aus einer Tabelle, die `customers` in der `sales` Datenbank benannt ist:

```
bcp sales.dbo.customers out /home/user/customers.txt \
    -S mydb.cluster-abc123.us-east-1.rds.amazonaws.com \
    -U admin \
    -P mypassword \
    -c \
    -t "|" \
```

```
    -r "\n"
```

Dieser Befehl:

- Exportiert Daten aus der `customers` Tabelle
- Speichert die Ausgabe in `/home/user/customers.txt`
- Verwendet das Zeichenformat (`-c`)
- Verwendet Pipe (|) als Feldtrennzeichen () `-t "|"`
- Verwendet Newline als Zeilentrennzeichen () `-r "\n"`

Daten nach RDS für SQL Server importieren

Sie können BCP verwenden, um Daten aus Dateien auf Ihrem Linux-System in Ihre RDS for SQL Server-DB-Instance zu importieren. Dies ist nützlich für die Datenmigration, das Laden von Testdaten oder regelmäßige Datenaktualisierungen.

Grundlegende Importsyntax

Die grundlegende Syntax für den Import von Daten mit BCP lautet:

```
bcp database.schema.table in input_file -S server_name -U username -P password
 [options]
```

Wobei gilt:

- `database.schema.table`- Der vollqualifizierte Name der Zieltabelle
- `input_file`- Der Pfad und der Name der Eingabedatei
- `server_name`- Ihr RDS für SQL Server-Endpunkt
- `username`- Ihr Datenbank-Benutzername
- `password`- Ihr Datenbank-Passwort

Beispiel importieren

Im folgenden Beispiel werden Daten aus einer Datei in eine Tabelle mit dem Namen importiert`customers`:

```
bcp sales.dbo.customers in /home/user/customers.txt \
    -S mydb.cluster-abc123.us-east-1.rds.amazonaws.com \
    -U admin \
    -P mypassword \
    -c \
    -t "|" \
    -r "\n" \
    -b 1000
```

Dieser Befehl:

- Importiert Daten in die `customers` Tabelle
- Liest Daten aus `/home/user/customers.txt`
- Verwendet das Zeichenformat (`-c`)
- Verwendet Pipe (|) als Feldtrennzeichen () `-t "|"`
- Verwendet Newline als Zeilentrennzeichen () `-r "\n"`
- Verarbeitet Daten in Stapeln von 1000 Zeilen () `-b 1000`

Allgemeine BCP-Optionen

BCP bietet zahlreiche Optionen zur Steuerung der Datenformatierung und des Übertragungsverhaltens. In der folgenden Tabelle werden häufig verwendete Optionen beschrieben:

Option	Beschreibung
`-c`	Verwendet den Zeichendatentyp für alle Spalten
`-n`	Verwendet native Datenbank-Datentypen
`-t`	Gibt das Feldtrennzeichen an (Standard ist Tab)
`-r`	Gibt das Zeilentrennzeichen an (Standard ist Newline)
`-b`	Gibt die Batchgröße für Massenoperationen an
`-F`	Gibt die erste Zeile an, die exportiert oder importiert werden soll
`-L`	Gibt die letzte Zeile an, die exportiert oder importiert werden soll

Option	Beschreibung
`-e`	Gibt eine Fehlerdatei zum Erfassen abgelehnter Zeilen an
`-f`	Gibt eine Formatdatei für die Datenformatierung an
`-q`	Verwendet Bezeichner in Anführungszeichen für Objektnamen

Bewährte Methoden und Überlegungen

Beachten Sie bei der Verwendung von BCP mit RDS für SQL Server unter Linux die folgenden bewährten Methoden:

- Stapelverarbeitung verwenden — Verwenden Sie bei großen Datensätzen die `-b` Option, Daten stapelweise zu verarbeiten. Dies verbessert die Leistung und ermöglicht eine bessere Fehlerbehebung.
- Fehler ordnungsgemäß behandeln — Verwenden Sie `-e` diese Option, um Fehlerinformationen und abgelehnte Zeilen zur Analyse in einer separaten Datei zu erfassen.
- Wählen Sie geeignete Datenformate — Verwenden Sie das Zeichenformat (`-c`) für plattformübergreifende Kompatibilität oder das systemeigene Format (`-n`) für eine bessere Leistung, wenn sowohl Quelle als auch Ziel SQL Server sind.
- Schützen Sie Ihre Anmeldeinformationen — Vermeiden Sie es, Passwörter direkt in Befehlszeilen einzugeben. Erwägen Sie die Verwendung von Umgebungsvariablen oder Konfigurationsdateien mit entsprechenden Berechtigungen.
- Testen Sie mit kleinen Datensätzen — Bevor Sie große Datenmengen verarbeiten, testen Sie Ihre BCP-Befehle mit kleineren Datensätzen, um die Formatierung und Konnektivität zu überprüfen.
- Überwachen Sie die Netzwerkkonnektivität — Sorgen Sie für stabile Netzwerkverbindungen, insbesondere bei großen Datenübertragungen. Erwägen Sie die Verwendung von Tools wie `screen` oder `tmux` für lang andauernde Operationen.
- Überprüfen Sie die Datenintegrität — Überprüfen Sie nach der Datenübertragung die Zeilenanzahl und die Stichprobendaten, um sicherzustellen, dass der Vorgang erfolgreich abgeschlossen wurde.

Behebung häufig auftretender Probleme

In der folgenden Tabelle werden häufig auftretende Probleme beschrieben, die bei der Verwendung von BCP unter Linux auftreten können, sowie deren Lösungen:

Problem	Lösung
Verbindungs-Timeout oder Netzwerkfehler	Überprüfen Sie Ihren Amazon RDS-Endpunkt, Ihre Sicherheitsgruppeneinstellungen und Ihre Netzwerkkonnektivität. Stellen Sie sicher, dass der SQL Server-Port (normalerweise 1433) von Ihrem Linux-System aus zugänglich ist.
Authentication failures (Authentifizierungsfehler)	Überprüfen Sie Ihren Benutzernamen und Ihr Passwort. Stellen Sie sicher, dass der Datenbankbenutzer über die entsprechenden Berechtigungen für die von Ihnen ausgeführten Operationen verfügt.
Fehler im Datenformat	Überprüfen Sie Ihre Feld- und Zeilentrennzeichen. Stellen Sie sicher, dass das Datenformat den Erwartungen von BCP entspricht. Verwenden Sie Formatdateien für komplexe Datenstrukturen.
Fehler bei abgelehnter Berechtigung	Stellen Sie sicher, dass Ihr Datenbankbenutzer über `INSERT` Berechtigungen für Importe oder `SELECT` Berechtigungen für Exporte in den Zieltabellen verfügt.
Probleme bei der Handhabung großer Dateien	Verwenden Sie die Stapelverarbeitung mit der `-b` Option. Erwägen Sie, große Dateien in kleinere Teile aufzuteilen, um die Leistung zu verbessern und Fehler zu beheben.
Probleme mit der Zeichenkodierung	Stellen Sie sicher, dass Ihre Datendateien eine kompatible Zeichenkodierung verwenden. Verwenden Sie die `-c` Option für das Zeichenformat oder geben Sie die entsprechenden Codepages an.

Arbeiten mit Read Replicas für Microsoft SQL Server in Amazon RDS

Sie verwenden Lesereplikate üblicherweise, um die Replikation zwischen Amazon RDS-DB-Instances zu konfigurieren. Allgemeine Informationen zu Lesereplikaten finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

In diesem Abschnitt finden Sie spezifische Informationen zum Arbeiten mit Lesereplikaten unter Amazon RDS for SQL Server.

- Synchronisieren von Datenbankbenutzern und Objekten mit einer SQL Server-Read Replica
- Behebung eines SQL Server-Read-Replica-Problems

Konfigurieren von Read Replicas für SQL Server

Bevor eine DB-Instance als Quell-Instance für die Replikation eingesetzt werden kann, müssen Sie automatische Sicherungen auf der Quell-DB-Instance aktivieren. Hierzu legen Sie für den Aufbewahrungszeitraum für Sicherungen einen anderen Wert als 0 fest. Durch das Festlegen dieses Bereitstellungstyps wird außerdem erzwungen, dass automatische Sicherungen aktiviert sind.

Das Erstellen eines SQL Server-Lesereplikats erfordert keinen Wartungsausfall für die primäre DB-Instance. Amazon RDS legt die erforderlichen Parameter und Berechtigungen für die Quelldatenbank-Instance und das Lesereplikat ohne Serviceunterbrechung fest. Ein Snapshot von der Quell-DB-Instance wird gemacht und dieser Snapshot wird zum Lesereplikat. Es findet kein Nutzungsausfall statt, wenn Sie ein Lesereplikat löschen.

Sie können bis zu 15 Lesereplikate aus einer Quell-DB-Instance erstellen. Damit die Replikation effektiv durchgeführt werden kann, empfehlen wir Ihnen, jedes Lesereplikat mit derselben Menge an Rechen- und Speicherressourcen wie die Quell-DB-Instance zu konfigurieren. Wenn Sie die Quell-DB-Instance skalieren, skalieren Sie auch die Lesereplikate.

Die SQL-Server-DB-Engine-Version der Quell-DB-Instance und alle Lese-Replikate müssen identisch sein. Amazon RDS aktualisiert die primäre Instance unmittelbar nach dem Upgrade der Lesereplikate, unabhängig vom Wartungsfenster. Weitere Informationen zum Aktualisieren der DB-Engine-Version finden Sie unter Upgrades der Microsoft SQL Server-DB-Engine.

Damit ein Lesereplikat Änderungen von der Quelle empfängt und anwendet, sollte es über ausreichende Rechen- und Speicherressourcen verfügen. Wenn ein Lesereplikat die Kapazität

von Rechen-, Netzwerk- und Speicherressourcen erreicht hat, stellt das Lesereplikat den Empfang und die Anwendung von Änderungen aus seiner Quelle ein. Sie können die Speicher- und CPU-Ressourcen eines Lesereplikats unabhängig von seiner Quelle und anderen Lesereplikaten ändern.

Weitere Hinweise zum Erstellen einer Read Replica finden Sie unter. Erstellen eines Lesereplikats

Read-Replica-Einschränkungen mit SQL Server

Die folgenden Einschränkungen gelten für SQL Server-Lesereplikate in Amazon RDS:

- Lesereplikate sind nur auf der SQL Server Enterprise Edition (EE)-Engine verfügbar.
- Read Replicas sind für die SQL Server-Versionen 2016—2022 verfügbar.
- Sie können bis zu 15 Lesereplikate aus einer Quell-DB-Instance erstellen. Die Replikation kann verzögert werden, wenn Ihre Quell-DB-Instance über mehr als 5 Read Replicas verfügt.
- Read Replicas sind nur für DB-Instances verfügbar, die auf DB-Instance-Klassen mit vier oder mehr v ausgeführt werden. CPUs
- Eine Read Replica unterstützt je nach Instance-Klassentyp und Verfügbarkeitsmodus bis zu 100 Datenbanken. Sie müssen Datenbanken auf der Quell-DB-Instance erstellen, um sie automatisch auf die Read Replicas zu replizieren. Sie können keine einzelnen Datenbanken für die Replikation auswählen. Weitere Informationen finden Sie unter Einschränkungen für Microsoft SQL Server-DB-Instances.
- Sie können eine Datenbank nicht aus einer Read Replica löschen. Um eine Datenbank zu löschen, löschen Sie sie mit der `rds_drop_database` gespeicherten Prozedur aus der Quell-DB-Instance. Weitere Informationen finden Sie unter Löschen einer Datenbank in einer Amazon RDS for Microsoft SQL Server Server-DB-Instance.
- Wenn die Quell-DB-Instance Transparent Data Encryption (TDE) zum Verschlüsseln von Daten verwendet, konfiguriert die Read Replica auch automatisch TDE.

  Wenn die Quell-DB-Instance einen KMS-Schlüssel zum Verschlüsseln von Daten verwendet, verwenden Lesereplikate in derselben Region denselben KMS-Schlüssel. Für regionsübergreifende Read Replicas müssen Sie beim Erstellen der Read Replica einen KMS-Schlüssel aus der Region der Read Replica angeben. Sie können den KMS-Schlüssel für eine Read Replica nicht ändern.
- Read Replicas haben dieselbe Zeitzone und Sortierung wie die Quell-DB-Instance, unabhängig von der Availability Zone, in der sie erstellt wurden.
- Folgendes wird in Amazon RDS for SQL Server nicht unterstützt:

  - Backup-Aufbewahrung von Lesereplikaten
  - Point-in-time Wiederherstellung von Read Replicas
  - Manuelle Snapshots von Lesereplikaten
  - Multi-AZ-Lesereplikate
  - Erstellen von Lesereplikaten aus Lesereplikaten
  - Synchronisierung von Benutzeranmeldungen bei Lesereplikaten
- Amazon RDS for SQL Server greift nicht ein, um eine hohe Replikationsverzögerung zwischen einer Quell-DB-Instance und ihren Lesereplikaten zu minimieren. Stellen Sie sicher, dass die Quell-DB-Instance und ihre Lesereplikate in Bezug auf Rechen- und Speicherkapazität die für ihre Betriebslast angemessene Größe aufweisen.
- Sie können zwischen den Regionen AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) replizieren, jedoch nicht innerhalb oder außerhalb. AWS GovCloud (US) Regions

Überlegungen zu Optionen für Replikate von RDS für SQL Server

Bevor Sie ein Replikat von RDS für SQL Server erstellen, sollten Sie die folgenden Anforderungen, Einschränkungen und Empfehlungen berücksichtigen:

- Wenn sich Ihr SQL-Server-Replikat in derselben Region befindet wie die Quell-DB-Instance, stellen Sie sicher, dass es zur gleichen Optionsgruppe gehört wie die Quell-DB-Instance. Änderungen an der Quell-Optionsgruppe oder der Quell-Optionsgruppenmitgliedschaft werden von den Replikaten übernommen. Diese Änderungen werden unmittelbar, nachdem sie auf die Quell-DB-Instance angewandt wurden, auf die Replikate angewandt, ungeachtet des Wartungsfensters des Replikats.

  Weitere Informationen über Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.
- Wenn Sie ein regionsübergreifendes SQL-Server-Replikat erstellen, erstellt Amazon RDS dafür eine dedizierte Optionsgruppe.

  Ein regionsübergreifendes SQL-Server-Replikat kann nicht aus seiner dedizierten Optionsgruppe entfernt werden. Die dedizierte Optionsgruppe eines regionsübergreifenden SQL-Server-Replikats kann nicht von anderen DB-Instances verwendet werden.

  Bei den folgenden Optionen handelt es sich um replizierte Optionen. Wenn Sie einem regionsübergreifenden SQL-Server-Replikat replizierte Optionen hinzufügen möchten, fügen Sie es der Optionsgruppe der Quell-DB-Instance hinzu. Die Option wird auch auf allen Replikaten der Quell-DB-Instance installiert.

- TDE

Bei den folgenden Optionen handelt es sich nicht um replizierte Optionen. Sie können nicht replizierte Optionen einer dedizierten Optionsgruppe hinzufügen oder daraus entfernen.

- MSDTC
- SQLSERVER_AUDIT
- Wenn Sie die SQLSERVER_AUDIT-Option für ein regionsübergreifendes Replikat aktivieren möchten, fügen Sie die SQLSERVER_AUDIT-Option der dedizierten Optionsgruppe für das regionsübergreifende Lesereplikat und die Optionsgruppe der Quell-Instance hinzu. Indem Sie die SQLSERVER_AUDIT-Option für die Quell-Instance des regionsübergreifenden Lesereplikats von SQL Server hinzufügen, können Sie für jedes der regionsübergreifenden Lesereplikate der Quell-Instance ein Prüfobjekt auf Serverebene und Prüfspezifikationen auf Serverebene erstellen. Wenn Sie den regionsübergreifenden Lesereplikaten Zugriffberechtigungen für das Hochladen der abgeschlossenen Prüfprotokolle in einen Amazon-S3-Bucket gewähren möchten, fügen Sie der dedizierten Optionsgruppe die SQLSERVER_AUDIT-Option hinzu und konfigurieren Sie die Optionseinstellungen. Der als Ziel für die Überwachungsdateien verwendete Amazon-S3-Bucket muss sich in derselben Region befinden wie das regionsübergreifende Lesereplikat. Sie können die Optionseinstellung der SQLSERVER_AUDIT-Option für jedes regionsübergreifende Lesereplikat unabhängig ändern, sodass jedes Replikat in seiner jeweiligen Region auf einen Amazon-S3-Bucket zugreifen kann.

Die folgenden Optionen werden für Read Replicas nicht unterstützt.

- SSRS
- SSAS
- SSIS

Die folgenden Optionen werden für regionsübergreifende Lesereplikate teilweise unterstützt.

- SQLSERVER_BACKUP_RESTORE
- Die Quell-DB-Instance eines regionsübergreifenden SQL-Server-Replikats kann über die SQLSERVER_BACKUP_RESTORE-Option verfügen, Sie können jedoch erst systemeigene Wiederherstellungen für die Quell-DB-Instance durchführen, wenn Sie alle ihre regionsübergreifenden Replikate gelöscht haben. Alle vorhandenen systemeigenen Wiederherstellungsaufgaben werden bei der Erstellung eines regionsübergreifenden Replikats abgebrochen. Sie können die SQLSERVER_BACKUP_RESTORE-Option nicht einer dedizierten Optionsgruppe hinzufügen.

Weitere Informationen zu systemeigenen Backups und Wiederherstellungen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Wenn Sie ein regionsübergreifendes SQL-Server-Lesereplikat hochstufen, verhält sich das hochgestufte Lesereplikat genau so wie andere DB-Instances von SQL Server, einschließlich der Verwaltung seiner Optionen. Weitere Informationen über Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Synchronisieren von Datenbankbenutzern und Objekten mit einer SQL Server-Read Replica

Es wird erwartet, dass alle Logins, benutzerdefinierten Serverrollen, SQL Agent-Jobs oder andere Objekte auf Serverebene, die zum Zeitpunkt der Erstellung einer Read Replica in der primären DB-Instance vorhanden sind, in der neu erstellten Read Replica vorhanden sind. Objekte auf Serverebene, die nach der Erstellung des Lesereplikats in der primären DB-Instance erstellt wurden, werden jedoch nicht automatisch repliziert. Sie müssen sie manuell im Lesereplikat erstellen.

Die Datenbankbenutzer werden automatisch von der primären DB-Instance in das Lesereplikat repliziert. Da sich die Read Replica-Datenbank im schreibgeschützten Modus befindet, kann die Sicherheits-ID (SID) des Datenbankbenutzers in der Datenbank nicht aktualisiert werden. Daher muss beim Erstellen von SQL Logins in der Read Replica unbedingt sichergestellt werden, dass SID der Anmeldename mit dem entsprechenden SQL Anmeldenamen in SID der primären DB-Instance übereinstimmt. Wenn Sie die SIDs SQL Logins nicht synchronisieren, können sie nicht auf die Datenbank in der Read Replica zugreifen. Bei authentifizierten Windows Active Directory (AD) - Anmeldungen tritt dieses Problem nicht auf, da der SQL Server die Daten aus dem Active Directory bezieht. SID

Um eine SQL Anmeldung von der primären DB-Instance mit der Read Replica zu synchronisieren

1. Stellen Sie eine Verbindung mit der primären DB-Instance her.
2. Erstellen Sie ein neues SQL Login in der primären DB-Instance.

```
USE [master]
GO
CREATE LOGIN TestLogin1
WITH PASSWORD = 'REPLACE WITH PASSWORD';
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

3. Erstellen Sie einen neuen Datenbankbenutzer für die SQL Anmeldung in der Datenbank.

```
USE [REPLACE WITH YOUR DB NAME]
GO
CREATE USER TestLogin1 FOR LOGIN TestLogin1;
GO
```

4. Überprüfen Sie den SID Namen des neu erstellten SQL Logins in der primären DB-Instance.

```
SELECT name, sid FROM sys.server_principals WHERE name =  TestLogin1;
```

5. Stellen Sie eine Verbindung mit dem Lesereplikat her. Erstellen Sie das neue SQL Login.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #4];
```

Wenn Sie Zugriff auf die Lesereplikat-Datenbank haben, können Sie den verwaisten Benutzer alternativ wie folgt korrigieren:

1. Stellen Sie eine Verbindung mit dem Lesereplikat her.
2. Identifizieren Sie die verwaisten Benutzer in der Datenbank.

```
USE [REPLACE WITH YOUR DB NAME]
GO
EXEC sp_change_users_login 'Report';
GO
```

3. Erstellen Sie ein neues SQL Login für den verwaisten Datenbankbenutzer.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #2];
```

   Beispiel:

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'TestPa$$word#1',
 SID=[0x1A2B3C4D5E6F7G8H9I0J1K2L3M4N5O6P];
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Behebung eines SQL Server-Read-Replica-Problems

Sie können die Replikationsverzögerung in Amazon überwachen, CloudWatch indem Sie sich die RDS `ReplicaLag` Amazon-Metrik ansehen. Weitere Informationen zur zeitlichen Verzögerung bei der Replikation finden Sie unter Überwachen der Lesereplikation.

Wenn Replikationsverzögerung zu groß ist, können Sie die folgende Abfrage verwenden, um Informationen über die Verzögerung abzurufen.

```
SELECT AR.replica_server_name
     , DB_NAME (ARS.database_id) 'database_name'
     , AR.availability_mode_desc
     , ARS.synchronization_health_desc
     , ARS.last_hardened_lsn
     , ARS.last_redone_lsn
     , ARS.secondary_lag_seconds
FROM sys.dm_hadr_database_replica_states ARS
INNER JOIN sys.availability_replicas AR ON ARS.replica_id = AR.replica_id
--WHERE DB_NAME(ARS.database_id) = 'database_name'
ORDER BY AR.replica_server_name;
```

Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL

Multi-AZ-Bereitstellungen bieten eine erhöhte Verfügbarkeit, eine längere Lebensdauer von Daten sowie eine höhere Fehlertoleranz für DB-Instances. Im Falle einer geplanten Datenbankwartung oder einer ungeplanten Serviceunterbrechung wechselt Amazon RDS automatisch zur up-to-date sekundären DB-Instance. Mit dieser Funktion können Datenbankoperationen schnell ohne manuellen Eingriff fortgesetzt werden. Die Primär- und Standby-Instances verwenden denselben Endpunkt, dessen physische Netzwerkadresse als Teil des Failoverprozesses am sekundären Replica gespiegelt wird. Sie müssen Ihre Anwendung nicht neu konfigurieren, wenn ein Failover auftritt.

Amazon RDS unterstützt Multi-AZ-Bereitstellungen für Microsoft SQL Server mithilfe von SQL Server Database Mirroring (DBM) oder Always On Availability Groups (). AGs Amazon RDS überwacht und sorgt für den Zustand Ihrer Multi-AZ-Bereitstellung. Wenn Probleme auftreten, repariert es RDS automatisch fehlerhafte DB-Instances, stellt die Synchronisation wieder her und leitet Failover ein. Failover treten nur auf, wenn Standby- und Primär-Instance vollständig synchron sind. Sie müssen nichts verwalten.

Wenn Sie SQL Server Multi-AZ einrichten, RDS werden automatisch alle Datenbanken auf der Instance so konfiguriert, dass sie oder verwenden. DBM AGs Amazon RDS verwaltet die primäre, die Zeugeninstanz und die sekundäre DB-Instance für Sie. Da die Konfiguration automatisch erfolgt, RDS wählen Sie AGs je nach der SQL Serverversion, die Sie bereitstellen, DBM oder Always On aus.

Amazon RDS unterstützt Multi-AZ mit Always On AGs für die folgenden SQL Serverversionen und Editionen:

- SQLServer 2022:
  - Standard Edition
  - Enterprise Edition
- SQLServer 2019:
  - Standard Edition 15.00.4073.23 und höher
  - Enterprise Edition
- SQLServer 2017:
  - Standard Edition 14.00.3401.7 und höher
  - Enterprise Edition 14.00.3049.1 und höher

- SQLServer 2016: Enterprise Edition 13.00.5216.0 und höher

Amazon RDS unterstützt Multi-AZ mit DBM für die folgenden SQL Serverversionen und Editionen, mit Ausnahme der zuvor genannten Versionen:

- SQLServer 2019: Standard Edition 15.00.4043.16
- SQLServer 2017: Standard- und Enterprise-Editionen
- SQLServer 2016: Standard- und Enterprise-Editionen

Sie können die folgende SQL Abfrage verwenden, um festzustellen, ob es sich bei Ihrer SQL Server-DB-Instance um eine Single-AZ-, Multi-AZ- oder Multi-AZ-Instance mit DBM AlwaysOn handelt. AGs

```
SELECT CASE WHEN dm.mirroring_state_desc IS NOT NULL THEN 'Multi-AZ (Mirroring)'
    WHEN dhdrs.group_database_id IS NOT NULL THEN 'Multi-AZ (AlwaysOn)'
    ELSE 'Single-AZ'
    END 'high_availability'
FROM sys.databases sd
LEFT JOIN sys.database_mirroring dm ON sd.database_id = dm.database_id
LEFT JOIN sys.dm_hadr_database_replica_states dhdrs ON sd.database_id =
 dhdrs.database_id AND dhdrs.is_local = 1
WHERE DB_NAME(sd.database_id) = 'rdsadmin';
```

Die Ausgabe sieht in etwa folgendermaßen aus:

```
high_availability
Multi-AZ (AlwaysOn)
```

Hinzufügen von Multi-AZ zu einer Microsoft SQL Server-DB-Instance

Wenn Sie mit dem eine neue SQL Server-DB-Instance erstellen AWS Management Console, können Sie Multi-AZ with Database Mirroring (DBM) oder Always On hinzufügen. AGs Dazu wählen Sie Yes (Mirroring / Always On) (Ja (Spiegelung/Always On)) unter Multi-AZ deployment (Multi-AZ-Bereitstellung) aus. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Wenn Sie eine bestehende SQL Server-DB-Instance mithilfe der Konsole ändern, können Sie Multi-AZ mit DBM oder hinzufügen, AGs indem Sie auf der Seite DB-Instance modifizieren unter Multi-AZ-

Bereitstellung Ja (Mirroring/Always On) auswählen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Note**
>
> Wenn auf Ihrer DB-Instance Database Mirroring (DBM) und nicht Always On Availability Groups (AGs) ausgeführt wird, müssen Sie möglicherweise die In-Memory-Optimierung deaktivieren, bevor Sie Multi-AZ hinzufügen. Deaktivieren Sie die In-Memory-Optimierung mit, DBM bevor Sie Multi-AZ hinzufügen, wenn Ihre DB-Instance SQL Server 2016 oder 2017 Enterprise Edition ausführt und die In-Memory-Optimierung aktiviert ist.
> Wenn Ihre DB-Instance läuftAGs, ist dieser Schritt nicht erforderlich.

Multi-AZ aus einer Microsoft SQL Server-DB-Instance entfernen

Wenn Sie eine bestehende SQL Server-DB-Instance mithilfe von ändern AWS Management Console, können Sie Multi-AZ mit DBM oder entfernen. AGs Sie können dies tun, indem Sie Nein (Spiegelung/Always On) von Multi-AZ deployment (Multi-AZ-Bereitstellung) auf der DB-Instance ändern-Seite auswählen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Einschränkungen, Hinweise und Empfehlungen für Microsoft SQL Server Multi-AZ-Bereitstellung

Im Folgenden sind einige Einschränkungen beim Arbeiten mit Multi-AZ-Bereitstellungen auf RDS für SQL Server-DB-Instances aufgeführt:

- Regionsübergreifende Multi-AZ wird nicht unterstützt.
- Das Beenden einer DB-Instance von RDS für SQL Server in einer Multi-AZ-Bereitstellung wird nicht unterstützt.
- Sie können die sekundäre DB-Instance nicht so konfigurieren, dass sie die Datenbankleseaktivität akzeptiert.
- Multi-AZ mit Always-On-Verfügbarkeitsgruppen (AGs) unterstützt die speicherinterne Optimierung.
- Multi-AZ mit Always-On-Verfügbarkeitsgruppen (AGs) unterstützt keine Kerberos-Authentifizierung für den Verfügbarkeitsgruppen-Listener. Dies liegt daran, dass der Listener keinen Dienstprinzipalnamen (SPN, Service Principal Name) hat.
- Sie können eine Datenbank in einer SQL Server-DB-Instance nicht umbenennen, die sich in einer SQL Server-Multi-AZ-Bereitstellung befindet. Falls Sie eine Datenbank in einer derartigen Instance

umbenennen müssen, deaktivieren Sie erst Multi-AZ für die DB-Instance und benennen dann die Datenbank um. Aktivieren Sie letztendlich Multi-AZ wieder für die DB-Instance.

- Sie können nur Multi-AZ-DB-Instances wiederherstellen, die mithilfe des vollständigen Wiederherstellungsmodells gesichert wurden.
- Multi-AZ-Bereitstellungen haben ein Limit von 10.000 SQL-Server-Agent-Aufträgen.

  Wenn Sie ein höheres Limit benötigen, fordern Sie eine Erhöhung an, indem Sie sich an uns wenden. SupportÖffnen Sie die Seite des AWS -Support -Centers, melden Sie sich an und wählen Sie Fall erstellen aus. Wählen Sie Service Limit increase (Erhöhung des Servicelimits). Füllen Sie das Formular aus und senden Sie es ab.
- Sie können keine Offline-Datenbank auf einer SQL Server-DB-Instance einrichten, die sich in einer SQL Server-Multi-AZ-Bereitstellung befindet.
- RDS für SQL Server repliziert keine MSDB-Datenbankberechtigungen auf die sekundäre Instanz. Wenn Sie diese Berechtigungen für die sekundäre Instanz benötigen, müssen Sie sie manuell neu erstellen.

Hinweise zur Arbeit mit Multi-AZ-Bereitstellungen auf RDS für SQL Server-DB-Instances:

- Amazon RDS macht den Listener-Endpunkt der AGs Always-On-Verfügbarkeitsgruppe verfügbar. Der Endpunkt ist in der Konsole sichtbar und wird vom `DescribeDBInstances`-API-Vorgang als Eintrag im Feld mit den Endpunkten zurückgegeben.
- Amazon RDS unterstützt Failover bei mehreren Subnetzen in Verfügbarkeitsgruppen.
- Zur Verwendung von SQL Server-Multi-AZ mit einer SQL Server-DB-Instance in einer Virtual Private Cloud (VPC) erstellen Sie zuerst eine DB-Subnetzgruppe, die Subnetze in mindestens zwei verschiedenen Availability Zones aufweist. Sie weisen anschließend die DB-Subnetzgruppe dem primären Replica der SQL Server-DB-Instance zu.
- Wenn eine DB-Instance in eine Multi-AZ-Bereitstellung geändert wird, hat sie während der Änderung den Status Modifying (Wird geändert …). Amazon RDS erstellt den Standby und erstellt ein Backup der primären DB-Instance. Wenn der Prozess abgeschlossen ist, ändert sich der Status der primären DB-Instance zu Available (Verfügbar).
- Multi-AZ-Bereitstellungen verwalten alle Datenbanken auf demselben Knoten. Bei einem Failover einer Datenbank auf dem primären Host erfolgt ein Failover für alle Ihre SQL Server-Datenbanken als Einheit auf Ihren Standby-Host. Amazon RDS stellt einen neuen fehlerfreien Host bereit und ersetzt den fehlerhaften Host.
- Multi-AZ mit DBM oder AGs unterstützt ein einzelnes Standby-Replikat.

- Benutzer, Logins und Berechtigungen werden auf der sekundären Instance automatisch für Sie repliziert. Sie müssen sie nicht erneut erstellen. Benutzerdefinierte Serverrollen werden nur in DB-Instances repliziert, die Always On für Multi-AZ-Bereitstellungen verwenden. AGs
- In Multi-AZ-Bereitstellungen erstellt RDS for SQL Server SQL Server-Logins, um AlwaysOn oder Datenbankspiegelung zu ermöglichen. AGs RDS erstellt Anmeldungen mit dem folgenden Muster:, und. `db_<dbiResourceId>_node1_login` `db_<dbiResourceId>_node2_login` `db_<dbiResourceId>_witness_login`
- RDS for SQL Server erstellt eine SQL Server-Anmeldung, um den Zugriff auf Read Replicas zu ermöglichen. RDS erstellt einen Anmeldenamen mit dem folgenden Muster:. `db_<readreplica_dbiResourceId>_node_login`
- In Multi-AZ-Bereitstellungen werden Aufträge von SQL Server Agent vom primären Host auf den sekundären Host repliziert, wenn die Auftragsreplikationsfunktion aktiviert ist. Weitere Informationen finden Sie unter Aktivieren der Auftragsreplikation von SQL Server Agent.
- Aufgrund der synchronen Datenreplikation kann es zu erhöhten Latenzen im Vergleich zur standardmäßigen Bereitstellung einer DB-Instance in einer einzigen Availability Zone kommen.
- Die Failover-Zeiten sind von der Zeit abhängig, die für den Wiederherstellungsprozess benötigt wird. Große Transaktionen erhöhen die Failover-Zeit.
- In SQL Server-Multi-AZ-Bereitstellungen wird bei einem Neustart mit Failover nur die primäre DB-Instance neu gestartet. Nach dem Failover wird die primäre DB-Instance zur neuen sekundären DB-Instance. Die Parameter werden für Multi-AZ-Instances möglicherweise nicht aktualisiert. Für einen Neustart ohne Failover starten sowohl die primäre als auch die sekundäre DB-Instance neu. Die Parameter werden nach dem Neustart aktualisiert. Wenn die DB-Instance nicht reagiert, empfehlen wir einen Neustart ohne Failover.

Empfehlungen für die Arbeit mit Multi-AZ-Bereitstellungen auf RDS für Microsoft SQL Server-DB-Instances:

- Für Datenbanken, die in der Produktion oder Vorproduktion verwendet werden, empfehlen wir die folgenden Optionen:
  - Multi-AZ-Bereitstellungen für Hochverfügbarkeit
  - Provisioned IOPS für schnelle, konsistente Leistung
  - „Speicheroptimiert“ statt „Universell“
- Sie können die Availability Zone (AZ) für die sekundäre Instance nicht auswählen. Berücksichtigen Sie dies daher bei der Bereitstellung von Anwendungshosts. Für Ihre Datenbank konnte kein

Failover auf eine andere AZ durchgeführt werden, und die Anwendungshosts befinden sich möglicherweise nicht in derselben AZ wie die Datenbank. Aus diesem Grund empfehlen wir, dass Sie Ihre Anwendungshosts auf alle Hosts AZs in der jeweiligen AWS Region verteilen.

- Um eine optimale Leistung zu erzielen, sollten Sie die Option Datenbankspiegelung oder Always On nicht aktivieren AGs , wenn große Datenmengen geladen werden. Falls der Datenladevorgang so schnell wie möglich ablaufen soll, schließen Sie den Datenladevorgang ab, bevor Sie Ihre DB-Instance in eine Multi-AZ-Bereitstellung konvertieren.
- Anwendungen, die SQL Server-Datenbanken aufrufen, sollten über eine Ausnahmebehandlung verfügen, die Verbindungsfehler erfasst. Das folgende Codebeispiel zeigt einen try/catch Block, der einen Kommunikationsfehler abfängt. In diesem Beispiel beendet die Anweisung `break` die `while`-Schleife, wenn die Verbindung erfolgreich ist, versucht es jedoch bis zu zehnmal neu, wenn eine Ausnahme ausgelöst wird.

```
int RetryMaxAttempts = 10;
int RetryIntervalPeriodInSeconds = 1;
int iRetryCount = 0;
while (iRetryCount < RetryMaxAttempts)
{
   using (SqlConnection connection = new SqlConnection(DatabaseConnString))
   {
      using (SqlCommand command = connection.CreateCommand())
      {
         command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue');";
         try
         {
            connection.Open();
            command.ExecuteNonQuery();
            break;
         }
         catch (Exception ex)
         {
            Logger(ex.Message);
            iRetryCount++;
         }
         finally {
            connection.Close();
         }
      }
   }
   Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
```

```
}
```

- Verwenden Sie den `Set Partner Off`-Befehl nicht, wenn Sie mit Multi-AZ-Instances arbeiten. Unterlassen Sie beispielsweise Folgendes:

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- Setzen Sie den Wiederherstellungsmodus nicht auf `simple`. Unterlassen Sie beispielsweise Folgendes:

```
--Don't do this
ALTER DATABASE db1 SET RECOVERY simple
```

- Verwenden Sie den `DEFAULT_DATABASE`-Parameter nicht, wenn Sie neue Logins für Multi-AZ-DB-Instances erstellen, da diese Einstellungen nicht in die Standby-Spiegelung übernommen werden können. Unterlassen Sie beispielsweise Folgendes:

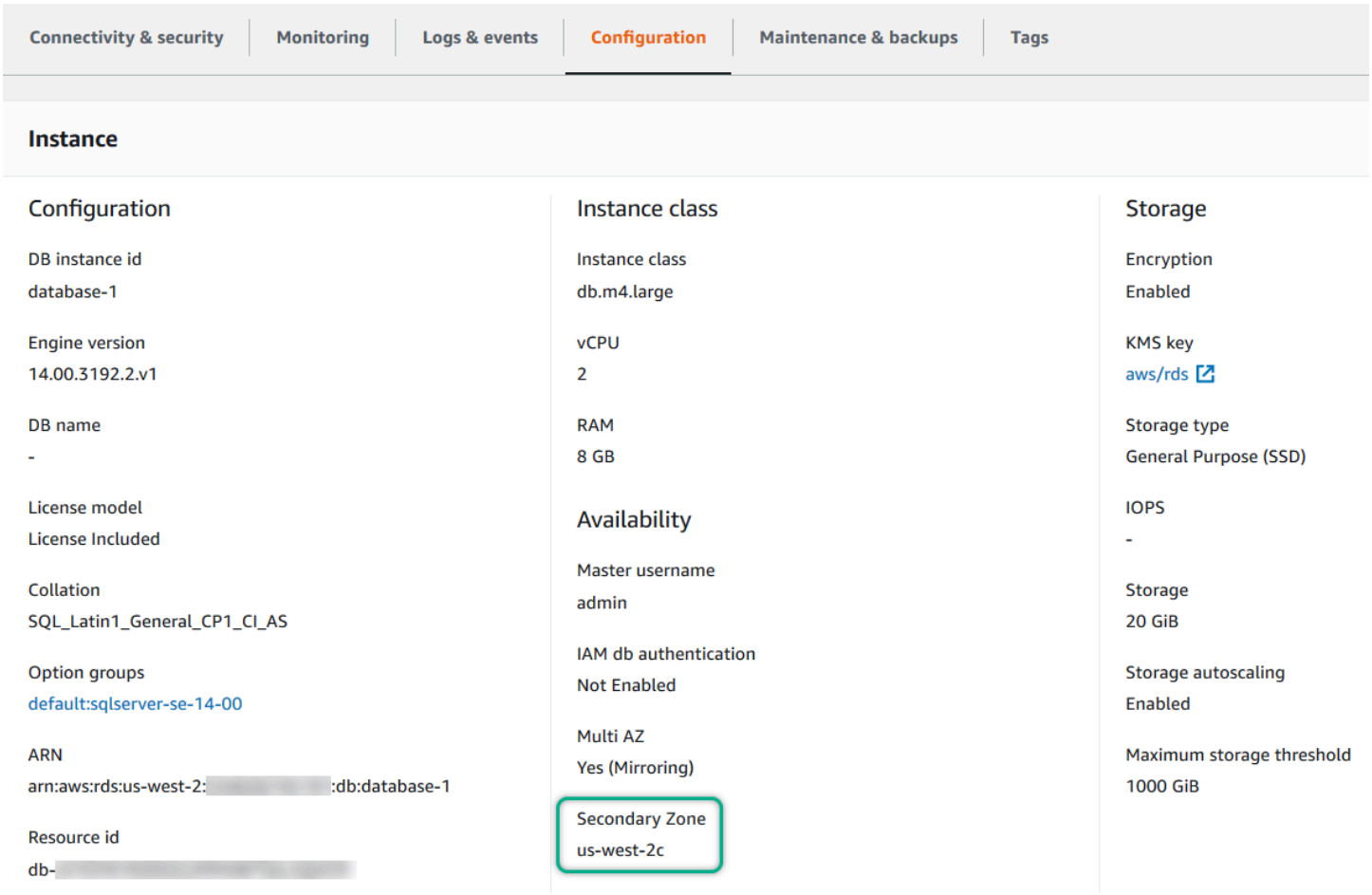
```
--Don't do this
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

  Unterlassen Sie zudem Folgendes:

```
--Don't do this
ALTER LOGIN [test_dba] WITH DEFAULT_DATABASE=[db3]
```

Festlegen des Standorts der sekundären Instance

Sie können den Standort des sekundären Replica mithilfe der festlege AWS Management Console. Sie müssen den Standort der sekundären Instanz kennen, wenn Sie Ihre primäre DB-Instance in einer einrichtenVPC.



Sie können die Availability Zone der sekundären auch mithilfe des AWS CLI Befehls `describe-db-instances` oder der RDS API Operation anzeigen`DescribeDBInstances`. Die Ausgabe zeigt die sekundäre AZ-Instance, in der sich der Standby-Spiegel befindet.

Migrieren von der Datenbankspiegelung zu AlwaysOn-Verfügbarkeitsgruppen

In Version 14.00.3049.1 von Microsoft SQL Server Enterprise Edition sind AlwaysOn-Verfügbarkeitsgruppen (AGs) standardmäßig aktiviert.

Um von Database Mirroring (DBM) zu migrieren, überprüfen Sie zunächst Ihre VersionAGs. Wenn Sie eine DB-Instance mit einer Version vor Enterprise Edition 13.00.5216.0 verwenden, patchen Sie die Instance zu Version 13.00.5216.0 oder höher. Wenn Sie eine DB-Instance mit einer Version vor Enterprise Edition 14.00.3049.1 verwenden, patchen Sie die Instance zu Version 14.00.3049.1 oder höher.

Wenn Sie eine zu verwendende gespiegelte DB-Instance aktualisieren möchtenAGs, führen Sie zuerst das Upgrade aus, ändern Sie die Instance, um Multi-AZ zu entfernen, und ändern Sie sie dann erneut, um Multi-AZ hinzuzufügen. Dadurch wird Ihre Instance auf Always On umgestellt. AGs

Zusätzliche Funktionen für Microsoft SQL Server auf Amazon RDS

In den folgenden Abschnitten finden Sie Informationen zur Erweiterung von RDS Amazon-Instances, auf denen die Microsoft SQL Server-DB-Engine ausgeführt wird.

Themen

- Verwenden der Kennwortrichtlinie für SQL Server-Logins auf RDS for SQL Server
- Integration einer Amazon RDS for SQL Server-DB-Instance mit Amazon S3
- Datenbank-Mail auf Amazon RDS for SQL Server verwenden
- Instance-Speicher-Support für die tempdb-Datenbank in Amazon RDS for SQL Server
- Verwenden erweiterter Ereignisse mit Amazon RDS für Microsoft SQL Server
- Zugriff auf Transaktionsprotokoll-Backups mit RDS für SQL Server

Verwenden der Kennwortrichtlinie für SQL Server-Logins auf RDS for SQL Server

Mit Amazon RDS können Sie die Passwortrichtlinie für Ihre Amazon RDS-DB-Instance festlegen, auf der Microsoft SQL Server ausgeführt wird. Verwenden Sie diese Option, um die Komplexität, Länge und Sperranforderungen für Anmeldungen festzulegen, die die SQL Server-Authentifizierung zur Authentifizierung bei Ihrer DB-Instance verwenden.

Wichtige Begriffe

Anmeldung

In SQL Server wird ein Prinzipal auf Serverebene, der sich bei einer Datenbankinstanz authentifizieren kann, als Anmeldung bezeichnet. Andere Datenbank-Engines bezeichnen diesen Prinzipal möglicherweise als Benutzer. In RDS für SQL Server kann eine Anmeldung mithilfe der SQL Server-Authentifizierung oder der Windows-Authentifizierung authentifiziert werden.

SQL Server-Anmeldung

Eine Anmeldung, die einen Benutzernamen und ein Passwort zur Authentifizierung mithilfe der SQL Server-Authentifizierung verwendet, ist eine SQL Server-Anmeldung. Die Kennwortrichtlinie, die Sie über DB-Parameter konfigurieren, gilt nur für SQL Server-Logins.

Windows-Anmeldung

Eine Anmeldung, die auf einem Windows-Prinzipal basiert und sich mithilfe der Windows-Authentifizierung authentifiziert, ist eine Windows-Anmeldung. Sie können die Kennwortrichtlinie für Ihre Windows-Anmeldungen in Active Directory konfigurieren. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.

Richtlinie für jede Anmeldung aktivieren und deaktivieren

Jede SQL Server-Anmeldung hat Flags für `CHECK_POLICY` und`CHECK_EXPIRATION`. Standardmäßig werden neue Logins mit `CHECK_POLICY` set to `ON` und `CHECK_EXPIRATION` set to `OFF` erstellt.

Wenn `CHECK_POLICY` es für eine Anmeldung aktiviert ist, überprüft RDS for SQL Server das Passwort anhand der Anforderungen an Komplexität und Mindestlänge. Es gelten auch Sperrrichtlinien. Ein Beispiel für eine T-SQL-Anweisung zum Aktivieren `CHECK_POLICY` und: `CHECK_EXPIRATION`

```
ALTER LOGIN [master_user] WITH CHECK_POLICY = ON, CHECK_EXPIRATION = ON;
```

Wenn diese Option aktiviert `CHECK_EXPIRATION` ist, unterliegen Passwörter den Richtlinien für das Alter von Passwörtern. Die T-SQL-Anweisung, mit der überprüft werden soll, ob `CHECK_POLICY` und gesetzt `CHECK_EXPIRATION` sind:

```
SELECT name, is_policy_checked, is_expiration_checked FROM sys.sql_logins;
```

Parameter für die Passwortrichtlinie

Alle Kennwortrichtlinienparameter sind dynamisch und erfordern keinen DB-Neustart, um wirksam zu werden. In der folgenden Tabelle sind die DB-Parameter aufgeführt, die Sie festlegen können, um die Kennwortrichtlinie für SQL Server-Logins zu ändern:

DB-Parameter	Beschreibung	Zulässige Werte	Standardwert	
rds.password_complexity_enabled	Beim Erstellen oder Ändern von Kennwörtern für SQL Server-Logins müssen die Anforderungen an die Kennwortkomplexität erfüllt sein. Die folgenden Einschränkungen müssen erfüllt sein: • Das Passwort muss Zeichen aus drei der folgenden Kategorien enthalten:	0,1	0	

DB-Parameter	Beschreibung	Zulässige Werte	Standardwert	
	• Lateinischer Kleinbuchstabe (A bis Z) • Lateinischer Großbuchstabe (A bis Z) • Nicht-alphanumerische Zeichen wie: Ausrufezeichen (!) , Dollarzeichen ($), Zahlenzeichen (#) oder Prozent (%). • Das Passwort enthält nicht den Kontonamen des Benutzers.			
rds.password_min_length	Die Mindestanzahl an Zeichen, die ein Kennwort für eine SQL Server-Anmeldung benötigt.	0-14	0	

DB-Parameter	Beschreibung	Zulässige Werte	Standardwert	
rds.password_min_age	Die Mindestanzahl von Tagen, für die ein SQL Server-Anmeldekennwort verwendet werden muss, bevor der Benutzer es ändern kann. Passwörter können sofort geändert werden, wenn sie auf 0 gesetzt sind.	0-998	0	
rds.password_max_age	Die maximale Anzahl von Tagen, für die ein SQL Server-Anmeldekennwort verwendet werden kann. Danach muss der Benutzer es ändern. Passwörter laufen nie ab, wenn sie auf 0 gesetzt sind.	0-999	42	

DB-Parameter	Beschreibung	Zulässige Werte	Standardwert	
rds.password_lockout_threshold	Die Anzahl aufeinander erfolgender fehlgeschlagener Anmeldeversuche, die dazu führen, dass eine SQL Server-Anmeldung gesperrt wird.	0-999	0	
rds.password_lockout_duration	Die Anzahl der Minuten, die ein gesperrter SQL Server-Anmeldename warten muss, bevor er entsperrt wird.	1—60	10	
rds.password_lockout_reset_counter_after	Die Anzahl der Minuten, die nach einem fehlgeschlagenen Anmeldeversuch vergehen müssen, bis der Zähler für fehlgeschlagene Anmeldeversuche auf 0 zurückgesetzt wird.	1—60	10	

> **Note**
>
> Weitere Informationen zur SQL Server-Passwortrichtlinie finden Sie unter Kennwortrichtlinie. Die Richtlinien für Kennwortkomplexität und Mindestlänge gelten auch für DB-Benutzer in geschlossenen Datenbanken. Weitere Informationen finden Sie unter Enthaltene Datenbanken.

Die folgenden Einschränkungen gelten für die Kennwortrichtlinienparameter:

- Der `rds.password_min_age` Parameter muss kleiner als sein`rds.password_max_age parameter`, sofern er nicht auf 0 gesetzt `rds.password_max_age` ist
- Der `rds.password_lockout_reset_counter_after` Parameter muss kleiner oder gleich dem `rds.password_lockout_duration` Parameter sein.
- Es `rds.password_lockout_threshold` ist auf 0 gesetzt `rds.password_lockout_duration` und gilt `rds.password_lockout_reset_counter_after` nicht.

Überlegungen zu bestehenden Logins

Nach der Änderung der Kennwortrichtlinie für eine Instanz werden bestehende Kennwörter für Anmeldungen nicht rückwirkend anhand der neuen Anforderungen an Komplexität und Länge von Passwörtern bewertet. Nur neue Passwörter werden anhand der neuen Richtlinie validiert.

SQL Server bewertet vorhandene Kennwörter auf ihre Altersanforderungen hin.

Es ist möglich, dass Kennwörter sofort ablaufen, sobald eine Kennwortrichtlinie geändert wurde. Wenn beispielsweise eine Anmeldung `CHECK_EXPIRATION` aktiviert wurde und ihr Passwort zuletzt vor 100 Tagen geändert wurde und Sie den `rds.password_max_age` Parameter auf 5 Tage setzen, läuft das Passwort sofort ab und der Anmelder muss sein Passwort beim nächsten Anmeldeversuch ändern.

> **Note**
>
> RDS für SQL Server unterstützt keine Richtlinien für den Kennwortverlauf. Verlaufsrichtlinien verhindern, dass Anmeldungen zuvor verwendete Passwörter wiederverwenden.

Überlegungen zu Multi-AZ-Bereitstellungen

Der Zähler für fehlgeschlagene Anmeldeversuche und der Sperrstatus für Multi-AZ-Instances werden nicht zwischen den Knoten repliziert. Falls eine Anmeldung beim Failover einer Multi-AZ-Instance gesperrt wird, ist es möglich, dass die Anmeldung auf dem neuen Knoten bereits entsperrt ist.

Überlegungen zum Passwort für den Master-Login

Wenn Sie eine RDS for SQL Server-DB-Instance erstellen, wird das Masterbenutzerkennwort nicht anhand der Kennwortrichtlinie bewertet. Ein neues Master-Passwort wird auch nicht anhand des Kennworts ausgewertet, wenn Operationen für den Masterbenutzer ausgeführt werden, insbesondere `MasterUserPassword` bei der Eingabe des `ModifyDBInstance` Befehls. In beiden Fällen können Sie ein Passwort für den Hauptbenutzer festlegen, das nicht Ihren Kennwortrichtlinien entspricht, und der Vorgang ist trotzdem erfolgreich. Wenn die Richtlinie nicht erfüllt ist, versucht RDS, ein RDS-Ereignis auszulösen, mit der Empfehlung, ein sicheres Passwort festzulegen. Achten Sie darauf, nur sichere Passwörter für den Masterbenutzer zu verwenden.

RDS versucht, die folgenden Ereignismeldungen zu generieren, wenn das Masterbenutzerkennwort nicht den Anforderungen der Kennwortrichtlinie entspricht:

- Der Masterbenutzer wurde erstellt, aber das Passwort entspricht nicht der Mindestlänge Ihrer Passwortrichtlinie. Erwägen Sie die Verwendung eines stärkeren Passworts.
- Der Hauptbenutzer wurde erstellt, aber das Passwort entspricht nicht den Komplexitätsanforderungen Ihrer Passwortrichtlinie. Erwägen Sie die Verwendung eines stärkeren Passworts.
- Das Masterbenutzerkennwort wurde zurückgesetzt, aber das Passwort entspricht nicht der Mindestlänge, die in Ihrer Passwortrichtlinie festgelegt ist. Erwägen Sie die Verwendung eines stärkeren Passworts.
- Das Masterbenutzerkennwort wurde zurückgesetzt, aber das Passwort entspricht nicht den Komplexitätsanforderungen Ihrer Passwortrichtlinie. Erwägen Sie die Verwendung eines stärkeren Passworts.

Standardmäßig wird der Masterbenutzer mit erstellt `CHECK_POLICY` und auf `CHECK_EXPIRATION` eingestellt`OFF`. Um die Passwortrichtlinie auf den Masterbenutzer anzuwenden, müssen Sie diese Flags nach der Erstellung der DB-Instance manuell für den Masterbenutzer aktivieren. Nachdem Sie diese Flags aktiviert haben, ändern Sie das Masterbenutzerkennwort direkt in SQL Server (z. B. über T-SQL-Anweisungen oder SSMS), um das neue Passwort anhand der Kennwortrichtlinie zu überprüfen.

> **Note**
>
> Wenn der Hauptbenutzer gesperrt wird, können Sie den Benutzer entsperren, indem Sie das Masterbenutzerpasswort mithilfe des Befehls zurücksetzen. `ModifyDBInstance`

Ändern des Master-Benutzerpassworts

Sie können das Masterbenutzerkennwort mit dem Befehl ModifyDBInstance ändern.

> **Note**
>
> Wenn Sie das Masterbenutzerkennwort zurücksetzen, setzt RDS verschiedene Berechtigungen für den Masterbenutzer zurück, und der Masterbenutzer verliert möglicherweise bestimmte Berechtigungen. Durch das Zurücksetzen des Masterbenutzerpassworts wird auch der Masterbenutzer entsperrt, falls dieser gesperrt war.

RDS validiert das neue Masterbenutzerkennwort und versucht, ein RDS-Ereignis auszulösen, wenn das Passwort nicht der Richtlinie entspricht. RDS legt das Passwort fest, auch wenn es die Passwortrichtlinie nicht erfüllt.

Integration einer Amazon RDS for SQL Server-DB-Instance mit Amazon S3

Sie können Dateien zwischen einer DB-Instance mit Amazon RDS for SQL Server und einem Amazon S3-Bucket übertragen. Dabei können Sie Amazon S3 mit SQL Server-Funktionen wie etwa BULK INSERT verwenden. Sie können beispielsweise .csv-, .xml-, .txt- und andere Dateien aus Amazon S3 zum Host der DB-Instance herunterladen und die Daten aus `D:\S3\` in die Datenbank laden. Alle Dateien werden in `D:\S3\` auf der DB-Instance gespeichert.

Die folgenden Einschränkungen gelten:

> **Note**
>
> Der Datenverkehr zwischen dem RDS-Host und S3 wird über VPC-Endpunkte in RDS intern VPCs für alle SQL Server-Funktionen geleitet, die S3 verwenden. Dieser Datenverkehr verwendet nicht den RDS-Instanzendpunkt ENI. S3-Bucket-Richtlinien können den RDS-Verkehr nicht durch Netzwerkbedingungen einschränken.

- Dateien im `D:\S3`-Ordner werden nach einem Failover auf Multi-AZ-Instances auf dem Standby-Replikat gelöscht. Weitere Informationen finden Sie unter Multi-AZ-Einschränkungen für die S3-Integration.
- Die DB-Instance und der S3-Bucket müssen sich in derselben AWS Region befinden.
- Wenn Sie mehr als eine S3-Integrationsaufgabe gleichzeitig ausführen, werden die Aufgaben sequenziell und nicht parallel ausgeführt.

  > **Note**
  >
  > S3-Integrationsaufgaben verwenden dieselbe Warteschlange wie native Backup- und Wiederherstellungsaufgaben. Sie können maximal zwei Aufgaben gleichzeitig in dieser Warteschlange haben. Daher blockieren zwei laufende native Backup- und Wiederherstellungsaufgaben alle S3-Integrationsaufgaben.

- Sie müssen die S3-Integrationsfunktion auf den wiederhergestellten Instances erneut aktivieren. Die S3-Integration wird nicht von der Quell-Instance auf die wiederhergestellte Instance übertragen. Dateien in `D:\S3` werden auf einer wiederhergestellten Instance gelöscht.
- Der Download auf die DB-Instance ist auf 100 Dateien begrenzt. Anders ausgedrückt: Es können sich nicht mehr als 100 Dateien in befinde `D:\S3\`.

- Nur Dateien ohne Dateierweiterungen oder mit den folgenden Dateierweiterungen werden zum Download
  unterstützt: .abf, .asdatabase, .bcp, .configsettings, .csv, .dat, .deploymentoptions, .deploymenttargets, .fm und .xmla.
- Der S3-Bucket muss denselben Besitzer haben wie die zugehörige Rolle AWS Identity and Access Management (IAM). Daher wird die kontoübergreifende S3-Integration nicht unterstützt.
- Der S3-Bucket kann nicht für die Öffentlichkeit zugänglich sein.
- Die Dateigröße für Uploads von RDS auf S3 ist auf 50 GB pro Datei begrenzt.
- Die Dateigröße für Downloads von S3 auf RDS ist auf das von S3 unterstützte Maximum beschränkt.

Themen

- Voraussetzungen für die Integration von RDS-for-SQL-Server mit S3
- Aktivierung RDS der SQL Serverintegration mit S3
- Dateien zwischen RDS for SQL Server und Amazon S3 übertragen
- Dateien auf der RDS DB-Instance auflisten
- Löschen von Dateien auf der RDS DB-Instance
- Überwachung des Status einer Dateiübertragungsaufgabe
- Abbrechen einer Aufgabe
- Multi-AZ-Einschränkungen für die S3-Integration
- Deaktivierung RDS für die SQL Serverintegration mit S3

Weitere Informationen zum Arbeiten mit Dateien in Amazon S3 finden Sie unter Erste Schritte mit Amazon Simple Storage Service.

Voraussetzungen für die Integration von RDS-for-SQL-Server mit S3

Bevor Sie beginnen, suchen oder erstellen Sie den S3-Bucket, den Sie verwenden möchten. Fügen Sie weiterhin Berechtigungen hinzu, so dass die RDS DB-Instance auf den S3-Bucket zugreifen kann. Zur Konfiguration dieses Zugriffs erstellen Sie eine IAM-Richtlinie und eine IAM-Rolle.

Konsole

So erstellen Sie eine IAM-Richtlinie für den Zugriff auf Amazon S3:

1. Wählen Sie in der IAM Management-Konsole im Navigationsbereich Policies (Richtlinien).
2. Erstellen Sie eine neue Richtlinie, und verwenden Sie die Registerkarte Visual editor für die folgenden Schritte.
3. Geben Sie unter Service **`S3`** ein, und wählen Sie dann den S3-Service.
4. Wählen Sie unter Actions (Aktionen) Folgendes, um den Zugang zu gewähren, den Ihre DB-Instance benötigt.

   - `ListAllMyBuckets` – erforderlich
   - `ListBucket` – erforderlich
   - `GetBucketAcl` – erforderlich
   - `GetBucketLocation` – erforderlich
   - `GetObject` – erforderlich für das Herunterladen von Dateien von S3 zu `D:\S3\`
   - `PutObject` – erforderlich für das Hochladen von Dateien von `D:\S3\` zu S3
   - `ListMultipartUploadParts` – erforderlich für das Hochladen von Dateien von `D:\S3\` zu S3
   - `AbortMultipartUpload` – erforderlich für das Hochladen von Dateien von `D:\S3\` zu S3
5. Für Resources (Ressourcen) hängen die angezeigten Optionen davon ab, welche Aktionen Sie im vorherigen Schritt ausgewählt haben. Möglicherweise sehen Sie Optionen für Bucket, Object (Objekt) oder beide. Geben Sie für beide den jeweiligen Amazon-Ressourcennamen (ARN) an.

   Geben Sie für Bucket den ARN für den Bucket an, den Sie verwenden möchten. Wenn Ihr Bucket beispielsweise den Namen *`amzn-s3-demo-bucket`* hat, setzen Sie den ARN auf `arn:aws:s3:::`*`amzn-s3-demo-bucket`*.

   Geben Sie für Object (Objekt) den ARN für den Bucket ein, und wählen Sie dann eine der folgenden Optionen:

   - Um Zugriff auf alle Dateien in dem angegebenen Bucket zu gewähren, wählen Sie Any (Beliebig) für Bucket name (Name des Buckets) und Object name (Name des Objekts).
   - Um Zugriff auf bestimmte Dateien oder Ordner im Bucket zu gewähren, geben ARNs Sie die spezifischen Buckets und Objekte an, auf die SQL Server zugreifen soll.
6. Befolgen Sie die Anweisungen in der Konsole, bis Sie die Richtlinie fertig erstellt haben.

Vorstehendes ist eine verkürzte Anleitung zur Einrichtung einer Richtlinie. Ausführlichere Anweisungen zum Erstellen von IAM-Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

So erstellen Sie eine IAM-Rolle, die die IAM-Richtlinie aus der vorherigen Prozedur verwendet:

1. Wählen Sie in der IAM Management-Konsole im Navigationsbereich Roles (Rollen).
2. Erstellen Sie eine neue IAM-Rolle, und wählen Sie die folgenden Optionen, wenn sie in der Konsole angezeigt werden:

   - AWS Service nicht zulässig
   - RDS
   - RDS – Add Role to Database (Rolle zu Datenbank hinzufügen)

   Wählen Sie dann unten Next permissions (Nächste Berechtigungen).
3. Geben Sie für Attach permissions policies (Berechtigungsrichtlinien anfügen) den Namen der vorher erstellten IAM-Richtlinie ein. Wählen Sie dann eine Richtlinie aus der Liste aus.
4. Befolgen Sie die Anweisungen in der Konsole, bis Sie die Rolle fertig erstellt haben.

   Vorstehendes ist eine verkürzte Anleitung zur Einrichtung einer Rolle. Wenn Sie eine ausführlichere Anleitung zum Erstellen einer Rolle wünschen, vgl. IAM-Rollen im IAM-Benutzerhandbuch.

AWS CLI

Sie können Amazon RDS wie folgt Zugriff auf einen Amazon-S3-Bucket gewähren:

1. Erstellen Sie eine IAM-Richtlinie, die Amazon RDS Zugriff auf einen S3-Bucket gewährt.
2. Erstellen Sie eine IAM-Rolle, die Amazon RDS in Ihrem Auftrag annehmen kann, um auf Ihre S3-Buckets zuzugreifen.

   Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.
3. Fügen Sie die erstellte IAM-Richtlinie der IAM-Rolle an, die Sie erstellt haben.

So erstellen Sie die -IAM-Richtlinie

Schließen Sie die jeweiligen Aktionen ein, um den Zugriff zu gewähren, den Ihre DB-Instance erfordert:

- `ListAllMyBuckets` – erforderlich
- `ListBucket` – erforderlich
- `GetBucketAcl` – erforderlich
- `GetBucketLocation` – erforderlich
- `GetObject` – erforderlich für das Herunterladen von Dateien von S3 zu `D:\S3\`
- `PutObject` – erforderlich für das Hochladen von Dateien von `D:\S3\` zu S3
- `ListMultipartUploadParts` – erforderlich für das Hochladen von Dateien von `D:\S3\` zu S3
- `AbortMultipartUpload` – erforderlich für das Hochladen von Dateien von `D:\S3\` zu S3

1. Mit dem folgenden AWS CLI Befehl wird eine IAM-Richtlinie `rds-s3-integration-policy` mit diesen Optionen erstellt. Sie gewährt Zugriff auf einen Bucket namens *`amzn-s3-demo-bucket`*.

   Example

   Für LinuxmacOS, oderUnix:

```
aws iam create-policy \
  --policy-name rds-s3-integration-policy \
  --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "s3:ListAllMyBuckets",
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
```

```
                "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:GetObject",
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/key_prefix/*"
            }
        ]
    }'
```

Windows:

Achten Sie darauf, dass Sie die Zeilenenden so ändern, dass Ihre Schnittstelle sie unterstützt (^ anstelle von \). Unter Windows müssen Sie dazu alle doppelten Anführungszeichen mit dem Escape-Zeichen versehe \. Um dies für alle Anführungszeichen in JSON zu vermeiden, können Sie sie stattdessen in eine Datei speichern und als Parameter einführen.

Erstellen Sie zuerst die `policy.json`-Datei mit der folgenden Berechtigungsrichtlinie:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/key_prefix/*"
        }
    ]
}
```

Verwenden Sie dann den folgenden Befehl, um die Richtlinie zu erstellen:

```
aws iam create-policy ^
     --policy-name rds-s3-integration-policy ^
     --policy-document file://file_path/assume_role_policy.json
```

2. Notieren Sie nach dem Erstellen der Richtlinie den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen den ARN bei einem späteren Schritt.

So erstellen Sie die IAM-Rolle

- Der folgende AWS CLI Befehl erstellt die `rds-s3-integration-role` IAM-Rolle für diesen Zweck.

  Example

  Für LinuxmacOS, oderUnix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
```

```
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

Windows:

Achten Sie darauf, dass Sie die Zeilenenden so ändern, dass Ihre Schnittstelle sie unterstützt (^ anstelle von \). Unter Windows müssen Sie dazu alle doppelten Anführungszeichen mit dem Escape-Zeichen versehe \. Um dies für alle Anführungszeichen in JSON zu vermeiden, können Sie sie stattdessen in eine Datei speichern und als Parameter einführen.

Erstellen Sie zuerst die `assume_role_policy.json`-Datei mit der folgenden Richtlinie:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Verwenden Sie anschließend den folgenden Befehl zum Erstellen der IAM-Rolle:

```
aws iam create-role ^
     --role-name rds-s3-integration-role ^
     --assume-role-policy-document file://file_path/assume_role_policy.json
```

Example die Verwendung des globalen Bedingungskontextschlüssels zum Erstellen der IAM-Rolle

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Sie können beide globalen Bedingungskontextschlüssel verwenden und der Wert `aws:SourceArn` enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der Wert `aws:SourceAccount` und das Konto im Wert `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Stellen Sie in der Richtlinie sicher, dass Sie den globalen Bedingungskontextschlüssel `aws:SourceArn` mit dem vollständigen Amazon-Ressourcennamen (ARN) der Ressourcen verwenden, die auf die Rolle zugreifen. Stellen Sie bei der S3-Integration sicher, dass Sie die DB-Instance einbeziehen ARNs, wie im folgenden Beispiel gezeigt.

Für LinuxmacOS, oderUnix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
```

```
"aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    }
                }
        }
      ]
   }'
```

Windows:

Fügen Sie den globalen Bedingungskontextschlüssel zu `assume_role_policy.json` hinzu.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                }
            }
        }
    ]
}
```

IAM-Richtlinie an die IAM-Rolle anfügen

- Mit dem folgenden AWS CLI Befehl wird die Richtlinie an die angegebene `rds-s3-integration-role` Rolle angehängt. Ersetzen Sie *`your-policy-arn`* durch den Richtlinien-ARN, den Sie im vorherigen Schritt notiert haben.

Example

Für LinuxmacOS, oderUnix:

```
aws iam attach-role-policy \
    --policy-arn your-policy-arn \
    --role-name rds-s3-integration-role
```

Windows:

```
aws iam attach-role-policy ^
    --policy-arn your-policy-arn ^
    --role-name rds-s3-integration-role
```

Aktivierung RDS der SQL Serverintegration mit S3

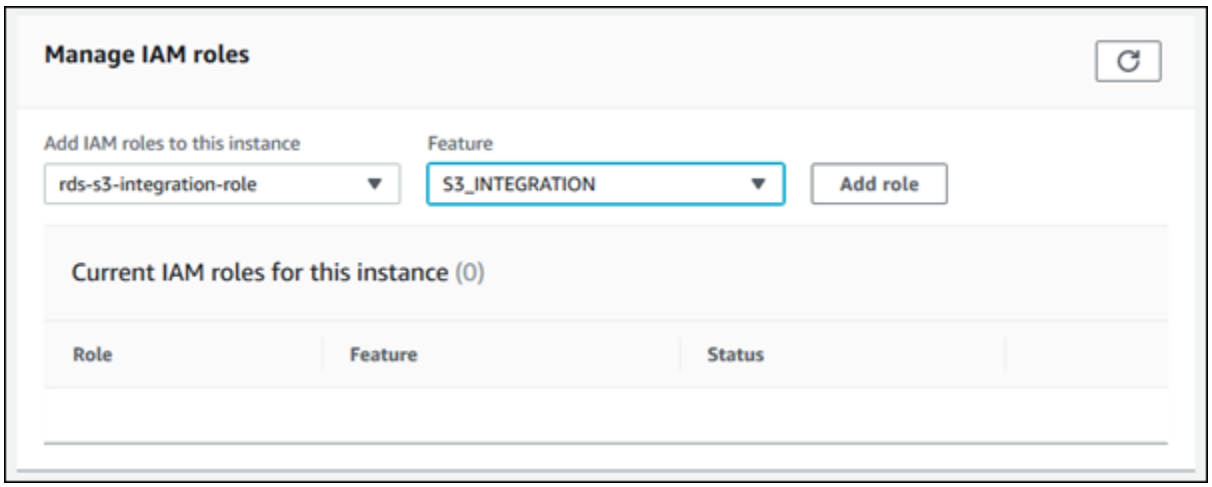
Im folgenden Abschnitt erfahren Sie, wie Sie die Amazon S3 S3-Integration mit Amazon RDS for SQL Server aktivieren. Um mit der S3-Integration arbeiten zu können, muss Ihre DB-Instance der IAM Rolle zugeordnet sein, die Sie zuvor erstellt haben, bevor Sie den `S3_INTEGRATION` Feature-Name-Parameter verwenden können.

> **Note**
>
> Um einer DB-Instance eine IAM Rolle hinzuzufügen, muss der Status der DB-Instance verfügbar sein.

Konsole

Um Ihre IAM Rolle mit Ihrer DB-Instance zu verknüpfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie RDS den Namen der SQL Server-DB-Instance, um deren Details anzuzeigen.
3. Wählen Sie auf der Registerkarte Konnektivität und Sicherheit im Abschnitt IAMRollen verwalten die IAM Rolle aus, die Sie für IAMRollen zu dieser Instance hinzufügen möchten.
4. Wählen Sie für Feature die Option S3_ INTEGRATION aus.



5. Wählen Sie Rolle hinzufügen.

AWS CLI

Um die IAM Rolle zur RDS for SQL Server-DB-Instance hinzuzufügen

- Der folgende AWS CLI Befehl fügt Ihre IAM Rolle einer RDS for SQL Server-DB-Instance mit dem Namen hinzu*mydbinstance*.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds add-role-to-db-instance \
      --db-instance-identifier mydbinstance \
      --feature-name S3_INTEGRATION \
      --role-arn your-role-arn
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds add-role-to-db-instance ^
      --db-instance-identifier mydbinstance ^
      --feature-name S3_INTEGRATION ^
      --role-arn your-role-arn
  ```

  *your-role-arn*Ersetzen Sie sie durch die RolleARN, die Sie in einem vorherigen Schritt notiert haben. `S3_INTEGRATION`muss für die `--feature-name` Option angegeben werden.

Dateien zwischen RDS for SQL Server und Amazon S3 übertragen

Sie können RDS gespeicherte Amazon-Prozeduren verwenden, um Dateien zwischen Amazon S3 und Ihrer RDS DB-Instance herunterzuladen und hochzuladen. Sie können auch RDS gespeicherte Amazon-Prozeduren verwenden, um Dateien auf der RDS Instance aufzulisten und zu löschen.

Die Dateien, die Sie von S3 herunter- und hochladen, werden im Ordner `D:\S3` gespeichert. Dies ist der einzige Ordner, den Sie zum Zugriff auf Ihre Dateien verwenden können. Sie können Ihre Dateien in Unterordnern organisieren, die für Sie erstellt werden, wenn Sie beim Download den Zielordner einschließen.

Für einige der gespeicherten Prozeduren müssen Sie einen Amazon-Ressourcennamen (ARN) für Ihren S3-Bucket und Ihre S3-Datei angeben. Das Format für Ihr ARN ist`arn:aws:s3:::`*`amzn-s3-demo-bucket`*`/file_name`. Amazon S3 benötigt keine Kontonummer oder AWS Region inARNs.

S3-Integrationsaufgaben werden sequenziell ausgeführt und nutzen dieselbe Warteschlange wie native Sicherungs- und Wiederherstellungsaufgaben. Sie können maximal zwei Aufgaben gleichzeitig in dieser Warteschlange haben. Der Verarbeitungsbeginn für jede Aufgabe kann bis zu fünf Minuten in Anspruch nehmen.

Dateien von einem Amazon S3 S3-Bucket auf eine SQL Server-DB-Instance herunterladen

Verwenden Sie die RDS gespeicherte Amazon-Prozedur `msdb.dbo.rds_download_from_s3` mit den folgenden Parametern, um Dateien von einem S3-Bucket auf eine RDS for SQL Server-DB-Instance herunterzuladen.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`@s3_arn_of_file`	NVARCHAR	–	Erforderlich	Das S3 ARN der herunterzuladenden Datei, zum Beispiel: `arn:aws:s3:::` *`amzn-s3-demo-bucket`* `/mydata.csv`
`@rds_file_path`	NVARCHAR	–	Optional	Der Dateipfad für die RDS Instanz. Wenn nichts angegeben ist, ist der Dateipfad `D:`

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				\S3*<filename in s3>*. RDSunterstützt absolute Pfade und relative Pfade. Wenn Sie einen Unterordner erstellen möchten, schließen Sie ihn in den Dateipfad ein.
`@overwrite_file`	INT	0	Optional	Die vorhandene Datei überschreiben: 0 Nicht überschreiben 1 = Überschreiben

Sie können Dateien ohne Dateierweiterung und Dateien mit den folgenden Dateierweiterungen herunterladen: .bcp, .csv, .dat, .fmt, .info, .lst, .tbl, .txt und .xml.

> **Note**
>
> Dateien mit der Dateierweiterung.ispac werden zum Herunterladen unterstützt, wenn SQL Server Integration Services aktiviert ist. Weitere Informationen zur Aktivierung SSIS finden Sie unter. SQL Server Integration Services
> Dateien mit den folgenden Dateierweiterungen werden zum Herunterladen unterstützt, wenn SQL Server Analysis Services aktiviert ist: .abf, .asdatabase, .configsettings, .deploymentoptions, .deploymenttargets und .xmla. SSASWeitere SQLServeranalyse-Services Informationen zur Aktivierung finden Sie unter.

Das folgende Beispiel zeigt die gespeicherte Prozedur für den Download von Dateien von S3.

```
exec msdb.dbo.rds_download_from_s3
     @s3_arn_of_file='arn:aws:s3:::amzn-s3-demo-bucket/bulk_data.csv',
     @rds_file_path='D:\S3\seed_data\data.csv',
     @overwrite_file=1;
```

Die Beispieloperation `rds_download_from_s3` erstellt einen Ordner mit der Bezeichnung `seed_data` in `D:\S3\`, wenn der Ordner noch nicht vorhanden ist. Dann lädt das Beispiel die Quelldatei `bulk_data.csv` von S3 zu einer neuen Datei mit der Bezeichnung `data.csv` auf der DB-Instance herunter. Wenn die Datei bereits vorhanden war, wird sie überschrieben, da der Parameter `@overwrite_file` auf 1 gesetzt ist.

Dateien von einer SQL Server-DB-Instance in einen Amazon S3 S3-Bucket hochladen

Verwenden Sie die RDS gespeicherte Amazon-Prozedur `msdb.dbo.rds_upload_to_s3` mit den folgenden Parametern, um Dateien von einer RDS for SQL Server-DB-Instance in einen S3-Bucket hochzuladen.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`@s3_arn_of_file`	NVARCHAR	–	Erforderlich	Das S3 ARN der Datei, die in S3 erstellt werden soll, zum Beispiel: `arn:aws:s3:::` *`amzn-s3-demo-bucket`* `/mydata.csv`
`@rds_file_path`	NVARCHAR	–	Erforderlich	Der Dateipfad der zu S3 hochzuladenden Datei. Es werden absolute und relative Pfade unterstützt.
`@overwrite_file`	INT	–	Optional	Die vorhandene Datei überschreiben: 0 Nicht überschreiben 1 = Überschreiben

Im folgenden Beispiel wird die `data.csv` vom angegebenen Speicherort benannte Datei in eine Datei `new_data.csv` im `D:\S3\seed_data\` S3-Bucket hochgeladen, der durch den ARN angegeben wird.

```
exec msdb.dbo.rds_upload_to_s3
```

```
  @rds_file_path='D:\S3\seed_data\data.csv',
  @s3_arn_of_file='arn:aws:s3:::amzn-s3-demo-bucket/new_data.csv',
  @overwrite_file=1;
```

Wenn die Datei in S3 bereits vorhanden war, wird sie überschrieben, da der Parameter @overwrite_file auf gesetzt is 1.

Dateien auf der RDS DB-Instance auflisten

Verwenden Sie zum Auflisten der auf der DB-Instance verfügbaren Dateien eine gespeicherte Prozedur und eine Funktion. Führen Sie zunächst die folgende gespeicherte Prozedur aus, um Dateidetails von den Dateien in zu erfasse `D:\S3\`.

```
exec msdb.dbo.rds_gather_file_details;
```

Die gespeicherte Prozedur gibt die ID der Aufgabe zurück. Wie andere Aufgaben wird diese gespeicherte Prozedur asynchron ausgeführt. Sobald der Status der Aufgabe `SUCCESS` ist, können Sie die Aufgaben-ID in der `rds_fn_list_file_details`-Funktion verwenden, um die vorhandenen Dateien und Verzeichnisse in D:\S3\ wie nachfolgend gezeigt aufzulisten.

```
SELECT * FROM msdb.dbo.rds_fn_list_file_details(TASK_ID);
```

Die `rds_fn_list_file_details`-Funktion gibt eine Tabelle mit den folgenden Spalten zurück:

Ausgabeparameter	Beschreibung
`filepath`	Absoluter Pfad der Datei (zum Beispie, `D:\S3\mydata.csv`)
`size_in_bytes`	Dateigröße (in Bytes)
`last_modified_utc`	Datum und Uhrzeit der letzten Änderung im UTC Format
`is_directory`	Option, die angibt, ob es sich bei dem Element um ein Verzeichnis handelt (`true`/`false`).

Löschen von Dateien auf der RDS DB-Instance

Um die auf der DB-Instance verfügbaren Dateien zu löschen, verwenden Sie die RDS gespeicherte Amazon-Prozedur `msdb.dbo.rds_delete_from_filesystem` mit den folgenden Parametern.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`@rds_file_path`	NVARCHAR	–	Erforderlich	Der Dateipfad der zu löschenden Datei. Es werden absolute und relative Pfade unterstützt.
`@force_delete`	INT	0	Optional	Zum Löschen eines Verzeichnisses muss diese Markierung eingeschlossen und auf gesetzt werde 1. 1 = Löschen eines Verzeichnisses Dieser Parameter wird ignoriert, wenn Sie eine Datei löschen.

Zum Löschen eines Verzeichnisses muss `@rds_file_path` mit einem umgekehrten Schrägstrich (\) enden, und `@force_delete` muss auf 1 gesetzt sein.

Im folgenden Beispiel wird die Datei gelösch `D:\S3\delete_me.txt`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\delete_me.txt';
```

Im folgenden Beispiel wird das Verzeichnis gelösch `D:\S3\example_folder\`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\example_folder\',
    @force_delete=1;
```

Überwachung des Status einer Dateiübertragungsaufgabe

Rufen Sie zum Nachverfolgen des Status Ihrer S3-Integrationsaufgabe die `rds_fn_task_status`-Funktion auf. Dazu sind zwei Parameter erforderlich. Der erste Parameter sollte immer `NULL` sein, da er sich nicht auf die S3-Integration bezieht. Der zweite Parameter akzeptiert eine Aufgaben-ID.

Um eine Liste aller Aufgaben anzuzeigen, setzen Sie den ersten Parameter auf `NULL` und den zweiten Parameter auf `0`, wie im folgenden Beispiel gezeigt.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Um eine bestimmte Aufgabe zu erhalten, setzen Sie den ersten Parameter auf `NULL` und den zweiten Parameter auf die Aufgaben-ID, wie im folgenden Beispiel gezeigt,

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Die `rds_fn_task_status`-Funktion gibt die folgenden Informationen zurück.

Ausgabeparameter	Beschreibung
`task_id`	Die ID der Aufgabe.
`task_type`	Für die S3-Integration können Aufgaben die folgenden Aufgabentypen haben: • `DOWNLOAD_FROM_S3` • `UPLOAD_TO_S3` • `LIST_FILES_ON_DISK` • `DELETE_FILES_ON_DISK`
`database_name`	Nicht verwendbar für S3-Integrationsaufgaben.
`% complete`	Verlauf der Aufgabe als Prozentwert.
`duration(mins)`	Zeitdauer für die Ausführung der Aufgabe (in Minuten).
`lifecycle`	Der Status der Aufgabe. Die folgenden Status sind möglich:

Ausgabeparameter	Beschreibung
	• CREATED – Nach dem Aufruf einer der gespeicherten Prozeduren für die S3-Integr ation wird eine Aufgabe erstellt, und der Status wird auf gesetzt CREATED. • IN_PROGRESS – Nach dem Start einer Aufgabe wird der Status auf gesetzt IN_PROGRESS . Es kann bis zu fünf Minuten dauern, bis sich der Status von CREATED zu IN_PROGRESS ändert. • SUCCESS – Nach dem Abschluss einer Aufgabe wird der Status auf gesetzt SUCCESS. • ERROR – Wenn eine Aufgabe fehlschlägt, wird der Status auf gesetzt ERROR. Weitere Informationen über den Fehler können Sie der Spalte task_info entnehmen. • CANCEL_REQUESTED – Sobald Sie rds_cancel_task aufrufen, wird der Status der Aufgabe auf CANCEL_RE QUESTED gesetzt. • CANCELLED – Nachdem die Aufgabe abgebrochen wurde, wird der Status der Aufgabe auf gesetzt CANCELLED .
task_info	Zusätzliche Informationen über die Aufgabe. Wenn bei der Verarbeitung ein Fehler auftritt, enthält diese Spalte Informationen zu dem Fehler.
last_updated	Datum und Uhrzeit der letzten Aktualisierung des Aufgabenstatus.
created_at	Datum und Uhrzeit, an denen die Aufgabe angelegt wurde.

Ausgabeparameter	Beschreibung
`S3_object_arn`	Das ARN des S3-Objekts, das von heruntergeladen oder hochgeladen wurde.
`overwrite_S3_backup_file`	Nicht verwendbar für S3-Integrationsaufgaben.
`KMS_master_key_arn`	Nicht verwendbar für S3-Integrationsaufgaben.
`filepath`	Der Dateipfad auf der RDS DB-Instance.
`overwrite_file`	Eine Option, die anzeigt, ob eine bestehende Datei überschrieben wird.
`task_metadata`	Nicht verwendbar für S3-Integrationsaufgaben.

Abbrechen einer Aufgabe

Verwenden Sie zum Abbrechen von S3-Integrationsaufgaben die gespeicherte `msdb.dbo.rds_cancel_task`-Prozedur mit dem Parameter `task_id`. Laufende Lösch- und Auflistungsaufgaben können nicht abgebrochen werden. Das folgende Beispiel zeigt eine Anforderung zum Abbrechen einer Aufgabe.

```
exec msdb.dbo.rds_cancel_task @task_id = 1234;
```

Um einen Überblick über alle Aufgaben und deren Aufgaben zu erhaltenIDs, verwenden Sie die `rds_fn_task_status` Funktion wie unter beschriebenÜberwachung des Status einer Dateiübertragungsaufgabe.

Multi-AZ-Einschränkungen für die S3-Integration

Bei Multi-AZ-Instances werden Dateien im `D:\S3`-Ordner nach einem Failover auf dem Standby-Replikat gelöscht. Ein Failover kann beispielsweise bei Änderungen der DB-Instance geplant werden, z. B. beim Ändern der Instance-Klasse oder beim Aktualisieren der Engine-Version. Oder ein Failover kann während eines Ausfalls der primären Instance ungeplant sein.

> **Note**
>
> Es wird nicht empfohlen, den `D:\S3`-Ordner für die Dateispeicherung zu verwenden. Die bewährte Methode besteht darin, erstellte Dateien in Amazon S3 hochzuladen, um sie beständig zu machen, und Dateien herunterzuladen, wenn Sie Daten importieren müssen.

Um die letzte Failover-Zeit zu bestimmen, können Sie das gespeicherte `msdb.dbo.rds_failover_time`-Verfahren verwenden. Weitere Informationen finden Sie unter Ermitteln der letzten Failover-Zeit für Amazon RDS for Server SQL.

Example Kein Failover in letzter Zeit

Dieses Beispiel zeigt die Ausgabe, wenn in den Fehlerprotokollen kein aktuelles Failover vorhanden ist. Seit 2020-04-29 23:59:00.01 ist kein Failover aufgetreten.

Daher sind alle nach dieser Zeit heruntergeladenen Dateien, die nicht mit dem gespeicherten `rds_delete_from_filesystem`-Verfahren gelöscht wurden, weiterhin auf dem aktuellen Host verfügbar. Dateien, die vor diesem Zeitpunkt heruntergeladen wurden, sind möglicherweise ebenfalls verfügbar.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	Null

Example Failover in letzter Zeit

Dieses Beispiel zeigt die Ausgabe, wenn ein Failover in den Fehlerprotokollen vorliegt. Das letzte Failover erfolgte am 2020-05-05 18:57:51.89.

Auf alle nach dieser Zeit heruntergeladenen Dateien, die nicht mit dem gespeicherten `rds_delete_from_filesystem`-Verfahren gelöscht wurden, kann auf dem aktuellen Host weiterhin zugegriffen werden.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

Deaktivierung RDS für die SQL Serverintegration mit S3

Im Folgenden erfahren Sie, wie Sie die Amazon S3 S3-Integration mit Amazon RDS for SQL Server deaktivieren können. Dateien in `D:\S3\` werden nicht gelöscht, wenn die S3-Integration deaktiviert wird.

> **Note**
>
> Um eine IAM Rolle aus einer DB-Instance zu entfernen, muss der Status der DB-Instance lauten`available`.

Konsole

Um Ihre IAM Rolle von Ihrer DB-Instance zu trennen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie RDS den Namen der SQL Server-DB-Instance, um deren Details anzuzeigen.
3. Wählen Sie auf der Registerkarte Konnektivität und Sicherheit im Abschnitt IAMRollen verwalten die IAM Rolle aus, die Sie entfernen möchten.
4. Wählen Sie Löschen.

AWS CLI

Um die IAM Rolle aus der RDS for SQL Server-DB-Instance zu entfernen

- Der folgende AWS CLI Befehl entfernt die IAM Rolle aus einer RDS for SQL Server-DB-Instance mit dem Namen*mydbinstance*.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds remove-role-from-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds remove-role-from-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

*your-role-arn*Ersetzen Sie es durch die entsprechende IAM Rolle ARN für die `--feature-name` Option.

Datenbank-Mail auf Amazon RDS for SQL Server verwenden

Sie können Database Mail verwenden, um E-Mail-Nachrichten von Ihrer Amazon RDS on SQL Server-Datenbank-Instance aus an Benutzer zu senden. Die Nachrichten können Dateien und Abfrageergebnisse enthalten. Database Mail enthält die folgenden Komponenten:

- Konfigurations- und Sicherheitsobjekte – Diese Objekte erstellen Profile und Konten und werden in der `msdb`-Datenbank gespeichert.
- Messaging-Objekte – Diese Objekte umfassen die gespeicherte Prozedur sp_send_dbmail, die zum Senden von Nachrichten verwendet wird, und Datenstrukturen, die Informationen über Nachrichten enthalten. Sie sind in der `msdb`-Datenbank gespeichert.
- Objekte protokollieren und prüfen – Database Mail schreibt Protokollierungsinformationen in die `msdb`-Datenbank und das Ereignisprotokoll der Microsoft Windows-Anwendung.
- Die ausführbare Datei von Database Mail – `DatabaseMail.exe` liest aus einer Warteschlange in der `msdb`-Datenbank und sendet E-Mail-Nachrichten.

RDSunterstützt Database Mail für alle SQL Serverversionen in den Web-, Standard- und Enterprise-Editionen.

Einschränkungen

Die folgenden Einschränkungen gelten für die Verwendung von Database Mail auf Ihrer SQL Server-DB-Instance:

- Database Mail wird für SQL Server Express Edition nicht unterstützt.
- Das Ändern der Database Mail-Konfigurationsparameter wird nicht unterstützt. Um die voreingestellten (Standard-)Werte zu sehen, verwenden Sie den gespeicherten Prozess sysmail_help_configure_sp.
- Dateianhänge werden nicht vollständig unterstützt. Weitere Informationen finden Sie unter Arbeiten mit Dateianlagen.
- Die maximale Größe des Dateianhangs beträgt 1 MB.
- Database Mail erfordert zusätzliche Konfiguration für Multi-AZ DB-Instances. Weitere Informationen finden Sie unter Überlegungen zu Multi-AZ-Bereitstellungen.
- Die Konfiguration des SQL Server-Agents für das Senden von E-Mail-Nachrichten an vordefinierte Operatoren wird nicht unterstützt.

Aktivieren von Database Mail

Verwenden Sie den folgenden Prozess, um Database Mail für Ihre DB-Instance zu aktivieren:

1. Neue Parametergruppe erstellen.
2. Ändern Sie die Parametergruppe, um den Parameter `database mail xps` auf „1“ einzustellen.
3. Ordnen Sie die neue Parametergruppe der DB-Instance zu.

Erstellen der Parametergruppe für Database Mail

Erstellen Sie eine Parametergruppe für den `database mail xps` Parameter, der der SQL Server-Edition und Version Ihrer DB-Instance entspricht.

> **Note**
>
> Sie können auch eine vorhandene Parametergruppe ändern. Folgen Sie dem Verfahren unter Ändern des Parameters, der Database Mail aktiviert.

Konsole

Im folgenden Beispiel wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie Create parameter group (Parametergruppe erstellen).
4. Führen Sie im Bereich Parametergruppe erstellen die folgenden Schritte aus:
   a. Wählen Sie für Parametergruppenfamilie die Option sqlserver-se-13.0 aus.
   b. Geben Sie unter Gruppenname einen Bezeichner für die Parametergruppe ein, z. B. **`dbmail-sqlserver-se-13`**.
   c. Geben Sie für Beschreibung den Text **`Database Mail XPs`** ein.
5. Wählen Sie Create (Erstellen) aus.

CLI

Im folgenden Beispiel wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Database Mail XPs"
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Database Mail XPs"
```

Ändern des Parameters, der Database Mail aktiviert

Ändern Sie den `database mail xps` Parameter in der Parametergruppe, die der SQL Server-Edition und Version Ihrer DB-Instance entspricht.

Um Database Mail zu aktivieren, setzen Sie den Parameter `database mail xps` auf 1.

Konsole

Im folgenden Beispiel wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben.

So ändern Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

3. Wählen Sie die Parametergruppe aus, z. B. dbmail-sqlserver-se-13.
4. Filtern Sie unter Parameter die Parameterliste nach **mail**.
5. Wählen Sie database mail xps aus.
6. Wählen Sie Parameter bearbeiten aus.
7. Geben Sie ei **1**.
8. Wählen Sie Änderungen speichern.

CLI

Im folgenden Beispiel wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben.

So ändern Sie die Parametergruppe

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

Die Parametergruppe der DB-Instance zuordnen

Sie können das AWS Management Console oder verwenden AWS CLI , um die Database-Mail-Parametergruppe mit der DB-Instance zu verknüpfen.

Konsole

Sie können die Database Mail-Parametergruppe einer neuen oder einer vorhandenen DB-Instance zuordnen.

- Bei einer neuen DB-Instance ordnen Sie diese zu, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Bei einer vorhandenen DB-Instance ordnen Sie diese zu, indem Sie die Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

CLI

Sie können die Database Mail-Parametergruppe einer neuen oder einer vorhandenen DB-Instance zuordnen.

So erstellen Sie eine DB-Instance mit der Parametergruppe Database Mail

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Parametergruppe verwendet haben.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li
    --db-parameter-group-name dbmail-sqlserver-se-13
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
```

```
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name dbmail-sqlserver-se-13
```

So ändern Sie eine DB-Instance und verknüpfen die Parametergruppe Database Mail

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds modify-db-instance \
      --db-instance-identifier mydbinstance \
      --db-parameter-group-name dbmail-sqlserver-se-13 \
      --apply-immediately
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds modify-db-instance ^
      --db-instance-identifier mydbinstance ^
      --db-parameter-group-name dbmail-sqlserver-se-13 ^
      --apply-immediately
  ```

Konfigurieren von Database Mail

Sie führen die folgenden Aufgaben aus, um Database Mail zu konfigurieren:

1. Erstellen Sie das Database Mail-Profil.
2. Erstellen Sie das Database Mail-Konto.
3. Fügen Sie das Database Mail-Konto dem Database Mail-Profil hinzu.
4. Fügen Sie dem Database Mail-Profil Benutzer hinzu.

> **Note**
>
> Stellen Sie zum Konfigurieren von Database Mail sicher, dass Sie die Berechtigung zum `execute` für die gespeicherten Prozesse in der `msdb`-Datenbank haben.

Erstellen des Database Mail-Profils

Um das Database Mail-Profil zu erstellen, verwenden Sie den gespeicherten Prozess sysmail_add_profile_sp. Im folgenden Beispiel wird ein Profil namens erstell `Notifications`.

So erstellen Sie das Profil

- Verwenden Sie die folgende SQL Anweisung.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name         = 'Notifications',
    @description          = 'Profile used for sending outgoing notifications using
 Amazon SES.';
GO
```

Erstellen des Database Mail-Kontos

Um das Database Mail-Konto zu erstellen, verwenden Sie den gespeicherten Prozess sysmail_add_account_sp. Im folgenden Beispiel wird mithilfe von Amazon Simple Email Service ein Konto mit dem Namen `SES` einer RDS for SQL Server-DB-Instance in einer privaten VPC Umgebung erstellt.

Für die Verwendung von Amazon SES sind die folgenden Parameter erforderlich:

- `@email_address`— Eine von Amazon SES verifizierte Identität. Weitere Informationen finden Sie unter Verifizierte Identitäten bei Amazon SES.
- `@mailserver_name`— Ein SES SMTP Amazon-Endpunkt. Weitere Informationen finden Sie unter Verbindung zu einem SES SMTP Amazon-Endpunkt herstellen.
- `@username`— Ein SES SMTP Amazon-Benutzername. Weitere Informationen finden Sie unter SESSMTPAmazon-Anmeldeinformationen abrufen.

Verwenden Sie keinen AWS Identity and Access Management Benutzernamen.

- `@password`— Ein SES SMTP Amazon-Passwort. Weitere Informationen finden Sie unter SESSMTPAmazon-Anmeldeinformationen abrufen.

So erstellen Sie das Konto

- Verwenden Sie die folgende SQL Anweisung.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name        = 'SES',
    @description         = 'Mail account for sending outgoing notifications.',
    @email_address       = 'nobody@example.com',
    @display_name        = 'Automated Mailer',
    @mailserver_name     = 'vpce-0a1b2c3d4e5f-01234567.email-smtp.us-
west-2.vpce.amazonaws.com',
    @port                = 587,
    @enable_ssl          = 1,
    @username            = 'Smtp_Username',
    @password            = 'Smtp_Password';
GO
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

Hinzufügen des Database Mail-Kontos zum Database Mail-Profil

Um das Database Mail-Konto dem Database Mail-Profil hinzuzufügen, verwenden Sie den gespeicherten Prozess sysmail_add_profileaccount_sp. Im folgenden Beispiel wird das `SES`-Konto dem `Notifications`-Profil hinzugefügt.

So fügen Sie das Konto dem Profil hinzu

- Verwenden Sie die folgende SQL Anweisung.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name        = 'Notifications',
    @account_name        = 'SES',
    @sequence_number     = 1;
GO
```

Hinzufügen von Benutzern zum Database Mail-Profil

Um einem msdb-Datenbankprinzipal die Berechtigung zur Verwendung eines Database Mail-Profils zu erteilen, verwenden Sie den gespeicherten Prozess sysmail_add_principalprofile_sp. Ein Principal ist eine Entität, die SQL Serverressourcen anfordern kann. Der Datenbankprinzipal muss einem SQL Serverauthentifizierungsbenutzer, einem Windows-Authentifizierungsbenutzer oder einer Windows-Authentifizierungsgruppe zugeordnet sein.

Im folgenden Beispiel wird öffentlicher Zugriff auf das Notifications-Profil gewährt.

So fügen Sie dem Profil einen Benutzer hinzu

- Verwenden Sie die folgende SQL Anweisung.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name       = 'Notifications',
    @principal_name     = 'public',
    @is_default         = 1;
GO
```

RDSGespeicherte Prozeduren und Funktionen von Amazon für Database Mail

Microsoft stellt gespeicherte Prozeduren für die Verwendung von Database Mail zur Verfügung, z. B. zum Erstellen, Auflisten, Aktualisieren und Löschen von Konten und Profilen. RDSStellt außerdem die in der folgenden Tabelle aufgeführten gespeicherten Prozeduren und Funktionen für Datenbank-E-Mail bereit.

Prozess/Funktion	Beschreibung
rds_fn_sysmail_allitems	Zeigt gesendete Nachrichten an, einschließlich der von anderen Benutzern übermittelten Nachrichten.
rds_fn_sysmail_event_log	Zeigt Ereignisse an, einschließlich solcher für Nachrichten, die von anderen Benutzern übermittelt wurden.
rds_fn_sysmail_mailanhänge	Zeigt Anhänge an, einschließlich solcher für Nachrichten, die von anderen Benutzern übermittelt wurden.
rds_sysmail_control	Startet und stoppt die E-Mail-Warteschlange (DatabaseMail.exe-Prozess).
rds_sysmail_delete_mailitems_sp	Löscht E-Mail-Nachrichten, die von allen Benutzern aus den internen Tabellen von Database Mail gesendet wurden.

Senden von E-Mails mit Database Mail

Sie verwenden den gespeicherten Prozess sp_send_dbmail, um E-Mails mit Database Mail zu versenden.

Verwendung

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'profile_name',
@recipients = 'recipient1@example.com[; recipient2; ... recipientn]',
@subject = 'subject',
@body = 'message_body',
[@body_format = 'HTML'],
[@file_attachments = 'file_path1; file_path2; ... file_pathn'],
[@query = 'SQL_query'],
[@attach_query_result_as_file = 0|1]';
```

Die folgenden Parameter sind erforderlich:

- `@profile_name` – Der Name des Database Mail-Profils, von dem die Nachricht gesendet werden soll.

- `@recipients` – Die durch Semikolons getrennte Liste der E-Mail-Adressen, an die die Nachricht gesendet werden soll.
- `@subject` – Der Betreff der Nachricht.
- `@body` – Der Text der Nachricht. Sie können auch eine angegebene Variable als Text verwenden.

Die folgenden Parameter sind optional:

- `@body_format`— Dieser Parameter wird mit einer deklarierten Variablen verwendet, um E-Mails im HTML Format zu senden.
- `@file_attachments` – Die durch Semikolons getrennte Liste von Nachrichtenanhängen. Dateipfade müssen absolute Pfade sein.
- `@query`— Eine SQL auszuführende Abfrage. Die Abfrageergebnisse können als Datei angehängt oder im Text der Nachricht enthalten sein.
- `@attach_query_result_as_file` – Ob das Abfrageergebnis als Datei angehängt werden soll. Setzen Sie auf 0 für nein, 1 für ja. Der Standardwert ist 0.

Beispiele

Die folgenden Beispiele veranschaulichen, wie Sie E-Mails versenden.

Example das Senden einer Nachricht an einen einzelnen Empfänger

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Automated DBMail message - 1',
     @body               = 'Database Mail configuration was successful.';
GO
```

Example das Senden einer Nachricht an mehrere Empfänger

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
```

```
     @profile_name       = 'Notifications',
     @recipients         = 'recipient1@example.com;recipient2@example.com',
     @subject            = 'Automated DBMail message - 2',
     @body               = 'This is a message.';
GO
```

Example — ein SQL Abfrageergebnis als Dateianhang zu senden

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test SQL query',
     @body               = 'This is a SQL query test.',
     @query              = 'SELECT * FROM abc.dbo.test',
     @attach_query_result_as_file = 1;
GO
```

Example des Sendens einer Nachricht im HTML Format

```
USE msdb
GO

DECLARE @HTML_Body as NVARCHAR(500) = 'Hi, <h4> Heading </h4> </br> See the report. <b>
 Regards </b>';

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test HTML message',
     @body               = @HTML_Body,
     @body_format        = 'HTML';
GO
```

Example des Sendens einer Nachricht mit einem Trigger, wenn ein bestimmtes Ereignis in der Datenbank auftritt

```
USE AdventureWorks2017
GO
IF OBJECT_ID ('Production.iProductNotification', 'TR') IS NOT NULL
```

```
DROP TRIGGER Purchasing.iProductNotification
GO

CREATE TRIGGER iProductNotification ON Production.Product
   FOR INSERT
   AS
   DECLARE @ProductInformation nvarchar(255);
   SELECT
   @ProductInformation = 'A new product, ' + Name + ', is now available for $' +
 CAST(StandardCost AS nvarchar(20)) + '!'
   FROM INSERTED i;

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'New product information',
     @body               = @ProductInformation;
GO
```

Anzeigen von Nachrichten, Protokollen und Anhängen

Sie verwenden RDS gespeicherte Prozeduren, um Nachrichten, Ereignisprotokolle und Anlagen anzuzeigen.

So zeigen Sie alle E-Mails an

- Verwenden Sie die folgende SQL Abfrage.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_allitems(); --WHERE sent_status='sent' or
 'failed' or 'unsent'
```

So zeigen Sie alle E-Mail-Ereignisprotokolle an

- Verwenden Sie die folgende SQL Abfrage.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_event_log();
```

So zeigen Sie alle E-Mail-Anhänge an

- Verwenden Sie die folgende SQL Abfrage.

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_mailattachments();
```

Löschen von Nachrichten

Sie verwenden die `rds_sysmail_delete_mailitems_sp` gespeicherte Prozedur, um Nachrichten zu löschen.

> **Note**
>
> RDSlöscht automatisch Mailtabellenelemente, wenn die DBMail Verlaufsdaten eine Größe von 1 GB erreichen, mit einer Aufbewahrungsdauer von mindestens 24 Stunden.
> Wenn Sie Postsendungen länger aufbewahren möchten, können Sie diese archivieren.
> Weitere Informationen finden Sie in der Microsoft-Dokumentation unter Erstellen eines SQL Server-Agent-Auftrags zum Archivieren von Datenbank-E-Mail-Nachrichten und Ereignisprotokollen.

So löschen Sie alle E-Mail-Nachrichten

- Verwenden Sie die folgende SQL Anweisung.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_before = @GETDATE;
GO
```

So löschen Sie alle E-Mail-Nachrichten mit einem bestimmten Status

- Verwenden Sie die folgende SQL Anweisung, um alle fehlgeschlagenen Nachrichten zu löschen.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_status = 'failed';
GO
```

E-Mail-Warteschlange starten und beenden

Verwenden Sie die folgenden Anweisungen, um die DB-Mail-Warteschlange zu starten und zu beenden:

Themen

- Starten der Mail-Warteschlange
- Stoppen der Mail-Warteschlange

Starten der Mail-Warteschlange

Sie verwenden die `rds_sysmail_control` gespeicherte Prozedur, um den Prozess Database Mail zu starten.

> **Note**
>
> Die Aktivierung von Database Mail startet automatisch die E-Mail-Warteschlange.

So starten Sie die Mail-Warteschlange

- Verwenden Sie die folgende SQL Anweisung.

```
EXECUTE msdb.dbo.rds_sysmail_control start;
GO
```

Stoppen der Mail-Warteschlange

Sie verwenden die `rds_sysmail_control` gespeicherte Prozedur, um den Prozess Database Mail zu beenden.

So stoppen Sie die Mail-Warteschlange

- Verwenden Sie die folgende SQL Anweisung.

```
EXECUTE msdb.dbo.rds_sysmail_control stop;
GO
```

Arbeiten mit Dateianlagen

Die folgenden Dateianhangserweiterungen werden in Datenbank-Mail-Nachrichten von RDS einem SQL Server nicht
unterstützt: .ade, .adp, .apk, .appx, .appxbundle, .bat, .bak, .cab, .chm, .cmd, .com, .cpl, .dll, .dmg, .exe, .hta, und .wsh.

Database Mail verwendet den Microsoft Windows-Sicherheitskontext des aktuellen Benutzers, um den Zugriff auf Dateien zu steuern. Benutzer, die sich mit der SQL Serverauthentifizierung anmelden, können keine Dateien mithilfe des `@file_attachments` Parameters mit der `sp_send_dbmail` gespeicherten Prozedur anhängen. Windows erlaubt SQL dem Server nicht, Anmeldeinformationen von einem Remotecomputer an einen anderen Remotecomputer weiterzugeben. Daher kann Datenbank-E-Mail keine Dateien von einer Netzwerkfreigabe anhängen, wenn der Befehl von einem anderen Computer als dem Computer ausgeführt wird, auf dem SQL Server ausgeführt wird.

Sie können jedoch SQL Server-Agent-Jobs verwenden, um Dateien anzuhängen. Weitere Informationen zum SQL Server-Agent finden Sie unter Verwenden des SQL Server-Agenten für Amazon RDS und SQLServer-Agent in der Microsoft-Dokumentation.

Überlegungen zu Multi-AZ-Bereitstellungen

Wenn Sie Database Mail auf einer Multi-AZ-DB-Instance konfigurieren, wird die Konfiguration nicht automatisch an die sekundäre weitergegeben. Wir empfehlen, die Multi-AZ-Instance in eine Single-AZ-Instance zu konvertieren, Database Mail zu konfigurieren und dann die DB-Instance wieder in Multi-AZ zu konvertieren. Dann haben sowohl der primäre als auch der sekundäre Knoten die Database Mail-Konfiguration.

Wenn Sie aus Ihrer Multi-AZ-Instance, für die Datenbank-Mail konfiguriert ist, ein Read Replica erstellen, erbt das Replikat die Konfiguration, jedoch ohne das Passwort für den Server. SMTP Aktualisieren Sie das Database Mail-Konto mit dem Passwort.

Aufhebung der Beschränkung SMTP (Port 25)

AWS Blockiert standardmäßig ausgehenden Datenverkehr auf SMTP (Port 25) RDS für SQL Server-DB-Instances. Dies geschieht, um Spam auf der Grundlage der Richtlinien des Besitzers der elastic network interface zu verhindern. Sie können diese Einschränkung bei Bedarf aufheben. Weitere Informationen finden Sie unter Wie entferne ich die Beschränkung für Port 25 aus meiner EC2 Amazon-Instance oder Lambda-Funktion? .

Instance-Speicher-Support für die tempdb-Datenbank in Amazon RDS for SQL Server

Ein Instance-Speicher stellt für Ihre DB-Instance temporären Speicher auf Blockebene bereit. Dieser Speicher befindet sich auf Laufwerken, die physisch mit dem Hostcomputer verbunden sind. Diese Laufwerke verfügen über einen nicht-flüchtigen Memory Express (NVMe)-Instance-Speicher, der auf Solid-State-Laufwerken (SSD) basiert. Dieser Speicher ist für niedrige Latenzen, eine sehr hohe Random-I/O-Leistung und einen hohen sequentiellen Lesedurchsatz optimiert.

Durch das Speichern von `tempdb`-Datendateien und `tempdb`-Protokolldateien im Instance-Speicher können Sie im Vergleich zum Standardspeicher basierend auf Amazon EBS niedrigere Lese- und Schreiblatenzen erzielen.

> **Note**
>
> SQL Server-Datenbankdateien und Datenbankprotokolldateien werden nicht im Instance-Speicher platziert.

Aktivieren des Instance-Speichers

Wenn RDS DB-Instances mit einem der folgenden Instance-Klassen bereitstellt, wird die `tempdb`-Datenbank automatisch in den Instance-Speicher eingefügt:

- db.m5d
- db.r5d
- db.x2iedn

Um den Instance-Speicher zu aktivieren, führen Sie einen der folgenden Schritte aus:

- Erstellen Sie eine SQL Server-DB-Instance mit einem dieser Instance-Typen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Modifizieren Sie eine vorhandene SQL Server-DB-Instance, um eine davon zu verwenden. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Der Instance-Speicher ist in allen AWS-Regionen verfügbar, in denen einer oder mehrere dieser Instance-Typen unterstützt werden. Weitere Informationen zu den Instance-Klassen `db.m5d` und

db.r5d finden Sie unter . Weitere Hinweise zu den von Amazon RDS for SQL Server unterstützten Instance-Klassen finden Sie unter Unterstützung für DB-Instance-Klassen für Microsoft SQL Server.

Überlegungen zum Speicherort und zur Größe der Datei

Auf Instances ohne Instance-Speicher speichert RDS die Daten und Protokolldateien tempdb im Verzeichnis D:\rdsdbdata\DATA. Beide Dateien beginnen standardmäßig bei 8 MB.

Auf Instances mit Instance-Speicher speichert RDS die Daten und Protokolldateien tempdb im Verzeichnis T:\rdsdbdata\DATA.

Wann tempdb hat nur eine Datendatei (tempdb.mdf) und eine Protokolldatei (templog.ldf) hat, beginnt templog.ldf standardmäßig bei 8 MB und tempdb.mdf beginnt bei mindestens 80 % der Speicherkapazität der Instance. Zwanzig Prozent der Speicherkapazität oder 200 GB – je nachdem, was geringer ist – werden für den Start freigehalten. Mehrere tempdb-Datendateien teilen den Speicherplatz von 80 % gleichmäßig auf, während Protokolldateien immer eine Anfangsgröße von 8 MB haben.

Wenn Sie zum Beispiel Ihre DB-Instance-Klasse von db.m5.2xlarge zu db.m5d.2xlarge ändern, erhöht sich die Größe von tempdb-Datendateien von jeweils 8 MB auf 234 GB.

> **Note**
>
> Neben den Daten- und Protokolldateien tempdb im Instance-Speicher (T:\rdsdbdata\DATA), können Sie immer noch extra Daten und Protokolldateien tempdb auf dem Datenvolume (D:\rdsdbdata\DATA) erstellen. Diese Dateien haben immer eine Anfangsgröße von 8 MB.

Überlegungen zu Backups

Möglicherweise müssen Sie Backups für lange Zeiträume aufbewahren, was im Laufe der Zeit Kosten verursacht. Die tempdb-Daten und -Protokollblöcke können sich je nach Workload sehr häufig ändern. Dies kann die Größe von DB-Snapshots erheblich erhöhen.

Wenn sich tempdb im Instance-Speicher befindet, enthalten Snapshots keine temporären Dateien. Dies bedeutet, dass die Snapshots kleiner sind und im Vergleich zu reinem EBS-Speicher weniger von der kostenlosen Sicherungszuweisung verbrauchen.

Fehler „Voller Datenträger“

Wenn Sie den gesamten verfügbaren Speicherplatz im Instance-Speicher verwenden, erhalten Sie möglicherweise folgende Fehler:

- Das Transaktionsprotokoll für die Datenbank 'tempdb' ist aufgrund von „ACTIVE_TRANSACTION“ voll.
- In der Datenbank 'tempdb' konnte kein Speicherplatz für das Objekt 'dbo.SORT temporärer Speicher ausgeführt: 140738941419520' zugewiesen werden, da die Dateigruppe 'PRIMARY' voll ist. Schaffen Sie Speicherplatz, indem Sie nicht benötigte Dateien löschen, Objekte in die Dateigruppe verschieben, der Dateigruppe zusätzliche Dateien hinzufügen oder das automatische Erweitern für vorhandene Dateien in der Dateigruppe aktivieren.

Wenn der Instance-Speicher voll ist, können Sie eine oder mehrere der folgenden Aktionen ausführen:

- Passen Sie Ihren Workload oder die Art der Nutzung von a `tempdb`.
- Skalieren Sie auf die Verwendung einer DB-Instance-Klasse mit mehr NVMe-Speicher.
- Verwenden Sie nicht länger den Instance-Speicher, sondern eine Instance-Klasse mit EBS-Speicher.
- Verwenden Sie einen gemischten Modus, indem Sie sekundäre Daten oder Protokolldateien für `tempdb` zum EBS-Volume hinzufügen.

Entfernen des Instance-Speichers

Um den Instance-Speicher zu entfernen, ändern Sie Ihre SQL Server-DB-Instance so, dass sie einen Instance-Typ verwendet, der keinen Instance-Speicher unterstützt, wie db.m5, db.r5 oder db.x1e.

> **Note**
>
> Wenn Sie den Instance-Speicher entfernen, werden die temporären Dateien in das Verzeichnis `D:\rdsdbdata\DATA` verschoben und auf 8 MB reduziert.

Verwenden erweiterter Ereignisse mit Amazon RDS für Microsoft SQL Server

Sie können erweiterte Ereignisse in Microsoft SQL Server verwenden, um Debugging- und Fehlerbehebungsinformationen für Amazon RDS for SQL Server zu erfassen. Erweiterte Ereignisse ersetzen SQL Trace und Server Profiler, die von Microsoft als veraltet eingestuft wurden. Erweiterte Ereignisse ähneln Profiler-Traces, haben jedoch eine genauere Kontrolle über die nachverfolgten Ereignisse. Erweiterte Ereignisse werden für SQL Serverversionen 2016 und höher bei Amazon unterstütztRDS. Weitere Informationen finden Sie unter Übersicht über erweiterte Ereignisse in der Microsoft-Dokumentation.

Erweiterte Ereignisse werden automatisch für Benutzer mit Master-Benutzerrechten in Amazon RDS for SQL Server aktiviert.

Themen

- Einschränkungen und Empfehlungen
- Konfiguration erweiterter Ereignisse auf RDS für Server SQL
- Überlegungen zu Multi-AZ-Bereitstellungen
- Abfragen von erweiterten Ereignisdateien

Einschränkungen und Empfehlungen

Bei der Verwendung von Extended Events auf RDS for SQL Server gelten die folgenden Einschränkungen:

- Erweiterte Ereignisse werden nur für die Enterprise und Standard Editions unterstützt.
- Sie können standardmäßige erweiterte Ereignissitzungen nicht ändern.
- Stellen Sie sicher, dass Sie die Sitzungsspeicherpartition auf einstelle `NONE`.
- Der Aufbewahrungsmodus für Sitzungsereignisse kann entweder `ALLOW_SINGLE_EVENT_LOSS` oder `ALLOW_MULTIPLE_EVENT_LOSS` sein.
- Event Tracing für Windows (ETW) -Ziele werden nicht unterstützt.
- Stellen Sie sicher, dass sich die Dateiziele im Verzeichnis `D:\rdsdbdata\log` befinden.
- Um zusammengehörige Ziele zu paaren, setzen Sie die `respond_to_memory_pressure`-Eigenschaft auf `1`.
- Der Zielspeicher des Ringpuffers darf nicht größer als 4 MB sein.

- Die folgenden Aktionen werden nicht unterstützt:
  - `debug_break`
  - `create_dump_all_threads`
  - `create_dump_single_threads`
- Das `rpc_completed` Ereignis wird in den folgenden Versionen und später unterstützt: 15.0.4083.2, 14.0.3370.1, 13.0.5865.1, 12.0.6433.1, 11.0.7507.2.

Konfiguration erweiterter Ereignisse auf RDS für Server SQL

Auf dem SQL Server können Sie die Werte bestimmter Parameter RDS für erweiterte Veranstaltungssitzungen konfigurieren. In der folgenden Tabelle werden die konfigurierbaren Parameter beschrieben.

Parametername	Beschreibung
`xe_session_max_memory`	Angabe der maximalen Speichermenge, die der Sitzung f Dieser Wert entspricht der `max_memory` -Einstellung der
`xe_session_max_event_size`	Angabe der maximalen Speichergröße an, die für große E `max_event_size` -Einstellung der Ereignissitzung.
`xe_session_max_dispatch_latency`	Angabe der Dauer der Pufferung von Ereignissen im Spe abgegeben werden. Dieser Wert entspricht der `max_dis`
`xe_file_target_size`	Festlegung der maximalen Größe des Dateiziels. Dieser W des Dateiziels.
`xe_file_retention`	Angabe der Aufbewahrungszeit für Dateien in Tagen, die werden.

> **Note**
>
> Wenn der Wert `xe_file_retention` auf Null gesetzt wird, werden XEL-Dateien automatisch entfernt, nachdem die Sperre für diese Dateien vom SQL Server aufgehoben wurde. Die Sperre wird aufgehoben, wenn eine .xel-Datei die in eingestellte Größenbeschränkung erreich `xe_file_target_size`.

Sie können die in `rdsadmin.dbo.rds_show_configuration` gespeicherte Prozedur verwenden, um die aktuellen Werte dieser Parameter anzuzeigen. Verwenden Sie beispielsweise die folgende SQL Anweisung, um die aktuelle Einstellung von `xe_session_max_memory` anzuzeigen.

```
exec rdsadmin.dbo.rds_show_configuration 'xe_session_max_memory'
```

Sie können die in `rdsadmin.dbo.rds_set_configuration` gespeicherte Prozedur verwenden, um sie zu ändern. Verwenden Sie beispielsweise die folgende SQL Anweisung, um den Wert `xe_session_max_memory` auf 4 MB festzulegen.

```
exec rdsadmin.dbo.rds_set_configuration 'xe_session_max_memory', 4
```

Überlegungen zu Multi-AZ-Bereitstellungen

Wenn Sie eine erweiterte Ereignissitzung auf einer primären DB-Instance erstellen, wird sie nicht auf das Standby-Replikat übertragen. Sie können einen Failover haben und die erweiterte Ereignissitzung für die neue primäre DB-Instance erstellen. Oder Sie können die Multi-AZ-Konfiguration entfernen und dann erneut hinzufügen, um die erweiterte Ereignissitzung auf das Standby-Replikat zu übertragen. RDSstoppt alle nicht standardmäßigen Extended-Event-Sitzungen auf dem Standby-Replikat, sodass diese Sitzungen keine Ressourcen auf dem Standby-Replikat verbrauchen. Nachdem ein Standby-Replikat zur primären DB-Instance wurde, sollten Sie daher die erweiterten Ereignissitzungen manuell auf der neuen primären DB-Instance starten.

> **Note**
>
> Dieser Ansatz gilt sowohl für Always On-Verfügbarkeitsgruppen als auch für die Datenbankspiegelung.

Sie können auch einen SQL Server-Agent-Job verwenden, um das Standby-Replikat zu verfolgen und die Sitzungen zu starten, falls das Standby-Replikat zum primären Replikat wird. Verwenden Sie beispielsweise die folgende Abfrage in Ihrem SQL Server-Agent-Jobschritt, um Event-Sitzungen auf einer primären DB-Instance neu zu starten.

```
BEGIN
    IF (DATABASEPROPERTYEX('rdsadmin','Updateability')='READ_WRITE'
    AND DATABASEPROPERTYEX('rdsadmin','status')='ONLINE'
    AND (DATABASEPROPERTYEX('rdsadmin','Collation') IS NOT NULL OR
 DATABASEPROPERTYEX('rdsadmin','IsAutoClose')=1)
```

```
    )
    BEGIN
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe1')
            ALTER EVENT SESSION xe1 ON SERVER STATE=START
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe2')
            ALTER EVENT SESSION xe2 ON SERVER STATE=START
    END
END
```

Diese Abfrage startet die Ereignissitzungen `xe1` und `xe2` auf einer primären DB-Instance neu, wenn sich diese Sitzungen in einem gestoppten Zustand befinden. Sie können dieser Abfrage auch einen Zeitplan mit einem passenden Intervall hinzufügen.

Abfragen von erweiterten Ereignisdateien

Sie können entweder SQL Server Management Studio oder die `sys.fn_xe_file_target_read_file` Funktion verwenden, um Daten von erweiterten Ereignissen anzuzeigen, die Dateiziele verwenden. Weitere Informationen zu dieser Funktion finden Sie in der Microsoft-Dokumentation unter sys.fn_xe_file_target_read_file (Transact-). SQL

Ziele mit erweiterten Ereignisdateien können nur Dateien in das Verzeichnis auf dem Server schreiben. `D:\rdsdbdata\log` RDS SQL

Verwenden Sie beispielsweise die folgende SQL Abfrage, um den Inhalt aller Dateien von Sessions mit erweitertem Ereignis aufzulisten, deren Namen mit beginnen`xe`.

```
SELECT * FROM sys.fn_xe_file_target_read_file('d:\rdsdbdata\log\xe*', null,null,null);
```

Zugriff auf Transaktionsprotokoll-Backups mit RDS für SQL Server

Mit Zugriff auf Transaktionsprotokoll-Backups für RDS für SQL Server können Sie die Transaktionsprotokoll-Backup-Dateien für eine Datenbank auflisten und sie in einen Ziel-Bucket von Amazon S3 kopieren. Indem Sie Transaktionsprotokoll-Backups in einen Amazon-S3-Bucket kopieren, können Sie sie in Kombination mit vollständigen und differentiellen Datenbank-Backups verwenden, um zeitpunktbezogene Datenbankwiederherstellungen durchzuführen. Sie verwenden gespeicherte RDS-Prozeduren, um den Zugriff auf Transaktionsprotokoll-Backups einzurichten, verfügbare Transaktionsprotokoll-Backups aufzulisten und sie in Ihren Amazon-S3-Bucket zu kopieren.

Der Zugriff auf Transaktionsprotokoll-Backups bietet die folgenden Funktionen und Vorteile:

- Sie können die Metadaten verfügbarer Transaktionsprotokoll-Backups für eine Datenbank auf einer DB-Instance von RDS für SQL Server auflisten und anzeigen.
- Sie können verfügbare Transaktionsprotokoll-Backups von RDS für SQL Server in einen Ziel-Bucket von Amazon S3 kopieren.
- Führen Sie point-in-time Wiederherstellungen von Datenbanken durch, ohne eine gesamte DB-Instance wiederherstellen zu müssen. Weitere Informationen zum Wiederherstellen einer DB-Instance für einen bestimmten Zeitpunkt finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Verfügbarkeit und Support

Der Zugriff auf Transaktionsprotokoll-Backups wird in allen AWS Regionen unterstützt. Der Zugriff auf Transaktionsprotokoll-Backups ist für alle in Amazon RDS unterstützten Editionen und Versionen von Microsoft SQL Server verfügbar.

Voraussetzungen

Es müssen die folgenden Anforderungen erfüllt sein, um den Zugriff auf Transaktionsprotokoll-Backups zu aktivieren:

- Automatisierte Backups müssen auf der DB-Instance aktiviert sein und die Backup-Aufbewahrung muss auf einen Wert von einem oder mehreren Tagen festgelegt werden. Weitere Informationen zur Aktivierung automatisierter Backups und zur Konfiguration einer Aufbewahrungsrichtlinie finden Sie unter Aktivieren von automatisierten Backups.

- Ein Amazon-S3-Bucket muss im gleichen Konto und in derselben Region wie die Quell-DB-Instance existieren. Bevor Sie den Zugriff auf Transaktionsprotokoll-Backups aktivieren, wählen Sie einen vorhandenen Amazon-S3-Bucket aus oder erstellen Sie einen neuen Bucket, der für Ihre Transaktionsprotokoll-Backup-Dateien verwendet werden soll.
- Eine Berechtigungsrichtlinie für den Amazon-S3-Bucket-muss wie folgt konfiguriert werden, damit Amazon RDS Transaktionsprotokolldateien hineinkopieren kann:

  1. Legen Sie die Eigenschaft für den Objektkontobesitz für den Bucket auf Bucket Owner Preferred (Bucket-Eigentümer bevorzugt) fest.
  2. Fügen Sie die folgende Richtlinie hinzu. In der Standardeinstellung ist keine Richtlinie vorhanden. Verwenden Sie daher die Zugriffssteuerungsliste (ACL) für den Bucket, um die Bucket-Richtlinie zu bearbeiten und hinzuzufügen.

  Im folgenden Beispiel wird ein ARN zur Angabe einer Ressource verwendet. Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `SourceArn` und `SourceAccount` in ressourcenbasierten Vertrauensbeziehungen, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Weitere Informationen zur Arbeit mit ARNs finden Sie unter Amazon-Ressourcennamen (ARNs) undAmazon-Ressourcennamen (ARNs) in Amazon RDS.

  Example einer Amazon-S3-Berechtigungsrichtlinie für den Zugriff auf Transaktionsprotokoll-Backups

  JSON

```
   {
   "Version": "2012-10-17",
   "Statement": [
       {
           "Sid": "Only allow writes to my bucket with bucket owner full
control",
           "Effect": "Allow",
           "Principal": {
               "Service": "backups.rds.amazonaws.com"
           },
           "Action": "s3:PutObject",
           "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/{customer_path}/*",
           "Condition": {
               "StringEquals": {
```

```
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:sourceAccount": "{customer_account}",
                    "aws:sourceArn": "{db_instance_arn}"
                }
            }
        }
    ]
}
```

- Eine AWS Identity and Access Management (IAM-) Rolle für den Zugriff auf den Amazon S3 S3-Bucket. Wenn Sie bereits über eine IAM-Rolle verfügen, können Sie diese verwenden. Alternativ können Sie wählen, dass eine neue IAM-Rolle für Sie erstellt wird, wenn Sie die Option `SQLSERVER_BACKUP_RESTORE` mithilfe der AWS Management Console hinzufügen. Alternativ können Sie manuell eine neue Rolle erstellen. Weitere Informationen zum Erstellen und Konfigurieren einer IAM-Rolle mit `SQLSERVER_BACKUP_RESTORE` finden Sie unter Manuelles Erstellen einer IAM-Rolle für native Backups und Wiederherstellungen.
- Die Option `SQLSERVER_BACKUP_RESTORE` muss einer Optionsgruppe auf Ihrer DB-Instance hinzugefügt werden. Weitere Informationen zum Hinzufügen der Option `SQLSERVER_BACKUP_RESTORE` finden Sie unter Unterstützung für native Sicherung und Backup in SQL Server.

> **Note**
>
> Wenn für Ihre DB-Instance die Speicherverschlüsselung aktiviert ist, müssen die AWS KMS (KMS) -Aktionen und der Schlüssel in der IAM-Rolle bereitgestellt werden, die in der Optionsgruppe für systemeigene Sicherung und Wiederherstellung bereitgestellt wird.

Wenn Sie die gespeicherte Prozedur `rds_restore_log` zur Durchführung von zeitpunktbezogenen Datenbankwiederherstellungen verwenden möchten, empfehlen wir optional, denselben Amazon-S3-Pfad für die systemeigene Sicherungs- und Wiederherstellungsoptionsgruppe und den Zugriff auf Transaktionsprotokoll-Backups zu verwenden. Diese Methode stellt sicher, dass Amazon RDS, wenn es die Rolle der Optionsgruppe zur Ausführung der Wiederherstellungsprotokollfunktionen übernimmt, Zugriff auf das Abrufen von Transaktionsprotokoll-Backups aus demselben Amazon-S3-Pfad hat.

- Wenn die DB-Instance unabhängig vom Verschlüsselungstyp (AWS verwalteter Schlüssel oder vom Kunden verwalteter Schlüssel) verschlüsselt ist, müssen Sie in der IAM-Rolle und in der

`rds_tlog_backup_copy_to_S3` gespeicherten Prozedur einen vom Kunden verwalteten KMS-Schlüssel angeben.

Einschränkungen und Empfehlungen

Für den Zugriff auf Transaktionsprotokoll-Backups gelten die folgenden Einschränkungen und Empfehlungen:

- Sie können die Transaktionsprotokoll-Backups der letzten sieben Tage für jede DB-Instance auflisten und kopieren, für die die Backup-Aufbewahrung zwischen 1 und 35 Tagen konfiguriert ist.
- Ein Amazon-S3-Bucket, der für den Zugriff auf Transaktionsprotokoll-Backups verwendet wird, muss sich im gleichen Konto und in derselben Region wie die Quell-DB-Instance befinden. Kontoübergreifendes und regionsübergreifendes Kopieren wird nicht unterstützt.
- Es kann nur ein Amazon-S3-Bucket als Ziel konfiguriert werden, in das Transaktionsprotokoll-Backups kopiert werden. Sie können mit der gespeicherten Prozedur `rds_tlog_copy_setup` einen neuen Ziel-Bucket von Amazon S3 auswählen. Weitere Informationen zur Auswahl eines neuen Ziel-Buckets von Amazon S3 finden Sie unter Einrichten des Zugriffs auf Transaktionsprotokoll-Backups.
- Sie können den KMS-Schlüssel bei Verwendung der gespeicherten Prozedur `rds_tlog_backup_copy_to_S3` nicht angeben, wenn Ihre RDS-Instance nicht für die Speicherverschlüsselung aktiviert ist.
- Das Kopieren mehrerer Konten wird nicht unterstützt. Die zum Kopieren verwendete IAM-Rolle erlaubt nur den Schreibzugriff auf Amazon-S3-Buckets innerhalb des Besitzerkontos der DB-Instance.
- Auf einer DB-Instance von RDS für SQL Server können nur zwei gleichzeitige Aufgaben eines beliebigen Typs ausgeführt werden.
- Für eine einzelne Datenbank kann zu einem bestimmten Zeitpunkt nur eine Kopieraufgabe ausgeführt werden. Wenn Sie Transaktionsprotokoll-Backups für mehrere Datenbanken auf der DB-Instance kopieren möchten, verwenden Sie für jede Datenbank eine separate Kopieraufgabe.
- Wenn Sie ein Transaktionsprotokoll-Backup kopieren, das bereits mit demselben Namen im Amazon-S3-Bucket existiert, wird das vorhandene Transaktionsprotokoll-Backup überschrieben.
- Sie können nur die gespeicherten Prozeduren ausführen, die Zugriff auf Transaktionsprotokoll-Backups auf der primären DB-Instance haben. Sie können diese gespeicherten Prozeduren nicht auf einem Lesereplikat von RDS für SQL Server oder auf einer sekundären Instance eines Multi-AZ-DB-Clusters ausführen.

- Wenn die DB-Instance von RDS für SQL Server neu gestartet wird, während die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` ausgeführt wird, wird die Aufgabe automatisch von Anfang an neu gestartet, wenn die DB-Instance wieder online ist. Alle Transaktionsprotokoll-Backups, die während der Ausführung der Aufgabe vor dem Neustart in den Amazon-S3-Bucket kopiert wurden, werden überschrieben.
- Die Systemdatenbanken von Microsoft SQL Server und die `RDSAdmin`-Datenbank können nicht für den Zugriff auf Transaktionsprotokoll-Backups konfiguriert werden.
- Das Kopieren in mit SSE-KMS verschlüsselte Buckets wird nicht unterstützt.

Einrichten des Zugriffs auf Transaktionsprotokoll-Backups

Wenn Sie den Zugriff auf Transaktionsprotokoll-Backups einrichten möchten, erfüllen Sie alle Anforderungen in der Liste im Abschnitt Voraussetzungen und führen Sie dann die gespeicherte Prozedur `rds_tlog_copy_setup` aus. Das Verfahren ermöglicht den Zugriff auf die Funktion für Transaktionsprotokoll-Backups auf DB-Instance-Ebene. Sie müssen es nicht für jede einzelne Datenbank auf der DB-Instance ausführen.

> **Important**
>
> Dem Datenbankbenutzer muss in jeder Datenbank die `db_owner` Rolle innerhalb von SQL Server zugewiesen werden, um die Funktion für den Zugriff auf Transaktionslog-Backups konfigurieren und verwenden zu können.

Example Verwendung:

```
exec msdb.dbo.rds_tlog_copy_setup
@target_s3_arn='arn:aws:s3:::amzn-s3-demo-bucket/myfolder';
```

Der folgende Parameter ist erforderlich:

- `@target_s3_arn`— Der ARN des Amazon S3 S3-Ziel-Buckets, in den die Transaktionsprotokoll-Backup-Dateien kopiert werden sollen.

Example Einrichten eines Ziel-Buckets von Amazon S3:

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::amzn-s3-demo-logging-bucket/mytestdb1';
```

Rufen Sie die gespeicherte Prozedur `rds_show_configuration` auf, um die Konfiguration zu überprüfen.

Example Überprüfen der Konfiguration:

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Um den Zugriff auf Transaktionsprotokoll-Backups so zu ändern, dass sie auf einen anderen Amazon-S3-Bucket verweisen, können Sie den aktuellen Amazon-S3-Bucket-Wert anzeigen und die gespeicherte Prozedur `rds_tlog_copy_setup` mit einem neuen Wert für `@target_s3_arn` erneut ausführen.

Example Anzeigen des vorhandenen Amazon-S3-Buckets, der für den Zugriff auf Transaktionsprotokoll-Backups konfiguriert ist

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Example Aktualisieren auf einen neuen Ziel-Bucket von Amazon S3

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::amzn-s3-demo-logging-bucket1/mynewfolder';
```

Verfügbare Transaktionsprotokoll-Backups auflisten

Bei RDS For SQL Server werden Transaktionsprotokollsicherungen für Datenbanken, die so konfiguriert sind, dass sie das vollständige Wiederherstellungsmodell verwenden und die Aufbewahrung von DB-Instance-Backups auf einen oder mehrere Tage festgelegt ist, automatisch aktiviert. Wenn Sie den Zugriff auf Transaktionsprotokoll-Backups aktivieren, stehen Ihnen bis zu sieben Tage dieser Transaktionsprotokoll-Backups zur Verfügung, damit Sie sie in Ihren Amazon-S3-Bucket kopieren können.

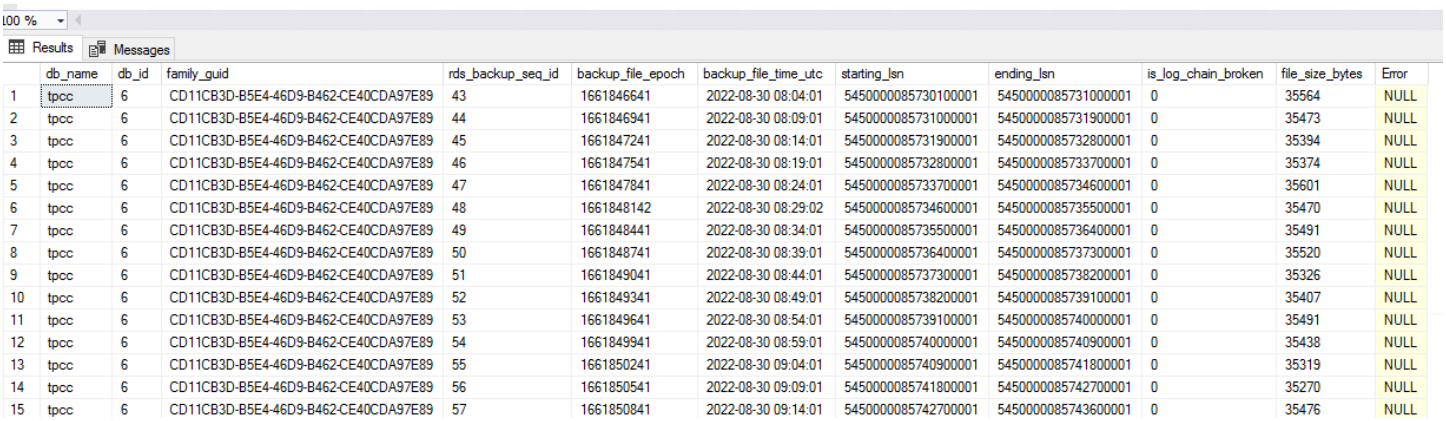
Nachdem Sie den Zugriff auf Transaktionsprotokoll-Backups aktiviert haben, können Sie damit beginnen, verfügbare Transaktionsprotokoll-Backup-Dateien aufzulisten und zu kopieren.

Auflisten von Transaktionsprotokoll-Backups

Rufen Sie die `rds_fn_list_tlog_backup_metadata`-Funktion auf, um alle für eine einzelne Datenbank verfügbaren Transaktionsprotokoll-Backups aufzulisten. Sie können eine `ORDER BY`- oder eine `WHERE`-Klausel verwenden, wenn Sie die Funktion aufrufen.

Example Auflistung und Filterung verfügbarer Transaktionsprotokoll-Backup-Dateien

```
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename');
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 rds_backup_seq_id = 3507;
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 backup_file_time_utc > '2022-09-15 20:44:01' ORDER BY backup_file_time_utc DESC;
```



100 %

Results Messages

	db_name	db_id	family_guid	rds_backup_seq_id	backup_file_epoch	backup_file_time_utc	starting_lsn	ending_lsn	is_log_chain_broken	file_size_bytes	Error
1	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	43	1661846641	2022-08-30 08:04:01	5450000085730100001	5450000085731000001	0	35564	NULL
2	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	44	1661846941	2022-08-30 08:09:01	5450000085731000001	5450000085731900001	0	35473	NULL
3	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	45	1661847241	2022-08-30 08:14:01	5450000085731900001	5450000085732800001	0	35394	NULL
4	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	46	1661847541	2022-08-30 08:19:01	5450000085732800001	5450000085733700001	0	35374	NULL
5	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	47	1661847841	2022-08-30 08:24:01	5450000085733700001	5450000085734600001	0	35601	NULL
6	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	48	1661848142	2022-08-30 08:29:02	5450000085734600001	5450000085735500001	0	35470	NULL
7	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	49	1661848441	2022-08-30 08:34:01	5450000085735500001	5450000085736400001	0	35491	NULL
8	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	50	1661848741	2022-08-30 08:39:01	5450000085736400001	5450000085737300001	0	35520	NULL
9	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	51	1661849041	2022-08-30 08:44:01	5450000085737300001	5450000085738200001	0	35326	NULL
10	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	52	1661849341	2022-08-30 08:49:01	5450000085738200001	5450000085739100001	0	35407	NULL
11	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	53	1661849641	2022-08-30 08:54:01	5450000085739100001	5450000085740000001	0	35491	NULL
12	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	54	1661849941	2022-08-30 08:59:01	5450000085740000001	5450000085740900001	0	35438	NULL
13	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	55	1661850241	2022-08-30 09:04:01	5450000085740900001	5450000085741800001	0	35319	NULL
14	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	56	1661850541	2022-08-30 09:09:01	5450000085741800001	5450000085742700001	0	35270	NULL
15	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	57	1661850841	2022-08-30 09:14:01	5450000085742700001	5450000085743600001	0	35476	NULL

Die Funktion `rds_fn_list_tlog_backup_metadata` gibt beispielsweise die folgende Ausgabe zurück:

Spaltenname	Datentyp	Beschreibung
db_name	sysname	Der Datenbankname, der für die Auflistung der Transaktionsprotokoll-Backups angegeben wurde.
db_id	int	Die interne Datenbank-ID für den Eingabepa rameter db_name.

Spaltenname	Datentyp	Beschreibung
`family_guid`	uniqueidentifier	Die eindeutige ID der ursprünglichen Datenbank zum Zeitpunkt der Erstellung. Dieser Wert bleibt unverändert, wenn die Datenbank wiederhergestellt wird, auch unter einem anderen Datenbanknamen.
`rds_backup_seq_id`	int	Die ID, die intern RDS verwendet wird, um eine Sequenznummer für jede Transaktionslog-Backup-Datei zu verwalten.
`backup_file_epoch`	bigint	Die Epochenzeit, zu der eine Transaktions-Backup-Datei generiert wurde.
`backup_file_time_utc`	datetime	Der in die UTC Zeit umgerechnete Wert für den `backup_file_epoch` Wert.
`starting_lsn`	numeric(25,0)	Die Protokollsequenznummer des ersten oder ältesten Protokolldatensatzes einer Transaktionsprotokoll-Backup-Datei.
`ending_lsn`	numeric(25,0)	Die Protokollsequenznummer des letzten oder nächsten Protokolldatensatzes einer Transaktionsprotokoll-Backup-Datei.
`is_log_chain_broken`	Bit	Ein boolescher Wert, der angibt, ob die Protokollkette zwischen der aktuellen Transaktionsprotokoll-Backup-Datei und der vorherigen Transaktionsprotokoll-Backup-Datei unterbrochen ist.
`file_size_bytes`	bigint	Die Größe des Transaktions-Backups in Byte.
`Error`	varchar(4000)	Fehlermeldung, wenn die `rds_fn_list_tlog_backup_metadata` -Funktion eine Ausnahme auslöst. `NULL`wenn es keine Ausnahmen gibt.

Kopieren von Transaktionsprotokoll-Backups

Rufen Sie die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` auf, um eine Reihe verfügbarer Transaktionsprotokoll-Backups für eine einzelne Datenbank in Ihren Amazon-S3-Bucket zu kopieren. Die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` initiiert eine neue Aufgabe zum Kopieren von Transaktionsprotokoll-Backups.

> **Note**
>
> Die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` kopiert die Transaktionsprotokoll-Backups, ohne sie anhand eines `is_log_chain_broken`-Attributs zu validieren. Aus diesem Grund sollten Sie eine ununterbrochene Protokollkette manuell bestätigen, bevor Sie di gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` ausführen. Weitere Erläuterungen finden Sie unter Validierung der Backup-Protokollkette des Transaktionsprotokolls.

Example Verwendung der gespeicherten Prozedur **`rds_tlog_backup_copy_to_S3`**

```
exec msdb.dbo.rds_tlog_backup_copy_to_S3
 @db_name='mydatabasename',
 [@kms_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@backup_file_start_time='2022-09-01 01:00:15'],
 [@backup_file_end_time='2022-09-01 21:30:45'],
 [@starting_lsn=149000000112100001],
 [@ending_lsn=149000000120400001],
 [@rds_backup_starting_seq_id=5],
 [@rds_backup_ending_seq_id=10];
```

Die folgenden Eingabeparameter sind verfügbar:

Parameter	Beschreibung
`@db_name`	Der Datenbankname, der zum Kopieren der Transaktionsprotokoll-Backups angegeben wurde
`@kms_key_arn`	Ein vom Kunden verwalteter KMS Schlüssel. Wenn Sie Ihre DB-Instance mit einem AWS verwalteten KMS Schlüssel verschlüsseln,

Parameter	Beschreibung
	müssen Sie einen vom Kunden verwalteten Schlüssel erstellen . Wenn Sie Ihre DB-Instance mit einem vom Kunden verwalteten Schlüssel verschlüsseln, können Sie denselben KMS Schlüssel verwenden. ARN
`@backup_file_start_time`	Der UTC Zeitstempel, wie er in der `[backup_file_time_utc]` Spalte der `rds_fn_list_tlog_backup_metadata` Funktion angegeben wurde.
`@backup_file_end_time`	Der UTC Zeitstempel, wie er aus der `[backup_file_time_utc]` Spalte der `rds_fn_list_tlog_backup_metadata` Funktion bereitgestellt wurde.
`@starting_lsn`	Die Protokoll-Sequenznummer (LSN), wie sie aus der `[starting_lsn]` Spalte der `rds_fn_list_tlog_backup_metadata` Funktion bereitgestellt wurde
`@ending_lsn`	Die Protokollsequenznummer (LSN), wie sie in der `[ending_lsn]` Spalte der `rds_fn_list_tlog_backup_metadata` Funktion angegeben wurde.
`@rds_backup_starting_seq_id`	Die Protokollsequenz-ID, wie sie in der Spalte `[rds_backup_seq_id]` der Funktion `rds_fn_list_tlog_backup_metadata` angegeben ist.
`@rds_backup_ending_seq_id`	Die Protokollsequenz-ID, wie sie in der Spalte `[rds_backup_seq_id]` der Funktion `rds_fn_list_tlog_backup_metadata` angegeben ist.

Sie können entweder einen Satz von Zeit- oder Sequenz-ID-Parametern angeben. LSN Es ist nur ein Satz von Parametern erforderlich.

Sie können auch nur einen einzelnen Parameter in einem der Sätze angeben. Wenn Sie beispielsweise nur einen Wert für den `backup_file_end_time`-Parameter angeben, werden alle verfügbaren Transaktionsprotokoll-Backupdateien vor diesem Zeitpunkt innerhalb des Sieben-Tage-Limits in Ihren Amazon-S3-Bucket kopiert.

Im Folgenden sind die gültigen Eingabeparameterkombinationen für die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` aufgeführt.

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_start_time='2022-08-23 00:00:00', @backup_file_end_time='2022-08-30 00:00:00';`	Kopiert Transaktionsprotokoll-Backups der letzten sieben Tage und liegt zwischen dem angegebenen Bereich von `backup_file_start_time` und `backup_file_end_time` . In diesem Beispiel kopiert die gespeicherte Prozedur Transaktionsprotokoll-Backups, die zwischen '2022-08-23 00:00:00 'und '2022-08-30 00:00:00' generiert wurden.	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_f`	Kopiert Transaktionsprotokoll-Backups der letzten sieben Tage und beginnt mit dem angegebenen Wert für `backup_file_start_`	

Angegebene Parameter	Erwartetes Ergebnis
`ile_start_time='2022-08-23 00:00:00';`	`time` . In diesem Beispiel kopiert die gespeicherte Prozedur die Transaktionsprotokoll-Backups von '2022-08-23 00:00:00' bis zum letzten Transaktionsprotokoll-Backup.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_end_time='2022-08-30 00:00:00';`	Kopiert Transaktionsprotokoll-Backups der letzten sieben Tage bis zum angegebenen Wert für `backup_file_end_time` . In diesem Beispiel kopiert die gespeicherte Prozedur Transaktionsprotokoll-Backups von '2022-08-23 00:00:00' bis '2022-08-30 00:00:00'.

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007, @ending_lsn = 149000000 0050009;`	Kopiert Transaktionsprotokoll-Backups, die in den letzten sieben Tagen verfügbar sind und zwischen dem angegebenen Bereich von `starting_lsn` und `ending_lsn` liegen. In diesem Beispiel kopiert die gespeicherte Prozedur Transaktionsprotokollsicherungen der letzten sieben Tage mit einem LSN Bereich zwischen 1490000000040007 und 149000000 0050009.	

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007;`	Kopiert Transaktionsprotokoll-Backups, die in den letzten sieben Tagen verfügbar sind, ab dem für `starting_lsn` angegebenen Wert. In diesem Beispiel kopiert die gespeicherte Prozedur Transaktionsprotokollsicherungen von 149000000 0040007 bis zur letzten Transaktionsprotokollsicherung. LSN	

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @ending_lsn=14900000 00050009;`	Kopiert Transaktionsprotokoll-Backups, die in den letzten sieben Tagen verfügbar sind, bis zu dem für `ending_lsn` angegebenen Wert. In diesem Beispiel kopiert die gespeicherte Prozedur die Transaktionsprotokoll-Backups ab der letzten sieben Tage bis zum LSN-Bereich 149000000 0050009.	

Angegebene Parameter	Erwartetes Ergebnis
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000, @rds_backup_ending_seq_id= 5000;`	Kopiert Transaktionsprotokoll-Backups, die in den letzten sieben Tagen verfügbar sind und sich im angegebenen Bereich von `rds_backup_starting_seq_id` bis `rds_backup_ending_seq_id` befinden. In diesem Beispiel kopiert die gespeicherte Prozedur Transaktionsprotokoll-Backups ab den letzten sieben Tagen und innerhalb des angegebenen Sequenz-ID-Bereichs für RDS-Backups, beginnend bei seq_id 2000 bis seq_id 5000.

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000;`	Kopiert Transaktionsprotokoll-Backups, die in den letzten sieben Tagen verfügbar sind, ab dem für `rds_backup_starting_seq_id` angegebenen Wert. In diesem Beispiel kopiert die gespeicherte Prozedur die Transaktionsprotokoll-Backups ab seq_id 2000 bis zum letzten Transaktionsprotokoll-Backup.	

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_ending_seq_id= 5000;`	Kopiert Transaktionsprotokoll-Backups, die in den letzten sieben Tagen verfügbar sind, bis zu dem für `rds_backup_ending_seq_id` angegebenen Wert. In diesem Beispiel kopiert die gespeicherte Prozedur die Transaktionsprotokoll-Backups ab der letzten sieben Tage bis zu seq_id 5000.	

Angegebene Parameter	Erwartetes Ergebnis	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000; @rds_backup_ending_seq_id= 2000;`	Kopiert ein einzelnes Transaktionsprotokoll-Backup mit der angegebenen `rds_backup_starting_seq_id`, sofern innerhalb der letzten sieben Tage verfügbar. Bei diesem Beispiel kopiert die gespeicherte Prozedur ein einzelnes Transaktionsprotokoll-Backup mit einer seq_id von 2000, falls es innerhalb der letzten sieben Tage existiert.	

Validierung der Backup-Protokollkette des Transaktionsprotokolls

Für Datenbanken, die für den Zugriff auf Transaktionsprotokoll-Backups konfiguriert sind, muss die automatische Aufbewahrung von Backups aktiviert sein. Durch die automatische Aufbewahrung von Backups werden die Datenbanken auf der DB-Instance auf das FULL-Wiederherstellungsmodell eingestellt. Um die zeitpunktbezogene Wiederherstellung für eine Datenbank zu unterstützen, sollten Sie das Datenbank-Wiederherstellungsmodell nicht ändern, da dies zu einer fehlerhaften Protokollkette führen kann. Wir empfehlen, die Datenbank auf das FULL-Wiederherstellungsmodell einzustellen.

Um die Protokollkette manuell zu validieren, bevor Sie die Transaktionsprotokoll-Backups kopieren, rufen Sie die `rds_fn_list_tlog_backup_metadata`-Funktion auf und überprüfen Sie die Werte

in der `is_log_chain_broken`-Spalte. Ein Wert von „1" gibt an, dass die Protokollkette zwischen dem aktuellen Protokoll-Backup und dem vorherigen Protokoll-Backup unterbrochen wurde.

Das folgende Beispiel zeigt eine defekte Protokollkette in der Ausgabe der gespeicherten Prozedur `rds_fn_list_tlog_backup_metadata`.

rds_sequence_id	first_lsn	last_lsn	is_log_chain_broken
43	90023	90457	0
44	90457	90985	0
45	90987	92034	1

In einer normalen Protokollkette sollte der Wert der Protokollsequenznummer (LSN) für first_lsn für die angegebene rds_sequence_id dem Wert von last_lsn in der vorherigen rds_sequence_id entsprechen. In dem Bild hat die rds_sequence_id 45 einen first_lsn-Wert von 90987, was nicht mit dem last_lsn-Wert von 90985 für den vorangegangenen Wert von rds_sequence_id 44 übereinstimmt.

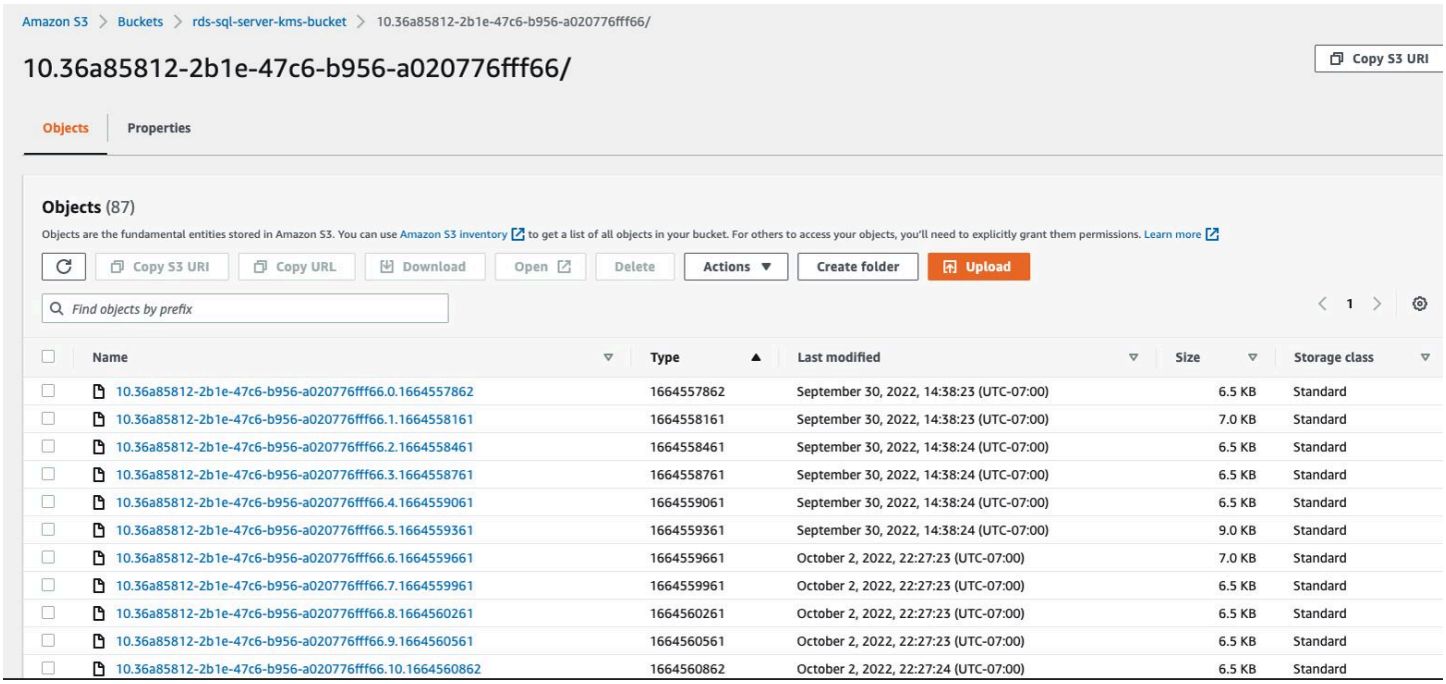
Weitere Informationen zur SQL Server-Transaktionsprotokollarchitektur und zu den Protokollsequenznummern finden Sie unter Logische Transaktionsprotokollarchitektur in der Microsoft SQL Server-Dokumentation.

Amazon S3 Bucket – Ordner und Dateistruktur

Transaktionsprotokoll-Backups haben innerhalb eines Amazon-S3-Buckets die folgende Standardstruktur und Namenskonvention:

- Unter dem `target_s3_arn`-Pfad wird für jede Datenbank ein neuer Ordner mit der Benennungsstruktur `{db_id}.{family_guid}` erstellt.
- Innerhalb des Ordners haben Transaktionsprotokoll-Backups eine Dateinamenstruktur, die `{db_id}.{family_guid}.{rds_backup_seq_id}.{backup_file_epoch}` lautet.
- Sie können die Details von `family_guid,db_id,rds_backup_seq_id and backup_file_epoch` mit der `rds_fn_list_tlog_backup_metadata`-Funktion einsehen.

Das folgende Beispiel veranschaulicht den Ordner und die Dateistruktur einer Reihe von Transaktionsprotokoll-Backups innerhalb eines Amazon-S3-Buckets.



Verfolgen des Status von Aufgaben

Um den Status Ihrer Kopieraufgaben zu verfolgen, rufen Sie die gespeicherte Prozedur `rds_task_status` auf. Wenn Sie keine Parameter angeben, gibt die gespeicherte Prozedur den Status aller Aufgaben zurück.

Example Verwendung:

```
exec msdb.dbo.rds_task_status
  @db_name='database_name',
  @task_id=ID_number;
```

Die folgenden Parameter sind optional:

- `@db_name` – Name der Datenbank, für die der Aufgabenstatus angezeigt werden soll
- `@task_id` – ID der Aufgabe, für die der Aufgabenstatus angezeigt werden soll

Example Auflistung des Status für eine bestimmte Aufgaben-ID:

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example Auflistung des Status für eine bestimmte Datenbank und Aufgabe:

```
exec msdb.dbo.rds_task_status@db_name='my_database',@task_id=5;
```

Example Auflistung aller Aufgaben und ihrer Status für eine bestimmte Datenbank:

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example Auflistung aller Aufgaben und ihrer Status auf der aktuellen DB-Instance:

```
exec msdb.dbo.rds_task_status;
```

Abbrechen einer Aufgabe

Zum Abbrechen einer laufenden Aufgabe rufen Sie die gespeicherte Prozedur `rds_cancel_task` auf.

Example Verwendung:

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

Der folgende Parameter ist erforderlich:

- `@task_id` – ID der abzubrechenden Aufgabe Sie können die ID der Aufgabe durch Aufrufen der gespeicherten Prozedur `rds_task_status` anzeigen.

Weitere Informationen zum Anzeigen und Abbrechen von laufenden Aufgaben finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Fehlerbehebung bei Problemen mit dem Zugriff auf Transaktionsprotokoll-Backups

Die folgenden Probleme können bei der Verwendung der gespeicherten Prozeduren für den Zugriff auf Transaktionsprotokoll-Backups auftreten.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_copy_setup	Backups sind auf dieser DB-Instance deaktiviert. Aktivieren Sie DB-Instance-Backups mit einem Aufbewahrungswert von mindestens „1" und versuchen Sie es erneut.	Automatische Backups sind für die DB-Instance nicht aktiviert.	Die Aufbewahrung von DB-Instance-Backups muss mit einer Aufbewahrungsdauer von mindestens einem Tag aktiviert werden. Weitere Informationen zur Aktivierung von automatischen Backups und zur Konfiguration der Backup-Aufbewahrung finden Sie unter Aufbewahrungszeitraum für Backups.
rds_tlog_copy_setup	Fehler beim Ausführen der gespeicherten Prozedur rds_tlog_copy_setup. Stellen Sie erneut eine Verbindung zum RDS Endpunkt	Es ist ein interner Fehler aufgetreten.	Stellen Sie erneut eine Verbindung zum RDS Endpunkt her und führen Sie die `rds_tlog_copy_setup` gespeicherte Prozedur erneut aus.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
	her und versuchen Sie es erneut.		
rds_tlog_copy_setup	Das Ausführen der gespeicherten Prozedur rds_tlog_backup_copy_setup innerhalb einer Transaktion wird nicht unterstützt. Stellen Sie sicher, dass in der Sitzung keine offenen Transaktionen vorhanden sind, und versuchen Sie es erneut.	Die gespeicherte Prozedur wurde innerhalb einer Transaktion mit `BEGIN` und `END` versucht.	Vermeiden Sie die Verwendung von `BEGIN` und `END` beim Ausführen der gespeicherten Prozedur `rds_tlog_copy_setup` .

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_copy_setup	Der S3-Bucket-Name für den Eingabeparameter `@target_s3_arn` sollte mindestens ein anderes Zeichen als ein Leerzeichen enthalten.	Für den Eingabeparameter `@target_s3_arn` wurde ein falscher Wert angegeben.	Stellen Sie sicher, dass der Eingabeparameter den vollständigen Amazon S3 S3-Bucket `@target_s3_arn` angibtARN.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_copy_setup	Die SQLSERVER_BACKUP_RESTORE -Option ist nicht aktiviert oder wird gerade aktiviert. Aktivieren Sie die Option oder versuchen Sie es später noch einmal.	Die SQLSERVER_BACKUP_RESTORE -Option ist auf der DB-Instance nicht aktiviert oder wurde gerade aktiviert und wartet auf die interne Aktivierung.	Aktivieren Sie die SQLSERVER_BACKUP_RESTORE -Option, wie im Abschnitt „Anforderungen“ angegeben. Warten Sie einige Minuten und führen Sie die gespeicherte Prozedur rds_tlog_copy_setup erneut aus.
rds_tlog_copy_setup	Der Ziel-S3-ARN für den Eingabeparameter @target_s3_arn darf nicht leer oder null sein.	Für den Eingabeparameter NULL wurde ein @target_s3_arn -Wert angegeben oder der Wert wurde nicht angegeben.	Stellen Sie sicher, dass der Eingabeparameter den vollständigen Amazon S3 S3-Bucket @target_s3_arn angibtARN.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_copy_setup	Der Ziel-S3-ARN für den Eingabeparameter `@target_s3_arn` muss mit arn:aws beginnen.	Der Eingabeparameter `@target_s3_arn` wurde ohne `arn:aws` am Anfang angegeben.	Stellen Sie sicher, dass der Eingabeparameter den vollständigen Amazon S3 S3-Bucket `@target_s3_arn` angibtARN.
rds_tlog_copy_setup	Das Ziel-S3 ARN ist bereits auf den angegebenen Wert eingestellt.	Die `rds_tlog_copy_setup` gespeicherte Prozedur wurde zuvor mit einem Amazon S3 S3-Bucket ausgeführt und konfiguriertARN.	Wenn Sie den Wert des Amazon-S3-Buckets für den Zugriff auf Transaktionsprotokoll-Backups ändern möchten, geben Sie einen anderen `target S3 ARN` an.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_copy_setup	Es konnten keine Anmeldeinformationen für die Aktivierung des Zugriffs auf Transaktionsprotokoll-Backups generiert werden. Bestätigen Sie den mit ARN `rds_tlog_copy_setup` angegebenen S3-Pfad und versuchen Sie es später erneut.	Beim Generieren der Anmeldeinformationen für den Zugriff auf Transaktionsprotokoll-Backups ist ein unbekannter Fehler aufgetreten.	Überprüfen Sie die Setup-Konfiguration und versuchen Sie es erneut.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_copy_setup	Sie können die gespeicherte Prozedur rds_tlog_copy_setup nicht ausführen, solange noch Aufgaben ausstehen. Warten Sie, bis die ausstehenden Aufgaben abgeschlossen sind, und versuchen Sie es erneut.	Es können immer nur zwei Aufgaben gleichzeitig ausgeführt werden. Es gibt ausstehende Aufgaben, die darauf warten, abgeschlossen zu werden.	Sehen Sie sich die ausstehenden Aufgaben an und warten Sie, bis diese abgeschlossen sind. Weitere Informationen zur Überwachung des Aufgabenstatus finden Sie unter Verfolgen des Status von Aufgaben.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Eine Aufgabe zum Kopieren der T-Log-Backup-Datei wurde bereits für die Datenbank : %s mit der Aufgaben-ID: %d ausgegeben. Bitte versuchen Sie es später erneut.	Für eine bestimmte Datenbank kann jeweils nur eine Kopieraufgabe ausgeführt werden. Es gibt eine ausstehende Kopieraufgabe, die darauf wartet, abgeschlossen zu werden.	Sehen Sie sich die ausstehenden Aufgaben an und warten Sie, bis diese abgeschlossen sind. Weitere Informationen zur Überwachung des Aufgabenstatus finden Sie unter Verfolgen des Status von Aufgaben.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Mindestens einer dieser drei Parametersätze muss angegeben werden. SET-1 : (@backup_file_start_time, @backup_file_end_time) | SET -2 : (@starting_lsn, @ending_lsn) | SET -3 : (@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Keiner der drei Parametersätze wurde bereitgestellt oder in einem bereitgestellten Parametersatz fehlt ein erforderlicher Parameter.	Sie können jeweils die Zeit-, LSN- oder Sequenz-ID-Parameter angeben. Ein Satz dieser drei Parametersätze ist erforderlich. Weitere Informationen zu erforderlichen Parametern finden Sie unter Kopieren von Transaktionsprotokoll-Backups.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Backups sind auf Ihrer Instance deaktiviert. Bitte aktivieren Sie Backups und versuchen Sie es nach einer Weile erneut.	Automatische Backups sind für die DB-Instance nicht aktiviert.	Weitere Informationen zur Aktivierung von automatischen Backups und zur Konfiguration der Backup-Aufbewahrung finden Sie unter Aufbewahrungszeitraum für Backups.
rds_tlog_backup_copy_to_S3	Die angegebene Datenbank %s kann nicht gefunden werden.	Der für den Eingabeparameter `@db_name` angegebene Wert entspricht keinem Datenbanknamen auf der DB-Instance.	Verwenden Sie den richtigen Datenbanknamen. Führen Sie `SELECT * from sys.databases` aus, um alle Datenbanken nach Namen aufzulisten.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Die gespeicherte Prozedur RDS_TLog_Backup_Copy_To_S3 für Serversystemdatenbanken oder die rdsadmin-Datenbank kann nicht ausgeführt werden. SQL	Der für den Eingabeparameter angegebene Wert entspricht dem Datenbanknamen eines Serversystems oder der Datenbank. `@db_name` SQL RDSAdmin	Die folgenden Datenbanken dürfen nicht für den Zugriff auf Transaktionsprotokoll-Backups verwendet werden: `master, model, msdb, tempdb, RDSAdmin.`
rds_tlog_backup_copy_to_S3	Der Datenbankname für den Eingabeparameter @db_name darf nicht leer oder null sein.	Für den Eingabeparameter `@db_name` wurde ein leerer Wert oder `NULL` angegeben.	Verwenden Sie den richtigen Datenbanknamen. Führen Sie `SELECT * from sys.databases` aus, um alle Datenbanken nach Namen aufzulisten.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Der Aufbewahrungszeitraum für DB-Instance-Backups muss auf mindestens 1 festgelegt werden, um die gespeicherte Prozedur rds_tlog_backup_copy_setup auszuführen.	Automatische Backups sind für die DB-Instance nicht aktiviert.	Weitere Informationen zur Aktivierung von automatischen Backups und zur Konfiguration der Backup-Aufbewahrung finden Sie unter Aufbewahrungszeitraum für Backups.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Fehler beim Ausführen der gespeicherten Prozedur rds_tlog_backup_copy_to_S3. Stellen Sie erneut eine Verbindung zum RDS Endpunkt her und versuchen Sie es erneut.	Es ist ein interner Fehler aufgetreten.	Stellen Sie erneut eine Verbindung zum RDS Endpunkt her und führen Sie die `rds_tlog_backup_copy_to_S3` gespeicherte Prozedur erneut aus.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Es kann nur einer dieser drei Parametersätze angegeben werden. SET-1 : (@backup_file_start_time, @backup_file_end_time) | SET -2 : (@starting_lsn, @ending_lsn) | SET -3 : (@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Es wurden mehrere Parametersätze angegeben.	Sie können jeweils die Zeit-, LSN- oder Sequenz-ID-Parameter angeben. Ein Satz dieser drei Parametersätze ist erforderlich. Weitere Informationen zu erforderlichen Parametern finden Sie unter Kopieren von Transaktionsprotokoll-Backups.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Das Ausführen der gespeicherten Prozedur rds_tlog_backup_copy_to_S3 stored innerhalb einer Transaktion wird nicht unterstützt. Stellen Sie sicher, dass in der Sitzung keine offenen Transaktionen vorhanden sind, und versuchen Sie es erneut.	Die gespeicherte Prozedur wurde innerhalb einer Transaktion mit `BEGIN` und `END` versucht.	Vermeiden Sie die Verwendung von `BEGIN` und `END` beim Ausführen der gespeicherten Prozedur `rds_tlog_backup_copy_to_S3` .

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Die angegebenen Parameter liegen außerhalb des Aufbewahrungszeitraums der Transaktionsprotokoll-Backups. Führen Sie die Funktion rds_fn_list_tlog_backup_metadata aus, um eine Liste der verfügbaren Transaktionsprotokoll-Backup-Dateien zu erhalten.	Für die angegebenen Eingabeparameter sind keine Transaktionsprotokoll-Backup verfügbar, die in das Aufbewahrungsfenster für Kopien passen.	Versuchen Sie es erneut mit einem gültigen Parametersatz. Weitere Informationen zu erforderlichen Parametern finden Sie unter Kopieren von Transaktionsprotokoll-Backups.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Bei der Verarbeitung der Anfrage ist ein Berechtigungsfehler aufgetreten. Stellen Sie sicher, dass sich der Bucket in demselben Konto und derselben Region wie die DB-Instance befindet, und überprüfen Sie die S3-Bucket-Richtlinienberechtigungen anhand der Vorlage in der öffentlichen Dokumentation.	Es wurde ein Problem mit dem bereitgestellten S3-Bucket oder seinen Richtlinienberechtigungen festgestellt.	Vergewissern Sie sich, dass Ihre Einstellungen für den Zugriff auf Transaktionsprotokoll-Backups korrekt sind. Weitere Informationen zu den Einrichtungsanforderungen für Ihren S3-Bucket finden Sie unter Voraussetzungen.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Das Ausführen der `rds_tlog_backup_copy_to_S3` gespeicherten Prozedur auf einer RDS Read Replica-Instanz ist nicht zulässig.	Es wurde versucht, die gespeicherte Prozedur auf einer RDS Read Replica-Instanz auszuführen.	Stellen Sie eine Verbindung mit der RDS primären DB-Instance her, um die `rds_tlog_backup_copy_to_S3` gespeicherte Prozedur auszuführen.
rds_tlog_backup_copy_to_S3	Der Wert LSN für den Eingabeparameter `@starting_lsn` muss kleiner als sein`@ending_lsn` .	Der für den Eingabeparameter `@starting_lsn` angegebene Wert war größer als der Wert, der für den Eingabeparameter `@ending_lsn` angegeben wurde.	Stellen Sie sicher, dass der für den Eingabeparameter `@starting_lsn` angegebene Wert kleiner ist als der Wert, der für den Eingabeparameter `@ending_lsn` angegeben wurde.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` kann nur von den Mitgliedern der `db_owner`-Rolle in der Quelldatenbank ausgeführt werden.	Dem Konto, das versucht, die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` mit dem angegebenen `db_name` auszuführen, wurde die `db_owner`-Rolle nicht zugewiesen.	Stellen Sie sicher, dass das Konto, das die gespeicherte Prozedur ausführt, über die `db_owner`-Rolle mit dem angegebenen `db_name` verfügt.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Die Sequenz-ID für den Eingabeparameter `@rds_backup_starting_seq_id` muss kleiner oder gleich `@rds_backup_ending_seq_id` sein.	Der für den Eingabeparameter `@rds_backup_starting_seq_id` angegebene Wert war größer als der Wert, der für den Eingabeparameter `@rds_backup_ending_seq_id` angegeben wurde.	Stellen Sie sicher, dass der für den Eingabeparameter `@rds_backup_starting_seq_id` angegebene Wert kleiner ist als der Wert, der für den Eingabeparameter `@rds_backup_ending_seq_id` angegeben wurde.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Die RESTORE Option SQLSERVER BACKUP _ _ ist nicht aktiviert oder wird gerade aktiviert. Aktivieren Sie die Option oder versuchen Sie es später noch einmal.	Die `SQLSERVER_BACKUP_RESTORE` -Option ist auf der DB-Instance nicht aktiviert oder wurde gerade aktiviert und wartet auf die interne Aktivierung.	Aktivieren Sie die `SQLSERVER_BACKUP_RESTORE` -Option, wie im Abschnitt „Anforderungen“ angegeben. Warten Sie einige Minuten und führen Sie die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` erneut aus.
rds_tlog_backup_copy_to_S3	Die Startzeit für den Eingabeparameter `@backup_file_start_time` muss kleiner als `@backup_file_end_time` sein.	Der für den Eingabeparameter `@backup_file_start_time` angegebene Wert war größer als der Wert, der für den Eingabeparameter `@backup_file_end_time` angegeben wurde.	Stellen Sie sicher, dass der für den Eingabeparameter `@backup_file_start_time` angegebene Wert kleiner ist als der Wert, der für den Eingabeparameter `@backup_file_end_time` angegeben wurde.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Wir konnten die Anfrage nicht bearbeiten, da wir keinen Zugriff hatten. Bitte überprüfen Sie Ihre Einstellungen und Berechtigungen für die Funktion.	Möglicherweise liegt ein Problem mit den Berechtigungen des Amazon-S3-Buckets vor oder der bereitgestellte Amazon-S3-Bucket befindet sich in einem anderen Konto oder einer anderen Region.	Stellen Sie sicher, dass die Amazon S3 S3-Bucket-Richtlinienberechtigungen berechtigt sind, um den Zugriff zu ermöglichenRDS. Vergewissern Sie sich, dass sich der Amazon-S3-Bucket im gleichen Konto und in derselben Region wie die DB-Instance befindet.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Für Instances, die nicht speicherverschlüsselt sind, können Sie keinen KMS Schlüssel ARN als Eingabeparameter für die gespeicherte Prozedur angeben.	Wenn die Speicherverschlüsselung auf der DB-Instance nicht aktiviert ist, sollte der Eingabeparameter `@kms_key_arn` nicht angegeben werden.	Geben Sie keinen Eingabeparameter für `@kms_key_arn` an.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Sie müssen einen KMS Schlüssel ARN als Eingabeparameter für die gespeicherte Prozedur für speicherverschlüsselte Instances angeben.	Wenn die Speicherverschlüsselung auf der DB-Instance nicht aktiviert ist, muss der Eingabeparameter `@kms_key_arn` angegeben werden.	Geben Sie einen Eingabeparameter für `@kms_key_arn` mit einem Wert an, der dem ARN des Amazon S3 S3-Buckets entspricht, der für Transaktionsprotokoll-Backups verwendet werden soll.

Gespeicherte Prozedur	Fehlermeldung	Problem	Vorschläge für die Fehlerbehebung
rds_tlog_backup_copy_to_S3	Sie müssen die gespeicherte Prozedur `rds_tlog_copy_setup` ausführen und den `@target_s3_arn` festlegen, bevor Sie die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` ausführen.	Der Einrichtungsvorgang für den Zugriff auf Transaktionsprotokoll-Backups wurde nicht abgeschlossen, bevor versucht wurde, die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` auszuführen.	Führen Sie die gespeicherte Prozedur `rds_tlog_copy_setup` aus, bevor Sie die gespeicherte Prozedur `rds_tlog_backup_copy_to_S3` aufrufen. Weitere Informationen zur Ausführung des Einrichtungsvorgangs für den Zugriff auf Transaktionsprotokoll-Backups finden Sie unter Einrichten des Zugriffs auf Transaktionsprotokoll-Backups.

Optionen für die Microsoft SQL Server-Datenbank-Engine

In diesem Abschnitt finden Sie Beschreibungen für Optionen, die für Amazon RDS-Instances verfügbar sind, welche die Microsoft SQL Server-DB-Engine ausführen. Damit diese Optionen aktiviert werden, fügen Sie diese einer Optionsgruppe hinzu und ordnen anschließend die Optionsgruppe Ihrer DB-Instance zu. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.

Wenn Sie nach optionalen Funktionen suchen, die nicht über RDS-Optionsgruppen wie SSL, Microsoft Windows-Authentifizierung oder Amazon S3-Integration hinzugefügt wurden, finden Sie Informationen unter Zusätzliche Funktionen für Microsoft SQL Server auf Amazon RDS.

Amazon RDS unterstützt die folgenden Optionen für Microsoft SQL Server-DB-Instances.

Option	Options-ID	Engine-Editionen
Mit Oracle OLEDB verknüpfte Server	`OLEDB_ORACLE`	SQL Server Enterprise Edition SQL Server Standard Edition
Native Sicherung und Backup	`SQLSERVER_BACKUP_RESTORE`	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition
Transparente Datenverschlüsselung in	`TRANSPARENT_DATA_ENCRYPTION` (RDS-Konsole)	SQL Server 2016—2022 Enterprise Edition SQL Server 2022 Standardausgabe

Option	Options-ID	Engine-Editionen
	TDE(AWS CLI und RDS-API)	
SQLServerprüfung	SQLSERVER_AUDIT	In RDS unterstützen ab SQL Server 2016 alle Editionen von SQL Server Audits auf Serverebene, und Enterprise Edition unterstützt auch Audits auf Datenbank ebene. Ab SQL Server SQL Server 2016 (13.x) SP1 unterstützen alle Editionen sowohl Audits auf Server- als auch auf Datenbank ebene. Weitere Informati onen finden Sie unter SQL Server-Audit (Datenbank-Engine) in der SQL Server-Do kumentation.
SQLServeranalyse-Services	SSAS	SQL Server Enterpris e Edition SQL Server Standard Edition

Option	Options-ID	Engine-Editionen
SQL Server Integration Services	SSIS	SQL Server Enterprise Edition SQL Server Standard Edition
SQLServer Reporting Services	SSRS	SQL Server Enterprise Edition SQL Server Standard Edition
Microsoft Distributed Transaction Coordinator	MSDTC	In RDS unterstützen ab SQL Server 2016 alle Editionen von SQL Server verteilte Transaktionen.

Auflisten der verfügbaren Optionen für Versionen und Editionen von SQL Server

Sie können den `describe-option-group-options` AWS CLI Befehl verwenden, um die verfügbaren Optionen für SQL Server-Versionen und -Editionen sowie die Einstellungen für diese Optionen aufzulisten.

Das folgende Beispiel zeigt die Optionen und Optionseinstellungen für SQL Server 2019 Enterprise Edition. Die Option `--engine-name` ist erforderlich.

```
aws rds describe-option-group-options --engine-name sqlserver-ee --major-engine-version
 15.00
```

Die Ausgabe sieht in etwa folgendermaßen aus:

```
{
    "OptionGroupOptions": [
        {
```

```
            "Name": "MSDTC",
            "Description": "Microsoft Distributed Transaction Coordinator",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": true,
            "DefaultPort": 5000,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": [
                {
                    "SettingName": "ENABLE_SNA_LU",
                    "SettingDescription": "Enable support for SNA LU protocol",
                    "DefaultValue": "true",
                    "ApplyType": "DYNAMIC",
                    "AllowedValues": "true,false",
                    "IsModifiable": true,
                    "IsRequired": false,
                    "MinimumEngineVersionPerAllowedValue": []
                },
        ...

        {
            "Name": "TDE",
            "Description": "SQL Server - Transparent Data Encryption",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": false,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": true,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": []
        }
    ]
}
```

Support für mit Oracle OLEDB verknüpfte Server in Amazon RDS für SQL Server

Durch die Verknüpfung von Servern mit dem Oracle Provider für OLEDB auf RDS für SQL Server können Sie auf externe Datenquellen in einer Oracle-Datenbank zugreifen. Sie können Daten aus Remote-Oracle-Datenquellen lesen und Befehle auf Remote-Oracle-Datenbankservern außerhalb Ihrer DB-Instance von RDS für SQL Server ausführen. Mit Oracle OLEDB verknüpfte Server bieten folgende Möglichkeiten:

- Direkter Zugriff auf andere Datenquellen als SQL Server
- Abfragen verschiedener Oracle-Datenquellen mit derselben Abfrage, ohne die Daten zu verschieben
- Ausgabe verteilter Abfragen, Aktualisierungen, Befehle und Transaktionen für Datenquellen in einem Unternehmens-Ökosystem
- Integration von Verbindungen mit einer Oracle-Datenbank aus der Microsoft Business Intelligence Suite (SSIS, SSRS, SSAS)
- Migration von einer Oracle-Datenbank zu RDS für SQL Server

Sie können einen oder mehrere verknüpfte Server für Oracle auf einer vorhandenen oder einer neuen DB-Instance von RDS für SQL Server aktivieren. Anschließend können Sie externe Oracle-Datenquellen in Ihre DB-Instance integrieren.

Inhalt

- Unterstützte Versionen und Regionen
- Einschränkungen und Empfehlungen
- Aktivieren von verknüpften Servern mit Oracle
  - Erstellen der Optionsgruppe für OLEDB_ORACLE
  - Hinzufügen der OLEDB_ORACLE-Option zur Optionsgruppe
  - Ändern der OLEDB_ORACLE Versionsoption auf eine andere Version
  - Zuordnen der Optionsgruppe zu Ihrer DB-Instance
- Ändern der Eigenschaften des OLEDB-Providers
- Ändern der Eigenschaften des OLEDB-Treibers
- Deaktivieren von verknüpften Servern mit Oracle

Unterstützte Versionen und Regionen

RDS für SQL Server unterstützt mit Oracle OLEDB verknüpfte Server in den folgenden Versionen für SQL Server Standard und Enterprise Edition in allen Regionen:

- SQL Server 2022, alle Versionen
- SQL Server 2019, alle Versionen
- SQL Server 2017, alle Versionen

Mit Oracle OLEDB verknüpfte Server werden für die folgenden Oracle-Database-Versionen unterstützt:

- Oracle Database 21c, alle Versionen
- Oracle Database 19c, alle Versionen
- Oracle Database 18c, alle Versionen

Verbindungsserver mit Oracle OLEDB werden für die folgenden OLEDB-Oracle-Treiberversionen unterstützt:

- 21.7
- 21,16

Einschränkungen und Empfehlungen

Beachten Sie die folgenden Einschränkungen und Empfehlungen, die für mit Oracle OLEDB verknüpfte Server gelten:

- Erlauben Sie Netzwerkverkehr, indem Sie der Sicherheitsgruppe für jede DB-Instance von RDS für SQL Server den entsprechenden TCP-Port hinzufügen. Wenn Sie beispielsweise einen Verbindungsserver zwischen einer EC2 Oracle-DB-Instance und einer RDS for SQL Server-DB-Instance konfigurieren, müssen Sie Datenverkehr von der IP-Adresse der EC2 Oracle-DB-Instance zulassen. Außerdem müssen Sie den Datenverkehr auf dem Port zulassen, den SQL Server zum Überwachen der Datenbankkommunikation verwendet. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
- Führen Sie einen Neustart der DB-Instance von RDS für SQL Server durch, nachdem Sie die `OLEDB_ORACLE`-Option in Ihrer Optionsgruppe aktiviert, deaktiviert oder geändert haben. Der

Optionsgruppenstatus zeigt `pending_reboot` für diese Ereignisse an und ist erforderlich. Bei RDS for SQL Server Multi-AZ-Instances mit aktivierter Spiegelungsoption wird ein Failover erwartet, wenn die Instanz nach der Erstellung AlwaysOn oder Wiederherstellung der neuen Instanz neu gestartet wird.

- Es wird nur die einfache Authentifizierung mit einem Benutzernamen und einem Passwort für die Oracle-Datenquelle unterstützt.
- Open Database Connectivity (ODBC)-Treiber werden nicht unterstützt. Nur die oben aufgeführten OLEDB-Treiberversionen werden unterstützt.
- Verteilte Transaktionen (XA) werden unterstützt. Zur Aktivierung verteilter Transaktionen aktivieren Sie die `MSDTC`-Option in der Optionsgruppe für Ihre DB-Instance und stellen Sie sicher, dass XA-Transaktionen aktiviert sind. Weitere Informationen finden Sie unter Support für Microsoft Distributed Transaction Coordinator RDS für SQL Server.
- Das Erstellen von Datenquellennamen (DSNs) zur Verwendung als Abkürzung für eine Verbindungszeichenfolge wird nicht unterstützt.
- Die OLEDB-Treiberverfolgung wird nicht unterstützt. Sie können erweiterte SQL-Server-Ereignisse verwenden, um OLEDB-Ereignisse zu verfolgen. Weitere Informationen finden Sie unter Set up Extended Events in RDS for SQL Server.
- Der Zugriff auf den Ordner „Catalogs“ (Kataloge) für einen verknüpften Oracle-Server wird mit SQL Server Management Studio (SSMS) nicht unterstützt.

Aktivieren von verknüpften Servern mit Oracle

Aktivieren Sie mit Oracle verknüpfte Server, indem Sie die `OLEDB_ORACLE`-Option Ihrer DB-Instance von RDS für SQL Server hinzufügen. Verwenden Sie den folgenden Prozess:

1. Erstellen Sie eine neue Optionsgruppe oder wählen Sie eine bestehende Optionsgruppe aus.
2. Fügen Sie die Option `OLEDB_ORACLE` zur Optionsgruppe hinzu.
3. Wählen Sie eine Version des zu verwendenden OLEDB-Treibers aus.
4. Ordnen Sie die Optionsgruppe der DB-Instance zu.
5. Wir starten die DB-Instance neu.

Erstellen der Optionsgruppe für OLEDB_ORACLE

Um mit verknüpften Servern mit Oracle zu arbeiten, erstellen Sie eine Optionsgruppe oder ändern Sie eine Optionsgruppe, die der SQL Server-Edition und der Version der DB-Instance entspricht, die Sie

verwenden möchten. Um dieses Verfahren abzuschließen, verwenden Sie den AWS Management Console oder den AWS CLI.

Konsole

Mit der folgenden Prozedur wird eine Optionsgruppe für SQL Server Standard Edition 2019 erstellt.

So erstellen Sie die Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Fenster Create option group (Optionsgruppe erstellen) Folgendes aus:
   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihres AWS Kontos eindeutig ist, z. **`oracle-oledb-se-2019`** B. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.
   b. Geben Sie unter Beschreibung eine kurze Beschreibung der Optionsgruppe ein, z. B. **`OLEDB_ORACLE option group for SQL Server SE 2019`**. Die Beschreibung ist nur zur Information.
   c. Wählen Sie für Engine die Option sqlserver-se aus.
   d. Wählen Sie im Feld Major Engine Version (Engine-Hauptversion) 15.00 aus.
5. Wählen Sie Erstellen aus.

CLI

Mit der folgenden Prozedur wird eine Optionsgruppe für SQL Server Standard Edition 2019 erstellt.

So erstellen Sie die Optionsgruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds create-option-group \
```

```
    --option-group-name oracle-oledb-se-2019 \
    --engine-name sqlserver-se \
    --major-engine-version 15.00 \
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

Windows:

```
aws rds create-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --engine-name sqlserver-se ^
    --major-engine-version 15.00 ^
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

Hinzufügen der **`OLEDB_ORACLE`**-Option zur Optionsgruppe

Verwenden Sie als Nächstes das AWS Management Console oder, AWS CLI um die `OLEDB_ORACLE` Option zu Ihrer Optionsgruppe hinzuzufügen.

Konsole

So fügen Sie die Option OLEDB_ORACLE hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe, die Sie gerade erstellt haben, in diesem Beispiel oracle-oledb-se-2019.
4. Wählen Sie Add option (Option hinzufügen).
5. Wählen Sie unter Option details (Optionsdetails) für Option name (Optionsname) die Option OLEDB_ORACLE aus.
6. Wählen Sie unter Version die Version des OLEDB-Oracle-Treibers aus, den Sie installieren möchten.
7. Wählen Sie unter Scheduling (Planung) aus, ob die Option sofort oder während des nächsten Wartungsfensters hinzugefügt werden soll.
8. Wählen Sie Add option (Option hinzufügen).

CLI

So fügen Sie die Option OLEDB_ORACLE hinzu

- Fügen Sie die Option `OLEDB_ORACLE` zur Optionsgruppe hinzu.

  Example

  FürLinux, odermacOS: Unix

  ```
  aws rds add-option-to-option-group \
      --option-group-name oracle-oledb-se-2019 \
      --options OptionName=OLEDB_ORACLE, OptionVersion=21.16 \
      --apply-immediately
  ```

  Windows:

  ```
  aws rds add-option-to-option-group ^
      --option-group-name oracle-oledb-se-2019 ^
      --options OptionName=OLEDB_ORACLE, OptionVersion=21.16 ^
      --apply-immediately
  ```

Ändern der **`OLEDB_ORACLE`** Versionsoption auf eine andere Version

Um die `OLEDB_ORACLE` Optionsversion in eine andere Version zu ändern, verwenden Sie den AWS Management Console oder den AWS CLI.

Konsole

Um die Option OLEDB_ORACLE zu ändern

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der `OLEDB_ORACLE` Option (oracle-oledb-se-2019 im vorherigen Beispiel).
4. Wählen Sie Modify option (Option ändern) aus.
5. Wählen Sie unter Option details (Optionsdetails) für Option name (Optionsname) die Option OLEDB_ORACLE aus.

6. Wählen Sie unter Version die Version des OLEDB-Oracle-Treibers aus, den Sie verwenden möchten.
7. Wählen Sie unter Planung aus, ob die Option sofort oder im nächsten Wartungsfenster geändert werden soll.
8. Wählen Sie Modify option (Option ändern) aus.

CLI

Um die `OLEDB_ORACLE` Optionsversion zu ändern, verwenden Sie den `rds add-option-to-option-group`AWS CLI Befehl mit der Optionsgruppe und Optionsversion, die Sie verwenden möchten.

Um die Option OLEDB_ORACLE zu ändern

- Example

  FürLinux, oder: macOS Unix

```
aws rds add-option-to-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --options OptionName=OLEDB_ORACLE, OptionVersion=21.7 \
    --apply-immediately
```

  Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --options OptionName=OLEDB_ORACLE, OptionVersion=21.7 ^
    --apply-immediately
```

Zuordnen der Optionsgruppe zu Ihrer DB-Instance

Um die `OLEDB_ORACLE` Optionsgruppe und die Parametergruppe mit Ihrer DB-Instance zu verknüpfen, verwenden Sie das AWS Management Console oder AWS CLI

Konsole

Um die Aktivierung von verknüpften Servern für Oracle abzuschließen, ordnen Sie Ihre `OLEDB_ORACLE`-Optionsgruppe einer neuen oder vorhandenen DB-Instance zu:

- Ordnen Sie sie bei einer neuen DB-Instance zu, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ordnen Sie sie für eine vorhandene DB-Instance zu, indem Sie die Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

CLI

Sie können die `OLEDB_ORACLE`-Optionsgruppe und die Parametergruppe einer neuen oder vorhandenen DB-Instance zuordnen.

So erstellen Sie eine Instance mit der **`OLEDB_ORACLE`**-Optionsgruppe und der Parametergruppe

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Optionsgruppe verwendet haben.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 15.0.4236.7.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name
```

  Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 15.0.4236.7.v1 ^
```

```
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name
```

So ändern Sie eine Instance, um die **OLEDB_ORACLE**-Optionsgruppe zuzuordnen

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name \
    --apply-immediately
```

  Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name ^
    --apply-immediately
```

Ändern der Eigenschaften des OLEDB-Providers

Sie können die Eigenschaften des OLEDB-Providers anzeigen und ändern. Nur der `master`-Benutzer kann diese Aufgabe ausführen. Alle verknüpften Server für Oracle, die auf der DB-Instance erstellt wurden, verwenden dieselben Eigenschaften dieses OLEDB-Providers. Rufen Sie die gespeicherte `sp_MSset_oledb_prop`-Prozedur auf, um die Eigenschaften des OLEDB-Providers zu ändern.

So ändern Sie die Eigenschaften des OLEDB-Providers

```
USE [master]
GO
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'AllowInProcess', 1
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'DynamicParameters', 0
GO
```

Die folgenden Eigenschaften können geändert werden:

Name der Eigenschaft	Empfohlener Wert (1 = Ein, 0 = Aus)	Beschreibung
`Dynamic parameter`	1	Erlaubt SQL-Platzhalter (dargestellt durch '?') in parametrisierten Abfragen.
`Nested queries`	1	Erlaubt verschachtelte `SELECT`-Anweisungen in der `FROM`-Klausel, z. B. Unterabfragen.
`Level zero only`	0	Nur OLEDB-Schnittstellen auf Basisebene werden für den Provider aufgerufen.
`Allow inprocess`	1	Wenn diese Option aktiviert ist, ermöglicht Microsoft SQL Server, dass der Provider als prozessinterner Server instanziiert wird. Legen Sie diese Eigenschaft auf 1 fest, um verknüpfte Oracle-Server zu verwenden.
`Non transacted updates`	0	Wenn ein Wert ungleich Null ist, erlaubt SQL Server Aktualisierungen.
`Index as access path`	False	Wenn ein Wert ungleich Null ist, versucht SQL Server, Indizes des Providers zum Abrufen von Daten zu verwenden.
`Disallow adhoc access`	False	Wenn diese Option festgelegt ist, erlaubt SQL Server keine Ausführung von Passthrough-Abfrag en für den OLEDB-Provider. Diese Option kann

Name der Eigenschaft	Empfohlener Wert (1 = Ein, 0 = Aus)	Beschreibung
		zwar aktiviert werden, es ist jedoch manchmal angebracht, Passthrough-Abfragen auszuführen.
`Supports LIKE operator`	1	Zeigt an, dass der Provider Abfragen mit dem Schlüsselwort LIKE unterstützt.

Ändern der Eigenschaften des OLEDB-Treibers

Sie können die Eigenschaften des OLEDB-Treibers anzeigen und ändern, wenn Sie einen verknüpften Server für Oracle erstellen. Nur der `master`-Benutzer kann diese Aufgabe ausführen. Treibereigenschaften definieren, wie der OLEDB-Treiber Daten verarbeitet, wenn er mit einer Remote-Oracle-Datenquelle arbeitet. Die Treibereigenschaften sind für jeden verknüpften Oracle-Server spezifisch, der auf der DB-Instance erstellt wurde. Rufen Sie die gespeicherte `master.dbo.sp_addlinkedserver`-Prozedur auf, um die Eigenschaften des OLEDB-Treibers zu ändern.

Beispiel: So erstellen Sie einen verknüpften Server und kändern die `FetchSize`-Eigenschaft des OLEDB-Treibers

```
EXEC master.dbo.sp_addlinkedserver
@server = N'Oracle_link2',
@srvproduct=N'Oracle',
@provider=N'OraOLEDB.Oracle',
@datasrc=N'my-oracle-test.cnetsipka.us-west-2.rds.amazonaws.com:1521/ORCL',
@provstr='FetchSize=200'
GO
```

```
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=N'Oracle_link2',
@useself=N'False',
@locallogin=NULL,
@rmtuser=N'master',
@rmtpassword='Test#1234'
GO
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Deaktivieren von verknüpften Servern mit Oracle

Wenn Sie mit Oracle verknüpfte Server deaktivieren möchten, entfernen Sie die `OLEDB_ORACLE`-Option aus der Optionsgruppe.

> **Important**
>
> Wenn Sie die Option entfernen, werden die vorhandenen verknüpften Serverkonfigurationen auf der DB-Instance nicht gelöscht. Sie müssen sie manuell löschen, um sie aus der DB-Instance zu entfernen.
> Sie können die `OLEDB_ORACLE`-Option nach dem Entfernen erneut aktivieren, um die zuvor auf der DB-Instance konfigurierten verknüpften Serverkonfigurationen wiederzuverwenden.

Konsole

Mit dem folgenden Verfahren wird die Option `OLEDB_ORACLE` entfernt.

So entfernen Sie die OLEDB_ORACLE-Option aus der Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der Option `OLEDB_ORACLE` (`oracle-oledb-se-2019` in den vorherigen Beispielen).
4. Wählen Sie Delete option (Option löschen) aus.
5. Wählen Sie unter Deletion options (Löschoptionen) für Options to delete (Zu löschende Optionen) die Option OLEDB_ORACLE aus.
6. Wählen Sie unter Apply immediately (Sofort anwenden) die Option Yes (Ja) aus, um die Option sofort zu löschen, oder No (Nein), um sie während des nächsten Wartungsfensters zu löschen.
7. Wählen Sie Löschen aus.

CLI

Mit dem folgenden Verfahren wird die Option `OLEDB_ORACLE` entfernt.

So entfernen Sie die OLEDB_ORACLE-Option aus der Optionsgruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --options OLEDB_ORACLE \
    --apply-immediately
```

  Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --options OLEDB_ORACLE ^
    --apply-immediately
```

Verbindungsserver mit Teradata ODBC in RDS für SQL Server

Durch die Support von Verbindungsservern mit dem Teradata ODBC-Treiber auf RDS for SQL Server können Sie auf externe Datenquellen in einer Teradata-Datenbank zugreifen. Sie können Daten lesen und Befehle von externen Teradata-Datenbankservern außerhalb Ihrer RDS for SQL Server-Instanz ausführen. Verwenden Sie Linked-Server mit Teradata ODBC, um die folgenden Funktionen zu aktivieren:

- Greifen Sie direkt auf andere Datenquellen als SQL Server zu.
- Abfragen verschiedener Teradata-Datenquellen mit derselben Abfrage, ohne die Daten zu verschieben.
- Führen Sie verteilte Abfragen, Aktualisierungen, Befehle und Transaktionen für Datenquellen im gesamten Unternehmensumfeld durch.
- Integrieren Sie Verbindungen zu einer Teradata-Datenbank aus der Microsoft Business Intelligence Suite (SSIS, SSRS, SSAS).
- Migrieren Sie von einer Teradata-Datenbank zu RDS für SQL Server.

Sie können wählen, ob Sie einen oder mehrere Verbindungsserver für Teradata auf einer vorhandenen oder einer neuen RDS for SQL Server-DB-Instance aktivieren möchten. Anschließend können Sie externe Teradata-Datenquellen in Ihre DB-Instance integrieren.

Themen

- Unterstützte Versionen und Regionen
- Einschränkungen und Empfehlungen
- Überlegungen zur Multi-AZ-Bereitstellung
- Aktivierung von Verbindungsservern mit Teradata
- Verbindungsserver mit Teradata erstellen
- Deaktivierung von Servern, die mit Teradata verbunden sind

Unterstützte Versionen und Regionen

RDS for SQL Server unterstützt Verbindungsserver mit Teradata ODBC in all AWS-Regionen für SQL Server Standard und Enterprise Edition für die folgenden Versionen:

- SQL Server 2022, alle Versionen

- SQL Server 2019, alle Versionen
- SQL Server 2017, alle Versionen

Die folgenden Teradata-Datenbankversionen unterstützen die Verknüpfung mit RDS für SQL Server

- Teradata 17.20, alle Versionen

Einschränkungen und Empfehlungen

Die folgenden Einschränkungen gelten für Verbindungsserver mit Teradata ODBC:

- RDS für SQL Server unterstützt nur die einfache Authentifizierung mit einem Benutzernamen und einem Passwort für die Teradata-Quelle.
- RDS für SQL Server unterstützt nur die Teradata ODBC-Treiberversion 17.20.0.33.
- RDS für SQL Server unterstützt nicht das Erstellen von Datenquellennamen (DSNs), die als Abkürzungen für eine Verbindungszeichenfolge verwendet werden sollen.
- RDS für SQL Server unterstützt die ODBC-Treiberablaufverfolgung nicht. Verwenden Sie SQL Server Extended Events, um ODBC-Ereignisse zu verfolgen. Weitere Informationen finden Sie unter Set up Extended Events in RDS for SQL Server.
- RDS für SQL Server unterstützt bei Verwendung von SQL Server Management Studio (SSMS) keinen Zugriff auf den Katalogordner für einen Teradata-Verbindungsserver.

Beachten Sie bei der Verwendung von Verbindungsservern mit Teradata ODBC die folgenden Empfehlungen:

- Erlauben Sie Netzwerkverkehr, indem Sie der Sicherheitsgruppe für jede DB-Instance von RDS für SQL Server den entsprechenden TCP-Port hinzufügen. Wenn Sie einen Verbindungsserver zwischen einer EC2 Teradata-DB-Instance und einer RDS for SQL Server-DB-Instance konfigurieren, müssen Sie Datenverkehr von der IP-Adresse der Teradata-DB-Instance zulassen. EC2 Sie müssen auch Datenverkehr auf dem Port zulassen, den die RDS for SQL Server-DB-Instance verwendet, um die Datenbankkommunikation abzuhören. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
- Verteilte Transaktionen (XA) werden unterstützt. Um verteilte Transaktionen zu aktivieren, aktivieren Sie die `MSDTC` Option in der Optionsgruppe für Ihre DB-Instance und stellen Sie sicher, dass XA-Transaktionen aktiviert sind. Weitere Informationen finden Sie unter Support für Microsoft Distributed Transaction Coordinator RDS für SQL Server.

- Linked Teradata ODBC unterstützt SSL/TLS, solange es auf dem Teradata Server konfiguriert ist. Weitere Informationen finden Sie unter Aktivieren der TLS-Konnektivität auf Teradata Vantage.

Überlegungen zur Multi-AZ-Bereitstellung

RDS für SQL Server repliziert derzeit keine Verbindungsserver auf den gespiegelten Datenbankserver (oder den sekundären Server der Always-On-Verfügbarkeitsgruppe) in einer Multi-AZ-Bereitstellung. Wenn die Verbindungsserver hinzugefügt werden, bevor die Konfiguration geändert wird, um Spiegelung oder Always-On hinzuzufügen, werden die Verbindungsserver für die vorhandenen Verbindungsserver kopiert.

Alternativ können Sie die Verbindungsserver auf der primären Instanz erstellen, ein Failover auf die Hochverfügbarkeits-Serverinstanz durchführen und dann die Verbindungsserver erneut erstellen, sodass sie sich auf beiden RDS für SQL Server-Instanzen befinden.

Aktivierung von Verbindungsservern mit Teradata

Aktivieren Sie Verbindungsserver mit Teradata, indem Sie die `ODBC_TERADATA` Option zu Ihrer RDS for SQL Server-DB-Instance hinzufügen. Verwenden Sie den folgenden Prozess:

Themen

- Erstellen Sie die Optionsgruppe für ODBC_TERADATA
- Hinzufügen der ODBC_TERADATA-Option zur Optionsgruppe
- Verknüpfen der ODBC_TERADATA Option mit Ihrer DB-Instance

Erstellen Sie die Optionsgruppe für **`ODBC_TERADATA`**

Um mit Verbindungsservern mit Teradata zu arbeiten, erstellen Sie eine Optionsgruppe oder ändern Sie eine Optionsgruppe, die der SQL Server-Edition und Version der DB-Instance entspricht, die Sie verwenden möchten. Verwenden Sie das oder das, um dieses Verfahren abzuschließen. AWS Management Console AWS CLI

Konsole

Gehen Sie wie folgt vor, um eine Optionsgruppe für SQL Server Standard Edition 2019 zu erstellen.

So erstellen Sie die Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Fenster Create option group (Optionsgruppe erstellen) Folgendes aus:
   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihrer Optionsgruppe einzigartig ist AWS-Konto, z. `teradata-odbc-se-2019` B. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.
   b. Geben Sie unter Beschreibung eine kurze Beschreibung der Optionsgruppe ein.
   c. Wählen Sie für Engine die Option sqlserver-se aus.
   d. Wählen Sie im Feld Major Engine Version (Engine-Hauptversion) 15.00 aus.
5. Wählen Sie Erstellen aus.

AWS CLI

Mit der folgenden Prozedur wird eine Optionsgruppe für SQL Server Standard Edition 2019 erstellt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-option-group \
    --option-group-name teradata-odbc-se-2019 \
    --engine-name sqlserver-se \
    --major-engine-version 15.00 \
    --option-group-description "ODBC_TERADATA option group for SQL Server SE 2019"
```

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-option-group ^
    --option-group-name teradata-odbc-se-2019 ^
    --engine-name sqlserver-se ^
    --major-engine-version 15.00 ^
```

```
    --option-group-description "ODBC_TERADATA option group for SQL Server SE 2019"
```

Hinzufügen der **ODBC_TERADATA**-Option zur Optionsgruppe

Verwenden Sie als Nächstes das AWS Management Console oder, AWS CLI um die `ODBC_Teradata` Option zu Ihrer Optionsgruppe hinzuzufügen.

Konsole

Verwenden Sie das folgende Verfahren, um eine Optionsgruppe für SQL Server Standard Edition 2019 zu erstellen.

Um die **ODBC_TERADATA** Option hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Ihre neue Optionsgruppe.
4. Wählen Sie Add option (Option hinzufügen).
5. Unter Optionsdetails:
    a. Wählen Sie ODBC_TERADATA als Optionsnamen.
    b. Für für die Optionsversion. **17.20.33.00**
6. Wählen Sie unter Planung aus, ob die Option sofort oder im nächsten Wartungsfenster hinzugefügt werden soll.
7. Wählen Sie Add option (Option hinzufügen).

AWS CLI

Mit dem folgenden Verfahren wird die `ODBC_TERADATA` Option zu Ihrer Optionsgruppe hinzugefügt.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --option-group-name teradata-odbc-se-2019 \
    --options "OptionName=ODBC_TERADATA,OptionVersion=17.20.33.00" \
    --apply-immediately
```

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name teradata-odbc-se-2019 ^
    --options "OptionName=ODBC_TERADATA,OptionVersion=17.20.33.00" ^
    --apply-immediately
```

Verknüpfen der **`ODBC_TERADATA`** Option mit Ihrer DB-Instance

Verwenden Sie AWS Management Console oder, um die `ODBC_TERADATA` Optionsgruppe Ihrer DB-Instance zuzuordnen AWS CLI.

Konsole

Um die Aktivierung von Verbindungsservern für Teradata abzuschließen, ordnen Sie Ihre Optionsgruppe einer neuen oder vorhandenen DB-Instance zu:

- Bei einer neuen DB-Instance ordnen Sie diese zu, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Bei einer vorhandenen DB-Instance ordnen Sie diese zu, indem Sie die Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Optionsgruppe verwendet haben.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mytestsqlserverteradataodbcinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 15.00 \
    --license-model license-included \
    --allocated-storage 100 \
    --master-username admin \
    --master-user-password password \
    --storage-type gp2 \
```

```
    --option-group-name teradata-odbc-se-2019
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mytestsqlserverteradataodbcinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 15.00 ^
    --license-model license-included ^
    --allocated-storage 100 ^
    --master-username admin ^
    --master-user-password password ^
    --storage-type gp2 ^
    --option-group-name teradata-odbc-se-2019
```

Um eine Instanz zu ändern und die neue Optionsgruppe zuzuordnen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mytestsqlserverteradataodbcinstance \
    --option-group-name teradata-odbc-se-2019 \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mytestsqlserverteradataodbcinstance ^
    --option-group-name teradata-odbc-se-2019 ^
    --apply-immediately
```

Verbindungsserver mit Teradata erstellen

Führen Sie die folgenden Befehle aus, um einen Verbindungsserver mit Teradata zu erstellen:

```
EXECUTE master.dbo.sp_addlinkedserver
    @server = N'LinkedServer_NAME',
    @srvproduct=N'',
    @provider=N'MSDASQL',
    @provstr=N'"PROVIDER=MSDASQL;DRIVER={Teradata Database ODBC Driver 17.20};
                DBCName=Server;UID=user_name;PWD=user_password;
```

```
                UseDataEncryption=YES/NO;SSLMODE=PREFER/ALLOW/DISABLE>;"',
    @catalog='database'
```

```
EXECUTE master.dbo.sp_addlinkedsrvlogin
    @rmtsrvname = N'LinkedServer_NAME',
    @locallogin = NULL ,
    @useself = N'False',
    @rmtuser = N'user_name',
    @rmtpassword = N'user_password'
```

Ein Beispiel für die oben genannten Befehle finden Sie hier:

```
EXECUTE master.dbo.sp_addlinkedserver
    @server = N'LinkedServerToTeradata',
    @srvproduct=N'',
    @provider=N'MSDASQL',
    @provstr=N'"PROVIDER=MSDASQL;DRIVER={Teradata Database ODBC Driver 17.20};
                DBCName=my-teradata-test.cnetsipka.us-west-2.rds.amazonaws.com;
                UID=master;
                PWD=Test#1234;
                UseDataEncryption=YES;
                SSLMODE=PREFER;"',
    @catalog='MyTestTeradataDB'

EXECUTE master.dbo.sp_addlinkedsrvlogin
    @rmtsrvname = N'LinkedServerToTeradata',
    @locallogin = NULL ,
    @useself = N'False',
    @rmtuser = N'master',
    @rmtpassword = N'Test#1234'
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Deaktivierung von Servern, die mit Teradata verbunden sind

Um mit Teradata verknüpfte Server zu deaktivieren, entfernen Sie die `ODBC_TERADATA` Option aus der entsprechenden Optionsgruppe.

> **Important**
>
> Durch das Löschen der Option werden die Verbindungsserverkonfigurationen auf der DB-Instance nicht gelöscht. Sie müssen sie manuell löschen, um sie aus der DB-Instance zu entfernen.
> Sie können die Option ODBC_TERADATA nach dem Entfernen erneut aktivieren, um die Verbindungsserverkonfigurationen wiederzuverwenden, die zuvor auf der DB-Instance konfiguriert wurden.

Konsole

Um die ODBC_TERADATA Option aus der Optionsgruppe zu entfernen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der ODBC_TERADATA Option aus.
4. Wählen Sie Löschen aus.
5. Wählen Sie ODBC_TERADATA unter Löschoptionen die Option Zu löschende Optionen aus.
6. Wählen Sie unter Apply immediately (Sofort anwenden) die Option Yes (Ja) aus, um die Option sofort zu löschen, oder No (Nein), um sie während des nächsten Wartungsfensters zu löschen.
7. Wählen Sie Löschen aus.

AWS CLI

Mit den folgenden Befehlen wird die ODBC_TERADATA Option entfernt.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name teradata-odbc-se-2019 \
    --options ODBC_TERADATA \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds remove-option-from-option-group ^
```

```
    --option-group-name teradata-odbc-se-2019 ^
    --options ODBC_TERADATA ^
    --apply-immediately
```

Unterstützung für native Sicherung und Backup in SQL Server

Durch die Verwendung der nativen Sicherungen und Wiederherstellungen für SQL Server-Datenbanken können Sie eine differenzielle oder vollständige Sicherung Ihrer lokalen Datenbank erstellen und die Sicherungsdateien auf Amazon S3 speichern. Sie können die Wiederherstellung dann auf einer vorhandenen Amazon RDS-DB-Instance durchführen, auf der SQL Server ausgeführt wird. Des Weiteren können Sie eine RDS for SQL Server-Datenbank sichern, in Amazon S3 speichern und an anderen Speicherorten wiederherstellen. Darüber hinaus können Sie die Sicherung auf einem lokalen Server oder einer anderen Amazon RDS DB-Instance mit SQL Server wiederherstellen. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.

Amazon RDS unterstützt die native Sicherung und Backup von Microsoft SQL Server-Datenbanken mithilfe von Dateien für differenzielle und vollständige Sicherungen (BAK-Dateien).

Hinzufügen der Option „Native Sicherung und Backup“

Der allgemeine Prozess zum Hinzufügen der nativen Sicherungs- und Wiederherstellungsoption zu einer DB-Instance ist der folgende:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie die Option `SQLSERVER_BACKUP_RESTORE` zur Optionsgruppe hinzu.
3. Verknüpfen Sie eine AWS Identity and Access Management (IAM)-Rolle mit der Option. Die IAM-Rolle muss Zugriff auf einen S3-Bucket haben, um die Datenbanksicherungen zu speichern.

   Das heißt, die Option muss als Optionseinstellung einen gültigen Amazon-Ressourcennamen (ARN) im Format habe `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) im Allgemeine AWS-Referenz.

   An die IAM-Rolle muss zudem eine Vertrauensbeziehung und eine Berechtigungsrichtlinie angehängt sein. Die Vertrauensbeziehung ermöglicht es RDS, die Rolle zu übernehmen, und die Berechtigungsrichtlinie definiert die Aktionen, welche die Rolle ausführen kann. Weitere Informationen finden Sie unter Manuelles Erstellen einer IAM-Rolle für native Backups und Wiederherstellungen.
4. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie die native Sicherungs- und Wiederherstellungsoption hinzugefügt haben, müssen Sie Ihre DB-Instance nicht neu starten. Sobald die Optionsgruppe aktiv ist, können Sie sofort mit dem Sichern und Wiederherstellen beginnen.

Konsole

So fügen Sie die native Sicherungs- und Wiederherstellungsoption hinzu:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Erstellen Sie eine neue Optionsgruppe oder verwenden Sie eine bestehende Optionsgruppe. Informationen zum Erstellen einer benutzerdefinierten DB-Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.

   Um eine vorhandene Optionsgruppe zu verwenden, gehen Sie zum nächsten Schritt über.
4. Fügen Sie der Optionsgruppe die Option SQLSERVER_BACKUP_RESTORE hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
5. Führen Sie eine der folgenden Aufgaben aus:

   - Um eine vorhandene IAM-Rolle und Amazon S3-Einstellungen zu verwenden, wählen Sie eine vorhandene IAM-Rolle für IAM Role (IAM-Rolle). Wenn Sie eine vorhandene IAM-Rolle verwenden, verwendet RDS die für diese Rolle konfigurierten Amazon S3-Einstellungen.
   - Um eine neue Rolle zu erstellen und neue Amazon S3-Einstellungen zu konfigurieren, gehen Sie wie folgt vor:
     1. Wählen Sie für IAM Role (IAM-Rolle) die Option Create a New Role (Neue Rolle erstellen) aus.
     2. Wählen Sie für S3-Bucket-Name, einen vorhandenen S3-Bucket aus der Liste aus.
     3. Geben Sie unter S3-Ordnerpfad-Präfix (optional) ein Präfix an, das für die in Ihrem Amazon S3-Bucket gespeicherten Dateien verwendet werden soll.

        Dieses Präfix kann einen Dateipfad beinhalten, muss es aber nicht. Wenn Sie ein Präfix angeben, hängt RDS dieses Präfix an alle Sicherungsdateien an. RDS verwendet dann das Präfix während einer Wiederherstellung, um verwandte Dateien zu identifizieren und irrelevante Dateien zu ignorieren. Beispielsweise können Sie den S3-Bucket für andere Zwecke als das Speichern von Backup-Dateien verwenden. In diesem Fall können Sie das

Präfix verwenden, um RDS eine native Sicherung und Backup nur für einen bestimmten Ordner und seine Unterordner durchführen zu lassen.

Wenn Sie das Präfix leer lassen, verwendet RDS kein Präfix, um Backup-Dateien oder wiederherzustellende Dateien zu identifizieren. Infolgedessen versucht RDS bei einer Wiederherstellung mit mehreren Dateien, jede Datei in jedem Ordner des S3-Buckets wiederherzustellen.

4. Für Verschlüsselung aktivieren wählen Sie das Käastchen aus, um die Sicherungsdatei zu verschlüsseln. Lassen Sie das Kontrollkästchen deaktiviert (Standardeinstellung), damit die Sicherungsdatei unverschlüsselt ist.

   Wenn Sie Enable encryption (Verschlüsselung aktivieren) gewählt haben, wählen Sie einen Verschlüsselungscode für AWS KMS key aus. Weitere Informationen zu Verschlüsselungsschlüsseln finden Sie unter Erste Schritte im AWS Key Management Service-Entwicklerhandbuch.

6. Wählen Sie Add option (Option hinzufügen).
7. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Weisen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

CLI

Dieses Verfahren geht von den folgenden Annahmen aus:

- Sie fügen die Option SQLSERVER_BACKUP_RESTORE zu einer bereits vorhandenen Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
- Sie verknüpfen die Option mit einer bereits vorhandenen IAM-Rolle, die Zugriff auf einen S3-Bucket zum Speichern der Sicherungen hat.
- Sie wenden die Optionsgruppe auf eine bereits existierende DB-Instance an. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

So fügen Sie die native Sicherungs- und Wiederherstellungsoption hinzu:

1. Fügen Sie die Option SQLSERVER_BACKUP_RESTORE zur Optionsgruppe hinzu.

   Example

   Für Linux, macOSoder Unix:

   ```
   aws rds add-option-to-option-group \
    --apply-immediately \
    --option-group-name mybackupgroup \
    --options "OptionName=SQLSERVER_BACKUP_RESTORE, \
      OptionSettings=[{Name=IAM_ROLE_ARN,Value=arn:aws:iam::account-id:role/role-
   name}]"
   ```

   Windows:

   ```
   aws rds add-option-to-option-group ^
    --option-group-name mybackupgroup ^
    --options "[{\"OptionName\": \"SQLSERVER_BACKUP_RESTORE\", ^
    \"OptionSettings\": [{\"Name\": \"IAM_ROLE_ARN\", ^
    \"Value\": \"arn:aws:iam::account-id:role/role-name"}]}]" ^
    --apply-immediately
   ```

   > **Note**
   >
   > Bei Verwendung der Windows-Befehlszeile müssen doppelte Anführungszeichen (") im JSON-Code mit einem umgekehrten Schrägstrich (\) als Escape-Zeichen versehen werden.

2. Wenden Sie die Optionsgruppe auf die DB-Instance an.

   Example

   Für Linux, macOSoder Unix:

   ```
   aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --option-group-name mybackupgroup \
    --apply-immediately
   ```

Windows:

```
aws rds modify-db-instance ^
 --db-instance-identifier mydbinstance ^
 --option-group-name mybackupgroup ^
 --apply-immediately
```

Einstellungen für Native Sicherungs- und Wiederherstellungsoption ändern

Nachdem Sie die native Sicherungs- und Wiederherstellungsoption aktiviert haben, können Sie die Einstellungen für die Option ändern. Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung.

Entfernen der Nativen Sicherungs- und Wiederherstellungsoption

Sie können die native Sicherung und Backup deaktivieren, indem Sie die Option aus Ihrer DB-Instance entfernen. Nachdem Sie die native Sicherungs- und Wiederherstellungsoption entfernt haben, müssen Sie Ihre DB-Instance nicht neu starten.

Um die native Sicherungs- und Wiederherstellungsoption aus einer DB-Instance zu entfernen, führen Sie einen der folgenden Schritte aus:

- Entfernen Sie die -Option aus der zugehörigen Optionsgruppe. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, welche die native Sicherungs- und Wiederherstellungsoption nicht enthält. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Support für transparente Datenverschlüsselung im SQL Server

Amazon RDS unterstützt die Verwendung von Transparent Data Encryption (TDE) zur Verschlüsselung von gespeicherten Daten auf Ihren DB-Instances, auf denen Microsoft SQL Server ausgeführt wird. TDEverschlüsselt Daten automatisch, bevor sie in den Speicher geschrieben werden, und entschlüsselt Daten automatisch, wenn die Daten aus dem Speicher gelesen werden.

Amazon RDS unterstützt TDE die folgenden SQL Serverversionen und -Editionen:

- SQLServer 2022 Standard- und Enterprise-Editionen
- SQLServer 2019 Standard- und Enterprise-Editionen
- SQLServer 2017 Enterprise Edition
- SQLServer 2016 Enterprise Edition

Transparent Data Encryption for SQL Server ermöglicht die Verwaltung von Verschlüsselungsschlüsseln mithilfe einer zweistufigen Schlüsselarchitektur. Zum Schutz der Datenverschlüsselungsschlüssel wird ein Zertifikat verwendet, das aus dem Datenbank-Hauptschlüssel generiert wird. Der Datenbankverschlüsselungsschlüssel führt die eigentliche Verschlüsselung und Entschlüsselung der Daten auf der Benutzerdatenbank aus. Amazon RDS sichert und verwaltet den Datenbank-Hauptschlüssel und das TDE Zertifikat.

Transparent Data Encryption wird in Szenarien verwendet, in denen Sie sensible Daten verschlüsseln müssen. So beispielsweise, wenn Sie einem Drittanbieter Datendateien und Sicherungen zur Verfügung stellen möchten, oder in Fällen, oder in denen es um sicherheitsbezogene Fragen zur Einhaltung gesetzlicher Vorschriften geht. Sie können die Systemdatenbanken für SQL Server nicht verschlüsseln, z. B. die `master` Datenbanken `model` oder.

Eine ausführliche Diskussion über Transparent Data Encryption ist nicht Gegenstand dieses Leitfadens, aber Sie sollten unbedingt die Sicherheitsstärken und -schwächen der einzelnen Verschlüsselungsalgorithmen und Schlüssel verstehen. Informationen zur transparenten Datenverschlüsselung für SQL Server finden Sie unter Transparente Datenverschlüsselung (TDE) in der Microsoft-Dokumentation.

Themen

- Wird TDE RDS für den SQL Server aktiviert
- Daten auf dem RDS Server verschlüsseln SQL
- Sichern und Wiederherstellen von TDE-Zertifikaten in RDS for SQL Server

- TDEZertifikate für lokale Datenbanken sichern und wiederherstellen
- Ausschalten TDE RDS für SQL Server

Wird TDE RDS für den SQL Server aktiviert

Um die transparente Datenverschlüsselung RDS für eine SQL Server-DB-Instance zu aktivieren, geben Sie die TDE Option in einer RDS Optionsgruppe an, die dieser DB-Instance zugeordnet ist:

1. Stellen Sie fest, ob Ihre DB-Instance bereits mit einer Optionsgruppe verknüpft ist, die über TDE diese Option verfügt. Um die Optionsgruppe anzuzeigen, der eine DB-Instance zugeordnet ist, verwenden Sie die RDS Konsole, den describe-db-instance AWS CLI Befehl oder die API Operation escribeDBInstancesD.
2. Wenn die DB-Instance keiner Optionsgruppe zugeordnet ist, die TDE aktiviert wurde, haben Sie zwei Möglichkeiten. Sie können eine Optionsgruppe erstellen und die TDE Option hinzufügen, oder Sie können die zugehörige Optionsgruppe ändern, um sie hinzuzufügen.

   > **Note**
   >
   > In der RDS Konsole hat die Option einen Namen`TRANSPARENT_DATA_ENCRYPTION`. In der AWS CLI Hand RDS API ist sie benannt`TDE`.

   Weitere Informationen zum Erstellen oder Ändern einer Optionsgruppe finden Sie unter Arbeiten mit Optionsgruppen. Weitere Informationen zum Hinzufügen einer Option zu einer Optionsgruppe finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die DB-Instance der Optionsgruppe zu, die die TDE Option enthält. Weitere Informationen zum Zuordnen einer DB-Instance zu einer Optionsgruppe finden Sie unter Ändern einer Amazon RDS DB-Instance.

Überlegungen zu Optionsgruppen

Die TDE Option ist eine persistente Option. Sie können sie nur dann aus einer Optionsgruppe entfernen, wenn keine DB-Instances und Sicherungen mehr mit der Optionsgruppe verknüpft sind. Nachdem Sie die TDE Option zu einer Optionsgruppe hinzugefügt haben, kann die Optionsgruppe nur DB-Instances zugeordnet werden, die sie verwendenTDE. Weitere Information zu persistenten Optionen in einer Optionsgruppe, finden Sie unter Übersicht über die Optionsgruppen.

Da es sich bei der TDE Option um eine persistente Option handelt, kann es zu einem Konflikt zwischen der Optionsgruppe und einer zugehörigen DB-Instance kommen. In folgenden Situationen können Konflikte auftreten:

- Die aktuelle Optionsgruppe hat die TDE Option, und Sie ersetzen sie durch eine Optionsgruppe, die TDE diese Option nicht hat.
- Sie führen eine Wiederherstellung von einem DB-Snapshot auf eine neue DB-Instance durch, die keine Optionsgruppe hat, die die TDE Option enthält. Weitere Informationen zu diesem Szenario finden Sie unter Überlegungen zu Optionsgruppen.

SQLÜberlegungen zur Serverleistung

Die Verwendung transparenter Datenverschlüsselung kann sich auf die Leistung einer SQL Server-DB-Instance auswirken.

Die Performance für unverschlüsselte Datenbanken kann ebenfalls beeinträchtigt werden, wenn sich die Datenbanken auf einer DB-Instance befinden, die mindestens eine verschlüsselte Datenbank besitzt. Daher empfehlen wir Ihnen, verschlüsselte und unverschlüsselte Datenbanken auf getrennten DB-Instances zu halten.

Daten auf dem RDS Server verschlüsseln SQL

Wenn die TDE Option zu einer Optionsgruppe hinzugefügt wird, RDS generiert Amazon ein Zertifikat, das im Verschlüsselungsprozess verwendet wird. Sie können das Zertifikat dann verwenden, um SQL Anweisungen auszuführen, die Daten in einer Datenbank auf der DB-Instance verschlüsseln.

Im folgenden Beispiel wird das mit RDS -created aufgerufene Zertifikat verwendet, um eine Datenbank `RDSTDECertificateName` mit dem Namen zu verschlüsseln. `myDatabase`

```
---------- Turning on TDE -------------

-- Find an RDS TDE certificate to use
USE [master]
GO
SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'
GO

USE [myDatabase]
GO
-- Create a database encryption key (DEK) using one of the certificates from the
 previous step
```

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE [RDSTDECertificateName]
GO

-- Turn on encryption for the database
ALTER DATABASE [myDatabase] SET ENCRYPTION ON
GO

-- Verify that the database is encrypted
USE [master]
GO
SELECT name FROM sys.databases WHERE is_encrypted = 1
GO
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO
```

Die Zeit, die zum Verschlüsseln einer SQL Serverdatenbank mit benötigt wird, TDE hängt von mehreren Faktoren ab. Dazu gehören die Größe der DB-Instance, ob die Instance bereitgestellten IOPS Speicher verwendet, die Datenmenge und andere Faktoren.

Sichern und Wiederherstellen von TDE-Zertifikaten in RDS for SQL Server

RDS for SQL Server bietet gespeicherte Prozeduren zum Sichern, Wiederherstellen und Löschen von TDE-Zertifikaten. RDS for SQL Server bietet auch eine Funktion zum Anzeigen wiederhergestellter Benutzer-TDE-Zertifikate.

Benutzer-TDE-Zertifikate werden verwendet, um Datenbanken in RDS for SQL Server wiederherzustellen, die lokal sind und TDE aktiviert haben. Diese Zertifikate haben das Präfix `UserTDECertificate_`. Nach dem Wiederherstellen von Datenbanken und bevor sie zur Verwendung verfügbar gemacht werden, ändert RDS die Datenbanken, für die TDE aktiviert ist, so, dass RDS-generierte TDE-Zertifikate verwenden werden können. Diese Zertifikate haben das Präfix `RDSTDECertificate`.

Benutzer-TDE-Zertifikate bleiben auf der DB-Instance von RDS for SQL Server gespeichert, es sei denn, Sie entfernen sie mit der gespeicherten Prozedur `rds_drop_tde_certificate`. Weitere Informationen finden Sie unter Entfernen wiederhergestellter TDE-Zertifikate.

Sie können ein Benutzer-TDE-Zertifikat verwenden, um andere Datenbanken aus der Quell-DB-Instance wiederherzustellen. Die wiederherzustellenden Datenbanken müssen dasselbe TDE-Zertifikat verwenden und TDE aktiviert haben. Sie müssen das entsprechende Zertifikat nicht erneut importieren (wiederherstellen).

Themen

- Voraussetzungen
- Einschränkungen
- Sichern eines TDE-Zertifikats
- Wiederherstellen eines TDE-Zertifikats
- Anzeigen wiederhergestellter TDE-Zertifikate
- Entfernen wiederhergestellter TDE-Zertifikate

Voraussetzungen

Bevor Sie TDE-Zertifikate in RDS for SQL Server sichern oder wiederherstellen können, müssen Sie die folgenden Schritte ausführen. Die ersten drei sind in Einrichtung für native Backups und Wiederherstellungen beschrieben.

1. Erstellen Sie Amazon-S3-Buckets zum Speichern von Dateien, die gesichert und wiederhergestellt werden sollen.

   Es wird empfohlen, separate Buckets für Datenbanksicherungen und für TDE-Zertifikatssicherungen zu verwenden.
2. Erstellen Sie eine IAM-Rolle zum Sichern und Wiederherstellen von Dateien.

   Die IAM-Rolle muss sowohl ein Benutzer als auch ein Administrator für AWS KMS key sein.

   Zusätzlich zu den Berechtigungen, die für die native Sicherung und Wiederherstellung von SQL Server erforderlich sind, benötigt die IAM-Rolle die folgenden Berechtigungen:
   - `s3:GetBucketAcl`, `s3:GetBucketLocation` und `s3:ListBucket` in der S3-Bucket-Ressource
   - `s3:ListAllMyBuckets` in der Ressource *
3. Fügen Sie die Option `SQLSERVER_BACKUP_RESTORE` einer Optionsgruppe auf Ihrer DB-Instance hinzu.

   Dies gilt zusätzlich zu der Option `TRANSPARENT_DATA_ENCRYPTION` (`TDE`).
4. Stellen Sie sicher, dass Sie einen symmetrischen KMS-Verschlüsselungsschlüssel zur Verfügung haben. Ihnen stehen folgende Optionen zur Verfügung:
   - Wenn ein KMS-Schlüssel in Ihrem Konto vorhanden ist, können Sie diesen verwenden. Es sind keine weiteren Maßnahmen erforderlich.

- Wenn Sie keinen vorhandenen symmetrischen KMS-Verschlüsselungsschlüssel in Ihrem Konto haben, erstellen Sie einen KMS-Schlüssel, indem Sie den Anweisungen unter Erstellen von Schlüsseln im AWS Key Management Service -Entwicklerhandbuch folgen.

5. Aktivieren Sie die Amazon-S3-Integration, um Dateien zwischen der DB-Instance und Amazon S3 zu übertragen.

   Weitere Informationen zum Aktivieren der Amazon-S3-Integration finden Sie unter Integration einer Amazon RDS for SQL Server-DB-Instance mit Amazon S3.

Einschränkungen

Die Verwendung von gespeicherten Prozeduren zum Sichern und Wiederherstellen von TDE-Zertifikaten unterliegt folgenden Einschränkungen:

- Beide Optionen, sowohl `SQLSERVER_BACKUP_RESTORE` als auch `TRANSPARENT_DATA_ENCRYPTION` (`TDE`), müssen der Optionsgruppe hinzugefügt werden, die mit Ihrer DB-Instance verbunden ist.
- Sicherung und Wiederherstellung von TDE-Zertifikaten werden auf Multi-AZ-DB-Instances nicht unterstützt.
- Backup- und Wiederherstellungsaufgaben für das TDE-Zertifikat können nicht abgebrochen werden.
- Sie können kein Benutzer-TDE-Zertifikat für die TDE-Verschlüsselung einer anderen Datenbank auf Ihrer DB-Instance von RDS for SQL Server verwenden. Sie können damit nur andere Datenbanken von der Quell-DB-Instance wiederherstellen, bei denen TDE aktiviert ist und die dasselbe TDE-Zertifikat verwenden.
- Sie können nur Benutzer-TDE-Zertifikate löschen.
- Die maximale Anzahl von Benutzer-TDE-Zertifikaten, die in RDS unterstützt werden, beträgt 10. Wenn die Anzahl 10 überschreitet, entfernen Sie nicht verwendete TDE-Zertifikate und versuchen Sie es erneut.
- Der Zertifikatsname darf nicht leer oder null sein.
- Beim Wiederherstellen eines Zertifikats darf der Zertifikatname das Schlüsselwort `RDSTDECERTIFICATE` nicht enthalten und muss mit dem Präfix `UserTDECertificate_` beginnen.
- Der `@certificate_name`-Parameter darf nur die folgenden Zeichen enthalten: a–z, 0–9, @, $, # und Unterstrich (_).

- Die Dateierweiterungen für `@certificate_file_s3_arn` muss .cer (Groß-/Kleinschreibung wird nicht berücksichtigt) lauten.
- Die Dateierweiterungen für `@private_key_file_s3_arn` muss .pvk (Groß-/Kleinschreibung wird nicht berücksichtigt) lauten.
- Die S3-Metadaten für die private Schlüsseldatei müssen das Tag `x-amz-meta-rds-tde-pwd` enthalten. Weitere Informationen finden Sie unter TDEZertifikate für lokale Datenbanken sichern und wiederherstellen.

Sichern eines TDE-Zertifikats

Verwenden Sie zum Sichern von TDE-Zertifikaten die gespeicherte Prozedur `rds_backup_tde_certificate`. Es hat die folgende Syntax.

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name |
 RDSTDECertificatetimestamp',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id',
    [@overwrite_s3_files=0|1];
```

Die folgenden Parameter sind erforderlich:

- `@certificate_name` – Der Name des zu sichernden TDE-Zertifikats.
- `@certificate_file_s3_arn` – Der Ziel-Amazon-Ressourcenname (ARN) für die Sicherungsdatei des Zertifikats in Amazon S3.
- `@private_key_file_s3_arn` – Der Ziel-S3-ARN der privaten Schlüsseldatei, die das TDE-Zertifikat sichert.
- `@kms_password_key_arn` – Der ARN des symmetrischen KMS-Schlüssels, der zum Verschlüsseln des Passworts des privaten Schlüssels verwendet wurde.

Der folgende Parameter ist optional:

- `@overwrite_s3_files` – Gibt an, ob das vorhandene Zertifikat und die privaten Schlüsseldateien in S3 überschrieben werden sollen:
  - `0` – Die vorhandenen Dateien werden nicht überschrieben. Dieser Wert ist der Standard.

Wenn Sie `@overwrite_s3_files` auf 0 festlegen, wird ein Fehler ausgegeben, wenn eine Datei bereits vorhanden ist.

- 1 – Eine vorhandene Datei mit dem angegebenen Namen wird überschrieben, auch wenn es sich nicht um eine Sicherungsdatei handelt.

Example Sichern eines TDE-Zertifikats

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='RDSTDECertificate20211115T185333',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE',
    @overwrite_s3_files=1;
```

Wiederherstellen eines TDE-Zertifikats

Verwenden Sie die gespeicherte Prozedur `rds_restore_tde_certificate` zum Wiederherstellen (Importieren) von Benutzer-TDE-Zertifikaten. Es hat die folgende Syntax.

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id';
```

Die folgenden Parameter sind erforderlich:

- `@certificate_name` – Der Name des TDE-Zertifikats, das wiederhergestellt werden soll. Der Name muss mit dem Präfix `UserTDECertificate_` beginnen.
- `@certificate_file_s3_arn` – Der S3-ARN der Sicherungsdatei, die zum Wiederherstellen des TDE-Zertifikats verwendet wurde.
- `@private_key_file_s3_arn` – Der S3-ARN der Sicherungsdatei des privaten Schlüssels, die zum Wiederherstellen des TDE-Zertifikats verwendet wurde.
- `@kms_password_key_arn` – Der ARN des symmetrischen KMS-Schlüssels, der zum Verschlüsseln des Passworts des privaten Schlüssels verwendet wurde.

Example Wiederherstellen eines TDE-Zertifikats

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_myTDEcertificate',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Anzeigen wiederhergestellter TDE-Zertifikate

Verwenden Sie die Funktion `rds_fn_list_user_tde_certificates` zum Wiederherstellen (Importieren) von Benutzer-TDE-Zertifikaten Es hat die folgende Syntax.

```
SELECT * FROM msdb.dbo.rds_fn_list_user_tde_certificates();
```

Die Ausgabe sieht in etwa folgendermaßen aus. Hier werden nicht alle Spalten angezeigt.

name	certif te_id	princi _id	pvt_ke ncrypt _type_ c	issuer me	cert_s al_num	thumbp t	subjec	start_ e	expiry te	pvt_key_l ast_backu p_date
UserTD rtific _tde_c	343	1	ENCRYP _BY_MA R_KEY	AnyCom y Shippi	79 3e 57 a3 69 fd 1d 9e 47 2c 32 67 1d 9c	0x6BB2 341103 80B FE1BA2 C69509 5B5	AnyCom y Shippi	2022-0 5 19:49: 000000	2023-0 5 19:49: 000000	NULL

					ca af					

Entfernen wiederhergestellter TDE-Zertifikate

Verwenden Sie die gespeicherte Prozedur `rds_drop_tde_certificate`, um wiederhergestellte (importierte) Benutzer-TDE-Zertifikate zu löschen, die Sie nicht verwenden. Es hat die folgende Syntax.

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_certificate_name';
```

Der folgende Parameter ist erforderlich:

- `@certificate_name` – Der Name des TDE-Zertifikats, das entfernt werden soll.

Sie können nur wiederhergestellte (importierte) TDE-Zertifikate löschen. Von RDS erstellte Zertifikate können nicht gelöscht werden.

Example Entfernen eines TDE-Zertifikats

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_myTDEcertificate';
```

TDEZertifikate für lokale Datenbanken sichern und wiederherstellen

Sie können TDE Zertifikate für lokale Datenbanken sichern und sie später auf RDS dem Server wiederherstellen. SQL Sie können ein RDS for SQL TDE Server-Zertifikat auch auf einer lokalen DB-Instance wiederherstellen.

Mit dem folgenden Verfahren werden ein TDE Zertifikat und ein privater Schlüssel gesichert. Der private Schlüssel wird mit einem Datenschlüssel verschlüsselt, der aus Ihrem symmetrischen KMS Verschlüsselungsschlüssel generiert wurde.

Um ein lokales Zertifikat zu sichern TDE

1. Generieren Sie den Datenschlüssel mit dem AWS CLI generate-data-keyBefehl.

   ```
   aws kms generate-data-key \
   ```

```
    --key-id my_KMS_key_ID \
    --key-spec AES_256
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
{
"CiphertextBlob": "AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vetng
2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==",
"Plaintext": "U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=",
"KeyId": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-00ee-99ff-88dd-
aa11bb22cc33"
}
```

Sie verwenden die Nur-Text-Ausgabe im nächsten Schritt als Passwort für den privaten Schlüssel.

2. Sichern Sie Ihr TDE Zertifikat, wie im folgenden Beispiel gezeigt.

```
BACKUP CERTIFICATE myOnPremTDEcertificate TO FILE = 'D:\tde-cert-backup.cer'
WITH PRIVATE KEY (
FILE = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\cert-
backup-key.pvk',
ENCRYPTION BY PASSWORD = 'U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=');
```

3. Speichern Sie die Sicherungsdatei des Zertifikats in Ihrem Amazon-S3-Zertifikat-Bucket.
4. Speichern Sie die Sicherungsdatei für den privaten Schlüssel in Ihrem S3-Zertifikat-Bucket mit dem folgenden Tag in den Metadaten der Datei:

   - Schlüssel: `x-amz-meta-rds-tde-pwd`
   - Wert: Der Wert `CiphertextBlob` aus der Generierung des Datenschlüssels (siehe folgendes Beispiel).

     ```
     AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
     XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
     BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vet
     2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==
     ```

Das folgende Verfahren stellt ein RDS for SQL TDE Server-Zertifikat auf einer lokalen DB-Instance wieder her. Sie kopieren und stellen das TDE Zertifikat auf Ihrer Ziel-DB-Instance mithilfe der Zertifikatsicherung, der entsprechenden privaten Schlüsseldatei und des Datenschlüssels wieder her. Das wiederhergestellte Zertifikat wird durch den Datenbank-Hauptschlüssel des neuen Servers verschlüsselt.

Um ein TDE Zertifikat wiederherzustellen

1. Kopieren Sie die Sicherungsdatei des TDE Zertifikats und die Datei mit dem privaten Schlüssel von Amazon S3 auf die Ziel-Instance. Weitere Informationen zum Kopieren von Dateien aus Amazon S3 finden Sie unter Dateien zwischen RDS for SQL Server und Amazon S3 übertragen.
2. Verwenden Sie Ihren KMS Schlüssel, um den ausgegebenen Chiffriertext zu entschlüsseln und den Klartext des Datenschlüssels abzurufen. Der Verschlüsselungstext befindet sich in den S3-Metadaten der Sicherungsdatei für private Schlüssel.

   ```
   aws kms decrypt \
       --key-id my_KMS_key_ID \
       --ciphertext-blob fileb://exampleCiphertextFile | base64 -d \
       --output text \
       --query Plaintext
   ```

   Sie verwenden die Nur-Text-Ausgabe im nächsten Schritt als Passwort für den privaten Schlüssel.
3. Verwenden Sie den folgenden SQL Befehl, um Ihr Zertifikat wiederherzustellen. TDE

   ```
   CREATE CERTIFICATE myOnPremTDEcertificate FROM FILE='D:\tde-cert-backup.cer'
   WITH PRIVATE KEY (FILE = N'D:\tde-cert-key.pvk',
   DECRYPTION BY PASSWORD = 'plain_text_output');
   ```

Weitere Informationen zur KMS Entschlüsselung finden Sie unter decrypt im KMS Abschnitt der AWS CLI Befehlsreferenz.

Nachdem das TDE Zertifikat auf der Ziel-DB-Instance wiederhergestellt wurde, können Sie verschlüsselte Datenbanken mit diesem Zertifikat wiederherstellen.

> **Note**
>
> Sie können dasselbe TDE Zertifikat verwenden, um mehrere SQL Serverdatenbanken auf der Quell-DB-Instance zu verschlüsseln. Um mehrere Datenbanken zu einer Ziel-Instance zu migrieren, kopieren Sie das ihnen zugeordnete TDE Zertifikat nur einmal in die Ziel-Instance.

Ausschalten TDE RDS für SQL Server

Um die Funktion TDE RDS für eine SQL Server-DB-Instance auszuschalten, stellen Sie zunächst sicher, dass keine verschlüsselten Objekte mehr auf der DB-Instance vorhanden sind. Entschlüsseln Sie dazu entweder die Objekte oder entfernen Sie sie. Wenn auf der DB-Instance verschlüsselte Objekte vorhanden sind, können Sie diese Funktion TDE für die DB-Instance nicht deaktivieren. Wenn Sie die Konsole verwenden, um die TDE Option aus einer Optionsgruppe zu entfernen, zeigt die Konsole an, dass sie verarbeitet wird. Zudem wird ein Fehlerereignis erstellt, wenn die Optionsgruppe mit einer verschlüsselten DB-Instance oder einem DB-Snapshot verknüpft ist.

Im folgenden Beispiel wird die TDE Verschlüsselung aus einer Datenbank mit dem Namen entfernt`customerDatabase`.

```
------------- Removing TDE ---------------

USE [customerDatabase]
GO

-- Turn off encryption of the database
ALTER DATABASE [customerDatabase]
SET ENCRYPTION OFF
GO

-- Wait until the encryption state of the database becomes 1. The state is 5
 (Decryption in progress) for a while
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO

-- Drop the DEK used for encryption
DROP DATABASE ENCRYPTION KEY
GO

-- Alter to SIMPLE Recovery mode so that your encrypted log gets truncated
USE [master]
```

```
GO
ALTER DATABASE [customerDatabase] SET RECOVERY SIMPLE
GO
```

Wenn alle Objekte entschlüsselt werden, haben Sie zwei Möglichkeiten:

1. Sie können die DB-Instance so ändern, dass sie einer Optionsgruppe ohne die TDE Option zugeordnet wird.
2. Sie können die TDE Option aus der Optionsgruppe entfernen.

SQLServerprüfung

In Amazon RDS können Sie Microsoft SQL Server-Datenbanken mithilfe des integrierten SQL Serverüberwachungsmechanismus prüfen. Sie können Überwachungen und Überwachungsspezifikationen genau so erstellen, wie Sie dies für lokale Datenbankserver tun.

RDSlädt die abgeschlossenen Audit-Logs mithilfe der von Ihnen angegebenen IAM Rolle in Ihren S3-Bucket hoch. Wenn Sie die Aufbewahrung aktivieren, werden RDS Ihre Audit-Logs für den konfigurierten Zeitraum auf Ihrer DB-Instance gespeichert.

Weitere Informationen finden Sie unter SQLServer Audit (Datenbankmodul) in der Microsoft SQL Server-Dokumentation.

SQLServeraudit mit Datenbankaktivitätsströmen

Sie können Database Activity Streams verwendenRDS, um SQL Server Audit-Ereignisse in die Tools zur Überwachung der Datenbankaktivität von Imperva, McAfee, und IBM zu integrieren. Weitere Informationen zur Überwachung mit Datenbankaktivitätsströmen für RDS SQL Server finden Sie unter Prüfungen in Microsoft SQL Server

Themen

- Support für SQL Server Audit
- Hinzufügen von SQL Server Audit zu den DB-Instance-Optionen
- SQLServer Audit verwenden
- Anzeigen von Audit-Protokollen
- Verwenden von SQL Server Audit mit Multi-AZ-Instances
- Konfigurieren eines S3-Buckets
- Manuelles Erstellen einer IAM-Rolle für SQL Server Audit

Support für SQL Server Audit

In Amazon RDS unterstützen ab SQL Server 2016 alle Editionen von Server Audits auf SQL Serverebene, und die Enterprise Edition unterstützt auch Audits auf Datenbankebene. Ab SQL Server 2016 (13.x) unterstützen alle Editionen sowohl SP1 Audits auf Server- als auch auf Datenbankebene. Weitere Informationen finden Sie unter SQLServer Audit (Datenbank-Engine) in der Serverdokumentation. SQL

RDSunterstützt die Konfiguration der folgenden Optionseinstellungen für SQL Server Audit.

Optionseinstellung	Zulässige Werte	Beschreibung
`IAM_ROLE_ARN`	Ein gültiger Amazon-Ressourcenname (ARN) im Format `arn:aws:iam::` *`account-id`* `:role/`*`role-name`*.	Die ARN IAM Rolle, die Zugriff auf den S3-Bucket gewährt, in dem Sie Ihre Audit-Logs speichern möchten. Weitere Informationen finden Sie unter Amazon Resource Names (ARNs) in der Allgemeine AWS-Referenz.
`S3_BUCKET_ARN`	A gültig ARN im Format `arn:aws:s3:::` *`amzn-s3-demo-bucket`* oder `arn:aws:s3:::` *`amzn-s3-demo-bucket`* `/key-prefix`	Das ARN für den S3-Bucket, in dem Sie Ihre Audit-Logs speichern möchten.
`ENABLE_COMPRESSION`	`true` oder `false`	Steuert die Komprimierung des Überwachungsprotokolls. Standardmäßig ist die Komprimierung aktiviert (auf gesetz `true`).
`RETENTION_TIME`	`0` auf `840`	Die Aufbewahrungszeit (in Stunden), für die SQL Server-Audit-Aufzeichnungen auf Ihrer RDS Instance aufbewahrt werden. Die Aufbewahrung ist standardmäßig aktiviert.

Hinzufügen von SQL Server Audit zu den DB-Instance-Optionen

Die Aktivierung von SQL Server Audit erfordert zwei Schritte: das Aktivieren der Option auf der DB-Instance und das Aktivieren der Funktion innerhalb des SQL Servers. Das Hinzufügen der Option SQL Server Audit zu einer DB-Instance erfolgt wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie alle erforderlichen Optionen hinzu und konfigurieren Sie diese.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie die Option SQL Server Audit hinzugefügt haben, müssen Sie Ihre DB-Instance nicht neu starten. Sobald die Optionsgruppe aktiv ist, können Sie Überwachungen erstellen und Audit-Protokolle in Ihrem S3-Bucket speichern.

Um SQL Server Audit zur Optionsgruppe einer DB-Instance hinzuzufügen und zu konfigurieren

1. Wählen Sie eine der folgenden Optionen aus:
   - Verwenden einer vorhandenen Optionsgruppe.
   - Erstellen einer benutzerdefinierten DB-Optionsgruppe und verwenden der Optionsgruppe. Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die AUDIT Option SQLSERVER_ zur Optionsgruppe hinzu und konfigurieren Sie die Optionseinstellungen. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
   - Wenn Sie bereits über eine IAM Rolle mit den erforderlichen Richtlinien verfügen, können Sie diese Rolle für die Rolle auswählen. IAM Um eine neue IAM Rolle zu erstellen, wählen Sie Neue Rolle erstellen. Informationen zu den erforderlichen Richtlinien finden Sie unter Manuelles Erstellen einer IAM-Rolle für SQL Server Audit.
   - Wenn Sie bereits einen S3-Bucket haben, den Sie verwenden möchten, wählen Sie diesen für Select S3 destination (S3-Ziel auswählen) aus.. Um einen S3-Bucket zu erstellen, wählen Sie Einen neuen S3-Bucket erstellen aus.
   - Lassen die Option Enable Compression (Komprimierung aktivieren) aktiviert, um Überwachungsdateien zu komprimieren. Die Komprimierung ist standardmäßig aktiviert. Deaktivieren Sie das Kontrollkästchen Enable Compression (Komprimierung aktivieren), wenn keine Komprimierung erfolgen soll.
   - Wählen Sie für die Aufbewahrung von Überwachungsdatensätzen auf der DB-Instance die Option Audit log retention (Aufbewahrung von Audit-Protokollen) aus. Geben Sie einen Aufbewahrungszeitraum in Stunden an. Die maximale Aufbewahrungsdauer beträgt 35 Tage.
3. Wenden Sie die Optionsgruppe auf eine neue oder vorhandene DB-Instance an. Wählen Sie eine der folgenden Optionen aus:

- Wenn Sie eine neue DB-Instance erstellen, weisen Sie die Optionsgruppe beim Start der Instance zu.
- Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Ändern der Option „SQLServer-Audit“

Nachdem Sie die Option SQL Server Audit aktiviert haben, können Sie die Einstellungen ändern. Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung.

SQLServer Audit aus den DB-Instance-Optionen entfernen

Sie können die SQL Server-Audit-Funktion ausschalten, indem Sie die Audits deaktivieren und dann die Option löschen.

So wird die Überwachung entfernt

1. Deaktivieren Sie alle Überwachungseinstellungen auf dem SQL Server. Um zu erfahren, wo Audits ausgeführt werden, fragen Sie in den Ansichten des SQL Serversicherheitskatalogs nach. Weitere Informationen finden Sie in der Microsoft SQL Server-Dokumentation unter Sicherheitskatalogansichten.
2. Löschen Sie die Option SQL Server Audit aus der DB-Instance. Wählen Sie eine der folgenden Optionen aus:

   - Löschen Sie die Option SQL Server Audit aus der Optionsgruppe, die die DB-Instance verwendet. Diese Änderung wirkt sich auf alle DB-Instances aus, die dieselbe Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
   - Ändern Sie die DB-Instance und wählen Sie dann eine Optionsgruppe ohne die Option SQL Server Audit aus. Diese Änderung wirkt sich nur auf die DB-Instance aus, die Sie modifizieren. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

3. Nachdem Sie die Option SQL Server Audit aus der DB-Instance gelöscht haben, müssen Sie die Instance nicht neu starten. Entfernen Sie nicht benötigte Überwachungsdateien aus Ihrem S3-Bucket.

SQLServer Audit verwenden

Sie können Serverüberwachungen, Serverüberwachungs-Spezifikationen und Datenbanküberwachungs-Spezifikationen auf dieselbe Art steuern, wie Sie dies auch für Ihre lokalen Datenbankserver tun.

Erstellen von Überwachungen

Sie erstellen Serverüberwachungen auf die gleiche Art wie Sie auch Überwachungen für lokale Datenbankserver erstellen. Informationen zum Erstellen von Serveraudits finden Sie CREATESERVERAUDITin der Microsoft SQL Server-Dokumentation.

Halten Sie sich an die folgenden Beschränkungen, um Fehler zu vermeiden:

- Überschreiten Sie nicht die maximale Anzahl unterstützter Serverüberwachungen von 50 pro Instance.
- Weisen Sie den SQL Server an, Daten in eine Binärdatei zu schreiben.
- Verwenden Sie `RDS_` nicht als Präfix im Serverüberwachungsnamen.
- Legen Sie für `FILEPATH` die Option `D:\rdsdbdata\SQLAudit` fest.
- Geben Sie für `MAXSIZE` eine Größe zwischen 2 MB und 50 MB an.
- Konfigurieren Sie nicht `MAX_ROLLOVER_FILES` oder `MAX_FILES`.
- Konfigurieren Sie den SQL Server nicht so, dass er die DB-Instance herunterfährt, wenn der Prüfdatensatz nicht geschrieben werden kann.

Erstellen von Überwachungsspezifikationen

Sie erstellen Spezifikationen für Server- und Datenbanküberwachungen auf dieselbe Art und Weise, wie Sie dies auch für lokale Datenbankserver tun. Informationen zum Erstellen von Auditspezifikationen finden Sie CREATEDATABASEAUDITSPECIFICATIONin CREATESERVERAUDITSPECIFICATIONund in der Microsoft SQL Server-Dokumentation.

Verwenden Sie `RDS_` nicht als Präfix im Namen der Spezifikation für die Server- oder Datenbanküberwachung, um Fehler zu vermeiden.

Anzeigen von Audit-Protokollen

Ihre Audit-Protokolle werden in gespeicher `D:\rdsdbdata\SQLAudit`.

Nachdem der SQL Server das Schreiben in eine Audit-Protokolldatei abgeschlossen hat und die Datei ihre Größenbeschränkung erreicht hat, RDS lädt Amazon die Datei in Ihren S3-Bucket hoch. Wenn die Aufbewahrung aktiviert ist, RDS verschiebt Amazon die Datei in den Aufbewahrungsordner:`D:\rdsdbdata\SQLAudit\transmitted`.

Informationen zu dem Konfigurieren der Aufbewahrung finden Sie unter Hinzufügen von SQL Server Audit zu den DB-Instance-Optionen.

Überwachungsdatensätze werden auf der DB-Instance aufbewahrt, bis die Audit-Protokolldatei hochgeladen wurde. Sie können die Überwachungsdatensätze anzeigen, wenn Sie den folgenden Befehl ausführen.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\*.sqlaudit'
             , default
             , default )
```

Sie können denselben Befehl zum Anzeigen der Überwachungsdatensätze in Ihrem Aufbewahrungsordner verwenden, indem Sie den Filter in änder `D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit`.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
             , default
             , default )
```

Verwenden von SQL Server Audit mit Multi-AZ-Instances

Bei Multi-AZ-Instances ähnelt der Prozess zum Senden von Audit-Protokolldateien an Amazon S3 dem Prozess bei Single-AZ-Instances. Es gibt jedoch einige wichtige Unterschiede:

- Objekte der Spezifikation der Datenbanküberwachung werden auf allen Knoten repliziert.
- Serverüberwachungen und Spezifikationen für Serverüberwachungen werden nicht auf sekundäre Knoten repliziert. Stattdessen müssen Sie diese manuell erstellen oder ändern.

So werden Serverüberwachungen oder eine Spezifikation für die Serverüberwachung von beiden Knoten erfasst:

1. Erstellen Sie eine Serverüberwachung oder eine Spezifikation für die Serverüberwachung auf dem primären Knoten.
2. Führen Sie einen Failover zum sekundären Knoten durch und erstellen Sie ein Server-Audit oder eine Server-Audit-Spezifikation mit demselben Namen und GUID auf dem sekundären Knoten. Verwenden Sie den `AUDIT_GUID` Parameter, um die anzugebenGUID.

Konfigurieren eines S3-Buckets

Die Audit-Protokolldateien werden automatisch von der DB-Instance in Ihren S3-Bucket hochgeladen. Für den S3-Bucket, den Sie als Ziel für die Überwachungsdateien verwenden, gelten folgende Einschränkungen:

- Sie muss sich in derselben AWS Region wie die DB-Instance befinden.
- Er darf nicht öffentlich zugänglich sein.
- Der Bucket-Besitzer muss auch der IAM Rollenbesitzer sein.
- Ihre IAM Rolle muss über Berechtigungen für den vom Kunden verwalteten KMS Schlüssel verfügen, der mit der serverseitigen S3-Bucket-Verschlüsselung verknüpft ist.

Der Zielschlüssel, der zum Speichern der Daten verwendet wird, unterliegt folgendem Benennungsschema: *`amzn-s3-demo-bucket`*`/key-prefix/instance-name/audit-name/node_file-name.ext`

> **Note**
>
> Sie legen sowohl den Bucket-Namen als auch die Schlüsselpräfixwerte mit der Optionseinstellung `S3_BUCKET_ARN` fest.

Das Schema besteht aus den folgenden Elementen:

- ***`amzn-s3-demo-bucket`*** – Der Name des S3-Buckets.
- **`key-prefix`** – Das benutzerdefinierte Schlüsselpräfix, das Sie für die Audit-Protokolle verwenden möchten.

- **`instance-name`**— Der Name Ihrer RDS Amazon-Instance.
- **`audit-name`** – Der Namen der Überwachung.
- **`node`** – Die Kennung des Knotens, der die Quelle der Audit-Protokolle ist (`node1` oder `node2`). Es gibt einen Knoten für die Single-AZ-Instance und zwei Replikationsknoten für eine Multi-AZ-Instance. Dies sind keine primären und sekundären Knoten, da sich die Rolle für primär und sekundär mit der Zeit ändert. Stattdessen ist die Knoten-ID eine einfache Beschriftung.
  - **`node1`** – Der erste Replikationsknoten (bei Single-AZ gibt es nur einen Knoten).
  - **`node2`** – Der zweite Replikationsknoten (bei Multi-AZ gibt es zwei Knoten).
- **`file-name`** – Der Name der Zieldatei. Der Dateiname wird unverändert vom SQL Server übernommen.
- **`ext`** – Die Erweiterung der Datei (`zip` oder `sqlaudit`):
  - **`zip`** – Bei aktivierter Komprimierung (Standard).
  - **`sqlaudit`** – Bei nicht aktivierter Komprimierung.

Manuelles Erstellen einer IAM-Rolle für SQL Server Audit

Wenn Sie eine neue Option erstellen, werden in der Regel die IAM-Rolle und die IAM-Vertrauensrichtlinie für Sie AWS Management Console erstellt. Sie können jedoch manuell eine neue IAM-Rolle für die Verwendung mit SQL Server Audits erstellen, sodass Sie diese an mögliche weitere Anforderungen anpassen können. Dazu erstellen Sie eine IAM-Rolle und delegieren Berechtigungen, sodass der Amazon RDS-Service Ihren Amazon S3-Bucket verwenden kann. Beim Anlegen dieser IAM-Rolle geben Sie Vertrauens- und Berechtigungsrichtlinien an. Die Vertrauensrichtlinie erlaubt es Amazon RDS, diese Rolle zu übernehmen. Die Berechtigungsrichtlinie definiert die Aktionen, die über diese Rolle ausgeführt werden können. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen für einen AWS Dienst im AWS Identity and Access Management-Benutzerhandbuch.

Sie können anhand der Beispiele in diesem Abschnitt die benötigten Vertrauensbeziehungen und Berechtigungsrichtlinien erstellen.

Das folgende Beispiel zeigt eine Vertrauensbeziehung für SQL Server Audit. Das Service-Prinzipal `rds.amazonaws.com` kommt zum Einsatz, um RDS das Schreiben in den S3-Bucket zu erlauben. EinService-Prinzipal ist eine Kennung, die verwendet wird, um einem Service Berechtigungen zu erteilen. Wann immer Sie auf diese Art Zugriff auf `rds.amazonaws.com` gewähren, erlauben Sie RDS, eine Aktion in Ihrem Namen auszuführen. Weitere Informationen zu Service-Prinzipalen finden Sie unter AWS JSON-Richtlinienelemente: Prinzipal.

Example Vertrauensbeziehung für SQL Server Audit

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Principal": {
                 "Service": "rds.amazonaws.com"
             },
             "Action": "sts:AssumeRole"
         }
     ]
 }
```

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Vertrauensbeziehungen, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Sie können beide globalen Bedingungskontextschlüssel verwenden und der Wert `aws:SourceArn` enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der Wert `aws:SourceAccount` und das Konto im Wert `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Anweisung verwendet werden.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Stellen Sie in der Vertrauensbeziehung sicher, dass Sie den globalen Bedingungskontextschlüssel `aws:SourceArn` mit dem vollständigen Amazon-Ressourcennamen (ARN) der Ressourcen verwenden, die auf die Rolle zugreifen. Stellen Sie bei SQL Server Audit sicher, dass Sie sowohl die DB-Optionsgruppe als auch die DB-Instances einschließen, wie im folgenden Beispiel gezeigt.

Example Vertrauensbeziehung mit dem globalen Bedingungskontextschlüssel für SQL Server Audit

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [

 "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
                        "arn:aws:rds:Region:my_account_ID:og:option_group_name"
                    ]
                }
            }
        }
    ]
}
```

Im folgenden Beispiel einer Berechtigungsrichtlinie für SQL Server Audit, wird ein ARN für den Amazon S3 Bucket angegeben. Sie können ARNs damit ein bestimmtes Konto, einen bestimmten Benutzer oder eine bestimmte Rolle identifizieren, für die Sie Zugriff gewähren möchten. Weitere Informationen zur Verwendung ARNs finden Sie unter Amazon-Ressourcennamen (ARNs).

Example Berechtigungsrichtlinie für SQL Server Audit

JSON

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
```

```
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/key_prefix/*"
        }
    ]
}
```

> **Note**
>
> Die `s3:ListAllMyBuckets` Aktion ist erforderlich, um zu überprüfen, ob dasselbe AWS Konto sowohl den S3-Bucket als auch die SQL Server-DB-Instance besitzt. Die Aktion listet die Namen der Buckets in dem Konto auf.
> S3-Bucket-Namespaces sind global. Wenn Sie Ihren Bucket versehentlich löschen, kann ein anderer Benutzer einen Bucket mit demselben Namen in einem anderen Konto erstellen. Dann werden die SQL Server-Prüfungsdaten in den neuen Bucket geschrieben.

Support für SQL Server Analysis Services in Amazon RDS for SQL Server

Microsoft SQL Server Analysis Services (SSAS) ist Teil der Microsoft Business Intelligence (MSBI) -Suite. SSASist ein Online-Tool zur analytischen Verarbeitung (OLAP) und Data Mining, das auf SQL Server installiert ist. Sie verwenden SSAS es, um Daten zu analysieren, um Geschäftsentscheidungen zu treffen. SSASunterscheidet sich von der relationalen SQL Serverdatenbank, da sie für Abfragen und Berechnungen optimiert SSAS ist, die in einer Business Intelligence-Umgebung üblich sind.

Sie können die Option SSAS für bestehende oder neue DB-Instances aktivieren. Es wird auf derselben DB-Instance wie Ihre Datenbank-Engine installiert. Weitere Informationen zu SSAS finden Sie in der Dokumentation zu Microsoft Analysis Services.

Amazon RDS unterstützt SSAS die SQL Server Standard und Enterprise Editions in den folgenden Versionen:

- Tabellarischer Modus:
  - SQLServer 2019, Version 15.00.4043.16.v1 und höher
  - SQLServer 2017, Version 14.00.3223.3.v1 und höher
  - SQLServer 2016, Version 13.00.5426.0.v1 und höher
- Mehrdimensionaler Modus:
  - SQLServer 2019, Version 15.00.4153.1.v1 und höher
  - SQLServer 2017, Version 14.00.3381.3.v1 und höher
  - SQLServer 2016, Version 13.00.5882.1.v1 und höher

Inhalt

- Einschränkungen
- Aktivieren von SSAS
  - Eine Optionsgruppe erstellen für SSAS
  - Die SSAS Option zur Optionsgruppe hinzufügen
  - Zuordnen der Optionsgruppe zu Ihrer DB-Instance
  - Zulassen des eingehenden Zugriffs auf Ihre VPC Sicherheitsgruppe
  - Aktivieren der Amazon-S3-Integration
- SSASProjekte auf Amazon bereitstellen RDS

- Überwachen des Status einer Bereitstellungsaufgabe
- Bei SSAS Amazon verwenden RDS
  - Einen Windows-authentifizierten Benutzer einrichten für SSAS
  - Hinzufügen eines Domänenbenutzers als Datenbankadministrator
  - Einen SSAS Proxy erstellen
  - Die SSAS Datenbankverarbeitung mithilfe des SQL Server-Agenten planen
  - Widerrufen des SSAS Zugriffs über den Proxy
- Eine SSAS Datenbank sichern
- Eine SSAS Datenbank wiederherstellen
  - Wiederherstellen einer DB-Instance zu einer bestimmten Zeit
- Den SSAS Modus ändern
- Deaktivieren von SSAS
- SSASProbleme beheben

Einschränkungen

Die folgenden Einschränkungen gelten für die Verwendung von on for Server: SSAS RDS SQL

- RDSfor SQL Server unterstützt die Ausführung SSAS im tabellarischen oder mehrdimensionalen Modus. Weitere Informationen finden Sie unter Comparing tabular and multidimensional solutions in der Microsoft-Dokumentation.
- Sie können jeweils nur einen SSAS Modus verwenden. Bevor Sie den Modus ändern, stellen Sie sicher, dass Sie alle SSAS Datenbanken löschen.

  Weitere Informationen finden Sie unter Den SSAS Modus ändern.
- Multi-AZ-Instances werden nicht unterstützt.
- Instanzen müssen selbstverwaltetes Active Directory oder AWS Directory Service for Microsoft Active Directory zur SSAS Authentifizierung verwenden. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.
- Benutzern wird kein SSAS Serveradministratorzugriff gewährt, ihnen kann jedoch Administratorzugriff auf Datenbankebene gewährt werden.
- Der einzige unterstützte Port für den Zugriff SSAS ist 2383.

- Projekte können nicht direkt bereitgestellt werden. Zu diesem Zweck stellen wir eine RDS gespeicherte Prozedur zur Verfügung. Weitere Informationen finden Sie unter SSASProjekte auf Amazon bereitstellen RDS.
- Die Verarbeitung während der Bereitstellung wird nicht unterstützt.
- Die Verwendung von XMLA-Dateien für die Bereitstellung wird nicht unterstützt.
- SSASProjekteingabedateien und Datenbank-Backup-Ausgabedateien können sich nur im `D:\S3` Ordner auf der DB-Instance befinden.

Aktivieren von SSAS

Gehen Sie wie folgt vor, SSAS um Ihre DB-Instance einzuschalten:

1. Erstellen Sie eine neue Optionsgruppe oder wählen Sie eine bestehende Optionsgruppe aus.
2. Fügen Sie die Option `SSAS` zur Optionsgruppe hinzu.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.
4. Erlauben Sie eingehenden Zugriff auf die Sicherheitsgruppe Virtual Private Cloud (VPC) für den SSAS Listener-Port.
5. Aktivieren der Amazon-S3-Integration

Eine Optionsgruppe erstellen für SSAS

Verwenden Sie AWS Management Console oder AWS CLI , um eine Optionsgruppe zu erstellen, die der SQL Server-Engine und der Version der DB-Instance entspricht, die Sie verwenden möchten.

> **Note**
>
> Sie können auch eine vorhandene Optionsgruppe verwenden, wenn sie für die richtige SQL Server-Engine und Version bestimmt ist.

Konsole

Mit dem folgenden Konsolenverfahren wird eine Optionsgruppe für SQL Server Standard Edition 2017 erstellt.

So erstellen Sie die Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Bereich Create option group (Optionsgruppe erstellen) Folgendes aus:
   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihres AWS Kontos eindeutig ist, z. **`ssas-se-2017`** B. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.
   b. Geben Sie unter Beschreibung eine kurze Beschreibung der Optionsgruppe ein, z. B. **`SSAS option group for SQL Server SE 2017`**. Die Beschreibung ist nur zur Information.
   c. Wählen Sie für Engine die Option sqlserver-se aus.
   d. Wählen Sie im Feld Major Engine Version (Engine-Hauptversion) 14.00 aus.
5. Wählen Sie Create (Erstellen) aus.

CLI

Im folgenden CLI Beispiel wird eine Optionsgruppe für SQL Server Standard Edition 2017 erstellt.

So erstellen Sie die Optionsgruppe

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-option-group \
    --option-group-name ssas-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-option-group ^
```

```
    --option-group-name ssas-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

Die SSAS Option zur Optionsgruppe hinzufügen

Verwenden Sie als Nächstes das AWS Management Console oder das AWS CLI , um die SSAS Option zur Optionsgruppe hinzuzufügen.

Konsole

Um die SSAS Option hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus, die Sie gerade erstellt haben.
4. Wählen Sie Add option (Option hinzufügen).
5. Wählen Sie unter Optionsdetails SSASden Optionsnamen aus.
6. Führen Sie unter Optionseinstellungen die folgenden Schritte aus:

    a. Geben Sie für Max memory (Maximaler Speicher) einen Wert im Bereich von 10 bis 80 ein.

    Max. Arbeitsspeicher gibt den oberen Schwellenwert an, SSAS ab dem Speicher aggressiver freigegeben wird, um Platz für laufende Anfragen und auch für neue Anfragen mit hoher Priorität zu schaffen. Die Zahl ist ein Prozentsatz des Gesamtspeichers der DB-Instance. Der erlaubte Wertebereich liegt zwischen 10–80; der Standardwert ist 45.

    b. Wählen Sie für Modus den SSAS Servermodus, Tabellarisch oder Mehrdimensional.

    Wenn Sie die Optionseinstellung Modus nicht sehen, bedeutet dies, dass der mehrdimensionale Modus in Ihrer Region nicht unterstützt wird. AWS Weitere Informationen finden Sie unter Einschränkungen.

    Tabular (Tabellarisch) ist die Standardeinstellung.

    c. Wählen Sie für Sicherheitsgruppen die VPC Sicherheitsgruppe aus, die der Option zugeordnet werden soll.

> **Note**
>
> Der Port für den ZugriffSSAS, 2383, ist vorbelegt.

7. Wählen Sie unter Scheduling (Planung) aus, ob die Option sofort oder während des nächsten Wartungsfensters hinzugefügt werden soll.
8. Wählen Sie Add option (Option hinzufügen).

CLI

Um die Option hinzuzufügen SSAS

1. Erstellen Sie beispielsweise `ssas-option.json` eine JSON Datei mit den folgenden Parametern:

   - `OptionGroupName` – Der Name der Optionsgruppe, die Sie zuvor erstellt oder ausgewählt haben (`ssas-se-2017` im folgenden Beispiel).
   - `Port`— Der Port, über den Sie zugreifenSSAS. Der einzige unterstützte Port ist 2383.
   - `VpcSecurityGroupMemberships`— Mitgliedschaften für VPC Sicherheitsgruppen für Ihre RDS DB-Instance.
   - `MAX_MEMORY`— Der obere Schwellenwert, ab dem Speicher aggressiver freigegeben werden SSAS sollte, um Platz für laufende Anfragen und auch für neue Anfragen mit hoher Priorität zu schaffen. Die Zahl ist ein Prozentsatz des Gesamtspeichers der DB-Instance. Der erlaubte Wertebereich liegt zwischen 10–80; der Standardwert ist 45.
   - `MODE`— Der SSAS Servermodus, entweder oder`Tabular`. `Multidimensional` `Tabular`ist die Standardeinstellung.

     Wenn Sie eine Fehlermeldung erhalten, dass die `MODE` Optionseinstellung nicht gültig ist, bedeutet dies, dass der mehrdimensionale Modus in Ihrer AWS Region nicht unterstützt wird. Weitere Informationen finden Sie unter Einschränkungen.

   Im Folgenden finden Sie ein Beispiel für eine JSON Datei mit SSAS Optionseinstellungen.

```
{
"OptionGroupName": "ssas-se-2017",
```

```
"OptionsToInclude": [
 {
 "OptionName": "SSAS",
 "Port": 2383,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [{"Name":"MAX_MEMORY","Value":"60"},
{"Name":"MODE","Value":"Multidimensional"}]
 }],
"ApplyImmediately": true
}
```

2. Fügen Sie die Option SSAS zur Optionsgruppe hinzu.

   Example

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   aws rds add-option-to-option-group \
       --cli-input-json file://ssas-option.json \
       --apply-immediately
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   aws rds add-option-to-option-group ^
       --cli-input-json file://ssas-option.json ^
       --apply-immediately
   ```

Zuordnen der Optionsgruppe zu Ihrer DB-Instance

Sie können die Konsole oder die verwendenCLI, um die Optionsgruppe Ihrer DB-Instance zuzuordnen.

Konsole

Ordnen Sie Ihre Optionsgruppe einer neuen oder vorhandenen DB-Instance zu:

- Ordnen Sie bei einer neuen DB-Instance die Optionsgruppe der DB-Instance beim Start zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie für eine vorhandene DB-Instance die Instance und ordnen Sie ihr die neue Optionsgruppe zu. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Note**
>
> Wenn Sie eine bestehende Instance verwenden, muss dieser bereits eine Active Directory-Domäne und AWS Identity and Access Management (IAM) -Rolle zugeordnet sein. Wenn Sie eine neue Instanz erstellen, geben Sie eine bestehende Active Directory-Domäne und IAM -Rolle an. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.

CLI

Sie können Ihre Optionsgruppe einer neuen oder vorhandenen DB-Instance zuordnen.

> **Note**
>
> Wenn Sie eine vorhandene Instanz verwenden, muss dieser bereits eine Active Directory-Domäne und IAM Rolle zugeordnet sein. Wenn Sie eine neue Instanz erstellen, geben Sie eine vorhandene Active Directory-Domäne und IAM -Rolle an. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.

So erstellen Sie eine DB-Instance, welche die Optionsgruppe verwendet

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Optionsgruppe verwendet haben.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssasinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
```

```
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssas-se-2017
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssasinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssas-se-2017
```

So ändern Sie eine DB-Instance, um die Optionsgruppe zuzuordnen

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssasinstance \
    --option-group-name ssas-se-2017 \
    --apply-immediately
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssasinstance ^
    --option-group-name ssas-se-2017 ^
    --apply-immediately
```

Zulassen des eingehenden Zugriffs auf Ihre VPC Sicherheitsgruppe

Erstellen Sie eine Regel für eingehenden Datenverkehr für den angegebenen SSAS Listener-Port in der VPC Sicherheitsgruppe, die Ihrer DB-Instance zugeordnet ist. Weitere Informationen zum Einrichten von Sicherheitsgruppen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

Aktivieren der Amazon-S3-Integration

Verwenden Sie die Amazon-S3-Integration, um Modellkonfigurationsdateien für die Bereitstellung auf Ihren Host herunterzuladen. Weitere Informationen finden Sie unter Integration einer Amazon RDS for SQL Server-DB-Instance mit Amazon S3.

SSASProjekte auf Amazon bereitstellen RDS

Wenn aktiviertRDS, können Sie SSAS Projekte nicht direkt mithilfe von SQL Server Management Studio (SSMS) bereitstellen. Verwenden Sie eine RDS gespeicherte Prozedur, um Projekte bereitzustellen.

> **Note**
>
> Die Verwendung von XMLA-Dateien für die Bereitstellung wird nicht unterstützt.

Stellen Sie vor der Bereitstellung von Projekten Folgendes sicher:

- Die Amazon-S3-Integration ist aktiviert. Weitere Informationen finden Sie unter Integration einer Amazon RDS for SQL Server-DB-Instance mit Amazon S3.
- Die `Processing Option`-Konfigurationseinstellung ist auf `Do Not Process` eingestellt. Diese Einstellung bedeutet, dass nach der Bereitstellung keine Verarbeitung erfolgt.
- Sie verfügen über die Dateien *`myssasproject`*`.asdatabase` und *`myssasproject`*`.deploymentoptions`. Sie werden automatisch generiert, wenn Sie das SSAS Projekt erstellen.

Um ein SSAS Projekt bereitzustellen auf RDS

1. Laden Sie die `.asdatabase` (SSASModell-) Datei von Ihrem S3-Bucket auf Ihre DB-Instance herunter, wie im folgenden Beispiel gezeigt. Weitere Informationen zu den Download-Parametern

finden Sie unter Dateien von einem Amazon S3 S3-Bucket auf eine SQL Server-DB-Instance herunterladen.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.asdatabase',
[@rds_file_path='D:\S3\myssasproject.asdatabase'],
[@overwrite_file=1];
```

2. Laden Sie die `.deploymentoptions`-Datei aus Ihrem S3-Bucket in Ihre DB-Instance herunter.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.deploymentoptions',
[@rds_file_path='D:\S3\myssasproject.deploymentoptions'],
[@overwrite_file=1];
```

3. Stellen Sie das Projekt bereit.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_DEPLOY_PROJECT',
@file_path='D:\S3\myssasproject.asdatabase';
```

Überwachen des Status einer Bereitstellungsaufgabe

Rufen Sie die Funktion `rds_fn_task_status` auf, um den Status Ihrer Bereitstellungs- (oder Download)-Aufgabe zu verfolgen. Dazu sind zwei Parameter erforderlich. Der erste Parameter sollte immer sein`NULL`, weil er nicht gilt für`SSAS`. Der zweite Parameter akzeptiert eine Aufgaben-ID.

Um eine Liste aller Aufgaben anzuzeigen, setzen Sie den ersten Parameter auf `NULL` und den zweiten Parameter auf `0`, wie im folgenden Beispiel gezeigt.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Um eine bestimmte Aufgabe zu erhalten, setzen Sie den ersten Parameter auf `NULL` und den zweiten Parameter auf die Aufgaben-ID, wie im folgenden Beispiel gezeigt,

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Die `rds_fn_task_status`-Funktion gibt die folgenden Informationen zurück.

Ausgabeparameter	Beschreibung
`task_id`	Die ID der Aufgabe.
`task_type`	Denn SSAS Aufgaben können die folgenden Aufgabentypen haben: • `SSAS_DEPLOY_PROJECT` • `SSAS_ADD_DB_ADMIN_MEMBER` • `SSAS_BACKUP_DB` • `SSAS_RESTORE_DB`
`database_name`	Gilt nicht für SSAS Aufgaben.
`% complete`	Verlauf der Aufgabe als Prozentwert.
`duration (mins)`	Zeitdauer für die Ausführung der Aufgabe (in Minuten).
`lifecycle`	Der Status der Aufgabe. Die folgenden Status sind möglich: • `CREATED`— Nachdem Sie eine der SSAS gespeicherten Prozeduren aufgerufen haben, wird eine Aufgabe erstellt und ihr Status wird auf gesetzt`CREATED`. • `IN_PROGRESS` – Nach dem Start einer Aufgabe wird der Status auf gesetzt `IN_PROGRESS` . Es kann bis zu fünf Minuten dauern, bis sich der Status von `CREATED` zu `IN_PROGRESS` ändert. • `SUCCESS` – Nach dem Abschluss einer Aufgabe wird der Status auf gesetzt `SUCCESS`.

Ausgabeparameter	Beschreibung
	• ERROR – Wenn eine Aufgabe fehlschlägt, wird der Status auf gesetzt ERROR. Weitere Informationen über den Fehler können Sie der Spalte task_info entnehmen. • CANCEL_REQUESTED – Sobald Sie rds_cancel_task aufrufen, wird der Status der Aufgabe auf CANCEL_REQUESTED gesetzt. • CANCELLED – Nachdem die Aufgabe abgebrochen wurde, wird der Status der Aufgabe auf gesetzt CANCELLED .
task_info	Zusätzliche Informationen über die Aufgabe. Wenn bei der Verarbeitung ein Fehler auftritt, enthält diese Spalte Informationen zu dem Fehler. Weitere Informationen finden Sie unter SSASProbleme beheben.
last_updated	Datum und Uhrzeit der letzten Aktualisierung des Aufgabenstatus.
created_at	Datum und Uhrzeit, an denen die Aufgabe angelegt wurde.
S3_object_arn	Gilt nicht für SSAS Aufgaben.
overwrite_S3_backup_file	Gilt nicht für SSAS Aufgaben.
KMS_master_key_arn	Gilt nicht für SSAS Aufgaben.

Ausgabeparameter	Beschreibung
`filepath`	Gilt nicht für SSAS Aufgaben.
`overwrite_file`	Gilt nicht für SSAS Aufgaben.
`task_metadata`	Mit der SSAS Aufgabe verknüpfte Metadaten.

Bei SSAS Amazon verwenden RDS

Nach der Bereitstellung des SSAS Projekts können Sie die OLAP Datenbank direkt weiterverarbeitenSSMS.

Zur Verwendung SSAS auf RDS

1. Stellen Sie SSAS unter Verwendung des Benutzernamens und des Kennworts für die Active Directory-Domäne eine Verbindung her. SSMS
2. Erweitern Sie Databases (Datenbanken). Die neu bereitgestellte SSAS Datenbank wird angezeigt.
3. Suchen Sie die Verbindungszeichenfolge und aktualisieren Sie den Benutzernamen und das Passwort, um Zugriff auf die SQL Quelldatenbank zu gewähren. Dies ist für die Verarbeitung von SSAS Objekten erforderlich.

    a. Führen Sie für den tabellarischen Modus folgende Schritte aus:

        1. Zeigen Sie die Registerkarte Connections (Verbindungen) an.
        2. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Verbindungsobjekt und wählen Sie dann Properties (Eigenschaften) aus.
        3. Aktualisieren Sie den Benutzernamen und das Passwort in der Verbindungszeichenfolge.

    b. Führen Sie für den multidimensionalen Modus folgende Schritte aus:

        1. Zeigen Sie die Registerkarte Data Sources (Datenquellen) an.
        2. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Datenquellobjekt und wählen Sie dann Properties (Eigenschaften) aus.
        3. Aktualisieren Sie den Benutzernamen und das Passwort in der Verbindungszeichenfolge.

4. Öffnen Sie das Kontextmenü (Rechtsklick) für die SSAS Datenbank, die Sie erstellt haben, und wählen Sie Datenbank verarbeiten.

   Abhängig vom Umfang der Eingabedaten kann der Verarbeitungsvorgang einige Minuten dauern.

Themen

- Einen Windows-authentifizierten Benutzer einrichten für SSAS
- Hinzufügen eines Domänenbenutzers als Datenbankadministrator
- Einen SSAS Proxy erstellen
- Die SSAS Datenbankverarbeitung mithilfe des SQL Server-Agenten planen
- Widerrufen des SSAS Zugriffs über den Proxy

Einen Windows-authentifizierten Benutzer einrichten für SSAS

Der Hauptadministrator (manchmal auch als Hauptnutzer bezeichnet) kann das folgende Codebeispiel verwenden, um eine Windows-authentifizierte Anmeldung einzurichten und die erforderlichen Prozessberechtigungen zu erteilen. Dadurch erhält der Domänenbenutzer die Berechtigung, SSAS Kundenaufgaben auszuführen, S3-Dateiübertragungsverfahren zu verwenden, Anmeldeinformationen zu erstellen und mit dem SQL Server-Agent-Proxy zu arbeiten. Weitere Informationen finden Sie in der Microsoft-Dokumentation unter Anmeldeinformationen (Datenbankmodul) und Erstellen eines SQL Server-Agent-Proxys.

Sie können Windows-authentifizierten Benutzern bei Bedarf einige oder alle der folgenden Berechtigungen erteilen.

Example

```
-- Create a server-level domain user login, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create domain user, if it doesn't already exist
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
```

```
GO

-- Grant necessary privileges to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO

USE [msdb]
GO
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] with grant option
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO
```

Hinzufügen eines Domänenbenutzers als Datenbankadministrator

Sie können einen Domänenbenutzer auf folgende Weise als SSAS Datenbankadministrator hinzufügen:

- Ein Datenbankadministrator kann SSMS damit eine Rolle mit `admin` Rechten erstellen und dann Benutzer zu dieser Rolle hinzufügen.
- Sie können die folgende gespeicherte Prozedur verwenden.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_ADD_DB_ADMIN_MEMBER',
@database_name='myssasdb',
@ssas_role_name='exampleRole',
@ssas_role_member='domain_name\domain_user_name';
```

Die folgenden Parameter sind erforderlich:

- `@task_type`— In diesem Fall der Typ der MSBI Aufgabe`SSAS_ADD_DB_ADMIN_MEMBER`.
- `@database_name`— Der Name der SSAS Datenbank, für die Sie Administratorrechte gewähren.
- `@ssas_role_name`— Der Name der SSAS Datenbankadministratorrolle. Wenn die Rolle noch nicht vorhanden ist, wird sie erstellt.
- `@ssas_role_member`— Der SSAS Datenbankbenutzer, den Sie der Administratorrolle hinzufügen.

Einen SSAS Proxy erstellen

Um die SSAS Datenbankverarbeitung mithilfe des SQL Server-Agenten planen zu können, müssen Sie SSAS Anmeldeinformationen und einen SSAS Proxy erstellen. Führen Sie diese Prozeduren als Windows-authentifizierter Benutzer aus.

Um die Anmeldeinformationen zu erstellen SSAS

- Erstellen Sie die Anmeldeinformationen für den Proxy. Dazu können Sie SSMS oder die folgende SQL Anweisung verwenden.

```
USE [master]
GO
CREATE CREDENTIAL [SSAS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> `IDENTITY` muss eine domänenauthentifizierter Anmeldung sein. Ersetzen Sie *`mysecret`* durch das Passwort für die domänenauthentifizierte Anmeldung.

Um den SSAS Proxy zu erstellen

1. Verwenden Sie die folgende SQL Anweisung, um den Proxy zu erstellen.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSAS_Proxy',@credential_name=N'SSAS_Credential',@description=N''
GO
```

2. Verwenden Sie die folgende SQL Anweisung, um anderen Benutzern Zugriff auf den Proxy zu gewähren.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSAS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Verwenden Sie die folgende SQL Anweisung, um dem SSAS Subsystem Zugriff auf den Proxy zu gewähren.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

So zeigen Sie den Proxy und die Berechtigungserteilungen für den Proxy an

1. Verwenden Sie die folgende SQL Anweisung, um die Empfänger des Proxys anzuzeigen.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Verwenden Sie die folgende SQL Anweisung, um die Subsystem-Zuschüsse anzuzeigen.

```
USE [msdb]
GO
```

```
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Die SSAS Datenbankverarbeitung mithilfe des SQL Server-Agenten planen

Nachdem Sie die Anmeldeinformationen und den Proxy erstellt und SSAS Zugriff auf den Proxy gewährt haben, können Sie einen SQL Server-Agent-Job erstellen, um die SSAS Datenbankverarbeitung zu planen.

Um die SSAS Datenbankverarbeitung zu planen

- Verwenden Sie SSMS oder T-, SQL um den SQL Server-Agent-Job zu erstellen. Im folgenden Beispiel wird T- verwendetSQL. Sie können seinen Job-Zeitplan über SSMS oder T- weiter konfigurierenSQL.
  - Der `@command` Parameter gibt den Befehl XML for Analysis (XMLA) an, der vom SQL Server-Agent-Job ausgeführt werden soll. In diesem Beispiel wird die SSAS mehrdimensionale Datenbankverarbeitung konfiguriert.
  - Der `@server` Parameter gibt den SSAS Zielservernamen des SQL Server-Agent-Jobs an.

    Um den SSAS Dienst innerhalb derselben RDS DB-Instance aufzurufen, in der sich der SQL Server-Agent-Job befindet, verwenden Sie`localhost:2383`.

    Um den SSAS Dienst von außerhalb der RDS DB-Instance aufzurufen, verwenden Sie den RDS Endpunkt. Sie können auch den Kerberos Active Directory (AD) -Endpunkt (*`your-DB-instance-name`*`.`*`your-AD-domain-name`*) verwenden, wenn die RDS DB-Instances über dieselbe Domäne verbunden sind. Stellen Sie bei externen DB-Instances sicher, dass Sie die der RDS DB-Instance zugeordnete VPC Sicherheitsgruppe für eine sichere Verbindung ordnungsgemäß konfigurieren.

  Sie können die Abfrage weiter bearbeiten, um verschiedene XMLA Operationen zu unterstützen. Nehmen Sie Änderungen vor, indem Sie entweder die SQL T-Abfrage direkt ändern oder indem Sie nach der Erstellung des SQL Server-Agent-Jobs die SSMS Benutzeroberfläche verwenden.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'SSAS_Job',
```

```
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=0,
    @notify_level_netsend=0,
    @notify_level_page=0,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver
    @job_name=N'SSAS_Job',
    @server_name = N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'SSAS_Job',
 @step_name=N'Process_SSAS_Object',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_success_step_id=0,
    @on_fail_action=2,
    @on_fail_step_id=0,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
    @command=N'<Batch xmlns="http://schemas.microsoft.com/analysisservices/2003/
engine">
        <Parallel>
            <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/
engine/2" xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/
engine/2/2"
                xmlns:ddl100_100="http://schemas.microsoft.com/
analysisservices/2008/engine/100/100" xmlns:ddl200="http://schemas.microsoft.com/
analysisservices/2010/engine/200"
                xmlns:ddl200_200="http://schemas.microsoft.com/
analysisservices/2010/engine/200/200" xmlns:ddl300="http://schemas.microsoft.com/
analysisservices/2011/engine/300"
                xmlns:ddl300_300="http://schemas.microsoft.com/
analysisservices/2011/engine/300/300" xmlns:ddl400="http://schemas.microsoft.com/
analysisservices/2012/engine/400"
                xmlns:ddl400_400="http://schemas.microsoft.com/
analysisservices/2012/engine/400/400" xmlns:ddl500="http://schemas.microsoft.com/
analysisservices/2013/engine/500"
```

```
                xmlns:ddl500_500="http://schemas.microsoft.com/
analysisservices/2013/engine/500/500">
                <Object>
                    <DatabaseID>Your_SSAS_Database_ID</DatabaseID>
                </Object>
                <Type>ProcessFull</Type>
                <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
            </Process>
        </Parallel>
    </Batch>',
    @server=N'localhost:2383',
    @database_name=N'master',
    @flags=0,
    @proxy_name=N'SSAS_Proxy'
GO
```

Widerrufen des SSAS Zugriffs über den Proxy

Sie können den Zugriff auf das SSAS Subsystem widerrufen und den SSAS Proxy mithilfe der folgenden gespeicherten Prozeduren löschen.

So entziehen Sie den Zugriff und löschen den Proxy

1. Widerrufen des Teilsystemzugriffs.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

2. Widerrufen Sie die für den erteilten Berechtigungen Proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSAS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Löschen Sie den Proxy.

```
USE [msdb]
```

```
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSAS_Proxy'
GO
```

Eine SSAS Datenbank sichern

Sie können SSAS Datenbanksicherungsdateien nur in dem `D:\S3` Ordner auf der DB-Instance erstellen. Verwenden Sie Amazon S3, um die Sicherungsdateien in Ihren S3-Bucket zu verschieben.

Sie können eine SSAS Datenbank wie folgt sichern:

- Ein Domänenbenutzer mit der `admin` Rolle für eine bestimmte Datenbank kann SSMS sie verwenden, um die Datenbank in dem `D:\S3` Ordner zu sichern.

  Weitere Informationen finden Sie unter Hinzufügen eines Domänenbenutzers als Datenbankadministrator.
- Sie können die folgende gespeicherte Prozedur verwenden. Der gespeicherte Prozess unterstützt keine Verschlüsselung.

  ```
  exec msdb.dbo.rds_msbi_task
  @task_type='SSAS_BACKUP_DB',
  @database_name='myssasdb',
  @file_path='D:\S3\ssas_db_backup.abf',
  [@ssas_apply_compression=1],
  [@ssas_overwrite_file=1];
  ```

  Die folgenden Parameter sind erforderlich:

  - `@task_type`— In diesem Fall der Typ der MSBI Aufgabe`SSAS_BACKUP_DB`.
  - `@database_name`— Der Name der SSAS Datenbank, die Sie sichern.
  - `@file_path`— Der Pfad für die SSAS Sicherungsdatei. Die Erweiterung `.abf` ist erforderlich.

  Die folgenden Parameter sind optional:

  - `@ssas_apply_compression`— Ob die SSAS Backup-Komprimierung angewendet werden soll. Gültige Werte sind 1 (Ja) und 0 (Nein).
  - `@ssas_overwrite_file`— Ob die SSAS Backup-Datei überschrieben werden soll. Gültige Werte sind 1 (Ja) und 0 (Nein).

Eine SSAS Datenbank wiederherstellen

Verwenden Sie die folgende gespeicherte Prozedur, um eine SSAS Datenbank aus einer Sicherung wiederherzustellen.

Sie können eine Datenbank nicht wiederherstellen, wenn bereits eine SSAS Datenbank mit demselben Namen vorhanden ist. Die gespeicherte Prozedur zum Wiederherstellen unterstützt keine verschlüsselten Sicherungsdateien.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_RESTORE_DB',
@database_name='mynewssasdb',
@file_path='D:\S3\ssas_db_backup.abf';
```

Die folgenden Parameter sind erforderlich:

- `@task_type`— In diesem Fall der Typ der MSBI Aufgabe`SSAS_RESTORE_DB`.
- `@database_name`— Der Name der neuen SSAS Datenbank, in der Sie wiederherstellen.
- `@file_path`— Der Pfad zur SSAS Sicherungsdatei.

Wiederherstellen einer DB-Instance zu einer bestimmten Zeit

Point-in-time recovery (PITR) gilt nicht für SSAS Datenbanken. Wenn Sie dies tunPITR, sind nur die SSAS Daten im letzten Snapshot vor dem angeforderten Zeitpunkt auf der wiederhergestellten Instanz verfügbar.

Um up-to-date SSAS Datenbanken auf einer wiederhergestellten DB-Instance zu haben

1. Sichern Sie Ihre SSAS Datenbanken in dem `D:\S3` Ordner auf der Quell-Instance.
2. Übertragen Sie die Sicherungsdateien in den S3-Bucket.
3. Übertragen Sie die Sicherungsdateien aus dem S3-Bucket in den Ordner `D:\S3` auf der wiederhergestellten Instance.
4. Führen Sie die gespeicherte Prozedur aus, um die SSAS Datenbanken auf der wiederhergestellten Instanz wiederherzustellen.

   Sie können das SSAS Projekt auch erneut verarbeiten, um die Datenbanken wiederherzustellen.

Den SSAS Modus ändern

Sie können den Modus ändern, in dem die SSAS Ausführung ausgeführt wird, entweder tabellarisch oder mehrdimensional. Um den Modus zu ändern, verwenden Sie AWS Management Console oder, AWS CLI um die Optionseinstellungen in der SSAS Option zu ändern.

> **Important**
>
> Sie können jeweils nur einen SSAS Modus verwenden. Stellen Sie sicher, dass Sie alle SSAS Datenbanken löschen, bevor Sie den Modus ändern. Andernfalls wird eine Fehlermeldung angezeigt.

Konsole

Das folgende RDS Amazon-Konsolenverfahren ändert den SSAS Modus auf Tabular und setzt den `MAX_MEMORY` Parameter auf 70 Prozent.

Um die Option zu ändern SSAS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der Option `SSAS` (`ssas-se-2017` in den vorherigen Beispielen), die Sie ändern möchten.
4. Wählen Sie Modify option (Option ändern) aus.
5. Ändern Sie die Optionseinstellungen:
   a. Geben Sie für Max memory (Maximaler Speicher) **`70`** ein.
   b. Wählen Sie für Mode (Modus) Tabular (Tabellarisch) aus.
6. Wählen Sie Modify option (Option ändern) aus.

AWS CLI

Im folgenden AWS CLI Beispiel wird der SSAS Modus auf Tabular geändert und der `MAX_MEMORY` Parameter auf 70 Prozent gesetzt.

Damit der CLI Befehl funktioniert, stellen Sie sicher, dass Sie alle erforderlichen Parameter angeben, auch wenn Sie sie nicht ändern.

Um die SSAS Option zu ändern

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --option-group-name ssas-se-2017 \
    --options
 "OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,
{Name=MODE,Value=Tabular}]" \
    --apply-immediately
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name ssas-se-2017 ^
    --options
 OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,V
{Name=MODE,Value=Tabular}] ^
    --apply-immediately
```

Deaktivieren von SSAS

Um sie zu deaktivierenSSAS, entfernen Sie die SSAS Option aus ihrer Optionsgruppe.

> **Important**
>
> Bevor Sie die SSAS Option entfernen, löschen Sie Ihre SSAS Datenbanken.
> Wir empfehlen dringend, dass Sie Ihre SSAS Datenbanken sichern, bevor Sie sie löschen und die SSAS Option entfernen.

Konsole

Um die SSAS Option aus ihrer Optionsgruppe zu entfernen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der Option `SSAS` (`ssas-se-2017` in den vorherigen Beispielen), die Sie entfernen möchten.
4. Wählen Sie Delete option (Option löschen) aus.
5. Wählen Sie unter Löschoptionen SSASdie Option Optionen zum Löschen aus.
6. Wählen Sie unter Apply immediately (Sofort anwenden) die Option Yes (Ja) aus, um die Option sofort zu löschen, oder No (Nein), um sie während des nächsten Wartungsfensters zu löschen.
7. Wählen Sie Löschen.

AWS CLI

Um die SSAS Option aus ihrer Optionsgruppe zu entfernen

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssas-se-2017 \
    --options SSAS \
    --apply-immediately
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssas-se-2017 ^
    --options SSAS ^
    --apply-immediately
```

SSASProbleme beheben

Bei der Verwendung können die folgenden Probleme auftretenSSAS.

Problem	Typ	Vorschläge für die Fehlerbehebung
Die SSAS Option konnte nicht konfiguriert werden. Der angeforderte SSAS Modus ist *new_mode*, aber die aktuelle DB-Instance hat *number current_mode* Datenbanken. Löschen Sie die vorhandenen Datenbanken, bevor Sie zu wechseln *new_mode* Modus. Um wieder Zugriff auf *current_mode* Modus zum Löschen der Datenbank: Aktualisieren Sie entweder die aktuelle DB-Optionsgruppe oder fügen Sie eine neue Optionsgruppe mit %s als MODE Optionseinstellungswert für die SSAS Option hinzu.	RDSEreignis	Sie können den SSAS Modus nicht ändern, wenn Sie noch SSAS Datenbanken haben, die den aktuellen Modus verwenden. Löschen Sie die SSAS Datenbanken, und versuchen Sie es erneut.
Die SSAS Option konnte nicht entfernt werden, da *number* vorhandene *mode* Datenbanken. Die SSAS Option kann erst entfernt werden, wenn alle SSAS Datenbanken gelöscht wurden. Fügen Sie die SSAS Option erneut hinzu, löschen Sie alle SSAS Datenbanken und versuchen Sie es erneut.	RDSEreignis	Sie können es nicht ausschaltenSSAS, wenn Sie noch SSAS Datenbanken haben. Löschen Sie die SSAS Datenbanken, und versuchen Sie es erneut.
Die SSAS Option ist nicht aktiviert oder wird gerade aktiviert. Bitte versuchen Sie es später erneut.	RDSgespeicherte Prozedur	Sie können keine SSAS gespeicherten Prozeduren ausführen, wenn die Option ausgeschaltet oder aktiviert ist.

Problem	Typ	Vorschläge für die Fehlerbehebung
Die SSAS Option ist falsch konfiguriert. Vergewissern Sie sich, dass der Mitgliedschaftsstatus der Optionsgruppe „synchron“ lautet, und überprüfen Sie die RDS Ereignisprotokolle auf relevante SSAS Konfigurationsfehlermeldungen. Versuchen Sie es im Anschluss erneut. Wenn weiterhin Fehler auftreten, wenden Sie sich an den AWS Support.	RDSgespeicherte Prozedur	Sie können keine SSAS gespeicherten Prozeduren ausführen, wenn sich Ihre Optionsgruppenmitgliedschaft nicht im `in-sync` Status befindet. Dadurch wird die SSAS Option in einen falschen Konfigurationsstatus versetzt. Wenn sich der Mitgliedschaftsstatus Ihrer Optionsgruppe `failed` aufgrund einer Änderung der SSAS Option auf ändert, gibt es zwei mögliche Gründe: 1. Die SSAS Option wurde entfernt, ohne dass die SSAS Datenbanken gelöscht wurden. 2. Der SSAS Modus wurde von Tabellarisch auf Mehrdimensional oder von Mehrdimensional auf Tabellarisch aktualisiert, ohne dass die vorhandenen Datenbanken gelöscht wurden. SSAS Konfigurieren Sie die SSAS Option neu, da RDS jeweils nur ein SSAS Modus zulässig ist und das Entfernen von Optionen nicht unterstützt wird, wenn Datenbanken vorhanden sind. SSAS SSAS Überprüfen Sie die RDS Ereignisprotokolle auf Konfigurationsfehler für Ihre SSAS Instanz und lösen Sie die Probleme entsprechend.

Problem	Typ	Vorschläge für die Fehlerbehebung
Die Bereitstellung ist fehlgeschlagen. Die Änderung kann nur auf einem Server bereitgestellt werden, der unter läuft *deployment_file_mode* Modus. Der aktuelle Servermodus ist *current_mode* .	RDSgespeicherte Prozedur	Sie können eine tabellarische Datenbank nicht auf einem mehrdimensionalen Server oder eine mehrdimensionale Datenbank nicht auf einem tabellarischen Server bereitstellen. Stellen Sie sicher, dass Sie Dateien mit dem richtigen Modus verwenden, und überprüfen Sie, ob für die Optionseinstellung `MODE` der entsprechende Wert festgelegt wurde.
Die Wiederherstellung ist fehlgeschlagen. Die Sicherungsdatei kann nur auf einem Server wiederhergestellt werden, der unter läuft *restore_file_mode* Modus. Der aktuelle Servermodus ist *current_mode* .	RDSgespeicherte Prozedur	Sie können eine tabellarische Datenbank nicht auf einem mehrdimensionalen Server oder eine mehrdimensionale Datenbank auf einem tabellarischen Server wiederherstellen. Stellen Sie sicher, dass Sie Dateien mit dem richtigen Modus verwenden, und überprüfen Sie, ob für die Optionseinstellung `MODE` der entsprechende Wert festgelegt wurde.
Die Wiederherstellung ist fehlgeschlagen. Die Versionen der Sicherungsdatei und der RDS DB-Instance sind nicht kompatibel.	RDSgespeicherte Prozedur	Sie können keine SSAS Datenbank wiederherstellen, deren Version nicht mit der SQL Serverinstanzversion kompatibel ist. Weitere Informationen finden Sie unter Compatibility levels for tabular models und unter Compatibility level of a multidimensional database in der Microsoft-Dokumentation.

Problem	Typ	Vorschläge für die Fehlerbehebung
Die Wiederherstellung ist fehlgeschlagen. Die beim Wiederherstellungsvorgang angegebene Sicherungsdatei ist beschädigt oder es handelt sich nicht um eine SSAS Sicherungsdatei. Stellen Sie sicher, dass @rds_file_path korrekt formatiert ist.	RDSgespeicherte Prozedur	Sie können eine SSAS Datenbank mit einer beschädigten Datei nicht wiederherstellen. Stellen Sie sicher, dass die Datei nicht beschädigt oder fehlerhaft ist. Dieser Fehler kann auch ausgelöst werden, wenn `@rds_file_path` nicht richtig formatiert ist (wenn beispielsweise doppelte Backslashes wie in `D:\S3\\incorrect_format.abf` enthalten sind).
Die Wiederherstellung ist fehlgeschlagen. Der Name der wiederhergestellten Datenbank darf keine reservierten Wörter oder die folgenden ungültigen Zeichen enthalten . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >, oder länger als 100 Zeichen sein.	RDSgespeicherte Prozedur	Der Name der wiederhergestellten Datenbank darf keine reservierten Wörter oder ungültigen Zeichen enthalten oder länger als 100 Zeichen sein. Konventionen zur SSAS Benennung von Objekten finden Sie in der Microsoft-Dokumentation unter Regeln zur Objektbenennung.
Es wurde ein ungültiger Rollenname angegeben. Der Rollenname darf keine reservierten Strings enthalten.	RDSgespeicherte Prozedur	Der Rollenname darf keine reservierten Strings enthalten. Konventionen zur SSAS Benennung von Objekten finden Sie in der Microsoft-Dokumentation unter Regeln zur Objektbenennung.
Es wurde ein ungültiger Rollenname angegeben. Der Rollenname darf keines der folgenden reservierten Zeichen enthalten: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >	RDSgespeicherte Prozedur	Der Rollenname darf keine reservierten Zeichen enthalten. Konventionen zur SSAS Benennung von Objekten finden Sie in der Microsoft-Dokumentation unter Regeln zur Objektbenennung.

Unterstützung für SQL Server Integration Services in Amazon RDS for SQL Server

Microsoft SQL Server Integration Services (SSIS) ist eine Komponente, mit der Sie eine breite Palette von Datenmigrationsaufgaben ausführen können. SSIS ist eine Plattform für Datenintegrations- und Workflow-Anwendungen. Sie verfügt über ein Data Warehousing-Tool, das für Datenextraktion, Transformation und Laden (ETL) verwendet wird. Sie können dieses Tool auch verwenden, um die Wartung von SQL Server-Datenbanken und Aktualisierungen von mehrdimensionalen Cubedaten zu automatisieren.

SSIS-Projekte werden in Pakete organisiert, die als XML-basierte DTSX-Dateien gespeichert werden. Pakete können Kontrollflüsse und Datenflüsse enthalten. Zur Darstellung von ETL-Vorgängen verwenden Sie Datenflüsse. Nach der Bereitstellung werden Pakete in SQL Server in der SSISDB-Datenbank gespeichert. SSISDB ist eine OLTP-Datenbank (Online Transaction Processing) im vollständigen Wiederherstellungsmodus.

Amazon RDS for SQL Server unterstützt das Ausführen von SSIS direkt auf RDS-DB-Instance. Sie können SSIS für eine vorhandene oder neue DB-Instance aktivieren. SSIS wird auf derselben DB-Instance wie Ihre Datenbank-Engine installiert.

RDS unterstützt SSIS für die SQL Server Standard und Enterprise Edition in den folgenden Versionen:

- SQL Server 2022, alle Versionen
- SQL Server 2019, Version 15.00.4043.16.v1 und höher
- SQL Server 2017, version 14.00.3223.3.v1 und höher
- SQL Server 2016, version 13.00.5426.0.v1 und höher

Inhalt

- Einschränkungen und Empfehlungen
- Aktivieren von SSIS
  - Erstellen der Optionsgruppe für SSIS
  - Hinzufügen der SSIS-Option zur Optionsgruppe
  - Erstellen der Parametergruppe für SSIS
  - Ändern des Parameters für SSIS
  - Zuordnen der Options- und Parametergruppe zu Ihrer DB-Instance

  - Aktivieren der S3-Integration
- Administratorberechtigungen für SSISDB
  - Einen Windows-authentifizierten Benutzer einrichten für SSIS
- Bereitstellen eines SSIS-Projekts
- Überwachen des Status einer Bereitstellungsaufgabe
- Benutzen SSIS
  - Datenbankverbindungsmanager für Projekte einrichten SSIS
  - Einen Proxy erstellen SSIS
  - Ein SSIS Paket mit dem SQL Server-Agent planen
  - SSISZugriff vom Proxy aus widerrufen
- SSISDatenbank deaktivieren und löschen
  - Deaktivierung SSIS
  - Löschen der SSISDB Datenbank

Einschränkungen und Empfehlungen

Die folgenden Einschränkungen und Empfehlungen gelten für die Ausführung von SSIS auf RDS for SQL Server:

- Der DB-Instance muss eine Parametergruppe zugeordnet sein, wobei der Parameter `clr enabled` auf „1“ gesetzt ist. Weitere Informationen finden Sie unter Ändern des Parameters für SSIS.

  > **Note**
  >
  > Wenn Sie den Parameter `clr enabled` auf SQL Server 2017 oder 2019 aktivieren, können Sie die Common Language Runtime (CLR) auf Ihrer DB-Instance nicht verwenden. Weitere Informationen finden Sie unter Nicht unterstützte Funktionen und Funktionen mit beschränkter Unterstützung.

- Die folgenden Kontrollflussaufgaben werden unterstützt:
  - Analysis Services-Aufgabe „DDL ausführen“
  - Analysis Services-Verarbeitungsaufgabe
  - Masseneinfügungsaufgabe

  - Aufgabe „Datenbankintegrität überprüfen“
  - Datenfluss-Aufgabe
  - Aufgabe „Data Mining abfragen“
  - Datenprofilerstellungsaufgabe
  - Aufgabe „Paket ausführen“
  - Aufgabe „SQL Server-Agent-Auftrag ausführen“
  - Aufgabe „SQL ausführen“
  - Aufgabe „T-SQL-Anweisung ausführen“
  - Aufgabe „Bediener benachrichtigen“
  - Aufgabe „Index neu erstellen“
  - Aufgabe „Index neu organisieren“
  - Aufgabe „Datenbank verkleinern“
  - Aufgabe „Datenbank übertragen“
  - Aufgabe „Aufträge übertragen“
  - Aufgabe „Anmeldungen übertragen“
  - Task „SQL Server-Objekte übertragen“
  - Aufgabe „Statistik aktualisieren“
- Es wird nur die Projektbereitstellung unterstützt.
- Das Ausführen von SSIS-Paketen mithilfe von SQL Server-Agent wird unterstützt.
- SSIS-Protokolldatensätze können nur in vom Benutzer erstellte Datenbanken eingefügt werden.
- Verwenden Sie für die Arbeit mit Dateien nur den Ordner `D:\S3`. Dateien, die in einem anderen Verzeichnis gespeichert sind, werden gelöscht. Beachten Sie einige andere Details zum Dateispeicherort:
  - Platzieren Sie SSIS-Projekteingabe- und Ausgabedateien im Ordner `D:\S3`.
  - Ändern Sie für die Datenflussaufgabe den Speicherort für `BLOBTempStoragePath` und `BufferTempStoragePath` in eine Datei innerhalb des Ordners `D:\S3`. Der Dateipfad muss mit beginne `D:\S3\`.
  - Stellen Sie sicher, dass alle Parameter, Variablen und Ausdrücke, die für Dateiverbindungen verwendet werden, auf den Ordner `D:\S3` verweisen.

  - Bei Multi-AZ-Instances werden Dateien, die von SSIS im Ordner `D:\S3` erstellt wurden, nach einem Failover gelöscht. Weitere Informationen finden Sie unter Multi-AZ-Einschränkungen für die S3-Integration.
  - Laden Sie die von SSIS erstellten Dateien im Ordner `D:\S3` in Ihren Amazon S3-Bucket hoch, um sie dauerhaft zu machen.
- Importieren von Spalten- und Exportspalten-Transformationen sowie die Skriptkomponente in der Datenflussaufgabe werden nicht unterstützt.
- Sie können keine Dump bei ausgeführten SSIS-Paketen aktivieren, und Sie können keine Datentippeingaben für SSIS-Pakete hinzufügen.
- Die SSIS-Scale-Out-Funktion wird nicht unterstützt.
- Projekte können nicht direkt bereitgestellt werden. Dazu stellen wir gespeicherte RDS-Prozeduren bereit. Weitere Informationen finden Sie unter Bereitstellen eines SSIS-Projekts.
- Erstellen Sie SSIS-Projektdateien (.ispac) mit dem Schutzmodus `DoNotSavePasswords` für die Bereitstellung auf RDS.
- SSIS wird auf immer eingeschalteten Instances mit Read Replicas nicht unterstützt.
- Sie können die SSISDB-Datenbank, die der Option `SSIS` zugeordnet ist, nicht sichern.
- Das Importieren und Wiederherstellen der SSISDB-Datenbank aus anderen SSIS-Instances wird nicht unterstützt.
- Sie können eine Verbindung mit anderen DB-Instances von SQL Server oder mit einer Oracle-Datenquelle herstellen. Die Verbindung zu anderen Datenbank-Engines wie MySQL oder PostgreSQL wird für SSIS auf RDS für SQL Server nicht unterstützt. Weitere Informationen zum Herstellen einer Verbindung mit einer Oracle-Datenquelle finden Sie unter Mit Oracle OLEDB verknüpfte Server.
- SSIS unterstützt keine in eine Domäne eingebundene Instanz mit einer ausgehenden Vertrauensstellung zu einer lokalen Domäne. Wenn Sie eine ausgehende Vertrauensstellung verwenden, führen Sie den SSIS-Job von einem Konto in der lokalen Domäne aus. AWS

Aktivieren von SSIS

Sie aktivieren SSIS, indem Sie der DB-Instance die SSIS-Option hinzufügen. Verwenden Sie den folgenden Prozess:

1. Erstellen Sie eine neue Optionsgruppe oder wählen Sie eine bestehende Optionsgruppe aus.
2. Fügen Sie die Option `SSIS` zur Optionsgruppe hinzu.

3. Erstellen Sie eine neue Parametergruppe oder wählen Sie eine vorhandene Parametergruppe aus.
4. Ändern Sie die Parametergruppe, um den Parameter `clr enabled` auf „1“ einzustellen.
5. Ordnen Sie die Optionsgruppe und die Parametergruppe der DB-Instance zu.
6. Aktivieren Sie die Amazon S3-Integration.

> **Note**
>
> Wenn auf der DB-Instance bereits eine Datenbank mit dem Namen „SSISDB“ oder eine reservierte SSIS-Anmeldung vorhanden ist, können Sie SSIS für die Instance nicht aktivieren.

Erstellen der Optionsgruppe für SSIS

Um mit SSIS zu arbeiten, erstellen Sie eine Optionsgruppe oder ändern Sie eine Optionsgruppe, die der SQL Server-Edition und der Version der DB-Instance entspricht, die Sie verwenden möchten. Verwenden Sie dazu den AWS Management Console oder den AWS CLI.

Konsole

Mit der folgenden Konsolenprozedur wird eine Optionsgruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Fenster Create option group (Optionsgruppe erstellen) Folgendes aus:
   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihres AWS Kontos eindeutig ist, z. **`ssis-se-2016`** B. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.
   b. Geben Sie unter Beschreibung eine kurze Beschreibung der Optionsgruppe ein, z. B. **`SSIS option group for SQL Server SE 2016`**. Die Beschreibung ist nur zur Information.
   c. Wählen Sie für Engine die Option sqlserver-se aus.
   d. Wählen Sie im Feld Engine-Hauptversion 13.00 aus.

5. Wählen Sie Erstellen aus.

CLI

Mit der folgenden Konsolenprozedur wird eine Optionsgruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Optionsgruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds create-option-group \
    --option-group-name ssis-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

  Windows:

```
aws rds create-option-group ^
    --option-group-name ssis-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Hinzufügen der SSIS-Option zur Optionsgruppe

Verwenden Sie als Nächstes das AWS Management Console oder, AWS CLI um die `SSIS` Option zu Ihrer Optionsgruppe hinzuzufügen.

Konsole

So fügen Sie die SSIS-Option hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie in diesem Beispiel die gerade erstellte Optionsgruppe ssis-se-2016 aus.
4. Wählen Sie Add option (Option hinzufügen).
5. Wählen Sie unter Optionsdetails für Optionsname die Option SSIS aus.
6. Wählen Sie unter Scheduling (Planung) aus, ob die Option sofort oder während des nächsten Wartungsfensters hinzugefügt werden soll.
7. Wählen Sie Add option (Option hinzufügen).

CLI

So fügen Sie die SSIS-Option hinzu

- Fügen Sie die Option `SSIS` zur Optionsgruppe hinzu.

  Example

  Für LinuxmacOS, oderUnix:

  ```
  aws rds add-option-to-option-group \
      --option-group-name ssis-se-2016 \
      --options OptionName=SSIS \
      --apply-immediately
  ```

  Windows:

  ```
  aws rds add-option-to-option-group ^
      --option-group-name ssis-se-2016 ^
      --options OptionName=SSIS ^
      --apply-immediately
  ```

Erstellen der Parametergruppe für SSIS

Erstellen oder ändern Sie eine Parametergruppe für den Parameter `clr enabled`, der der SQL Server-Edition und der Version der DB-Instance entspricht, die Sie für SSIS verwenden möchten.

Konsole

Im folgenden Verfahren wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie Create parameter group (Parametergruppe erstellen).
4. Führen Sie im Bereich Parametergruppe erstellen die folgenden Schritte aus:
   a. Wählen Sie für Parametergruppenfamilie die Option sqlserver-se-13.0 aus.
   b. Geben Sie unter Gruppenname einen Bezeichner für die Parametergruppe ein, z. B. **`ssis-sqlserver-se-13`**.
   c. Geben Sie für Beschreibung den Text **`clr enabled parameter group`** ein.
5. Wählen Sie Create aus.

CLI

Im folgenden Verfahren wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "clr enabled parameter group"
```

  Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "clr enabled parameter group"
```

Ändern des Parameters für SSIS

Ändern Sie den `clr enabled`-Parameter in der Parametergruppe, die der SQL Server-Edition und der Version Ihrer DB-Instance entspricht. Stellen Sie für SSIS den Parameter `clr enabled` auf „1“ ein.

Konsole

Im folgenden Verfahren wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben.

So ändern Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie die Parametergruppe, z. B. ssis-sqlserver-se-13.
4. Filtern Sie unter Parameter die Parameterliste nach **`clr`**.
5. Wählen Sie clr enabled.
6. Wählen Sie Parameter bearbeiten aus.
7. Wählen Sie unter Werte die Option 1 aus.
8. Wählen Sie Änderungen speichern aus.

CLI

Im folgenden Verfahren wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben.

So ändern Sie die Parametergruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
```

```
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Zuordnen der Options- und Parametergruppe zu Ihrer DB-Instance

Um die SSIS-Optionsgruppe und die Parametergruppe mit Ihrer DB-Instance zu verknüpfen, verwenden Sie den AWS Management Console oder AWS CLI

> **Note**
>
> Wenn Sie eine vorhandene Instance verwenden, muss dieser bereits eine Active Directory-Domäne und AWS Identity and Access Management (IAM) -Rolle zugeordnet sein. Wenn Sie eine neue Instance erstellen, geben Sie eine vorhandene Active Directory-Domäne und IAM-Rolle an. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.

Konsole

Um die Aktivierung von SSIS abzuschließen, ordnen Sie Ihre SSIS-Optionsgruppe und Parametergruppe einer neuen oder vorhandenen DB-Instance zu:

- Ordnen Sie sie bei einer neuen DB-Instance zu, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ordnen Sie sie für eine vorhandene DB-Instance zu, indem Sie die Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

CLI

Sie können die SSIS-Optionsgruppe und die Parametergruppe einer neuen oder vorhandenen DB-Instance zuordnen.

So erstellen Sie eine Instance mit der SSIS-Optionsgruppe und der Parametergruppe

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Optionsgruppe verwendet haben.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds create-db-instance \
    --db-instance-identifier myssisinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13
```

  Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssisinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13
```

So ändern Sie eine Instance und ordnen die SSIS-Optionsgruppe und die Parametergruppe zu

- Führen Sie einen der folgenden Befehle aus.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssisinstance \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --apply-immediately
```

  Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssisinstance ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --apply-immediately
```

Aktivieren der S3-Integration

Um SSIS-Projektdateien (.ispac) für die Bereitstellung auf Ihren Host herunterzuladen, verwenden Sie die S3-Dateiintegration. Weitere Informationen finden Sie unter Integration einer Amazon RDS for SQL Server-DB-Instance mit Amazon S3.

Administratorberechtigungen für SSISDB

Wenn die Instanz mit der SSIS Option erstellt oder geändert wird, ist das Ergebnis eine SSISDB Datenbank mit den Rollen ssis_admin und ssis_logreader, die dem Masterbenutzer zugewiesen wurden. Der Masterbenutzer hat die folgenden Rechte in: SSISDB

- auf ssis_admin Rolle ändern
- auf ssis_logreader-Rolle ändern
- jeden Benutzer ändern

Da der Hauptbenutzer ein SQL -authentifizierter Benutzer ist, können Sie den Hauptbenutzer nicht zum Ausführen SSIS von Paketen verwenden. Der Hauptbenutzer kann diese Rechte verwenden, um neue SSISDB Benutzer zu erstellen und sie den Rollen ssis_admin und ssis_logreader hinzuzufügen. Dies ist nützlich, um Ihren Domänenbenutzern Zugriff zur Nutzung zu gewähren. SSIS

Einen Windows-authentifizierten Benutzer einrichten für SSIS

Der Masterbenutzer kann das folgende Codebeispiel verwenden, um eine Windows-authentifizierte Anmeldung einzurichten SSISDB und die erforderlichen Prozedurberechtigungen zu erteilen. Dadurch erhält der Domänenbenutzer die Berechtigung, SSIS Pakete bereitzustellen und auszuführen, S3-Dateiübertragungsverfahren zu verwenden, Anmeldeinformationen zu erstellen und mit dem SQL Server-Agent-Proxy zu arbeiten. Weitere Informationen finden Sie in der Microsoft-Dokumentation unter Anmeldeinformationen (Datenbankmodul) und Erstellen eines SQL Server-Agent-Proxys.

> **Note**
>
> Sie können Windows-authentifizierten Benutzern bei Bedarf einige oder alle der folgenden Berechtigungen erteilen.

Example

```
-- Create a server-level SQL login for the domain user, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create a database-level account for the domain user, if it doesn't already exist

USE [SSISDB]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Add SSIS role membership to the domain user
ALTER ROLE [ssis_admin] ADD MEMBER [mydomain\user_name]
ALTER ROLE [ssis_logreader] ADD MEMBER [mydomain\user_name]
GO

-- Add MSDB role membership to the domain user
```

```
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Grant MSDB stored procedure privileges to the domain user
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name]  with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] WITH GRANT OPTION


-- Add the SQLAgentUserRole privilege to the domain user
USE [msdb]
GO
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO

-- Grant the ALTER ANY CREDENTIAL privilege to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO
```

Bereitstellen eines SSIS-Projekts

Auf RDS können Sie SSIS-Projekte nicht direkt mithilfe von SQL Server Management Studio (SSMS) oder SSIS-Verfahren bereitstellen. Verwenden Sie gespeicherte RDS-Prozeduren, um Projektdateien aus Amazon S3 herunterzuladen und anschließend bereitzustellen.

Melden Sie sich zum Ausführen der gespeicherten Prozeduren als beliebiger Benutzer an, dem Sie Berechtigungen für das Ausführen der gespeicherten Prozeduren erteilt haben. Weitere Informationen finden Sie unter Einen Windows-authentifizierten Benutzer einrichten für SSIS.

So stellen Sie das SSIS-Projekt bereit

1. Laden Sie die Projektdatei (.ispac) herunter.

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/ssisproject.ispac',
@rds_file_path='D:\S3\ssisproject.ispac',
@overwrite_file=1;
```

2. Übermitteln Sie die Bereitstellungsaufgabe und stellen Sie sicher, dass Sie Folgendes beachten:

   - Der Ordner ist im SSIS-Katalog vorhanden.
   - Der Projektname stimmt mit dem Projektnamen überein, den Sie beim Entwickeln des SSIS-Projekts verwendet haben.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSIS_DEPLOY_PROJECT',
@folder_name='DEMO',
@project_name='ssisproject',
@file_path='D:\S3\ssisproject.ispac';
```

Überwachen des Status einer Bereitstellungsaufgabe

Rufen Sie die Funktion `rds_fn_task_status` auf, um den Status Ihrer Bereitstellungsaufgabe zu verfolgen. Dazu sind zwei Parameter erforderlich. Der erste Parameter sollte immer sein`NULL`, weil er nicht gilt für`SSIS`. Der zweite Parameter akzeptiert eine Aufgaben-ID.

Um eine Liste aller Aufgaben anzuzeigen, setzen Sie den ersten Parameter auf `NULL` und den zweiten Parameter auf `0`, wie im folgenden Beispiel gezeigt.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Um eine bestimmte Aufgabe zu erhalten, setzen Sie den ersten Parameter auf NULL und den zweiten Parameter auf die Aufgaben-ID, wie im folgenden Beispiel gezeigt,

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Die `rds_fn_task_status`-Funktion gibt die folgenden Informationen zurück.

Ausgabeparameter	Beschreibung
`task_id`	Die ID der Aufgabe.
`task_type`	`SSIS_DEPLOY_PROJECT`
`database_name`	Gilt nicht für SSIS Aufgaben.
`% complete`	Verlauf der Aufgabe als Prozentwert.
`duration (mins)`	Zeitdauer für die Ausführung der Aufgabe (in Minuten).
`lifecycle`	Der Status der Aufgabe. Die folgenden Status sind möglich: • `CREATED` – Nachdem Sie die gespeicherte Prozedur `msdb.dbo.rds_msbi_task` aufgerufen haben, wird eine Aufgabe erstellt und der Status wird auf `CREATED` gesetzt. • `IN_PROGRESS` – Nach dem Start einer Aufgabe wird der Status auf gesetzt `IN_PROGRESS` . Es kann bis zu fünf Minuten dauern, bis sich der Status von `CREATED` zu `IN_PROGRESS` ändert. •

Ausgabeparameter	Beschreibung
	SUCCESS – Nach dem Abschluss einer Aufgabe wird der Status auf gesetzt SUCCESS. • ERROR – Wenn eine Aufgabe fehlschlägt, wird der Status auf gesetzt ERROR. Weitere Informationen über den Fehler können Sie der Spalte task_info entnehmen. • CANCEL_REQUESTED – Sobald Sie rds_cancel_task aufrufen, wird der Status der Aufgabe auf CANCEL_RE QUESTED gesetzt. • CANCELLED – Nachdem die Aufgabe abgebrochen wurde, wird der Status der Aufgabe auf gesetzt CANCELLED .
task_info	Zusätzliche Informationen über die Aufgabe. Wenn bei der Verarbeitung ein Fehler auftritt, enthält diese Spalte Informationen zu dem Fehler.
last_updated	Datum und Uhrzeit der letzten Aktualisierung des Aufgabenstatus.
created_at	Datum und Uhrzeit, an denen die Aufgabe angelegt wurde.
S3_object_arn	Gilt nicht für SSIS Aufgaben.
overwrite_S3_backup_file	Gilt nicht für SSIS Aufgaben.
KMS_master_key_arn	Gilt nicht für SSIS Aufgaben.

Ausgabeparameter	Beschreibung
`filepath`	Gilt nicht für SSIS Aufgaben.
`overwrite_file`	Gilt nicht für SSIS Aufgaben.
`task_metadata`	Mit der SSIS Aufgabe verknüpfte Metadaten.

Benutzen SSIS

Nachdem Sie das SSIS Projekt im SSIS Katalog bereitgestellt haben, können Sie Pakete direkt vom Server Agent aus ausführen SSMS oder sie mithilfe des SQL Server-Agents planen. Für die Ausführung von Paketen müssen Sie eine Windows-authentifizierte Anmeldung verwenden. SSIS Weitere Informationen finden Sie unter Einen Windows-authentifizierten Benutzer einrichten für SSIS.

Themen

- Datenbankverbindungsmanager für Projekte einrichten SSIS
- Einen Proxy erstellen SSIS
- Ein SSIS Paket mit dem SQL Server-Agent planen
- SSISZugriff vom Proxy aus widerrufen

Datenbankverbindungsmanager für Projekte einrichten SSIS

Bei Einsatz eines Verbindungs-Managers können Sie die folgenden Authentifizierungstypen verwenden:

- Für lokale Datenbankverbindungen, die AWS Managed Active Directory verwenden, können Sie die SQL Authentifizierung oder die Windows-Authentifizierung verwenden. Verwenden Sie für die Windows-Authentifizierung *`DB_instance_name`*`.`*`fully_qualified_domain_name`* als Servername der Verbindungszeichenfolge.

  Ein Beispiel ist `myssisinstance.corp-ad.example.com`, wobei `myssisinstance` der Name der DB-Instance und `corp-ad.example.com` der vollqualifizierte Domänenname ist.
- Verwenden Sie für Remoteverbindungen immer die SQL Authentifizierung.

- Für lokale Datenbankverbindungen, die selbstverwaltetes Active Directory verwenden, können Sie die SQL Authentifizierung oder die Windows-Authentifizierung verwenden. Verwenden Sie für die Windows-Authentifizierung `.` oder *`LocalHost`* als Servername der Verbindungszeichenfolge.

Einen Proxy erstellen SSIS

Um SSIS Pakete mit dem SQL Server-Agent planen zu können, müssen Sie SSIS Anmeldeinformationen und einen SSIS Proxy erstellen. Führen Sie diese Prozeduren als Windows-authentifizierter Benutzer aus.

Um die Anmeldeinformationen zu erstellen SSIS

- Erstellen Sie die Anmeldeinformationen für den Proxy. Dazu können Sie SSMS oder die folgende SQL Anweisung verwenden.

```
USE [master]
GO
CREATE CREDENTIAL [SSIS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> `IDENTITY` muss eine domänenauthentifizierter Anmeldung sein. Ersetzen Sie *`mysecret`* durch das Passwort für die domänenauthentifizierte Anmeldung. Wenn der SSISDB primäre Host geändert wird, ändern Sie die SSIS Proxy-Anmeldeinformationen, damit der neue Host darauf zugreifen kann.

Um den SSIS Proxy zu erstellen

1. Verwenden Sie die folgende SQL Anweisung, um den Proxy zu erstellen.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSIS_Proxy',@credential_name=N'SSIS_Credential',@description=N''
GO
```

2. Verwenden Sie die folgende SQL Anweisung, um anderen Benutzern Zugriff auf den Proxy zu gewähren.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSIS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Verwenden Sie die folgende SQL Anweisung, um dem SSIS Subsystem Zugriff auf den Proxy zu gewähren.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

So zeigen Sie den Proxy und die Berechtigungserteilungen für den Proxy an

1. Verwenden Sie die folgende SQL Anweisung, um die Empfänger des Proxys anzuzeigen.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Verwenden Sie die folgende SQL Anweisung, um die Subsystem-Zuschüsse anzuzeigen.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Ein SSIS Paket mit dem SQL Server-Agent planen

Nachdem Sie die Anmeldeinformationen und den Proxy erstellt und SSIS Zugriff auf den Proxy gewährt haben, können Sie einen SQL Server-Agent-Job erstellen, um das SSIS Paket zu planen.

Um das Paket zu planen SSIS

- Sie können SSMS oder T- verwenden, SQL um den SQL Server-Agent-Job zu erstellen. Im folgenden Beispiel wird T- verwendetSQL.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'MYSSISJob',
@enabled=1,
@notify_level_eventlog=0,
@notify_level_email=2,
@notify_level_page=2,
@delete_level=0,
@category_name=N'[Uncategorized (Local)]',
@job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'MYSSISJob',@server_name=N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep
 @job_name=N'MYSSISJob',@step_name=N'ExecuteSSISPackage',
@step_id=1,
@cmdexec_success_code=0,
@on_success_action=1,
@on_fail_action=2,
@retry_attempts=0,
@retry_interval=0,
@os_run_priority=0,
@subsystem=N'SSIS',
@command=N'/ISSERVER "\"\SSISDB\MySSISFolder\MySSISProject\MySSISPackage.dtsx\"" /
SERVER "\"my-rds-ssis-instance.corp-ad.company.com/\""
/Par "\"$ServerOption::LOGGING_LEVEL(Int16)\"";1 /Par
 "\"$ServerOption::SYNCHRONIZED(Boolean)\"";True /CALLERINFO SQLAGENT /REPORTING
 E',
@database_name=N'master',
@flags=0,
@proxy_name=N'SSIS_Proxy'
GO
```

SSISZugriff vom Proxy aus widerrufen

Sie können den Zugriff auf das SSIS Subsystem widerrufen und den SSIS Proxy mithilfe der folgenden gespeicherten Prozeduren löschen.

So entziehen Sie den Zugriff und löschen den Proxy

1. Widerrufen des Teilsystemzugriffs.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

2. Widerrufen Sie die für den erteilten Berechtigungen Proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSIS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Löschen Sie den Proxy.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSIS_Proxy'
GO
```

SSISDatenbank deaktivieren und löschen

Gehen Sie wie folgt vor, um SSIS Datenbanken zu deaktivieren oder zu löschen:

Themen

- Deaktivierung SSIS
- Löschen der SSISDB Datenbank

Deaktivierung SSIS

Um sie zu deaktivierenSSIS, entfernen Sie die `SSIS` Option aus ihrer Optionsgruppe.

> **Important**
>
> Durch das Entfernen der Option wird die SSISDB Datenbank nicht gelöscht. Sie können die Option also problemlos entfernen, ohne die SSIS Projekte zu verlieren.
> Sie können die `SSIS` Option nach dem Entfernen wieder aktivieren, um die SSIS Projekte wiederzuverwenden, die zuvor im SSIS Katalog bereitgestellt wurden.

Konsole

Mit dem folgenden Verfahren wird die Option `SSIS` entfernt.

Um die SSIS Option aus ihrer Optionsgruppe zu entfernen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der Option `SSIS` (`ssis-se-2016` in den vorherigen Beispielen).
4. Wählen Sie Delete option (Option löschen) aus.
5. Wählen Sie unter Löschoptionen SSISdie Option Zu löschende Optionen aus.
6. Wählen Sie unter Apply immediately (Sofort anwenden) die Option Yes (Ja) aus, um die Option sofort zu löschen, oder No (Nein), um sie während des nächsten Wartungsfensters zu löschen.
7. Wählen Sie Löschen.

CLI

Mit dem folgenden Verfahren wird die Option `SSIS` entfernt.

Um die SSIS Option aus ihrer Optionsgruppe zu entfernen

- Führen Sie einen der folgenden Befehle aus.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssis-se-2016 \
    --options SSIS \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssis-se-2016 ^
    --options SSIS ^
    --apply-immediately
```

Löschen der SSISDB Datenbank

Nach dem Entfernen der SSIS Option wird die SSISDB Datenbank nicht gelöscht. Um die SSISDB Datenbank zu löschen, verwenden Sie die `rds_drop_ssis_database` gespeicherte Prozedur, nachdem Sie die SSIS Option entfernt haben.

Um die SSIS Datenbank zu löschen

- Verwenden Sie die folgende gespeicherte Prozedur.

```
USE [msdb]
GO
EXEC dbo.rds_drop_ssis_database
GO
```

Wenn Sie die SSIS Option nach dem Löschen der SSISDB Datenbank erneut aktivieren, erhalten Sie einen neuen SSISDB Katalog.

Support für SQL Server Reporting Services in Amazon RDS for SQL Server

Microsoft SQL Server Reporting Services (SSRS) ist eine serverbasierte Anwendung, die für die Generierung und Verteilung von Berichten verwendet wird. Es ist Teil einer Suite von SQL Serverdiensten, zu der auch SQL Server Analysis Services (SSAS) und SQL Server Integration Services (SSIS) gehören. SSRSist ein Dienst, der auf SQL Server aufbaut. Sie können damit Daten aus verschiedenen Datenquellen sammeln und auf eine Weise präsentieren, die leicht verständlich und für die Analyse bereit ist.

Amazon RDS for SQL Server unterstützt die SSRS direkte Ausführung auf RDS DB-Instances. Sie können es SSRS mit vorhandenen oder neuen DB-Instances verwenden.

RDSunterstützt SSRS SQL Server Standard und Enterprise Editions in den folgenden Versionen:

- SQLServer 2022, alle Versionen
- SQLServer 2019, Version 15.00.4043.16.v1 und höher
- SQLServer 2017, Version 14.00.3223.3.v1 und höher
- SQLServer 2016, Version 13.00.5820.21.v1 und höher

Inhalt

- Einschränkungen und Empfehlungen
- Aktivieren von SSRS
  - Eine Optionsgruppe erstellen für SSRS
  - Hinzufügen der SSRS Option zu Ihrer Optionsgruppe
  - Zuordnen Ihrer Optionsgruppe zu Ihrer DB-Instance
  - Zulassen des eingehenden Zugriffs auf Ihre VPC Sicherheitsgruppe
- Berichtsserverdatenbanken
- SSRSProtokolldateien
- Zugreifen auf das SSRS Webportal
  - Benutzt SSL ein RDS
  - Gewähren des Zugriffs für Domänenbenutzer
  - Zugriff auf das Webportal
- Bereitstellen von Berichten und Konfigurieren von Berichtsdatenquellen
  - Bereitstellen von Berichten für SSRS

  - Konfigurieren der Berichtsdatenquelle
- Verwenden von SSRS E-Mail zum Senden von Berichten
- Widerrufen von Berechtigungen auf Systemebene
- Überwachung des Status einer Aufgabe
- Datenbanken deaktivieren und löschen SSRS
  - Deaktivieren von SSRS
  - Löschen der SSRS Datenbanken

Einschränkungen und Empfehlungen

Die folgenden Einschränkungen und Empfehlungen gelten für die Ausführung auf dem Server: SSRS RDS SQL

- Sie können es nicht SSRS auf DB-Instances verwenden, die Read Replicas haben.
- Instanzen müssen selbstverwaltetes Active Directory oder AWS Directory Service for Microsoft Active Directory für die SSRS Webportal- und Webserverauthentifizierung verwenden. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.
- Sie können die Berichtsserver-Datenbanken, die mit dieser SSRS Option erstellt wurden, nicht sichern.
- Das Importieren und Wiederherstellen von Berichtsserver-Datenbanken aus anderen Instanzen von wird SSRS nicht unterstützt. Weitere Informationen finden Sie unter Berichtsserverdatenbanken.
- Sie können nicht so konfigurierenSSRS, dass der SSL Standardport (443) abgehört wird. Die zulässigen Werte sind 1150–49511, außer 1234, 1434, 3260, 3343, 3389 und 47001.
- Abonnements per Microsoft Windows-Dateifreigabe werden nicht unterstützt.
- Die Verwendung von Reporting Services Configuration Manager wird nicht unterstützt.
- Das Erstellen und Ändern von Rollen wird nicht unterstützt.
- Das Ändern der Eigenschaften des Berichtsservers wird nicht unterstützt.
- Systemadministrator- und Systembenutzerrollen werden nicht erteilt.
- Sie können Rollenzuweisungen auf Systemebene nicht über das Webportal bearbeiten.

Aktivieren von SSRS

Gehen Sie wie folgt vor, SSRS um Ihre DB-Instance einzuschalten:

1. Erstellen Sie eine neue Optionsgruppe oder wählen Sie eine bestehende Optionsgruppe aus.
2. Fügen Sie die Option SSRS zur Optionsgruppe hinzu.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.
4. Erlauben Sie eingehenden Zugriff auf die Sicherheitsgruppe Virtual Private Cloud (VPC) für den SSRS Listener-Port.

Eine Optionsgruppe erstellen für SSRS

Um damit zu arbeitenSSRS, erstellen Sie eine Optionsgruppe, die der SQL Server-Engine und der Version der DB-Instance entspricht, die Sie verwenden möchten. Verwenden Sie dazu die AWS Management Console oder die AWS CLI.

> **Note**
>
> Sie können auch eine vorhandene Optionsgruppe verwenden, wenn sie für die richtige SQL Server-Engine und Version bestimmt ist.

Konsole

Mit dem folgenden Verfahren wird eine Optionsgruppe für SQL Server Standard Edition 2017 erstellt.

So erstellen Sie die Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Bereich Create option group (Optionsgruppe erstellen) Folgendes aus:
   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihrer Optionsgruppe einzigartig ist AWS-Konto, z. **`ssrs-se-2017`** B. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.

b. Geben Sie unter Beschreibung eine kurze Beschreibung der Optionsgruppe ein, z. B. **SSRS option group for SQL Server SE 2017**. Die Beschreibung ist nur zur Information.

c. Wählen Sie für Engine die Option sqlserver-se aus.

d. Wählen Sie im Feld Major Engine Version (Engine-Hauptversion) 14.00 aus.

5. Wählen Sie Create (Erstellen) aus.

CLI

Mit dem folgenden Verfahren wird eine Optionsgruppe für SQL Server Standard Edition 2017 erstellt.

So erstellen Sie die Optionsgruppe

- Führen Sie einen der folgenden Befehle aus.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-option-group \
    --option-group-name ssrs-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-option-group ^
    --option-group-name ssrs-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Hinzufügen der SSRS Option zu Ihrer Optionsgruppe

Verwenden Sie als Nächstes das AWS Management Console oder, AWS CLI um die SSRS Option zu Ihrer Optionsgruppe hinzuzufügen.

Konsole

Um die SSRS Option hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die soeben erstellte Optionsgruppe aus, und wählen Sie dann Option hinzufügen.
4. Wählen Sie unter Optionsdetails SSRSden Optionsnamen aus.
5. Führen Sie unter Optionseinstellungen die folgenden Schritte aus:

   a. Geben Sie den Port ein, den der SSRS Dienst abhören soll. Der Standardwert ist 8443. Eine Liste der zulässigen Werte finden Sie unter Einschränkungen und Empfehlungen.

   b. Geben Sie einen Wert für Max. Speicher ein.

      Max. Speicher gibt den oberen Schwellenwert an, über dem keine neuen Speicherzuordnungsanforderungen für Berichtsserveranwendungen erteilt werden. Die Zahl ist ein Prozentsatz des Gesamtspeichers der DB-Instance. Die zulässigen Werte sind 10–80.

   c. Wählen Sie für Sicherheitsgruppen die VPC Sicherheitsgruppe aus, die der Option zugeordnet werden soll. Verwenden Sie dieselbe Sicherheitsgruppe, die Ihrer DB-Instance zugeordnet ist.

6. Um Berichte SSRS per E-Mail zu versenden, aktivieren Sie in Reporting Services unter E-Mail-Versand das Kontrollkästchen E-Mail-Zustellungsoptionen konfigurieren, und gehen Sie dann wie folgt vor:

   a. Geben Sie unter E-Mail-Adresse des Absenders die E-Mail-Adresse ein, die im Feld Von für per SSRS E-Mail gesendete Nachrichten verwendet werden soll.

      Geben Sie ein Benutzerkonto an, das berechtigt ist, E-Mails vom SMTP Server zu senden.

   b. Geben Sie als SMTPServer den SMTP Server oder das Gateway an, der verwendet werden soll.

      Dabei kann es sich um eine IP-Adresse, den BIOS Netznamen eines Computers in Ihrem Unternehmensintranet oder um einen vollqualifizierten Domänennamen handeln.

   c. Geben Sie als SMTPPort den Port ein, der für die Verbindung mit dem Mailserver verwendet werden soll. Der Standardwert ist 25.

   d. So verwenden Sie die Authentifizierung:

i. Aktivieren Sie das Kontrollkästchen Authentifizierung verwenden.

ii. Geben Sie für Secret Amazon Resource Name (ARN) die AWS Secrets Manager ARN für die Benutzeranmeldedaten ein.

Verwenden Sie das folgende Format:

arn:aws:secretsmanager:*Region*:*AccountId*:secret:*SecretName*-*6RandomChara*

Beispiel:

arn:aws:secretsmanager:*us-west-2*:*123456789012*:secret:*MySecret-a1b2c3*

Weitere Informationen über „creating the secret“ finden Sie in Verwenden von SSRS E-Mail zum Senden von Berichten.

e. Aktivieren Sie das Kontrollkästchen Secure Sockets Layer (SSL) verwenden, um E-Mail-Nachrichten zu verschlüsseln mitSSL.

7. Wählen Sie unter Scheduling (Planung) aus, ob die Option sofort oder während des nächsten Wartungsfensters hinzugefügt werden soll.
8. Wählen Sie Add option (Option hinzufügen).

CLI

Um die Option hinzuzufügen SSRS

1. Erstellen Sie beispielsweise eine JSON Dateissrs-option.json.

a. Setzen Sie die folgenden erforderlichen Parameter:

- `OptionGroupName` – Der Name der Optionsgruppe, die Sie zuvor erstellt oder ausgewählt haben (`ssrs-se-2017` im folgenden Beispiel).
- `Port`— Der Port, auf dem der SSRS Dienst lauschen soll. Der Standardwert ist 8443. Eine Liste der zulässigen Werte finden Sie unter Einschränkungen und Empfehlungen.
- `VpcSecurityGroupMemberships`— Mitgliedschaften in VPC Sicherheitsgruppen für Ihre RDS DB-Instance.
- `MAX_MEMORY` – Der obere Schwellenwert, über dem keine neuen Speicherzuordnungsanforderungen für Berichtsserveranwendungen erteilt werden. Die

Zahl ist ein Prozentsatz des Gesamtspeichers der DB-Instance. Die zulässigen Werte sind 10–80.

b. (Optional) Legen Sie die folgenden Parameter für die Verwendung von SSRS E-Mail fest:

- `SMTP_ENABLE_EMAIL`— Stellen Sie ein, `true` dass SSRS E-Mail verwendet werden soll. Der Standardwert ist `false`.
- `SMTP_SENDER_EMAIL_ADDRESS`— Die E-Mail-Adresse, die im Absenderfeld von per SSRS E-Mail gesendeten Nachrichten verwendet werden soll. Geben Sie ein Benutzerkonto an, das berechtigt ist, E-Mails vom SMTP Server zu senden.
- `SMTP_SERVER`— Der zu verwendende SMTP Server oder das Gateway. Dabei kann es sich um eine IP-Adresse, den BIOS Netznamen eines Computers in Ihrem Unternehmensintranet oder um einen vollqualifizierten Domänennamen handeln.
- `SMTP_PORT` – Der Port, der für die Verbindung mit dem Mailserver verwendet werden soll. Der Standardwert ist 25.
- `SMTP_USE_SSL`— Auf einstellen, `true` um E-Mail-Nachrichten zu verschlüsseln mit. SSL Der Standardwert ist `true`.
- `SMTP_EMAIL_CREDENTIALS_SECRET_ARN`— Der Secrets ManagerARN, der die Benutzeranmeldedaten enthält. Verwenden Sie das folgende Format:

  `arn:aws:secretsmanager:`*`Region`*`:`*`AccountId`*`:secret:`*`SecretName`*`-6RandomCharact`

  Weitere Informationen über „creating the secret“ finden Sie in Verwenden von SSRS E-Mail zum Senden von Berichten.
- `SMTP_USE_ANONYMOUS_AUTHENTICATION` – Einstellung auf `true` und nicht einschließen, `SMTP_EMAIL_CREDENTIALS_SECRET_ARN` wenn Sie keine Authentifizierung verwenden möchten.

  Der Standardwert ist `false`, wenn `SMTP_ENABLE_EMAIL` ist `true`.

Das folgende Beispiel enthält die SSRS E-Mail-Parameter unter Verwendung des GeheimnissesARN.

```
{
"OptionGroupName": "ssrs-se-2017",
"OptionsToInclude": [
 {
 "OptionName": "SSRS",
```

```
 "Port": 8443,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [
           {"Name": "MAX_MEMORY","Value": "60"},
           {"Name": "SMTP_ENABLE_EMAIL","Value": "true"}
           {"Name": "SMTP_SENDER_EMAIL_ADDRESS","Value": "nobody@example.com"},
           {"Name": "SMTP_SERVER","Value": "email-smtp.us-west-2.amazonaws.com"},
           {"Name": "SMTP_PORT","Value": "25"},
           {"Name": "SMTP_USE_SSL","Value": "true"},
           {"Name": "SMTP_EMAIL_CREDENTIALS_SECRET_ARN","Value":
 "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3"}
           ]
 }],
"ApplyImmediately": true
}
```

2. Fügen Sie die Option SSRS zur Optionsgruppe hinzu.

   Example

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssrs-option.json \
    --apply-immediately
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssrs-option.json ^
    --apply-immediately
```

Zuordnen Ihrer Optionsgruppe zu Ihrer DB-Instance

Verwenden Sie das AWS Management Console oder das AWS CLI , um Ihre Optionsgruppe mit Ihrer DB-Instance zu verknüpfen.

Wenn Sie eine vorhandene DB-Instance verwenden, muss dieser bereits eine Active Directory-Domäne und AWS Identity and Access Management (IAM) -Rolle zugeordnet sein. Wenn Sie eine neue Instance erstellen, geben Sie eine bestehende Active Directory-Domäne und IAM -Rolle an. Weitere Informationen finden Sie unter Arbeiten mit Active Directory mit RDS für SQL Server.

Konsole

Sie können Ihre Optionsgruppe einer neuen oder vorhandenen DB-Instance zuordnen:

- Ordnen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie für eine vorhandene DB-Instance die Instance und ordnen Sie die neue Optionsgruppe zu. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

CLI

Sie können Ihre Optionsgruppe einer neuen oder vorhandenen DB-Instance zuordnen.

So erstellen Sie eine DB-Instance, die Ihre Optionsgruppe verwendet

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Optionsgruppe verwendet haben.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier myssrsinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password  \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssrs-se-2017
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssrsinstance ^
    --db-instance-class db.m5.2xlarge ^
```

```
    --engine sqlserver-se ^
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssrs-se-2017
```

So ändern Sie eine DB-Instance für die Verwendung Ihrer Optionsgruppe

- Führen Sie einen der folgenden Befehle aus.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds modify-db-instance \
      --db-instance-identifier myssrsinstance \
      --option-group-name ssrs-se-2017 \
      --apply-immediately
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds modify-db-instance ^
      --db-instance-identifier myssrsinstance ^
      --option-group-name ssrs-se-2017 ^
      --apply-immediately
  ```

Zulassen des eingehenden Zugriffs auf Ihre VPC Sicherheitsgruppe

Um eingehenden Zugriff auf die VPC Sicherheitsgruppe zu ermöglichen, die Ihrer DB-Instance zugeordnet ist, erstellen Sie eine eingehende Regel für den angegebenen SSRS Listener-Port. Weitere Informationen zum Einrichten von Sicherheitsgruppen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

Berichtsserverdatenbanken

Wenn Ihre DB-Instance mit der SSRS Option verknüpft ist, werden zwei neue Datenbanken auf Ihrer DB-Instance erstellt:

- `rdsadmin_ReportServer`
- `rdsadmin_ReportServerTempDB`

Diese Datenbanken fungieren als die ReportServer und ReportServerTemp DB-Datenbanken. SSRSspeichert seine Daten in der ReportServer Datenbank und speichert seine Daten in der ReportServerTemp DB-Datenbank zwischen. Weitere Informationen finden Sie unter Berichtsserverdatenbank in der Microsoft-Dokumentation.

RDSbesitzt und verwaltet diese Datenbanken, sodass Datenbankoperationen auf ihnen wie ALTER und DROP nicht zulässig sind. Der Zugriff auf die Datenbank `rdsadmin_ReportServerTempDB` ist nicht zulässig. Sie können jedoch Lesevorgänge in der Datenbank `rdsadmin_ReportServer` ausführen.

SSRSProtokolldateien

Sie können SSRS Protokolldateien auflisten, anzeigen und herunterladen. SSRSProtokolldateien folgen der Namenskonvention ReportServerService *`_timestamp`*.log. Diese Berichtsserver-Protokolle befinden sich im Verzeichnis `D:\rdsdbdata\Log\SSRS`. (Das `D:\rdsdbdata\Log` Verzeichnis ist auch das übergeordnete Verzeichnis für Fehlerprotokolle und SQL Server-Agent-Protokolle.) Weitere Informationen finden Sie unter Anzeigen und Auflisten von Datenbank-Protokolldateien.

Bei vorhandenen SSRS Instanzen muss der SSRS Dienst möglicherweise neu gestartet werden, um auf die Berichtsserverprotokolle zuzugreifen. Sie können den Service neu starten, indem Sie die `SSRS`-Option aktualisieren.

Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS for Microsoft SQL Server Server-Logs.

Zugreifen auf das SSRS Webportal

Gehen Sie wie folgt vor, um auf das SSRS Webportal zuzugreifen:

1. Aktivieren Sie Secure Sockets Layer (SSL).

2. Gewähren Sie Domänenbenutzern Zugriff.
3. Greifen Sie mit einem Browser und den Anmeldeinformationen des Domänenbenutzers auf das Webportal zu.

Benutzt SSL ein RDS

SSRSverwendet das HTTPS SSL Protokoll für seine Verbindungen. Um mit diesem Protokoll zu arbeiten, importieren Sie ein SSL Zertifikat in das Microsoft Windows-Betriebssystem auf Ihrem Client-Computer.

Weitere Informationen zu SSL Zertifikaten finden Sie unter. Weitere Hinweise zur Verwendung SSL mit SQL Server finden Sie unterVerwendung SSL mit einer Microsoft SQL Server-DB-Instance.

Gewähren des Zugriffs für Domänenbenutzer

Bei einer neuen SSRS Aktivierung gibt es keine Rollenzuweisungen inSSRS. Um einem Domänenbenutzer oder einer Benutzergruppe Zugriff auf das Webportal zu gewähren, steht eine gespeicherte Prozedur zur RDS Verfügung.

So erteilen Sie einem Domänenbenutzer Zugriff auf das Webportal

- Verwenden Sie die folgende gespeicherte Prozedur.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_GRANT_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

Dem Domänenbenutzer oder der Benutzergruppe wird die `RDS_SSRS_ROLE`-Systemrolle erteilt. Diese Rolle hat die folgenden Aufgaben auf Systemebene erteilt:

- Führen Sie Berichte aus
- Verwalten von Aufträgen
- Verwalten von freigegebenen Zeitplänen
- Anzeigen von freigegebenen Zeitplänen

Die Rolle `Content Manager` auf Elementebene im Stammordner wird ebenfalls gewährt.

Zugriff auf das Webportal

Nachdem die SSRS_GRANT_PORTAL_PERMISSION-Aufgabe erfolgreich abgeschlossen wurde, haben Sie Zugriff auf das Portal über einen Webbrowser. Das Webportal URL hat das folgende Format.

```
https://rds_endpoint:port/Reports
```

In diesem Format gilt Folgendes:

- *rds_endpoint* — Der Endpunkt für die RDS DB-Instance, mit der Sie arbeiten. SSRS

  Sie finden den Endpunkt auf der Registerkarte Konnektivität und Sicherheit für Ihre DB-Instance. Weitere Informationen finden Sie unter Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen.
- *port*— Der Listener-PortSSRS, den Sie in der SSRS Option festgelegt haben.

So greifen Sie auf das Webportal zu

1. Rufen Sie das Webportal URL in Ihrem Browser auf.

   ```
   https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/Reports
   ```
2. Melden Sie sich mit den Anmeldeinformationen für einen Domänenbenutzer an, dem Sie Zugriff mit der SSRS_GRANT_PORTAL_PERMISSION-Aufgabe gewährt haben.

Bereitstellen von Berichten und Konfigurieren von Berichtsdatenquellen

Gehen Sie wie folgt vor, um Berichte für Berichtsdatenquellen bereitzustellen SSRS und zu konfigurieren:

Themen

- Bereitstellen von Berichten für SSRS
- Konfigurieren der Berichtsdatenquelle

Bereitstellen von Berichten für SSRS

Nachdem Sie Zugriff auf das Webportal haben, können Sie Berichte für das Webportal bereitstellen. Sie können das Upload-Tool im Webportal verwenden, um Berichte hochzuladen, oder die Bereitstellung direkt über die SQLServerdatentools (SSDT) durchführen. Stellen Sie bei der Bereitstellung von SSDT Folgendes sicher:

- Der Benutzer, der gestartet SSDT hat, hat Zugriff auf das SSRS Webportal.
- Der `TargetServerURL` Wert in den SSRS Projekteigenschaften ist auf den HTTPS Endpunkt der RDS DB-Instance gesetzt`ReportServer`, an den beispielsweise Folgendes angehängt wird:

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/ReportServer
```

Konfigurieren der Berichtsdatenquelle

Nachdem Sie einen Bericht bereitgestellt habenSSRS, sollten Sie die Berichtsdatenquelle konfigurieren. Achten Sie beim Konfigurieren der Berichtsdatenquelle auf Folgendes:

- Verwenden Sie RDS für SQL Server-DB-Instances AWS Directory Service for Microsoft Active Directory, denen Sie beigetreten sind, den vollqualifizierten Domänennamen (FQDN) als Datenquellennamen der Verbindungszeichenfolge. Ein Beispiel ist *`myssrsinstance.corp-ad.example.com`*, wobei *`myssrsinstance`* der Name der DB-Instance und *`corp-ad.example.com`* der vollqualifizierte Domänenname ist.
- Verwenden Sie RDS für SQL Server-DB-Instances, die mit dem selbstverwalteten Active Directory verknüpft sind`.`, oder *`LocalHost`* als Datenquellennamen der Verbindungszeichenfolge.

Verwenden von SSRS E-Mail zum Senden von Berichten

SSRS enthält die SSRS E-Mail-Erweiterung, mit der Sie Berichte an Benutzer senden können.

Um SSRS Email zu konfigurieren, verwenden Sie die SSRS-Optionseinstellungen. Weitere Informationen finden Sie unter Hinzufügen der SSRS Option zu Ihrer Optionsgruppe.

Nachdem Sie SSRS Email konfiguriert haben, können Sie Berichte auf dem Berichtsserver abonnieren. Weitere Informationen finden Sie unter E-Mail-Versand in Reporting Services in der Microsoft-Dokumentation.

Die Integration mit AWS Secrets Manager ist erforderlich, damit SSRS Email auf RDS funktioniert. Um mit Secrets Manager zu integrieren, erstellen Sie ein Secret.

> **Note**
>
> Wenn Sie das Geheimnis später ändern, müssen Sie auch die SSRS-Option in der Optionsgruppe aktualisieren.

Ein Secret für SSRS-E-Mail erstellen

1. Befolgen Sie die Schritte unter Erstellen eines Geheimnisses im AWS Secrets Manager Benutzerhandbuch.

   a. Wählen Sie für Select secret type (Secret-Typ auswählen) die Option Other type of secrets (Anderer Secret-Typ) aus.

   b. Für Schlüssel-Wert-Paare geben Sie Folgendes ein:

      - **`SMTP_USERNAME`** – Geben Sie einen Benutzer ein, der berechtigt ist, E-Mails vom SMTP-Server zu senden.
      - **`SMTP_PASSWORD`** – Geben Sie ein Passwort für den SMTP-Benutzer ein.

   c. Verwenden Sie für den Verschlüsselungscode nicht die Standardeinstellung AWS KMS key. Verwenden Sie Ihren eigenen vorhandenen Schlüssel oder erstellen Sie einen neuen.

      Die KMS-Schlüsselrichtlinie muss die `kms:Decrypt`-Aktion zulassen, zum Beispiel:

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

2. Führen Sie die Schritte unter Anhängen einer Berechtigungsrichtlinie an ein Geheimnis im AWS Secrets Manager Benutzerhandbuch aus. Die Berechtigungsrichtlinie gibt die `secretsmanager:GetSecretValue`-Aktion an den `rds.amazonaws.com` Service Principal weiter.

   Wir empfehlen Ihnen, die `aws:sourceAccount` und `aws:sourceArn` Bedingungen in der Police zu verwenden, um das Problem des verwirrten Vertreters zu vermeiden. Verwenden Sie Ihren AWS-Konto for `aws:sourceAccount` und den ARN für die Optionsgruppe`aws:sourceArn`. Weitere Informationen finden Sie unter Vermeidung des dienstübergreifenden Confused-Deputy-Problems.

   Das folgende Beispiel zeigt eine Berechtigungsrichtlinie.

   JSON

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Principal" : {
      "Service" : "rds.amazonaws.com"
    },
    "Action" : "secretsmanager:GetSecretValue",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "aws:sourceAccount" : "123456789012"
      },
      "ArnLike" : {
        "aws:sourceArn" : "arn:aws:rds:us-west-2:123456789012:og:ssrs-
se-2017"
      }
    }
  } ]
}
```

   Weitere Beispiele finden Sie unter Beispiele für Berechtigungsrichtlinien für AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

Widerrufen von Berechtigungen auf Systemebene

Die RDS_SSRS_ROLE-Systemrolle verfügt nicht über ausreichende Berechtigungen zum Löschen von Rollenzuweisungen auf Systemebene. Verwenden Sie zum Entfernen eines Benutzers oder einer Benutzergruppe aus RDS_SSRS_ROLE dieselbe gespeicherte Prozedur, die Sie zum Erteilen der Rolle verwendet haben, verwenden Sie jedoch den SSRS_REVOKE_PORTAL_PERMISSION-Aufgabentyp.

So widerrufen Sie den Zugriff eines Domänenbenutzers für das Webportal

- Verwenden Sie die folgende gespeicherte Prozedur.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_REVOKE_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

Dadurch wird der Benutzer aus der RDS_SSRS_ROLE-Systemrolle gelöscht. Außerdem wird der Benutzer aus der Content Manager-Rolle auf Elementebene gelöscht, wenn der Benutzer sie hat.

Überwachung des Status einer Aufgabe

Rufen Sie die rds_fn_task_status-Funktion auf, um den Status Ihrer Erteilungs- oder Widerrufsaufgabe zu verfolgen. Dazu sind zwei Parameter erforderlich. Der erste Parameter sollte immer seinNULL, weil er nicht gilt fürSSRS. Der zweite Parameter akzeptiert eine Aufgaben-ID.

Um eine Liste aller Aufgaben anzuzeigen, setzen Sie den ersten Parameter auf NULL und den zweiten Parameter auf 0, wie im folgenden Beispiel gezeigt.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Um eine bestimmte Aufgabe zu erhalten, setzen Sie den ersten Parameter auf NULL und den zweiten Parameter auf die Aufgaben-ID, wie im folgenden Beispiel gezeigt,

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

Die rds_fn_task_status-Funktion gibt die folgenden Informationen zurück.

Ausgabeparameter	Beschreibung
`task_id`	Die ID der Aufgabe.
`task_type`	Denn SSRS Aufgaben können die folgenden Aufgabentypen haben: • `SSRS_GRANT_PORTAL_PERMISSION` • `SSRS_REVOKE_PORTAL_PERMISSION`
`database_name`	Gilt nicht für SSRS Aufgaben.
`% complete`	Verlauf der Aufgabe als Prozentwert.
`duration (mins)`	Zeitdauer für die Ausführung der Aufgabe (in Minuten).
`lifecycle`	Der Status der Aufgabe. Die folgenden Status sind möglich: • `CREATED`— Nachdem Sie eine der SSRS gespeicherten Prozeduren aufgerufen haben, wird eine Aufgabe erstellt und ihr Status wird auf gesetzt`CREATED`. • `IN_PROGRESS` – Nach dem Start einer Aufgabe wird der Status auf gesetzt `IN_PROGRESS` . Es kann bis zu fünf Minuten dauern, bis sich der Status von `CREATED` zu `IN_PROGRESS` ändert. • `SUCCESS` – Nach dem Abschluss einer Aufgabe wird der Status auf gesetzt `SUCCESS`. • `ERROR` – Wenn eine Aufgabe fehlschlägt, wird der Status auf gesetzt `ERROR`. Weitere

Ausgabeparameter	Beschreibung
	Informationen über den Fehler können Sie der Spalte `task_info` entnehmen. • `CANCEL_REQUESTED` – Nachdem Sie die gespeicherte Prozedur `rds_cancel_task` aufgerufen haben, wird der Status der Aufgabe auf `CANCEL_REQUESTED` gesetzt. • `CANCELLED` – Nachdem die Aufgabe abgebrochen wurde, wird der Status der Aufgabe auf gesetzt `CANCELLED` .
`task_info`	Zusätzliche Informationen über die Aufgabe. Wenn bei der Verarbeitung ein Fehler auftritt, enthält diese Spalte Informationen zu dem Fehler.
`last_updated`	Datum und Uhrzeit der letzten Aktualisierung des Aufgabenstatus.
`created_at`	Datum und Uhrzeit, an denen die Aufgabe angelegt wurde.
`S3_object_arn`	Gilt nicht für SSRS Aufgaben.
`overwrite_S3_backup_file`	Gilt nicht für SSRS Aufgaben.
`KMS_master_key_arn`	Gilt nicht für SSRS Aufgaben.
`filepath`	Gilt nicht für SSRS Aufgaben.
`overwrite_file`	Gilt nicht für SSRS Aufgaben.
`task_metadata`	Mit der SSRS Aufgabe verknüpfte Metadaten.

Datenbanken deaktivieren und löschen SSRS

Gehen Sie wie folgt vor, um SSRS Datenbanken zu deaktivieren SSRS und zu löschen:

Themen

- Deaktivieren von SSRS
- Löschen der SSRS Datenbanken

Deaktivieren von SSRS

Um sie zu deaktivierenSSRS, entfernen Sie die `SSRS` Option aus ihrer Optionsgruppe. Durch das Entfernen der Option werden die SSRS Datenbanken nicht gelöscht. Weitere Informationen finden Sie unter Löschen der SSRS Datenbanken.

Sie können sie wieder SSRS einschalten, indem Sie die `SSRS` Option wieder hinzufügen. Wenn Sie auch die SSRS Datenbanken gelöscht haben, werden durch das Lesen der Option auf derselben DB-Instance neue Berichtsserver-Datenbanken erstellt.

Konsole

Um die SSRS Option aus ihrer Optionsgruppe zu entfernen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der Option `SSRS` (`ssrs-se-2017` in den vorherigen Beispielen).
4. Wählen Sie Delete option (Option löschen) aus.
5. Wählen Sie unter Löschoptionen SSRSdie Option Optionen zum Löschen aus.
6. Wählen Sie unter Apply immediately (Sofort anwenden) die Option Yes (Ja) aus, um die Option sofort zu löschen, oder No (Nein), um sie während des nächsten Wartungsfensters zu löschen.
7. Wählen Sie Löschen.

CLI

Um die SSRS Option aus ihrer Optionsgruppe zu entfernen

- Führen Sie einen der folgenden Befehle aus.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssrs-se-2017 \
    --options SSRS \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssrs-se-2017 ^
    --options SSRS ^
    --apply-immediately
```

Löschen der SSRS Datenbanken

Wenn Sie die SSRS-Option entfernen, werden die Berichtsserverdatenbanken nicht gelöscht. Verwenden Sie die folgende gespeicherte Prozedur, um sie zu löschen.

Um die Berichtsserverdatenbanken zu löschen, müssen Sie zuerst die SSRS-Option entfernen.

Um die SSRS Datenbanken zu löschen

- Verwenden Sie die folgende gespeicherte Prozedur.

```
exec msdb.dbo.rds_drop_ssrs_databases
```

Support für Microsoft Distributed Transaction Coordinator RDS für SQL Server

Eine verteilte Transaktion ist eine Datenbanktransaktion, an der zwei oder mehr Netzwerkhosts beteiligt sind. RDSfor SQL Server unterstützt verteilte Transaktionen zwischen Hosts, wobei ein einzelner Host einer der folgenden sein kann:

- RDSfür SQL Server-DB-Instance
- Lokaler SQL Serverhost
- EC2Amazon-Host mit installiertem SQL Server
- Jeder andere EC2 Host oder jede andere RDS DB-Instance mit einer Datenbank-Engine, die verteilte Transaktionen unterstützt

Ab SQL Server 2012 (Version 11.00.5058.0.v1 und höher) unterstützen alle Editionen von RDS for Server verteilte Transaktionen. RDS SQL Der Support wird mithilfe von Microsoft Distributed Transaction Coordinator (MSDTC) bereitgestellt. Ausführliche Informationen zu MSDTC finden Sie unter Distributed Transaction Coordinator in der Microsoft-Dokumentation.

Inhalt

- Einschränkungen
- Aktivieren MSDTC
  - Die Optionsgruppe wird erstellt für MSDTC
  - Die MSDTC Option zur Optionsgruppe hinzufügen
  - Erstellen der Parametergruppe für MSDTC
  - Ändern des Parameters für MSDTC
  - Zuordnen der Options- und Parametergruppe zur DB-Instance
  - Änderung der MSDTC Option
- Transaktionen verwenden
  - Verwenden verteilter Transaktionen
  - XA-Transaktionen verwenden
  - Verwenden der Transaktionsnachverfolgung
- Deaktivierung MSDTC
- Problembehandlung MSDTC für RDS für SQL Server

Einschränkungen

Die folgenden Einschränkungen gelten für die Verwendung von MSDTC on RDS for SQL Server:

- MSDTCwird auf Instanzen, die SQL Server Database Mirroring verwenden, nicht unterstützt. Weitere Informationen finden Sie unter Transaktionen - Verfügbarkeitsgruppen und Datenbankspiegelung.
- Der Parameter `in-doubt xact resolution` muss auf 1 oder 2 gesetzt werden. Weitere Informationen finden Sie unter Ändern des Parameters für MSDTC.
- MSDTCsetzt voraus, dass alle Hosts, die an verteilten Transaktionen teilnehmen, mit ihren Hostnamen auflösbar sind. RDSbehält diese Funktionalität für domänengebundene Instanzen automatisch bei. Stellen Sie bei eigenständigen Instanzen jedoch sicher, dass Sie den DNS Server manuell konfigurieren.
- Java Database Connectivity (JDBC) XA-Transaktionen werden für SQL Server 2017 Version 14.00.3223.3 und höher sowie für Server 2019 unterstützt. SQL
- Verteilte Transaktionen, die von Client-Dynamic-Link-Bibliotheken (DLLs) auf RDS Instanzen abhängen, werden nicht unterstützt.
- Die Verwendung benutzerdefinierter dynamischer XA-Verknüpfungsbibliotheken wird nicht unterstützt.

Aktivieren MSDTC

Gehen Sie wie folgt vor, um die Aktivierung MSDTC für Ihre DB-Instance durchzuführen:

1. Erstellen Sie eine neue Optionsgruppe oder wählen Sie eine bestehende Optionsgruppe aus.
2. Fügen Sie die Option `MSDTC` zur Optionsgruppe hinzu.
3. Erstellen Sie eine neue Parametergruppe oder wählen Sie eine vorhandene Parametergruppe aus.
4. Ändern Sie die Parametergruppe, um den Parameter `in-doubt xact resolution` auf 1 oder 2 festzulegen.
5. Ordnen Sie die Optionsgruppe und die Parametergruppe der DB-Instance zu.

Die Optionsgruppe wird erstellt für MSDTC

Verwenden Sie AWS Management Console oder AWS CLI , um eine Optionsgruppe zu erstellen, die der SQL Server-Engine und der Version Ihrer DB-Instance entspricht.

> **Note**
>
> Sie können auch eine vorhandene Optionsgruppe verwenden, wenn sie für die richtige SQL Server-Engine und Version bestimmt ist.

Konsole

Mit dem folgenden Verfahren wird eine Optionsgruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Optionsgruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie Create group (Gruppe erstellen) aus.
4. Führen Sie im Bereich Create option group (Optionsgruppe erstellen) Folgendes aus:
   a. Geben Sie unter Name einen Namen für die Optionsgruppe ein, der innerhalb Ihres AWS Kontos eindeutig ist, z. **`msdtc-se-2016`** B. Der Name darf nur Buchstaben, Ziffern und Bindestriche enthalten.
   b. Geben Sie unter Beschreibung eine kurze Beschreibung der Optionsgruppe ein, z. B. **`MSDTC option group for SQL Server SE 2016`**. Die Beschreibung ist nur zur Information.
   c. Wählen Sie für Engine die Option sqlserver-se aus.
   d. Wählen Sie im Feld Engine-Hauptversion 13.00 aus.
5. Wählen Sie Create aus.

CLI

Im folgenden Beispiel wird eine Optionsgruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Optionsgruppe

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-option-group \
    --option-group-name msdtc-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-option-group ^
    --option-group-name msdtc-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Die MSDTC Option zur Optionsgruppe hinzufügen

Verwenden Sie als Nächstes das AWS Management Console oder das AWS CLI , um die `MSDTC` Option zur Optionsgruppe hinzuzufügen.

Die folgenden Optionseinstellungen sind erforderlich:

- Port — Der Port, den Sie für den Zugriff verwendenMSDTC. Zulässige Werte sind 1150–49151 mit Ausnahme von 1234, 1434, 3260, 3343, 3389 und 47001. Der Standardwert ist 5000.

  Stellen Sie sicher, dass der zu verwendende Port in den Firewall-Regeln aktiviert ist. Stellen Sie außerdem sicher, dass dieser Port bei Bedarf in den ein- und ausgehenden Regeln für die Sicherheitsgruppe aktiviert ist, die Ihrer DB-Instance zugeordnet ist. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.
- Sicherheitsgruppen — Die VPC Sicherheitsgruppenmitgliedschaften für Ihre RDS DB-Instance.
- Authentifizierungstyp – Der Authentifizierungsmodus zwischen Hosts. Die folgenden Authentifizierungstypen werden unterstützt:
  - Gegenseitig — Die RDS Instances werden mithilfe der integrierten Authentifizierung gegenseitig authentifiziert. Wenn diese Option ausgewählt ist, müssen alle Instances, die dieser Optionsgruppe zugeordnet sind, einer Domäne zugeordnet sein.
  - Keine – Es wird keine Authentifizierung zwischen Hosts durchgeführt. Es wird nicht empfohlen, diesen Modus in Produktionsumgebungen zu verwenden.

- Größe des Transaktionsprotokolls — Die Größe des MSDTC Transaktionsprotokolls. Zulässige Werte sind 4–1024 MB. Die Standardgröße beträgt 4 MB.

Die folgenden Optionseinstellungen sind optional:

- Eingehende Verbindungen aktivieren — Gibt an, ob eingehende MSDTC Verbindungen zu Instances zugelassen werden sollen, die dieser Optionsgruppe zugeordnet sind.
- Ausgehende Verbindungen aktivieren — Gibt an, ob ausgehende MSDTC Verbindungen von Instances zugelassen werden sollen, die dieser Optionsgruppe zugeordnet sind.
- XA aktivieren – Gibt an, ob XA-Transaktionen zugelassen werden sollen. Weitere Informationen zum XA-Protokoll finden Sie unter XA-Spezifikation.
- SNALU aktivieren — Gibt an, ob das SNA LU-Protokoll für verteilte Transaktionen verwendet werden darf. Weitere Informationen zur SNA LU-Protokollunterstützung finden Sie in der Microsoft-Dokumentation unter IBM CICS LU 6.2-Transaktionen verwalten.

Konsole

Um die MSDTC Option hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe aus, die Sie gerade erstellt haben.
4. Wählen Sie Add option (Option hinzufügen).
5. Wählen Sie unter Optionsdetails MSDTCden Optionsnamen aus.
6. Unter Optionseinstellungen:
    a. Geben Sie unter Port die Portnummer für den Zugriff einMSDTC. Der Standardwert ist 5000.
    b. Wählen Sie für Sicherheitsgruppen die VPC Sicherheitsgruppe aus, die der Option zugeordnet werden soll.
    c. Wählen Sie für Authentifizierungstyp die Option Gegenseitig oder Keine aus.
    d. Geben Sie für Transaktionsprotokollgröße einen Wert von 4–1024 ein. Der Standardwert ist 4.
7. Führen Sie unter Zusätzliche Konfiguration die folgenden Schritte aus:

a. Wählen Sie für Verbindungen nach Bedarf die Option Eingehende Verbindungen aktivieren und Ausgehende Verbindungen aktivieren aus.

b. Wählen Sie bei Bedarf unter Zulässige Protokolle die Optionen XA aktivieren und SNALU aktivieren aus.

8. Wählen Sie unter Scheduling (Planung) aus, ob die Option sofort oder während des nächsten Wartungsfensters hinzugefügt werden soll.

9. Wählen Sie Add option (Option hinzufügen).

   Um diese Option hinzuzufügen, ist kein Neustart erforderlich.

CLI

Um die MSDTC Option hinzuzufügen

1. Erstellen Sie beispielsweise `msdtc-option.json` eine JSON Datei mit den folgenden erforderlichen Parametern.

```
{
"OptionGroupName":"msdtc-se-2016",
"OptionsToInclude": [
 {
 "OptionName":"MSDTC",
 "Port":5000,
 "VpcSecurityGroupMemberships":["sg-0abcdef123"],
 "OptionSettings":[{"Name":"AUTHENTICATION","Value":"MUTUAL"},
{"Name":"TRANSACTION_LOG_SIZE","Value":"4"}]
 }],
"ApplyImmediately": true
}
```

2. Fügen Sie die Option `MSDTC` zur Optionsgruppe hinzu.

   Example

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://msdtc-option.json \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://msdtc-option.json ^
    --apply-immediately
```

Es ist kein Neustart erforderlich.

Erstellen der Parametergruppe für MSDTC

Erstellen oder ändern Sie eine Parametergruppe für den `in-doubt xact resolution` Parameter, der der SQL Server-Edition und Version Ihrer DB-Instance entspricht.

Konsole

Im folgenden Beispiel wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie Create parameter group (Parametergruppe erstellen).
4. Führen Sie im Bereich Parametergruppe erstellen die folgenden Schritte aus:
   a. Wählen Sie für Parametergruppenfamilie die Option sqlserver-se-13.0 aus.
   b. Geben Sie unter Gruppenname einen Bezeichner für die Parametergruppe ein, z. B. **`msdtc-sqlserver-se-13`**.
   c. Geben Sie für Beschreibung den Text **`in-doubt xact resolution`** ein.
5. Wählen Sie Create (Erstellen) aus.

CLI

Im folgenden Beispiel wird eine Parametergruppe für SQL Server Standard Edition 2016 erstellt.

So erstellen Sie die Parametergruppe

- Verwenden Sie einen der folgenden Befehle.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "in-doubt xact resolution"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "in-doubt xact resolution"
```

Ändern des Parameters für MSDTC

Ändern Sie den `in-doubt xact resolution` Parameter in der Parametergruppe, die der SQL Server-Edition und Version Ihrer DB-Instance entspricht.

Stellen Sie für MSDTC den `in-doubt xact resolution` Parameter einen der folgenden Werte ein:

- 1—`Presume commit`. Bei allen MSDTC zweifelhaften Transaktionen wird davon ausgegangen, dass sie begangen wurden.
- 2—. `Presume abort` Es wird davon ausgegangen, dass alle MSDTC zweifelhaften Transaktionen eingestellt wurden.

Weitere Informationen finden Sie unter Lösung für unklare Transaktion (Serverkonfigurationsoption) in der Microsoft-Dokumentation.

Konsole

Im folgenden Beispiel wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben.

So ändern Sie die Parametergruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie die Parametergruppe aus, z. B. msdtc-sqlserver-se-13.
4. Filtern Sie unter Parameter die Parameterliste nach **`xact`**.
5. Wählen Sie in-doubt xact resolution aus.
6. Wählen Sie Parameter bearbeiten aus.
7. Geben Sie **1** oder **2** ein.
8. Wählen Sie Änderungen speichern.

CLI

Im folgenden Beispiel wird die Parametergruppe geändert, die Sie für SQL Server Standard Edition 2016 erstellt haben.

So ändern Sie die Parametergruppe

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

Zuordnen der Options- und Parametergruppe zur DB-Instance

Sie können das AWS Management Console oder das verwenden AWS CLI , um die MSDTC Optionsgruppe und die Parametergruppe mit der DB-Instance zu verknüpfen.

Konsole

Sie können die MSDTC Optionsgruppe und die Parametergruppe einer neuen oder vorhandenen DB-Instance zuordnen.

- Ordnen Sie sie bei einer neuen DB-Instance zu, wenn Sie die Instance starten. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ordnen Sie sie für eine vorhandene DB-Instance zu, indem Sie die Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Note**
>
> Wenn Sie eine bestehende DB-Instance verwenden, die in eine Domäne eingebunden ist, muss dieser bereits eine Active Directory-Domäne und eine AWS Identity and Access Management (IAM) -Rolle zugeordnet sein. Wenn Sie eine neue in die Domäne eingebundene Instanz erstellen, geben Sie eine bestehende Active Directory-Domäne und -Rolle an. IAM Weitere Informationen finden Sie unter Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server.

CLI

Sie können die MSDTC Optionsgruppe und die Parametergruppe einer neuen oder vorhandenen DB-Instance zuordnen.

> **Note**
>
> Wenn Sie eine bestehende DB-Instance verwenden, die in eine Domäne eingebunden ist, muss dieser bereits eine Active Directory-Domäne und eine zugeordnete IAM Rolle vorhanden sein. Wenn Sie eine neue in eine Domäne eingebundene Instance erstellen, geben Sie eine bestehende Active Directory-Domäne und -Rolle an. IAM Weitere Informationen finden Sie unter Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server.

Um eine DB-Instance mit der MSDTC Optionsgruppe und der Parametergruppe zu erstellen

- Geben Sie denselben DB-Engine-Typ und dieselbe Hauptversion an, die Sie beim Erstellen der Optionsgruppe verwendet haben.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13
```

Um eine DB-Instance zu ändern und die MSDTC Optionsgruppe und die Parametergruppe zuzuordnen

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --apply-immediately
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --apply-immediately
```

Änderung der MSDTC Option

Nachdem Sie die Option `MSDTC` aktiviert haben, können Sie ihre Einstellungen ändern. Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung.

> **Note**
>
> Bei einigen Änderungen an den MSDTC Optionseinstellungen muss der MSDTC Dienst neu gestartet werden. Diese Anforderung kann sich auf die Ausführung verteilter Transaktionen auswirken.

Transaktionen verwenden

Verwenden verteilter Transaktionen

In Amazon RDS for SQL Server führen Sie verteilte Transaktionen genauso aus wie verteilte Transaktionen, die lokal ausgeführt werden:

- Verwenden von. NETFramework `System.Transactions` Promotable Transactions, das verteilte Transaktionen optimiert, indem ihre Erstellung verzögert wird, bis sie benötigt werden.

  In diesem Fall erfolgt die Heraufstufung automatisch und erfordert keine Intervention. Wenn nur ein Ressourcenmanager innerhalb der Transaktion vorhanden ist, wird keine Heraufstufung durchgeführt. Weitere Informationen zu impliziten Transaktionsbereichen finden Sie unter Implementieren einer impliziten Transaktion mit Transaktionsbereich in der Microsoft-Dokumentation.

  Damit werden förderbare Transaktionen unterstützt. NETImplementierungen:

  - Beginnend mitADO. NET2.0, `System.Data.SqlClient` unterstützt werbbare Transaktionen mit dem SQL Server. Weitere Informationen finden Sie unter System.Transactions integration with SQL Server in der Microsoft-Dokumentation.
  - ODP. NETunterstütztSystem.Transactions. Für die erste Verbindung, die im `TransactionsScope`-Bereich zu Oracle Database 11g Release 1 (Version 11.1) und höher geöffnet wurde, wird eine lokale Transaktion erstellt. Wenn eine zweite Verbindung geöffnet wird, wird diese Transaktion automatisch zu einer verteilten Transaktion heraufgestuft. Weitere Informationen zur Unterstützung verteilter Transaktionen finden Sie unterODP. NET, siehe Integration von Microsoft Distributed Transaction Coordinator in der Microsoft-Dokumentation.
- Verwenden der `BEGIN DISTRIBUTED TRANSACTION`-Anweisung. Weitere Informationen finden Sie unter BEGINDISTRIBUTEDTRANSACTION(Transact-SQL) in der Microsoft-Dokumentation.

XA-Transaktionen verwenden

Ab Version 14.00.3223.3 RDS für SQL Server 2017 können Sie verteilte Transaktionen mithilfe von steuern. JDBC Wenn Sie die `Enable XA` Optionseinstellung `true` in der Option auf setzen, RDS werden JDBC Transaktionen automatisch aktiviert und dem `MSDTC` Benutzer die Rolle zugewiesen. `SqlJDBCXAUser guest` Dies ermöglicht die Ausführung verteilter Transaktionen überJDBC. Weitere Informationen, einschließlich eines Code-Beispiels, finden Sie unter unter.XA-Transaktionenin der Microsoft-Dokumentation.

Verwenden der Transaktionsnachverfolgung

RDSunterstützt die Steuerung von MSDTC Transaktionsablaufverfolgungen und deren Herunterladen von der RDS DB-Instance zur Fehlerbehebung. Sie können Transaktionsablaufverfolgungssitzungen steuern, indem Sie die folgende RDS gespeicherte Prozedur ausführen.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'trace_action',
[@traceall='0|1'],
[@traceaborted='0|1'],
[@tracelong='0|1'];
```

Der folgende Parameter ist erforderlich:

- `trace_action` – Die Nachverfolgungsaktion. Sie kann `START`, `STOP` oder `STATUS` sein.

Die folgenden Parameter sind optional:

- `@traceall` – Setzen Sie auf 1, um alle verteilten Transaktionen nachzuverfolgen. Der Standardwert ist 0.
- `@traceaborted` – Setzen Sie auf 1, um abgebrochene verteilte Transaktionen nachzuverfolgen. Der Standardwert ist 0.
- `@tracelong` – Setzen Sie auf 1, um zeitintensive verteilte Transaktionen zu verfolgen. Der Standardwert ist 0.

Example der START Verfolgungsaktion

Um eine neue Transaktionsnachverfolgungssitzung zu starten, führen Sie die folgende Beispielanweisung aus.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'START',
@traceall='0',
@traceaborted='1',
@tracelong='1';
```

> **Note**
>
> Es kann nur eine Transaktionsnachverfolgungssitzung gleichzeitig aktiv sein. Wenn ein neuer START-Befehl für die Nachverfolgungssitzung ausgegeben wird, während

eine Nachverfolgungssitzung aktiv ist, wird ein Fehler zurückgegeben, und die aktive Nachverfolgungssitzung bleibt unverändert.

Example der STOP Rückverfolgungshandlung

Um eine Transaktionsnachverfolgungssitzung zu beenden, führen Sie die folgende Anweisung aus.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STOP'
```

Diese Anweisung beendet die aktive Transaktionsablaufverfolgungssitzung und speichert die Transaktionsverfolgungsdaten im Protokollverzeichnis der RDS DB-Instance. Die erste Zeile der Ausgabe enthält das Gesamtergebnis und die folgenden Zeilen zeigen Details der Operation an.

Im Folgenden finden Sie ein Beispiel für einen erfolgreichen Sitzungsstopp der Nachverfolgung.

```
OK: Trace session has been successfully stopped.
Setting log file to: D:\rdsdbdata\MSDTC\Trace\dtctrace.log
Examining D:\rdsdbdata\MSDTC\Trace\msdtctr.mof for message formats,  8 found.
Searching for TMF files on path: (null)
Logfile D:\rdsdbdata\MSDTC\Trace\dtctrace.log:
 OS version    10.0.14393  (Currently running on 6.2.9200)
 Start Time    <timestamp>
 End Time      <timestamp>
 Timezone is   @tzres.dll,-932 (Bias is 0mins)
 BufferSize            16384 B
 Maximum File Size     10 MB
 Buffers  Written      Not set (Logger may not have been stopped).
 Logger Mode Settings (11000002) ( circular paged
 ProcessorCount         1
Processing completed   Buffers: 1, Events: 3, EventsLost: 0 :: Format Errors: 0,
 Unknowns: 3
Event traces dumped to d:\rdsdbdata\Log\msdtc_<timestamp>.log
```

Mit den detaillierten Informationen können Sie den Namen der generierten Protokolldatei abfragen. Weitere Hinweise zum Herunterladen von Protokolldateien aus der RDS DB-Instance finden Sie unterÜberwachung von Amazon RDS Amazon.

Die Nachverfolgungssitzungsprotokolle verbleiben 35 Tage lang auf der Instance. Ältere Nachverfolgungssitzungsprotokolle werden automatisch gelöscht.

Example der STATUS Tracing-Aktion

Um den Status einer Transaktionsnachverfolgungssitzung nachzuverfolgen, führen Sie die folgende Anweisung aus.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STATUS'
```

Diese Anweisung gibt Folgendes als separate Zeilen der Ergebnismenge aus.

```
OK
SessionStatus: <Started|Stopped>
TraceAll: <True|False>
TraceAborted: <True|False>
TraceLongLived: <True|False>
```

Die erste Zeile gibt das Gesamtergebnis der Operation an: `OK` oder gegebenenfalls `ERROR` mit Details. Die folgenden Zeilen zeigen Details zum Status der Nachverfolgungssitzung an:

- `SessionStatus` kann einer der folgenden sein:
  - `Started`, wenn eine Nachverfolgungssitzung ausgeführt wird.
  - `Stopped`, wenn keine Nachverfolgungssitzung ausgeführt wird.
- Die Protokollierungssitzungs-Flags können `True` oder `False` abhängig davon sein, wie sie im `START`-Befehl festgelegt wurden.

Deaktivierung MSDTC

Um sie zu deaktivierenMSDTC, entfernen Sie die `MSDTC` Option aus ihrer Optionsgruppe.

Konsole

Um die MSDTC Option aus ihrer Optionsgruppe zu entfernen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Wählen Sie die Optionsgruppe mit der Option `MSDTC` (`msdtc-se-2016` in den vorherigen Beispielen).
4. Wählen Sie Delete option (Option löschen) aus.

5. Wählen Sie unter Löschoptionen MSDTCdie Option Zu löschende Optionen aus.
6. Wählen Sie unter Apply immediately (Sofort anwenden) die Option Yes (Ja) aus, um die Option sofort zu löschen, oder No (Nein), um sie während des nächsten Wartungsfensters zu löschen.
7. Wählen Sie Löschen.

CLI

Um die MSDTC Option aus ihrer Optionsgruppe zu entfernen

- Verwenden Sie einen der folgenden Befehle.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds remove-option-from-option-group \
    --option-group-name msdtc-se-2016 \
    --options MSDTC \
    --apply-immediately
```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name msdtc-se-2016 ^
    --options MSDTC ^
    --apply-immediately
```

Problembehandlung MSDTC für RDS für SQL Server

In einigen Fällen können Sie Probleme haben, eine Verbindung zwischen der MSDTC Ausführung auf einem Client-Computer und dem MSDTC Dienst herzustellen, der auf einer RDS for SQL Server-DB-Instance ausgeführt wird. Wenn ja, stellen Sie Folgendes sicher:

- Die eingehenden Regeln für die Sicherheitsgruppe, die der DB-Instance zugeordnet ist, sind korrekt konfiguriert. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.
- Der Client-Computer ist korrekt konfiguriert.
- Die MSDTC Firewallregeln auf Ihrem Client-Computer sind aktiviert.

So konfigurieren Sie den Client-Computer

1. Öffnen Sie Komponentendienste.

   Oder wählen Sie im Server-Manager die Option Tools und dann Komponentendienste aus.
2. Erweitern Sie Komponentendienste, erweitern Sie Computer, erweitern Sie Arbeitsplatz und erweitern Sie dann Distributed Transaction Coordinator.
3. Öffnen Sie das Kontextmenü (Rechtsklick) für Lokal DTC und wählen Sie Eigenschaften.
4. Wählen Sie die Registerkarte Sicherheit aus.
5. Wählen Sie alle der folgenden Optionen aus:

   - DTCNetzwerkzugriff
   - Eingehende zulassen
   - Ausgehende zulassen
6. Stellen Sie sicher, dass der richtige Authentifizierungsmodus gewählt ist:

   - Gegenseitige Authentifizierung erforderlich – Der Client-Computer ist mit derselben Domäne verbunden wie andere Knoten, die an einer verteilten Transaktion beteiligt sind, oder es ist eine Vertrauensstellung zwischen Domänen konfiguriert.
   - Keine Authentifizierung erforderlich – Alle anderen Fälle.
7. Wählen Sie OK aus, um Ihre Änderungen zu speichern.
8. Wenn Sie aufgefordert werden, den Dienst neu zu starten, wählen Sie Ja.

Um MSDTC Firewallregeln zu aktivieren

1. Öffnen Sie die Windows-Firewall und wählen Sie dann Erweiterte Einstellungen aus.

   Oder wählen Sie im Server-Manager die Option Tools und dann Windows-Firewall mit erweiterter Sicherheit aus.

   > **Note**
   >
   > Abhängig von Ihrem Betriebssystem wird die Windows-Firewall möglicherweise als Windows Defender Firewall bezeichnet.
2. Wählen Sie im linken Bereich die Option Eingehende Regeln aus.

3. Aktivieren Sie die folgenden Firewall-Regeln, wenn sie noch nicht aktiviert sind:

   - Koordinator für verteilte Transaktionen (RPC)
   - Koordinator für verteilte Transaktionen (RPC) - EPMAP
   - Koordinator für verteilte Transaktionen (TCP-In)

4. Schließen Sie die Windows-Firewall.

Allgemeine DBA-Aufgaben für Amazon RDS for Microsoft SQL Server

Dieser Abschnitt beschreibt die Amazon RDS-spezifischen Implementierungen einiger häufiger DBA-Aufgaben für DB-Instances, auf denen die Microsoft SQL Server-Datenbank-Engine ausgeführt wird. Um eine verwaltete Service-Erfahrung zu bieten, stellt Amazon RDS keinen Shell-Zugriff auf DB-Instances bereit, und beschränkt den Zugriff auf bestimmte Systemprozeduren und -tabellen, die erweiterte Sonderrechte erfordern.

> **Note**
>
> Wenn Sie mit einer SQL Server DB-Instance arbeiten, können Sie Skripts für die Änderung einer neu erstellten Datenbank ausführen. Sie können jedoch nicht die [Modell]-Datenbank ändern, die von der Datenbank als Modell für neue Datenbanken verwendet wurde.

Themen

- Zugreifen auf die tempdb-Datenbank auf Microsoft SQL Server-DB-Instances auf Amazon RDS
- Analysieren Ihrer Datenbank-Arbeitslast auf einer Amazon RDS for SQL Server-DB-Instance mit Database Engine Tuning Advisor
- Ändern db_owner Sie das auf das rdsa Konto für Ihre Amazon RDS for SQL Server-Datenbank
- Verwaltung von Sortierungen und Zeichensätzen für Amazon RDS for Microsoft SQL Server
- Einen Datenbankbenutzer für Amazon RDS for SQL Server erstellen
- Ermitteln eines Wiederherstellungsmodells für Ihre Amazon RDS for SQL Server-Datenbank
- Ermitteln der letzten Failover-Zeit für Amazon RDS for Server SQL
- Behebung von point-in-time-recovery Fehlern aufgrund einer Lücke in der Protokollsequenznummer
- Anzeige von Datenbanknamen für Amazon RDS for SQL Server verweigern oder zulassen
- Deaktivierung von Fast Inserts beim Massenladen für Amazon RDS for Server SQL
- Löschen einer Datenbank in einer Amazon RDS for Microsoft SQL Server Server-DB-Instance
- Umbenennen einer Amazon RDS for Microsoft SQL Server Server-Datenbank in einer Multi-AZ-Bereitstellung
- Zurücksetzen der db_owner-Rollenmitgliedschaft für den Master-Benutzer für Amazon for Server RDS SQL

- Wiederherstellung von DB-Instances mit gekündigter Lizenz für Amazon for Server RDS SQL
- Umstellung einer Amazon RDS for SQL Server-Datenbank von zu OFFLINE ONLINE
- Verwenden von Change Data Capture für Amazon RDS for SQL Server
- Verwenden des SQL Server-Agenten für Amazon RDS
- Arbeiten mit Amazon RDS for Microsoft SQL Server Server-Logs
- Arbeiten mit Trace- und Dump-Dateien für Amazon RDS for Server SQL

Zugreifen auf die tempdb-Datenbank auf Microsoft SQL Server-DB-Instances auf Amazon RDS

Sie können auf die `tempdb` Datenbank Ihrer Microsoft SQL Server-DB-Instances bei Amazon zugreifenRDS. Sie können Code auf ausführen, `tempdb` indem Sie Transact- SQL über Microsoft SQL Server Management Studio (SSMS) oder eine andere SQL Standard-Client-Anwendung verwenden. Weitere Informationen zum Herstellen einer Verbindung zur DB-Instance finden Sie unter Verbindung zu Ihrer Microsoft SQL Server-DB-Instance herstellen.

Der Hauptbenutzer der DB-Instance erhält `CONTROL`-Zugriff auf `tempdb`, so dass dieser Benutzer die `tempdb`-Datenbankoptionen ändern kann. Der Hauptbenutzer ist nicht der Besitzer der `tempdb`-Datenbank. Falls notwendig, kann der Hauptbenutzer anderen Benutzern den `CONTROL`-Zugriff gewähren, damit diese ebenfalls die `tempdb`-Datenbankoptionen ändern können.

> **Note**
>
> Sie können keine Datenbankkonsolenbefehle (DBCC) in der `tempdb` Datenbank ausführen.

Ändern der Datenbankoptionen für tempdb

Sie können die Datenbankoptionen in der `tempdb` Datenbank Ihrer Amazon RDS DB-Instances ändern. Weitere Informationen darüber, welche Optionen geändert werden können, finden Sie unter tempdb Database in der Microsoft-Dokumentation.

Datenbankoptionen wie z. B. die Optionen für die maximale Dateigröße bleiben nach einem Neustart der DB-Instance bestehen. Sie können die Datenbankoptionen ändern, um die Leistung beim Datenimport zu optimieren und um Speicherplatzmangel vorzubeugen.

Optimieren der Leistung beim Import von Daten

Setzen Sie die Eigenschaften `SIZE` und `FILEGROWTH` der Datenbank "tempdb" auf hohe Zahlenwerte, um beim Importieren von großen Datenmengen in die DB-Instance die Leistung zu optimieren. Weitere Informationen zur Optimierung von `tempdb` finden Sie unter Optimieren der Leistung von tempdb in der Microsoft-Dokumentation.

Im folgenden Beispiel wird die Größe auf 100 GB und das Datenwachstum auf 10 Prozent eingestellt.

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

Vorbeugen von Speicherproblemen

Legen Sie einen Wert für die Eigenschaft `tempdb` fest, damit die `MAXSIZE`-Datenbank nicht den gesamten verfügbaren Speicherplatz belegt. Im folgenden Beispiel wird diese Eigenschaft auf 2048 MB festgelegt.

```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

Verkleinern der Datenbank tempdb

Es gibt zwei Möglichkeiten, die `tempdb` Datenbank auf Ihrer Amazon RDS DB-Instance zu verkleinern. Sie können die Prozedur `rds_shrink_tempdbfile` verwenden oder einen Wert für die Eigenschaft `SIZE` definieren.

Verwenden der Prozedur rds_shrink_tempdbfile

Sie können das RDS Amazon-Verfahren verwenden`msdb.dbo.rds_shrink_tempdbfile`, um die `tempdb` Datenbank zu verkleinern. Sie können `rds_shrink_tempdbfile` nur aufrufen, wenn Sie `CONTROL`-Zugriff auf `tempdb` haben. Durch den Aufruf von `rds_shrink_tempdbfile` wird keine Betriebsunterbrechung der DB-Instance verursacht.

Die Prozedur `rds_shrink_tempdbfile` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`@temp_filename`	SYSNAME	—	Erforderlich	Der logische Name der Datei, die verkleinert werden soll.
`@target_size`	int	Null	optional	Die neue Größe für diese Datei in Megabytes.

Im folgenden Beispiel werden die Namen der Dateien für die `tempdb`-Datenbank abgerufen.

```
use tempdb;
GO

select name, * from sys.sysfiles;
```

```
GO
```

Im folgenden Beispiel wird eine tempdb-Datenbankdatei mit dem Namen test_file verkleinert und die neue Größe auf 10 MB festgelegt:

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

Einstellung der Eigenschaft SIZE

Sie können die tempdb-Datenbank auch verkleinern, indem Sie die Eigenschaft SIZE festlegen und anschließend die DB-Instance neu starten. Weitere Informationen zum Neustart der DB-Instance finden Sie unter Neustarten einer DB-Instance.

Im folgenden Beispiel wird die Eigenschaft SIZE auf 1024 MB festgelegt.

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

TempDB-Konfiguration für Multi-AZ-Bereitstellungen

Wenn sich Ihre RDS SQL Server-DB-Instance in einer Multi-AZ-Bereitstellung befindet, die Database Mirroring (DBM) oder Always On Availability Groups (AGs) verwendet, sollten Sie bei der Verwendung der Datenbank die folgenden Überlegungen berücksichtigen. tempdb

Sie können keine tempdb Daten von Ihrer primären DB-Instance auf Ihre sekundäre DB-Instance replizieren. Wenn Sie ein Failover zu einer sekundären DB-Instance durchführen, tempdb ist diese sekundäre DB-Instance leer.

Sie können die Konfiguration der tempdb Datenbankoptionen, einschließlich der Einstellungen für Dateigröße und automatische Vergrößerung, von Ihrer primären DB-Instance mit Ihrer sekundären DB-Instance synchronisieren. Das Synchronisieren der tempDB Konfiguration wird auf allen RDS For SQL Server-Versionen unterstützt. Sie können die automatische Synchronisation der tempdb Konfiguration mithilfe der folgenden gespeicherten Prozedur aktivieren:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'TempDbFile';
```

> **Important**
>
> Bevor Sie die rds_set_system_database_sync_objects gespeicherte Prozedur verwenden, stellen Sie sicher, dass Sie Ihre bevorzugte tempdb Konfiguration auf Ihrer

primären DB-Instance und nicht auf Ihrer sekundären DB-Instance festgelegt haben. Wenn Sie die Konfigurationsänderung auf Ihrer sekundären DB-Instance vorgenommen haben, könnte Ihre bevorzugte `tempdb` Konfiguration gelöscht werden, wenn Sie die automatische Synchronisation aktivieren.

Sie können die folgende Funktion verwenden, um zu überprüfen, ob die automatische Synchronisation der `tempdb` Konfiguration aktiviert ist:

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Wenn die automatische Synchronisation der `tempdb` Konfiguration aktiviert ist, gibt es einen Rückgabewert für das `object_class` Feld. Wenn es ausgeschaltet ist, wird kein Wert zurückgegeben.

Sie können die folgende Funktion verwenden, um zu ermitteln, wann Objekte das letzte Mal synchronisiert wurden, und zwar UTC rechtzeitig:

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Wenn Sie beispielsweise die `tempdb` Konfiguration um 01:00 Uhr geändert und dann die `rds_fn_server_object_last_sync_time` Funktion ausgeführt haben, `last_sync_time` sollte der zurückgegebene Wert nach 01:00 Uhr liegen, was darauf hinweist, dass eine automatische Synchronisation stattgefunden hat.

Wenn Sie auch die SQL Server-Agent-Jobreplikation verwenden, können Sie die Replikation sowohl für SQL Agent-Jobs als auch für die `tempdb` Konfiguration aktivieren, indem Sie sie im `@object_type` folgenden Parameter angeben:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Weitere Informationen zur Replikation von SQL Server-Agent-Aufträgen finden Sie unter[Aktivieren der Auftragsreplikation von SQL Server Agent](#).

Als Alternative zur Verwendung der `rds_set_system_database_sync_objects` gespeicherten Prozedur, um sicherzustellen, dass `tempdb` Konfigurationsänderungen automatisch synchronisiert werden, können Sie eine der folgenden manuellen Methoden verwenden:

> **Note**
>
> Wir empfehlen, die automatische Synchronisation der `tempdb` Konfiguration mithilfe der `rds_set_system_database_sync_objects` gespeicherten Prozedur zu aktivieren. Durch die automatische Synchronisation müssen Sie diese manuellen Aufgaben nicht jedes Mal ausführen, wenn Sie Ihre `tempdb` Konfiguration ändern.

- Ändern Sie zuerst die DB-Instance und deaktivieren Sie Multi-AZ. Dann modifizieren Sie "tempdb" und aktivieren abschließend wieder Multi-AZ. Durch diese Methode entsteht keine Ausfallzeit.

  Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Ändern Sie zuerst `tempdb` auf der ursprünglichen primären Instance, führen Sie dann manuell ein Failover durch, und ändern Sie dann `tempdb` auf der neuen primären Instance. Bei dieser Methode kommt es zu einem Ausfall.

  Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Analysieren Ihrer Datenbank-Arbeitslast auf einer Amazon RDS for SQL Server-DB-Instance mit Database Engine Tuning Advisor

Database Engine Tuning Advisor ist eine von Microsoft bereitgestellte Client-Anwendung, die die Datenbank-Arbeitslast analysiert und basierend auf den Arten von Abfragen, die Sie ausführen, einen optimalen Satz von Indizes für Ihre Microsoft SQL Server-Datenbanken empfiehlt. Wie SQL Server Management Studio führen Sie Tuning Advisor von einem Client-Computer aus, der eine Verbindung zu Ihrer Amazon RDS DB-Instance herstellt, auf der SQL Server ausgeführt wird. Der Client-Computer kann ein lokaler Computer sein, den Sie lokal in Ihrem eigenen Netzwerk ausführen, oder es kann sich um eine Amazon EC2 Windows-Instance handeln, die in derselben Region wie Ihre Amazon RDS DB-Instance läuft.

In diesem Abschnitt wird gezeigt, wie ein Workload für eine Analyse im Optimierungshelfer erfasst wird. Dies ist das bevorzugte Verfahren zur Erfassung eines Workloads, da Amazon den Hostzugriff auf die SQL Server-Instance RDS einschränkt. Weitere Informationen finden Sie unter Database Engine Tuning Advisor in der Microsoft-Dokumentation.

Um den Optimierungshelfer zu verwenden, müssen Sie dem Helfer eine sogenannte Workload zur Verfügung stellen. Ein Workload besteht aus einer Reihe von SQL Transact-Anweisungen,

die für eine oder mehrere Datenbanken ausgeführt werden, die Sie optimieren möchten. Der Datenbankmodul-Optimierungsratgeber verwendet bei der Optimierung von Datenbanken Ablaufverfolgungsdateien, Ablaufverfolgungstabellen, SQL Transact-Skripten oder XML Dateien als Workload-Eingabe. Wenn Sie mit Amazon arbeitenRDS, kann ein Workload eine Datei auf einem Client-Computer oder eine Datenbanktabelle in einer Amazon RDS for SQL Server-DB sein, auf die Ihr Client-Computer zugreifen kann. Die Datei oder Tabelle muss Abfragen für die zu optimierenden Datenbanken enthalten, die in einem Format vorliegen, das für eine erneute Wiedergabe geeignet ist.

Damit der Optimierungshelfer optimal arbeiten kann, muss eine Workload so realistisch sein wie möglich. Sie können eine Workload-Datei oder -Tabelle erzeugen, indem Sie ein Trace für Ihre DB-Instance durchführen. Sie können während der Ausführung eines Trace entweder eine Belastung Ihrer DB-Instance simulieren oder Ihre Anwendungen mit normaler Belastung ausführen.

Es gibt zwei Arten von Traces: clientseitig und serverseitig. Eine clientseitige Ablaufverfolgung ist einfacher einzurichten, und Sie können verfolgen, wie Trace-Ereignisse in Echtzeit im SQL Server Profiler erfasst werden. Eine serverseitige Ablaufverfolgung ist komplexer einzurichten und erfordert etwas Transact-Scripting. SQL Da der Trace in eine Datei auf der RDS Amazon-DB-Instance geschrieben wird, wird außerdem Speicherplatz durch den Trace verbraucht. Es ist wichtig, zu beachten, wie viel Speicherplatz die Ausführung eines serverseitigen Trace benötigt, da die DB-Instance in den Storage-Full-Status wechseln könnte und dann nicht mehr verfügbar ist, sobald der Speicherplatz ausgeht.

Bei einem clientseitigen Trace können Sie, wenn eine ausreichende Menge an Trace-Daten im SQL Server Profiler erfasst wurde, die Workload-Datei generieren, indem Sie den Trace entweder in einer Datei auf Ihrem lokalen Computer oder in einer Datenbanktabelle auf einer DB-Instance speichern, die Ihrem Client-Computer zur Verfügung steht. Der Hauptnachteil bei der Verwendung eines clientseitigen Trace ist der, dass unter großer Auslastung eventuell nicht alle Abfragen vom Trace erfasst werden. Das könnte die Effektivität der vom Datenbank-Engine-Optimierungshelfer durchgeführten Analyse negativ beeinträchtigen. Muss ein Trace unter großer Auslastung ausgeführt werden und Sie möchten sicherstellen, dass während einer Trace-Sitzung alle Abfragen erfasst werden, sollte ein serverseitiger Trace zum Einsatz kommen.

Für einen serverseitigen Trace müssen die Trace-Dateien in der DB-Instance in einer geeigneten Workload-Datei gespeichert oder der Trace in einer Tabelle in der DB-Instance gespeichert werden, nachdem der Trace abgeschlossen ist. Sie können den SQL Server Profiler verwenden, um den Trace in einer Datei auf Ihrem lokalen Computer zu speichern, oder den Tuning Advisor aus der Trace-Tabelle auf der DB-Instance lesen lassen.

Einen clientseitigen Trace auf einer SQL Server-DB-Instance ausführen

Um einen clientseitigen Trace auf einer Server-DB-Instance auszuführen SQL

1. Starten Sie SQL Server Profiler. Es ist im Ordner Performance Tools Ihres SQL Serverinstanzordners installiert. Um einen clientseitigen Trace zu starten, muss eine Trace-Definitionsvorlage geladen oder definiert werden.
2. Wählen Sie im Menü Datei von SQL Server Profiler die Option New Trace aus. Im Dialogfeld Connect to Server (Mit Server verbinden) geben Sie DB-Instance-Endpunkt, Masterbenutzernamen und das Passwort für die Datenbank ein, für die eine Nachverfolgung ausgeführt werden soll.
3. Im Dialogfeld Trace Properties (Eigenschaften der Nachverfolgung) geben Sie einen Namen für die Nachverfolgung ein und wählen eine Definitionsvorlage für die Nachverfolgung aus. Eine Standardvorlage, TSQL _Replay, ist im Lieferumfang der Anwendung enthalten. Diese Vorlage kann für das Definieren Ihres Trace bearbeitet werden. Ereignisse und Ereignisinformationen können unter der Registerkarte Events Selection (Auswahl von Ereignissen) im Dialogfeld Trace Properties (Eigenschaften der Nachverfolgung) bearbeitet werden.

   Weitere Informationen zu Vorlagen für Ablaufverfolgungsdefinitionen und zur Verwendung des SQL Server Profilers zur Angabe einer clientseitigen Ablaufverfolgung finden Sie in der Microsoft-Dokumentation unter Database Engine Tuning Advisor.
4. Starten Sie die clientseitige Ablaufverfolgung und beobachten Sie SQL Abfragen in Echtzeit, während sie in Ihrer DB-Instance ausgeführt werden.
5. Wählen Sie im Menü File (Datei) die Option Stop Trace (Nachverfolgung beenden) aus, sobald die Nachverfolgung abgeschlossen ist. Speichern Sie die Ergebnisse als Datei oder als Trace-Tabelle in Ihrer DB-Instance.

Einen serverseitigen Trace auf einer SQL Server-DB-Instance ausführen

Das Schreiben eines Skripts für das Erstellen eines serverseitigen Trace kann eine komplexe Angelegenheit sein und geht über dieses Dokument hinaus. Dieser Abschnitt enthält Beispiel-Skripts, die Sie als Beispiele verwenden können. Wie auch bei einem clientseitigen Trace liegt das Ziel darin, eine Workload-Datei oder eine Trace-Tabelle zu erstellen, die mit dem Datenbank-Engine-Optimierungshelfer geöffnet werden kann.

Hier ist ein gekürztes Beispiel-Script, das einen serverseitigen Trace startet und die Details in einer Workload-Datei erfasst. Der Trace wird zunächst in der Datei RDSTrace .trc im Verzeichnis D:

\RDSDBDATA\ Log gespeichert und alle 100 MB wird ein Rollover ausgeführt, sodass nachfolgende Trace-Dateien die Namen RDSTrace _1.trc, _2.trc usw. erhalten. RDSTrace

```
DECLARE @file_name NVARCHAR(245) = 'D:\RDSDBDATA\Log\RDSTrace';
DECLARE @max_file_size BIGINT = 100;
DECLARE @on BIT = 1
DECLARE @rc INT
DECLARE @traceid INT

EXEC @rc = sp_trace_create @traceid OUTPUT, 2, @file_name, @max_file_size
IF (@rc = 0) BEGIN
   EXEC sp_trace_setevent @traceid, 10, 1, @on
   EXEC sp_trace_setevent @traceid, 10, 2, @on
   EXEC sp_trace_setevent @traceid, 10, 3, @on
 . . .
   EXEC sp_trace_setfilter @traceid, 10, 0, 7, N'SQL Profiler'
   EXEC sp_trace_setstatus @traceid, 1
   END
```

Das folgende Beispiel ist ein Script, mit dem ein Trace angehalten wird. Beachten Sie, dass ein vom vorherigen Script erstellter Trace solange ausgeführt wird, bis Sie den Trace explizit anhalten oder die Festplattenkapazität für den Vorgang nicht mehr ausreicht.

```
DECLARE @traceid INT
SELECT @traceid = traceid FROM ::fn_trace_getinfo(default)
WHERE property = 5 AND value = 1 AND traceid <> 1

IF @traceid IS NOT NULL BEGIN
   EXEC sp_trace_setstatus @traceid, 0
   EXEC sp_trace_setstatus @traceid, 2
END
```

Die Ergebnisse des serverseitigen Trace können in einer Datenbank-Tabelle gespeichert, und die Datenbank-Tabelle mithilfe der Funktion fn_trace_gettable als Workload für den Optimierungshelfer verwendet werden. Mit den folgenden Befehlen werden die Ergebnisse aller Dateien mit dem Namen RDSTrace .trc im Verzeichnis D:\rdsdbdata\Log, einschließlich aller Rollover-Dateien wie _1.trc, in eine Tabelle mit dem Namen in der aktuellen Datenbank geladen. RDSTrace RDSTrace

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

Um eine bestimmte Rollover-Datei in einer Tabelle zu speichern, z. B. die Datei RDSTrace _1.trc, geben Sie den Namen der Rollover-Datei an und ersetzen Sie den Wert 1 anstelle von default als letzten Parameter für fn_trace_gettable.

```
SELECT * INTO RDSTrace_1
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

Ausführung des Optimierungshelfers mit einem Trace

Sobald Sie einen Trace erstellen, entweder als lokale Datei oder als Datenbank-Tabelle, können Sie den Optimierungshelfer für Ihre DB-Instance ausführen. Die Verwendung von Tuning Advisor mit Amazon RDS ist derselbe Vorgang wie bei der Arbeit mit einer eigenständigen SQL Remote-Serverinstanz. Sie können entweder die Benutzerfläche des Optimierungshelfers auf Ihrem Client-Computer oder das Hilfsprogramm dta.exe von der Befehlszeile aus verwenden. In beiden Fällen müssen Sie über den Endpunkt der RDS DB-Instance eine Verbindung zur Amazon DB-Instance herstellen und Ihren Master-Benutzernamen und Ihr Master-Benutzerpasswort angeben, wenn Sie Tuning Advisor verwenden.

Das folgende Codebeispiel zeigt die Verwendung des Befehlszeilenprogramms dta.exe für eine RDS Amazon-DB-Instance mit einem Endpunkt von**`dta.cnazcmklsdei.us-east-1.rds.amazonaws.com`**. Das Beispiel enthält den Master-Benutzernamen **`admin`** und das Master-Benutzerkennwort **`test`**, die zu tunende Beispieldatenbank heißt „machine“ **`C:\RDSTrace.trc`**. Der Beispiel-Befehlszeilencode spezifiziert auch eine Trace-Sitzung mit dem Namen **`RDSTrace1`** und gibt Ausgabedateien **`RDSTrace.sql`** für den lokalen Computer an, **`RDSTrace.txt`** die nach dem SQL Ausgabeskript, einer Ergebnisdatei und **`RDSTrace.xml`** einer XML Analysedatei benannt sind. In der RDSDTA Datenbank mit dem Namen ist auch eine Fehlertabelle angegeben**`RDSTraceErrors`**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -
if C:\RDSTrace.trc -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Hier ist derselbe Beispiel-Befehlszeilencode, außer dass der Eingabe-Workload eine Tabelle auf der benannten entfernten RDS Amazon-Instance ist**`RDSTrace`**, die sich in der **`RDSDTA`** Datenbank befindet.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -it
 RDSDTA.dbo.RDSTrace -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Eine vollständige Liste der Befehlszeilenparameter des dta-Dienstprogramms finden Sie unter dta Utility in der Microsoft-Dokumentation.

Ändern **db_owner** Sie das auf das **rdsa** Konto für Ihre Amazon RDS for SQL Server-Datenbank

Wenn Sie eine Datenbank in einer RDS for SQL Server-DB-Instance erstellen oder wiederherstellen, RDS setzt Amazon den Eigentümer der Datenbank auf`rdsa`. Wenn Sie eine Multi-AZ-Bereitstellung mit SQL Server Database Mirroring (DBM) oder Always On Availability Groups (AGs) haben, RDS setzt Amazon den Besitzer der Datenbank auf der sekundären DB-Instance auf. `NT AUTHORITY\SYSTEM` Der Besitzer der sekundären Datenbank kann erst geändert werden, wenn die sekundäre DB-Instance zur primären Rolle heraufgestuft wurde. In den meisten Fällen ist es bei der Ausführung von Abfragen unproblematisch, wenn der Besitzer der Datenbank auf `NT AUTHORITY\SYSTEM` festgelegt ist. Es kann dabei jedoch zu Fehlern kommen, wenn gespeicherte Systemprozeduren wie `sys.sp_updatestats` ausgeführt werden, für deren Ausführung erhöhte Berechtigungen erforderlich sind.

Sie können die folgende Abfrage verwenden, um den Besitzer der Datenbanken von `NT AUTHORITY\SYSTEM` zu identifizieren:

```
SELECT name FROM sys.databases WHERE SUSER_SNAME(owner_sid) = 'NT AUTHORITY\SYSTEM';
```

Sie können die RDS gespeicherte Amazon-Prozedur verwenden`rds_changedbowner_to_rdsa`, um den Besitzer der Datenbank zu ändern`rdsa`. Die folgenden Datenbanken dürfen nicht mit `rds_changedbowner_to_rdsa` verwendet werden: `master, model, msdb, rdsadmin, rdsadmin_ReportServer, rdsadmin_ReportServerTempDB, SSISDB`.

Um den Besitzer der Datenbank zu ändern`rdsa`, rufen Sie die `rds_changedbowner_to_rdsa` gespeicherte Prozedur auf und geben Sie den Namen der Datenbank ein.

Example Verwendung:

```
exec msdb.dbo.rds_changedbowner_to_rdsa 'TestDB1';
```

Der folgende Parameter ist erforderlich:

- `@db_name` – Der Name der Datenbank, deren Besitzer in `rdsa` geändert werden soll.

> **Important**
>
> Sie können nicht verwenden`rds_changedbowner_to_rdsa`, um den Besitz einer Datenbank in einen anderen Anmeldenamen zu ändern als`rdsa`. Beispielsweise können Sie den Eigentümer des Anmeldenamens, mit dem Sie die Datenbank erstellt haben, nicht ändern. Um die verloren gegangene Mitgliedschaft in der `db_owner` Rolle für Ihren Masterbenutzer wiederherzustellen, wenn kein anderer Datenbankbenutzer verwendet werden kann, um die Mitgliedschaft zu gewähren, setzen Sie das Masterbenutzerpasswort zurück, um die Mitgliedschaft in der `db_owner` Rolle zu erhalten. Weitere Informationen finden Sie unter Zurücksetzen der db_owner-Rollenmitgliedschaft für den Master-Benutzer für Amazon for Server RDS SQL.

Verwaltung von Sortierungen und Zeichensätzen für Amazon RDS for Microsoft SQL Server

Dieses Thema enthält Anleitungen zur Verwaltung von Sortierungen und Zeichensätzen für Microsoft SQL Server in Amazon RDS. Es wird erklärt, wie Sortierungen während der Datenbankerstellung konfiguriert und später geändert werden, um sicherzustellen, dass Textdaten entsprechend den Sprach- und Gebietsschemaanforderungen ordnungsgemäß behandelt werden. Darüber hinaus werden bewährte Methoden zur Aufrechterhaltung der Kompatibilität und Leistung in SQL Server-Umgebungen in Amazon RDS behandelt.

SQL Server unterstützt das Sortieren auf verschiedenen Ebenen. Sie legen die Standardsortierung für den Server fest, wenn Sie die DB-Instance erstellen. Sie können die Sortierung auf der Ebene von Datenbanken, Tabellen und Spalten überschreiben.

Themen

- Sortierung auf Serverebene bei Microsoft SQL Server
- Sortierung auf Datenbankebene bei Microsoft SQL Server

Sortierung auf Serverebene bei Microsoft SQL Server

Wenn Sie eine Microsoft SQL Server-DB-Instance erstellen können Sie die zu verwendende Sortierung des Servers festlegen. Wenn Sie keine andere Sortierung wählen, wird für die Sortierung auf Serverebene standardmäßig SQL_Latin1_General_ _CI_AS verwendet. CP1 Die für den Server festgelegte Sortierung wird standardmäßig für alle Datenbanken und Datenbankobjekte verwendet.

> **Note**
>
> Sie können die Sortierung nicht ändern, wenn Sie aus einem DB-Snapshot wiederherstellen.

Derzeit unterstützt Amazon RDS die folgenden Sortierungen für Server:

Kollation	Beschreibung
Arabic_CI_AS	Arabisch, keine Beachtung der Groß-/Kle inschreibung, Beachtung von Akzenten, keine Beachtung des Kana-Typs, keine Beachtung der Breite
Chinesisch_PRC_ BIN2	Chinesisch-VR China, Sortierreihenfolge für binäre Codepunkte
Chinese_PRC_CI_AS	Chinesisch (vereinfacht), Groß-/Kleinschreib ung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Chinese_Taiwan_Stroke_CI_AS	Chinesisch (traditionell), Groß-/Kleinschreib ung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Dänisch_Norwegisch_CI_AS	Dänisch-Norwegisch, Groß-/Kleinschreibung irrelevant, Akzente relevant, Kanatyp irrelevant, Breite irrelevant
Dänisch_Norwegisch_CI_AS_KS	Dänisch-Norwegisch, ohne Berücksic htigung von Groß- und Kleinschreibung, ohne Berücksichtigung von Akzenten, ohne

Kollation	Beschreibung
	Berücksichtigung von Kanatypen, ohne Berücksichtigung der Breite
Dänisch_Norwegisch_CI_AS_KS_WS	Dänisch-Norwegisch, ohne Berücksichtigung von Groß- und Kleinschreibung, Berücksichtigung von Akzenten, Berücksichtigung von Kanatypen und Breiten
Dänisch_Norwegisch_CI_AS_WS	Dänisch-Norwegisch, ohne Berücksichtigung von Groß- und Kleinschreibung, ohne Berücksichtigung von Akzenten, ohne Berücksichtigung von Kanatypen, ohne Berücksichtigung der Breite
Dänisch_Norwegisch_CS_AI	Dänisch-Norwegisch, Groß- und Kleinschreibung wird nicht berücksichtigt, ohne Berücksichtigung von Akzenten, ohne Berücksichtigung von Kanatypen, ohne Berücksichtigung der Breite
Dänisch_Norwegisch_CS_AI_KS	Dänisch-Norwegisch, Groß- und Kleinschreibung wird nicht berücksichtigt, ohne Berücksichtigung von Akzenten, ohne Berücksichtigung von Kanatypen, ohne Berücksichtigung der Breite
Finnisch_Schwedisch_100_BIN	Finnisch-Schwedisch-100, binäre Sortierung
Finnisch_Schwedisch_100_ BIN2	Finnisch-Schwedisch-100, Vergleichssortierung nach Binärcodepunkten
Finnisch_Schwedisch_100_CI_AI	Finnisch-Schwedisch-100, ohne Berücksichtigung von Groß- und Kleinschreibung, ohne Berücksichtigung von Akzenten, ohne Berücksichtigung von Kanatypen, ohne Berücksichtigung der Breite

Kollation	Beschreibung
Finnisch_Schwedisch_100_CI_AS	Finnisch-Schwedisch-100, ohne Berücksichtigung von Groß- und Kleinschreibung, ohne Berücksichtigung von Akzenten, ohne Berücksichtigung von Kanatypen, ohne Berücksichtigung der Breite
Finnish_Swedish_CI_AS	Finnisch, Schwedisch und Schwedisch (Finnland), Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
French_CI_AS	Französisch, Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Hebrew_BIN	Hebräisch, binäre Sortierung
Hebrew_CI_AS	Hebräisch, Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Japanese_BIN	Japanisch, binäre Sortierung
Japanese_CI_AS	Japanisch, Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Japanese_CS_AS	Japanisch, Groß-/Kleinschreibung relevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Japanese_XJIS_140_CI_AS	Japanisch, keine Beachtung der Groß-/Kleinschreibung, Beachtung von Akzenten, keine Beachtung des Kana-Typs, keine Beachtung der Breite, zusätzliche Zeichen, keine Beachtung der Variierungsauswahlzeichen

Kollation	Beschreibung
Japanese_XJIS_140_CI_AS_KS_VSS	Japanisch, keine Beachtung der Groß-/Kle inschreibung, Beachtung von Akzenten, Beachtung des Kana-Typs, keine Beachtung der Breite, zusätzliche Zeichen, Beachtung der Variierungsauswahlzeichen
Japanese_XJIS_140_CI_AS_VSS	Japanisch, keine Beachtung der Groß-/Kle inschreibung, Beachtung von Akzenten, keine Beachtung des Kana-Typs, keine Beachtung der Breite, zusätzliche Zeichen, Beachtung der Variierungsauswahlzeichen
Japanese_XJIS_140_CS_AS_KS_WS	Japanisch, Beachtung der Groß-/Kleinschreib ung, Beachtung von Akzenten, Beachtung des Kana-Typs, Beachtung der Breite, zusätzlic he Zeichen, keine Beachtung der Variierun gsauswahlzeichen
Korean_Wansung_CI_AS	Koreanisch (Wansung), Groß-/Kleinschreib ung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Latin1_General_100_BIN	Latin1-General-100, binäre Sortierung
Lateinisch1_Allgemein_100_ BIN2	Latin1-General-100, binäre Codepunkt-Sortierr eihenfolge
Lateinisch1_Allgemein_100_ _ BIN2 UTF8	Latin1-General-100, binäre Codepunkt-Sortierr eihenfolge, UTF-8-kodiert
Latin1_General_100_CI_AS	Latin1-General-100, Groß-/Kleinschreibung irrelevant, Diakritika irrelevant, Kana-Typ irrelevant, Breite irrelevant
Lateinisch1_Allgemein_100_CI_AS_SC_ UTF8	Latin1-General-100, keine Beachtung der Groß-/Kleinschreibung, Beachtung von Akzenten, zusätzliche Zeichen, UTF-8-kodiert

Kollation	Beschreibung
Latin1_General_BIN	Latin1-General, binäre Sortierung
Latin1_Allgemein_ BIN2	Latin1-General, binäre Codepunkt-Sortierr eihenfolge
Latin1_General_CI_AI	Lateinisch 1 (allgemein), Groß-/Kleinschreib ung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Latin1_General_CI_AS	Lateinisch 1 (allgemein), Groß-/Kleinschreib ung irrelevant, Diakritika irrelevant, Kana-Typ irrelevant, Breite irrelevant
Latin1_General_CI_AS_KS	Lateinisch 1 (allgemein), Groß-/Kleinschreib ung irrelevant, Diakritika irrelevant, Kana-Typ relevant, Breite irrelevant
Latin1_General_CS_AS	Latin1-General, Groß-/Kleinschreibung berücksichtigt, Akzent berücksichtigt, Kanatyp nicht berücksichtigt, Breite nicht berücksichtigt
Modern_Spanish_CI_AS	Spanisch (modern), Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant
Polish_CI_AS	Polnisch, keine Beachtung der Groß-/Kle inschreibung, Beachtung von Akzenten, keine Beachtung des Kana-Typs, keine Beachtung der Breite
SQL_1xCompat_0_CI_AS CP85	Lateinisch 1 (allgemein), Groß-/Kleinschreib ung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-D aten, SQL Server-Sortierreihenfolge 49 auf Codepage 850 für Daten, die nicht in Unicode kodiert sind

Kollation	Beschreibung
SQL_Latin1_Allgemein_ _CI_AI CP1	Lateinisch 1 (allgemein), Groß-/Kleinschreibung irrelevant, Diakritika irrelevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-Daten, SQL Server-Sortierreihenfolge 54 auf Codepage 1252 für Daten, die nicht in Unicode kodiert sind
SQL_Latin1_General_ CP1 _CI_AS (Standard)	Lateinisch 1 (allgemein), Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-Daten, SQL Server-Sortierreihenfolge 52 auf Codepage 1252 für Daten, die nicht in Unicode kodiert sind
SQL_Latin1_General_ CP1 _CS_AS	Lateinisch 1 (allgemein), Groß-/Kleinschreibung relevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-Daten, SQL Server-Sortierreihenfolge 51 auf Codepage 1252 für Daten, die nicht in Unicode kodiert sind
CP437SQL_Latin1_General_ _CI_AI	Lateinisch 1 (allgemein), Groß-/Kleinschreibung irrelevant, Diakritika irrelevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-Daten, SQL Server-Sortierreihenfolge 34 auf Codepage 437 für Daten, die nicht in Unicode kodiert sind
CP85SQL_Latin1_General_0_BIN	Latin1-General, binäre Sortierreihenfolge für Unicode-Daten, SQL-Server-Sortierreihenfolge 40 auf Codepage 850 für Daten, die nicht in Unicode codiert sind

Kollation	Beschreibung
CP85SQL_Latin1_Allgemein_ 0_ BIN2	Lateinisch 1 (allgemein), binäre Codepoint-Sortierreihenfolge für Unicode-Daten, SQL Server-Sortierreihenfolge 40 auf Codepage 850 für Daten, die nicht in Unicode kodiert sind
CP85SQL_Latin1_Allgemein_ 0_CI_AI	Latin1-General, keine Beachtung der Groß-/Kleinschreibung, keine Beachtung von Akzenten, keine Beachtung des Kana-Typs, keine Beachtung der Breite für Unicode-Daten, SQL-Server-Sortierreihenfolge 44 auf Codepage 850 für Daten, die nicht in Unicode codiert sind
CP85SQL_Latin1_General_0_CI_AS	Lateinisch 1 (allgemein), Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-Daten, SQL Server-Sortierreihenfolge 42 auf Codepage 850 für Daten, die nicht in Unicode kodiert sind
CP1256SQL_Latin1_General_ _CI_AS	Lateinisch 1 (allgemein), Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant für Unicode-Daten, SQL Server-Sortierreihenfolge 146 auf Codepage 1256 für Daten, die nicht in Unicode kodiert sind
CP1255SQL_Latin1_General_ _CS_AS	Latin1-Allgemein, Berücksichtigung von Groß- und Kleinschreibung, Berücksichtigung von Akzenten, unabhängig von Kanatyp und Breite für Unicode-Daten, SQL Server-Sortierreihenfolge 137 auf Codepage 1255 für Nicht-Unicode-Daten
Thai_CI_AS	Thai, Groß-/Kleinschreibung irrelevant, Diakritika relevant, Kana-Typ irrelevant, Breite irrelevant

Kollation	Beschreibung
Turkish_CI_AS	Türkisch, keine Beachtung der Groß-/Kle inschreibung, Beachtung von Akzenten, keine Beachtung des Kana-Typs, keine Beachtung der Breite

Auswahl der Sortierung:

- Wenn Sie die Amazon-RDS-Konsole verwenden, wählen Sie beim Erstellen einer neuen DB-Instance Additional configuration (Zusätzliche Konfiguration) aus und geben Sie die Sortierung im Feld Collation (Sortierung) ein. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Wenn Sie die verwenden AWS CLI, verwenden `--character-set-name` Sie die `create-db-instance` Option zusammen mit dem Befehl. Weitere Informationen finden Sie unter create-db-instance.
- Wenn Sie die Amazon RDS-API verwenden, verwenden Sie den Parameter `CharacterSetName` mit der `CreateDBInstance`-Operation. Weitere Informationen finden Sie unter Erstellen DBInstance.

Sortierung auf Datenbankebene bei Microsoft SQL Server

Die Standardsortierung kann auf Datenbank-, Tabellen- oder Spaltenebene durch außer Kraft setzen der Sortierung beim Erstellen einer neuer Datenbank oder eines Datenbankobjekts geändert werden. Wenn Ihre Standard-Serversortierung beispielsweise SQL_Latin1_General_ _CI_AS ist, können Sie sie zur Unterstützung der Mohawk-Sortierung in CP1 Mohawk_100_CI_AS ändern. Selbst Argumente in einer Abfrage können einer Typumwandlung unterzogen werden, um bei Bedarf eine andere Sortierung zu verwenden.

Die folgende Abfrage würde beispielsweise die Standardsortierung für die Spalte in Mohawk_100_CI_AS ändern AccountName

```
CREATE TABLE [dbo].[Account]
 (
     [AccountID] [nvarchar](10) NOT NULL,
     [AccountName] [nvarchar](100) COLLATE Mohawk_100_CI_AS NOT NULL
 ) ON [PRIMARY];
```

Die Microsoft SQL Server-DB-Engine unterstützt durch die integrierten Datentypen NCHAR, NVARCHAR und NTEXT auch Unicode. Wenn Sie z. B. CJK-Unterstützung benötigen, verwenden Sie diese Unicode-Datentypen für die Zeichenspeicherung und setzen bei der Erstellung Ihrer Datenbanken und Tabellen die Server-Standardsortierung außer Kraft. Hier sind verschiedene Links von Microsoft, die das Thema Sortierung und Unicode-Support für SQL Server behandeln:

- Arbeiten mit Sortierungen
- Sortierung und internationale Terminologie
- Verwenden von SQL Server-Sortierungen
- Internationale Erwägungen für Datenbanken und Datenbank-Engine-Anwendungen

Einen Datenbankbenutzer für Amazon RDS for SQL Server erstellen

Sie können einen Datenbankbenutzer für Ihre DB-Instance von Amazon RDS for Microsoft SQL Server erstellen, indem Sie ein T-SQL-Skript wie im folgenden Beispiel ausführen. Verwenden Sie eine Anwendung wie SQL Server Management Suite (SSMS). Sie melden sich bei der DB-Instance als Hauptbenutzer an, der beim Erstellen der DB-Instance erstellt wurde.

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
```

Ein Beispiel für das Hinzufügen eines Datenbankbenutzers zu einer Rolle finden Sie unter Einen Benutzer zur SQLAgentUser Rolle hinzufügen.

> **Note**
>
> Wenn Sie beim Hinzufügen eines Benutzers Berechtigungsfehler erhalten, können Sie die Berechtigungen wiederherstellen, indem Sie das Passwort für den DB-Instance-

Hauptbenutzer ändern. Weitere Informationen finden Sie unter Zurücksetzen der db_owner-Rollenmitgliedschaft für den Master-Benutzer für Amazon for Server RDS SQL.
Es ist keine bewährte Methode, Masterbenutzerberechtigungen in Ihren Anwendungen zu klonen. Weitere Informationen finden Sie unter So klonen Sie Masterbenutzerberechtigungen in Amazon RDS for SQL Server.

Ermitteln eines Wiederherstellungsmodells für Ihre Amazon RDS for SQL Server-Datenbank

In Amazon RDS sind das Wiederherstellungsmodell, der Aufbewahrungszeitraum und der Datenbankstatus miteinander verknüpft.

Daher ist es wichtig, sich über die Konsequenzen klar zu werden, bevor eine Änderung an diesen Einstellungen vorgenommen wird. Jede Einstellung kann sich auf andere auswirken. Beispielsweise:

- Wenn Sie das Wiederherstellungsmodell einer Datenbank auf SIMPLE oder BULK _ ändern, LOGGED während die Aufbewahrung von Backups aktiviert ist, RDS setzt Amazon das Wiederherstellungsmodell auf FULL innerhalb von fünf Minuten zurück. Dies führt auch dazu, RDS dass ein Snapshot der DB-Instance erstellt wird.
- Wenn Sie die Aufbewahrungsdauer von Backups auf `0` Tage festlegen, wird der Wiederherstellungsmodus auf RDS festgelegtSIMPLE.
- Wenn Sie das Wiederherstellungsmodell einer Datenbank von einer anderen Option SIMPLE auf ändern, während die Aufbewahrung von Backups auf `0` Tage festgelegt ist, wird das Wiederherstellungsmodell auf RDS SIMPLE zurückgesetzt.

Important

Ändern Sie niemals das Wiederherstellungsmodell auf Multi-AZ-Instances, auch wenn es den Anschein hat, dass Sie dazu in der Lage sind, z. B. durch ALTER DATABASE Für Multi-AZ ist die Aufbewahrung von Backup und damit der FULL Wiederherstellungsmodus erforderlich. Wenn Sie das Wiederherstellungsmodell ändern, wird es RDS sofort wieder auf FULL geändert.
Durch diesen automatischen Reset RDS muss der Spiegel vollständig neu erstellt werden. Während der Neuerstellung ist die Verfügbarkeit der Datenbank ca. 30–90 Minuten lang beeinträchtigt, bis die Spiegelung für ein Failover bereit ist. Bei der DB-Instance

kann es genauso wie während der Konvertierung von Einzel-AZ in Multi-AZ ebenso zu Leistungseinschränkungen kommen. Wie lange die Leistung beeinträchtigt ist, ist von der Speichergröße der Datenbank abhängig – je größer die gespeicherte Datenbank, desto länger dauert die Beeinträchtigung an.

Weitere Informationen zu SQL Serverwiederherstellungsmodellen finden Sie unter Wiederherstellungsmodelle (SQLServer) in der Microsoft-Dokumentation.

Ermitteln der letzten Failover-Zeit für Amazon RDS for Server SQL

Um die letzte Failover-Zeit zu bestimmen, verwenden Sie das folgende gespeicherte Verfahren:

```
execute msdb.dbo.rds_failover_time;
```

Dieses Verfahren gibt die folgenden Informationen zurück.

Ausgabeparameter	Beschreibung
errorlog_available_from	Zeigt die Zeit an, ab der Fehlerprotokolle im Protokollverzeichnis verfügbar sind.
recent_failover_time	Zeigt die letzte Failover-Zeit an, wenn sie in den Fehlerprotokollen verfügbar ist. Andernfalls wird angezeig `null`.

Note

Die gespeicherte Prozedur durchsucht alle verfügbaren SQL Serverfehlerprotokolle im Protokollverzeichnis, um den letzten Failover-Zeitpunkt abzurufen. Wenn die Failover-Meldungen vom SQL Server überschrieben wurden, ruft das Verfahren die Failover-Zeit nicht ab.

Example Kein Failover in letzter Zeit

Dieses Beispiel zeigt die Ausgabe, wenn in den Fehlerprotokollen kein aktuelles Failover vorhanden ist. Seit 2020-04-29 23:59:00.01 ist kein Failover aufgetreten.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	Null

Example Failover in letzter Zeit

Dieses Beispiel zeigt die Ausgabe, wenn ein Failover in den Fehlerprotokollen vorliegt. Das letzte Failover erfolgte am 2020-05-05 18:57:51.89.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

Behebung von point-in-time-recovery Fehlern aufgrund einer Lücke in der Protokollsequenznummer

Beim Versuch point-in-time-recovery (PITR) in RDS für SQL Server können Fehler auftreten, die auf Lücken in den Protokollsequenznummern () zurückzuführen sind. LSNs Diese Lücken verhindern, dass RDS Ihre Datenbank zum angeforderten Zeitpunkt wiederherstellt, und RDS versetzt Ihre wiederherstellende Instanz in den `incompatible-restore` Status.

Zu den häufigsten Ursachen für dieses Problem gehören:

- Manuelle Änderungen am Datenbank-Wiederherstellungsmodell.
- Änderungen des automatischen Wiederherstellungsmodells durch RDS aufgrund unzureichender Ressourcen für die Durchführung von Transaktionsprotokollsicherungen.

Führen Sie diese Abfrage aus, um LSN-Lücken in Ihrer Datenbank zu identifizieren:

```
SELECT * FROM msdb.dbo.rds_fn_list_tlog_backup_metadata(database_name)
ORDER BY backup_file_time_utc desc;
```

Wenn Sie eine LSN-Lücke entdecken, können Sie:

- Wählen Sie einen Wiederherstellungspunkt vor der LSN-Lücke.
- Warten Sie und stellen Sie die Wiederherstellung bis zu einem Punkt wieder her, an dem das nächste Instanz-Backup abgeschlossen ist.

Um dieses Problem zu vermeiden, empfehlen wir, das Wiederherstellungsmodell Ihrer RDS for SQL Server-Datenbanken nicht manuell zu ändern, da dies die Instanzbeständigkeit unterbricht. Wir empfehlen Ihnen außerdem, einen Instanztyp mit ausreichenden Ressourcen für Ihre Arbeitslast zu wählen, um regelmäßige Backups der Transaktionsprotokolle sicherzustellen.

Weitere Informationen zur Transaktionsprotokollverwaltung finden Sie im Leitfaden zur Architektur und Verwaltung von SQL Server-Transaktionsprotokollen in der Microsoft SQL Server-Dokumentation.

Anzeige von Datenbanknamen für Amazon RDS for SQL Server verweigern oder zulassen

Der Hauptbenutzer kann keine Einstellungen vornehmen `DENY VIEW ANY DATABASE TO LOGIN` um Datenbanken vor einem Benutzer zu verbergen.
Verwenden Sie stattdessen die folgende gespeicherte Prozedur, um diese Berechtigung zu ändern:

- Der Zugriff auf die Datenbankansicht wird verweigert für *LOGIN*:

```
EXEC msdb.dbo.rds_manage_view_db_permission @permission='DENY',
 @server_principal='LOGIN'

go
```

- Zugriff auf die Datenbankansicht wird zugelassen für *LOGIN*:

```
EXEC msdb.dbo.rds_manage_view_db_permission @permission='GRANT',
 @server_principal='LOGIN'

go
```

Beachten Sie Folgendes, wenn Sie diese gespeicherte Prozedur verwenden:

- Datenbanknamen sind in den SSMS und internen Ansichten DMV (dynamische Verwaltungsansichten) verborgen. Datenbanknamen sind jedoch weiterhin in Audit-, Protokoll- und

Metadatentabellen sichtbar. Dies sind gesicherte `VIEW ANY DATABASE` Serverberechtigungen. Weitere Informationen finden Sie unter DENYServerberechtigungen.

- Sobald die Berechtigung wieder auf `GRANT` (erlaubt) zurückgesetzt wurde, *`LOGIN`* kann alle Datenbanken einsehen.
- Wenn Sie löschen und neu erstellen *`LOGIN`*, die damit verbundene Anzeigeberechtigung wurde auf LOGIN zurückgesetzt`ALLOW`.
- Legen Sie für Multi-AZ-Instances die `GRANT` Berechtigung `DENY` oder nur für *`LOGIN`* auf dem primären Host. Die Änderungen werden automatisch auf den sekundären Host übertragen.
- Diese Berechtigung ändert nur, ob ein Anmeldename die Datenbanknamen einsehen kann. Der Zugriff auf Datenbanken und die darin enthaltenen Objekte wird jedoch separat verwaltet.

Deaktivierung von Fast Inserts beim Massenladen für Amazon RDS for Server SQL

Ab SQL Server 2016 sind Fast Inserts standardmäßig aktiviert. Schnelle Einfügungen nutzen die minimale Protokollierung, die auftritt, während sich die Datenbank im einfachen oder in massenprotokolliertem Wiederherstellungsmodell befindet, um die Einfügeleistung zu optimieren. Bei schnellen Einfügungen erhält jeder Massenladestapel neue Ausdehnungen, wobei die Zuweisungssuche nach vorhandenen Ausdehnungen mit freiem Speicherplatz umgangen wird, um die Einfügeleistung zu optimieren.

Bei schnellen Einfügungen können Massenlasten mit kleinen Stapelgrößen jedoch zu einem erhöhten ungenutzten Speicherplatz führen, der von Objekten genutzt wird. Wenn eine Erhöhung der Stapelgröße nicht möglich ist, kann das Aktivieren des Ablaufverfolgungs-Flags 692 dazu beitragen, ungenutzten reservierten Speicherplatz zu reduzieren, jedoch auf Kosten der Leistung. Durch das Aktivieren dieses Ablaufverfolgungs-Flags werden schnelle Einfügungen beim Massenladen von Daten in Heap- oder Cluster-Indizes deaktiviert.

Sie aktivieren den Ablaufverfolgungs-Flag 692 als Startup-Parameter mittels DB-Parametergruppen. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

Trace Flag 692 wird für Amazon RDS on SQL Server 2016 und höher unterstützt. Weitere Informationen zu Trace-Flags finden Sie unter DBCCTRACEON- Trace-Flags in der Microsoft-Dokumentation.

Löschen einer Datenbank in einer Amazon RDS for Microsoft SQL Server Server-DB-Instance

Sie können eine Datenbank einer Amazon RDS-DB-Instance verwerfen, auf der Microsoft SQL Server in einer Single-AZ- oder Multi-AZ-Bereitstellung ausgeführt wird. Verwenden Sie den folgenden Befehl, um die Datenbank zu verwerfen:

```
--replace your-database-name with the name of the database you want to drop
EXECUTE msdb.dbo.rds_drop_database  N'your-database-name'
```

> **Note**
>
> Verwenden Sie im Befehl einfache gerade Anführungszeichen. Typographische Anführungszeichen verursachen einen Fehler.

Nachdem Sie diese Vorgehensweise zum Löschen der Datenbank verwendet haben, löscht Amazon RDS alle vorhandenen Verbindungen zur Datenbank und entfernt den Sicherungsverlauf der Datenbank.

Gehen Sie wie folgt vor, um anderen Benutzern die Möglichkeit zur Sicherung und Wiederherstellung zu gewähren:

```
USE master
GO
CREATE LOGIN user1 WITH PASSWORD=N'changeThis', DEFAULT_DATABASE=master,
 CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
USE msdb
GO
CREATE USER user1 FOR LOGIN user1
GO
use msdb
GO
GRANT EXECUTE ON msdb.dbo.rds_backup_database TO user1
GO
GRANT EXECUTE ON msdb.dbo.rds_restore_database TO user1
GO
```

Umbenennen einer Amazon RDS for Microsoft SQL Server Server-Datenbank in einer Multi-AZ-Bereitstellung

Gehen Sie wie folgt vor, um eine Multi-AZ nutzende Microsoft SQL Server-Datenbank-Instance umzubenennen:

1. Deaktivieren Sie zunächst Multi-AZ für die DB-Instance.
2. Ändern Sie den Datenbanknamen, indem Sie den Befehl ausführe `rdsadmin.dbo.rds_modify_db_name`.
3. Aktivieren Sie anschließend die Multi-AZ-Spiegelung oder AlwaysOn-Verfügbarkeitsgruppen für die DB-Instance, um den Ausgangszustand wiederherzustellen.

Weitere Informationen finden Sie unter Hinzufügen von Multi-AZ zu einer Microsoft SQL Server-DB-Instance.

> **Note**
>
> Falls Ihre Instance keine Multi-AZ verwendet, müssen vor oder nach dem Ausführen von keinerlei Einstellungen geändert werde `rdsadmin.dbo.rds_modify_db_name`.
> Sie können eine Datenbank auf einer Read Replica-Quell-Instance nicht umbenennen.

Beispiel: Im folgenden Beispiel wird mithilfe der gespeicherten Prozedur `rdsadmin.dbo.rds_modify_db_name` die Datenbank **`MOO`** umbenannt in **`ZAR`**. Dies entspricht der ausgeführten Anweisung `DDL ALTER DATABASE [MOO] MODIFY NAME = [ZAR]`.

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'
GO
```

Zurücksetzen der db_owner-Rollenmitgliedschaft für den Master-Benutzer für Amazon for Server RDS SQL

Wenn Sie Ihren Masterbenutzer von der `db_owner` Rollenmitgliedschaft in Ihrer RDS for SQL Server-Datenbank ausschließen und kein anderer Datenbankbenutzer die Mitgliedschaft gewähren kann, können Sie die verloren gegangene Mitgliedschaft wiederherstellen, indem Sie das Passwort für den DB-Instance-Masterbenutzer ändern.

Durch die Änderung des Masterbenutzerpassworts der DB-Instance wird den Datenbanken in der DB-Instance die `db_owner` Mitgliedschaft RDS gewährt, die möglicherweise versehentlich gesperrt wurde. Sie können das DB-Instance-Passwort ändern, indem Sie die RDS Amazon-Konsole modify-db-instance, den AWS CLI Befehl oder die odifyDBInstanceAPIM-Operation verwenden. Weitere Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Wiederherstellung von DB-Instances mit gekündigter Lizenz für Amazon for Server RDS SQL

Microsoft hat einige RDS Amazon-Kunden, die ihre Microsoft-Lizenzmobilitätsinformationen nicht gemeldet haben, aufgefordert, ihre DB-Instance zu beenden. Amazon erstellt RDS Snapshots dieser DB-Instances, und Sie können aus dem Snapshot eine Wiederherstellung auf einer neuen DB-Instance durchführen, für die das Modell License Included gilt.

Sie können aus einem Snapshot der Standard Edition die Standard Edition oder Enterprise Edition wiederherstellen.

Sie können aus einem Snapshot der Enterprise Edition die Standard Edition oder Enterprise Edition wiederherstellen.

So stellen Sie Daten von einem SQL Server-Snapshot wieder her, nachdem Amazon einen endgültigen Snapshot Ihrer Instance erstellt RDS hat

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den Snapshot Ihrer SQL Server-DB-Instance aus. Amazon RDS erstellt einen endgültigen Snapshot Ihrer DB-Instance. Der Name des beendeten Instance-Snapshots weist das Format au *`instance_name`*`-final-snapshot`. Wenn Ihr DB-Instance-Name beispielsweise lautet**`mytest.cdxgahslksma.us-east-1.rds.com`**, wird der endgültige Snapshot aufgerufen **`mytest-final-snapshot`** und befindet sich in derselben AWS Region wie die ursprüngliche DB-Instance.
4. Wählen Sie unter Actions (Aktionen) die Option Restore Snapshot (Snapshot wiederherstellen).

   Das Fenster Restore DB Instance (DB-Instance wiederherstellen) wird angezeigt.
5. Für License Model (Lizenzmodell) wählen Sie license-included (Lizenz enthalten) aus.
6. Wählen Sie die SQL Server-DB-Engine aus, die Sie verwenden möchten.

7. Geben Sie für DB Instance Identifier (DB-Instance-Kennung) den Namen Ihrer wiederhergestellten DB-Instance ein.
8. Klicken Sie auf Restore DB Instance (DB-Instance wiederherstellen).

Weitere Informationen zum Wiederherstellen aus einem Snapshot finden Sie unter Wiederherstellung auf einer DB-Instance.

Umstellung einer Amazon RDS for SQL Server-Datenbank von zu OFFLINE ONLINE

Sie können Ihre Microsoft SQL Server-Datenbank auf einer Amazon RDS DB-Instance von `OFFLINE` auf umstellen`ONLINE`.

SQLServer-Methode	RDSAmazon-Methode
ALTER DATABASE *db_name* SET ONLINE;	EXECrdsadmin.dbo.rds_set_database_online *db_name*

Verwenden von Change Data Capture für Amazon RDS for SQL Server

Amazon RDS unterstützt auch die Erfassung von Datenänderungen (Change Data Capture, CDC) für Ihre DB-Instances, die auf Microsoft SQL Server laufen. CDC erfasst Änderungen an Daten in Ihren Tabellen. Es speichert Metadaten über jede Änderung, auf die Sie später zugreifen können. Weitere Informationen über die Arbeitsweise von CDC finden Sie unter Change Data Capture in der Microsoft-Dokumentation. Bevor Sie CDC für Ihre Amazon RDS-DB-Instances verwenden, aktivieren Sie es in der Datenbank, indem Sie ausführe `msdb.dbo.rds_cdc_enable_db`. Nachdem CDC aktiviert wurde, kann jeder Benutzer, der `db_owner` dieser Datenbank ist, CDC für Tabellen in dieser Datenbank aktivieren oder deaktivieren.

> **Important**
>
> Während einer Wiederherstellung wird CDC deaktiviert. Alle zugehörigen Metadaten werden automatisch aus der Datenbank entfernt. Dies gilt für point-in-time Snapshot-Wiederherstellungen, Wiederherstellungen und SQL Server Native-Wiederherstellungen aus S3. Nachdem Sie eine dieser Wiederherstellungsarten durchgeführt haben, können Sie CDC wieder aktivieren und nachzuverfolgende Tabellen neu festlegen.

Um CDC für eine DB-Instance zu aktivieren, führen Sie die gespeicherte `msdb.dbo.rds_cdc_enable_db`-Prozedur aus.

```
exec msdb.dbo.rds_cdc_enable_db 'database_name'
```

Um CDC für eine DB-Instance zu deaktivieren, führen Sie die gespeicherte `msdb.dbo.rds_cdc_disable_db`-Prozedur aus.

```
exec msdb.dbo.rds_cdc_disable_db 'database_name'
```

Gehen Sie wie folgt vor, um einem Benutzer CDC-Berechtigungen zu erteilen:

```
go
  GRANT EXECUTE ON msdb.dbo.rds_cdc_enable_db TO User1
  GRANT EXECUTE ON msdb.dbo.rds_cdc_disable_db TO User1
```

Themen

- Nachverfolgen von Tabellen mit Change Data Capture
- Change Data Capture-Aufträge
- Change Data Capture für Multi-AZ-Instances

Nachverfolgen von Tabellen mit Change Data Capture

Nachdem CDC für die Datenbank aktiviert wurde, können Sie mit der Nachverfolgung spezifischer Tabellen beginnen. Sie wählen die nachzuverfolgenden Tabellen durch Ausführen von sys.sp_cdc_enable_table.

```
--Begin tracking a table
exec sys.sp_cdc_enable_table
   @source_schema           = N'source_schema'
,  @source_name             = N'source_name'
,  @role_name               = N'role_name'

--The following parameters are optional:

--, @capture_instance       = 'capture_instance'
--, @supports_net_changes   = supports_net_changes
--, @index_name             = 'index_name'
```

```
--, @captured_column_list   = 'captured_column_list'
--, @filegroup_name         = 'filegroup_name'
--, @allow_partition_switch = 'allow_partition_switch'
;
```

Um die CDC-Konfiguration für Ihre Tabellen anzuzeigen, führen Sie sys.sp_cdc_help_change_data_capture aus.

```
--View CDC configuration
exec sys.sp_cdc_help_change_data_capture

--The following parameters are optional and must be used together.
--  'schema_name', 'table_name'
;
```

Weitere Informationen zu CDC-Tabellen, Funktionen und gespeicherten Prozeduren in der SQL Server-Dokumentation finden Sie im Folgenden:

- Change Data Capture – gespeicherte Prozeduren (Transact-SQL)
- Change Data Capture – Funktionen (Transact-SQL)
- Change Data Capture – Tabellen (Transact-SQL)

Change Data Capture-Aufträge

Wenn Sie CDC aktivieren, erstellt SQL Server die CDC-Aufträge. Datenbankbesitzer (`db_owner`) können die CDC-Aufträge anzeigen, erstellen, ändern und löschen. Sie gehören jedoch dem RDS-Systemkonto. Aus diesem Grund sind die Aufträge in nativen Ansichten, Prozeduren oder in SQL Server Management Studio nicht sichtbar.

Für die Kontrolle des Verhaltens von CDC in einer Datenbank verwenden Sie native SQL Server-Prozeduren wie sp_cdc_enable_table und sp_cdc_start_job. Für die Änderung der Parameter von CDC-Aufträgen wie `maxtrans` und `maxscans` verwenden Sie sp_cdc_change_job.

Um weitere Informationen zu den CDC-Aufträgen zu erhalten, können Sie die folgenden dynamischen Verwaltungsansichten abfragen:

- sys.dm_cdc_errors
- sys.dm_cdc_log_scan_sessions
- sysjobs

- sysjobhistory

Change Data Capture für Multi-AZ-Instances

Wenn Sie CDC für eine Multi-AZ-Instance verwenden, vergewissern Sie sich, dass die CDC-Auftragskonfiguration der Spiegelung mit derjenigen auf dem Prinzipal übereinstimmt. CDC-Aufträge werden auf die abgebilde `database_id`. Wenn sich die Datenbank IDs auf der Sekundärseite vom Prinzipal unterscheidet, werden die Jobs nicht der richtigen Datenbank zugeordnet. Um Fehler nach dem Failover zu vermeiden, verwirft RDS die Aufträge auf dem neuen Prinzipal und erstellt sie neu. Die neu erstellten Aufträge verwenden die Parameter, die der Prinzipal vor dem Failover aufgezeichnet hat.

Obwohl dieser Prozess schnell abläuft, ist es immer noch möglich, dass die CDC-Aufträge ausgeführt werden, bevor RDS sie korrigieren kann. Hier gibt es drei Möglichkeiten zum Erzwingen der Konsistenz der Parameter von primären und sekundären Replicas:

- Verwenden Sie die gleichen Auftragsparameter für alle Datenbanken, die CDC aktiviert haben.
- Bevor Sie die CDC-Auftragskonfiguration ändern, konvertieren Sie die Multi-AZ-Instance zu Single-AZ.
- Übertragen Sie die Parameter manuell, wenn Sie sie auf dem Prinzipal ändern.

Um die CDC-Parameter anzuzeigen und zu definieren, mit denen die CDC-Aufträge nach einem Failover neu erstellt werden, verwenden Sie `rds_show_configuration` und `rds_set_configuration`.

Im folgenden Beispiel wird der Wert für zurückgegebe `cdc_capture_maxtrans`. Für alle Parameter, die auf `RDS_DEFAULT` gesetzt sind, konfiguriert RDS den Wert automatisch.

```
-- Show configuration for each parameter on either primary and secondary replicas.
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans';
```

Um die Konfiguration auf dem sekundären Server einzurichten, führen Sie au `rdsadmin.dbo.rds_set_configuration`. Dieses Verfahren legt die Parameterwerte für alle Datenbanken auf dem sekundären Server fest. Diese Einstellungen werden nur nach einem Failover verwendet. Im folgenden Beispiel wird `maxtrans` für alle CDC-Erfassungsaufträge der Wert wie folgt festgelegt: *1000*

```
--To set values on secondary. These are used after failover.
```

```
exec rdsadmin.dbo.rds_set_configuration 'cdc_capture_maxtrans', 1000;
```

Um die CDC-Auftragsparameter auf dem Prinzipal festzulegen, verwenden Sie stattdessen sys.sp_cdc_change_job.

Verwenden des SQL Server-Agenten für Amazon RDS

Mit Amazon RDS können Sie SQL Server Agent in einer DB-Instance verwenden, in der Microsoft SQL Enterprise Edition, Standard Edition oder Web Edition ausgeführt wird. SQL Server Agent ist ein Service von Microsoft Windows für die Ausführung geplanter administrativer Aufgaben, so genannte Jobs. Sie können SQL Server Agent für die Ausführung von T-SQL-Jobs für den erneuten Aufbau von Indizes, die Ausführung von Beschädigungsprüfungen und für die Zusammenfassung von Daten in einer SQL Server-DB-Instance verwenden.

Wenn Sie eine SQL-Server-DB-Instance erstellen, wird der Hauptbenutzer mit der Rolle `SQLAgentUserRole` registriert.

SQL Server Agent kann einen Job nach Plan, als Antwort auf ein bestimmtes Ereignis oder auf Abruf ausführen. Weitere Informationen finden Sie unter SQL Server Agent in der Microsoft-Dokumentation.

> **Note**
>
> Vermeiden Sie es, Jobs zu planen, die während der Wartungs- und Backup-zeitfenster für Ihre DB-Instance ausgeführt werden. Die Wartungs- und Sicherungsprozesse, die von gestartet werden, AWS könnten einen Job unterbrechen oder dazu führen, dass er abgebrochen wird.
>
> In Multi-AZ-Bereitstellungen werden Aufträge von SQL Server Agent vom primären Host auf den sekundären Host repliziert, wenn die Auftragsreplikationsfunktion aktiviert ist. Weitere Informationen finden Sie unter Aktivieren der Auftragsreplikation von SQL Server Agent.
>
> Multi-AZ-Bereitstellungen haben ein Limit von 10.000 SQL-Server-Agent-Aufträgen. Wenn Sie ein höheres Limit benötigen, fordern Sie eine Erhöhung an, indem Sie sich an uns wenden Support. Öffnen Sie die Seite des AWS -Support -Centers, melden Sie sich an und wählen Sie Fall erstellen aus. Wählen Sie Service Limit increase (Erhöhung des Servicelimits). Füllen Sie das Formular aus und senden Sie es ab.

Um den Verlauf eines einzelnen SQL Server Agent-Auftrags in SQL Server Management Studio (SSMS) anzuzeigen, öffnen Sie den Objektexplorer, klicken mit der rechten Maustaste auf den Auftrag und wählen dann View History (Verlauf anzeigen).

Da der SQL Server-Agent auf einem verwalteten Host in einer DB-Instance ausgeführt wird, werden einige Aktionen nicht unterstützt:

- Das Ausführen von Replikationsaufträgen und das Ausführen von Befehlszeilenskripts mithilfe von ActiveX, der Windows-Befehlsshell oder Windows PowerShell werden nicht unterstützt.
- Sie können SQL Server-Agent nicht manuell starten, stoppen oder neu starten.
- E-Mail-Benachrichtigungen über SQL Server Agent sind von einer DB-Instance aus nicht verfügbar.
- Warnungen und Operatoren von SQL Server-Agenten werden nicht unterstützt.
- Die Verwendung von SQL Server-Agent zum Erstellen von Backups wird nicht unterstützt. Verwenden Sie Amazon RDS, um Ihre DB-Instance zu sichern.
- Derzeit unterstützt RDS für SQL Server die Verwendung von SQL Server-Agent-Token nicht.

Aktivieren der Auftragsreplikation von SQL Server Agent

Sie können die Auftragsreplikation von SQL Server Agent mithilfe der folgenden gespeicherten Prozedur aktivieren:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'SQLAgentJob';
```

Sie können die gespeicherte Prozedur in allen SQL-Server-Versionen ausführen, die von Amazon RDS for SQL Server unterstützt werden. Es werden Aufträge in den folgenden Kategorien repliziert:

- [Uncategorized (Local)]
- [Uncategorized (Multi-Server)]
- [Uncategorized]
- Datenauflister
- Database Engine Tuning Advisor
- Datenbankwartung
- Volltext

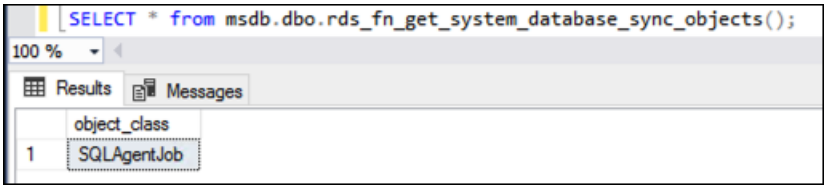
Es werden nur Aufträge, die T-SQL-Auftragsschritte verwenden, repliziert. Jobs mit Schritttypen wie SQL Server Integration Services (SSIS), SQL Server Reporting Services (SSRS), Replikation und PowerShell werden nicht repliziert. Aufträge, die Datenbank-E-Mail und Objekte auf Serverebene verwenden, werden nicht repliziert.

> **Important**
>
> Der primäre Host ist die Informationsquelle für die Replikation. Stellen Sie vor dem Aktivieren der Auftragsreplikation sicher, dass sich Ihre SQL-Server-Agent-Aufträge auf dem primären Host befinden. Andernfalls werden Ihre SQL-Server-Agent-Aufträge ggf. gelöscht, wenn Sie die Funktion aktivieren und sich neuere Aufträge auf dem sekundären Host befinden.

Sie können die folgende Funktion verwenden, um zu überprüfen, ob die Replikation aktiviert ist.

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Die T-SQL-Abfrage gibt Folgendes zurück, wenn die Aufträge von SQL Server Agent repliziert werden. Wenn sie nicht repliziert werden, wird für `object_class` kein Wert zurückgegeben.

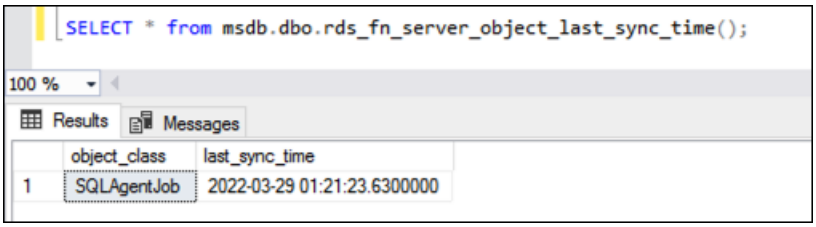


Sie können die folgende Funktion verwenden, um zu ermitteln, wann Objekte zuletzt synchronisiert (UTC) wurden.

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Angenommen, Sie ändern einen Auftrag von SQL Server Agent um 01:00 Uhr. Sie erwarten, dass die letzte Synchronisationszeit nach 01:00 Uhr liegt, was darauf hinweist, dass die Synchronisation stattgefunden hat.

Nach der Synchronisation wird erwartet, dass die für `date_created` und `date_modified` auf dem sekundären Knoten zurückgegebenen Werte übereinstimmen.



Wenn Sie auch die `tempdb` Replikation verwenden, können Sie die Replikation sowohl für SQL Agent-Jobs als auch für die `tempdb` Konfiguration aktivieren, indem Sie sie im folgenden Parameter angeben: `@object_type`

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Weitere Hinweise zur `tempdb` Replikation finden Sie unterTempDB-Konfiguration für Multi-AZ-Bereitstellungen.

Rollen des SQL Server-Agents

RDS for SQL Server unterstützt die folgenden SQL Server-Agent-Rollen mit unterschiedlichen Berechtigungsebenen für die Verwaltung von Aufträgen:

- SQLAgentUserRole

  Berechtigungen
  - Erstellen und verwalten Sie ihre eigenen Jobs, Zeitpläne und Operatoren
  - Sehen Sie sich die Eigenschaften ihrer eigenen Jobs und Zeitpläne an
  - Jobs, die von anderen Benutzern erstellt wurden, können nicht angezeigt oder verwaltet werden

  Diese Rolle eignet sich für Benutzer, die ihre eigenen Jobs erstellen und verwalten müssen, aber keinen Zugriff auf Jobs benötigen, die von anderen Benutzern erstellt wurden.
- SQLAgentReaderRole

  Berechtigungen
  - Alle Berechtigungen von SQLAgent UserRole
  - Eine Liste aller Jobs und Zeitpläne anzeigen, einschließlich der von anderen erstellten
  - Sehen Sie sich die Eigenschaften aller Jobs an
  - Überprüfen Sie den Jobverlauf

  Diese Rolle eignet sich für Benutzer, die den Status aller Jobs überwachen, diese aber nicht verwalten müssen.
- SQLAgentOperatorRole

  Berechtigungen
  - Alle Berechtigungen von SQLAgent UserRole und SQLAgent ReaderRole

- Jobs ausführen, beenden oder starten
- Auftragsverlauf verwalten
- Aktivieren/deaktivieren Sie Jobs und Zeitpläne
- Operatoren und Proxys anzeigen

Diese Rolle bietet die umfassendsten Berechtigungen und eignet sich für Benutzer, die die volle Kontrolle über alle Jobs haben müssen.

Verwenden Sie den folgenden Befehl, um Ihrem SQL Server-Anmeldenamen die Rollen zuzuweisen:

```
USE msdb;
EXEC sp_addrolemember 'SQLAgentOperatorRole', 'username';
```

Verwaltung SQLAgent OperatorRole in RDS für SQL Server

Um die aktuellen Jobs anzuzeigen, müssen Sie den SQLAgent OperatorRole zu Ihrem SQL Server-Anmeldenamen hinzufügen und ihn entfernen, bevor Sie die Verbindung zu Ihrer Datenbank trennen.

Gehen Sie folgendermaßen vor, um die SQL Server-Agent-Struktur im SQL Server Management Studio zu visualisieren:

Zeigen Sie den SQL Server-Agent auf SQL Server Management Studio (SSMS) an

1. Melden Sie sich mit den RDS-Master-Anmeldeinformationen bei der RDS-SQL Server-Instanz an und gewähren Sie dem gewünschten Benutzer die SQLAgentUserRole.

   ```
   USE msdb
   GO
   IF NOT EXISTS(SELECT name FROM sys.database_principals WHERE name = 'UserName')
   BEGIN
   CREATE USER UserName FROM LOGIN UserName
   END
   GO
   ALTER ROLE SQLAgentUserRole ADD MEMBER UserName
   GO
   GRANT ALTER ON ROLE::[SQLAgentOperatorRole] to UserName
   GO
   ```

Mit diesen Befehlen wird der Benutzer in der `msdb` Datenbank erstellt, falls er nicht existiert. Außerdem wird der Benutzer auf dem hinzugefügt SQLAgentUserRole, sodass die SQL Server-Agent-Struktur auf SSMS sichtbar ist. Schließlich gewährt es dem Benutzer Änderungsberechtigungen SQLAgent OperatorRole für. Dies ermöglicht es dem Benutzer, sich add/remove selbst von dieser Rolle aus zu öffnen.

2. Um sich selbst der oben genannten Rolle hinzuzufügen, stellen Sie mit dem Benutzer, der die Jobs sehen muss, eine Verbindung zur RDS-SQL Server-Instanz her und führen Sie das folgende Skript aus.

```
use msdb
go
ALTER ROLE SQLAgentOperatorRole ADD MEMBER UserName
GO
```

Danach klicken Sie mit der rechten Maustaste auf den Ordner Jobs und wählen Sie Refresh.

3. Wenn Sie diese Aktion ausführen, wird auf der Registerkarte Jobs eine Schaltfläche mit dem Pluszeichen (+) angezeigt. Klicken Sie hier, um die Liste der SQL Server-Agent-Jobs zu erweitern.

4. **Important**

   Bevor Sie die Verbindung zur RDS-SQL Server-Instanz trennen, müssen Sie sich selbst aus der entfernen SQLAgentOperatorRole.

   Um Ihren Anmeldenamen aus dem zu entfernen SQLAgentOperatorRole, führen Sie die folgende Abfrage aus, bevor Sie die Verbindung zu Management Studio trennen oder schließen:

```
USE msdb
GO
ALTER ROLE SQLAgentOperatorRole DROP MEMBER UserName
GO
```

Weitere Informationen finden Sie unter Leveraging SQLAgent OperatorRole in RDS SQL Server.

Einen Benutzer zur SQLAgentUser Rolle hinzufügen

Um einem zusätzlichen Anmeldenamen oder Benutzer die Verwendung des SQL Server-Agents zu ermöglichen, melden Sie sich als Masterbenutzer an und gehen Sie wie folgt vor:

1. Erstellen Sie mithilfe des Befehls `CREATE LOGIN` eine weitere Anmeldung auf Serverebene.
2. Erstellen Sie mithilfe des Befehls `msdb` einen Benutzer in `CREATE USER` und verknüpfen Sie dann diesen Benutzer mit der Anmeldung, die Sie im vorherigen Schritt erstellt haben.
3. Fügen Sie den Benutzer`SQLAgentUserRole` mit der gespeicherten Systemprozedur `sp_addrolemember` hinzu.

Nehmen wir zum Beispiel an, Ihr Master-Benutzername lautet **`admin`** und Sie möchten einem Benutzer **`theirname`** mit einem Kennwort Zugriff auf den SQL Server-Agent gewähren**`theirpassword`**. In diesem Fall können Sie das folgende Verfahren verwenden.

Um der SQLAgentUser Rolle einen Benutzer hinzuzufügen

1. Melden Sie sich als Hauptbenutzer an.
2. Führen Sie die folgenden Befehle aus:

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login
 theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
--Added database user theirname in msdb to SQLAgentUserRole in msdb
EXEC sp_addrolemember [SQLAgentUserRole], [theirname];
```

Löschen eines SQL Server-Agent-Jobs

Sie verwenden die gespeicherte Prozedur `sp_delete_job`, um SQL Server-Agent-Jobs auf Amazon RDS für Microsoft SQL Server zu löschen.

Sie können SSMS nicht verwenden, um SQL Server-Agent-Jobs zu löschen. Falls Sie dies versuchen, erhalten Sie eine Fehlermeldung ähnlich der folgenden:

```
The EXECUTE permission was denied on the object 'xp_regread', database
 'mssqlsystemresource', schema 'sys'.
```

Als verwalteter Service ist RDS von der Ausführung von Prozeduren ausgeschlossen, die auf die Windows-Registrierung zugreifen. Wenn Sie SSMS verwenden, versucht es, einen Prozess (`xp_regread`) auszuführen, für den RDS nicht autorisiert ist.

> **Note**
>
> Bei RDS für SQL Server dürfen nur Mitglieder der sysadmin-Rolle Jobs aktualisieren oder löschen, die einem anderen Anmeldenamen gehören. Weitere Informationen finden Sie unter Leveraging SQLAgent OperatorRole in RDS SQL Server.

So löschen Sie einen SQL Server-Agent-Job

- Führen Sie die folgende T-SQL-Anweisung aus:

```
EXEC msdb..sp_delete_job @job_name = 'job_name';
```

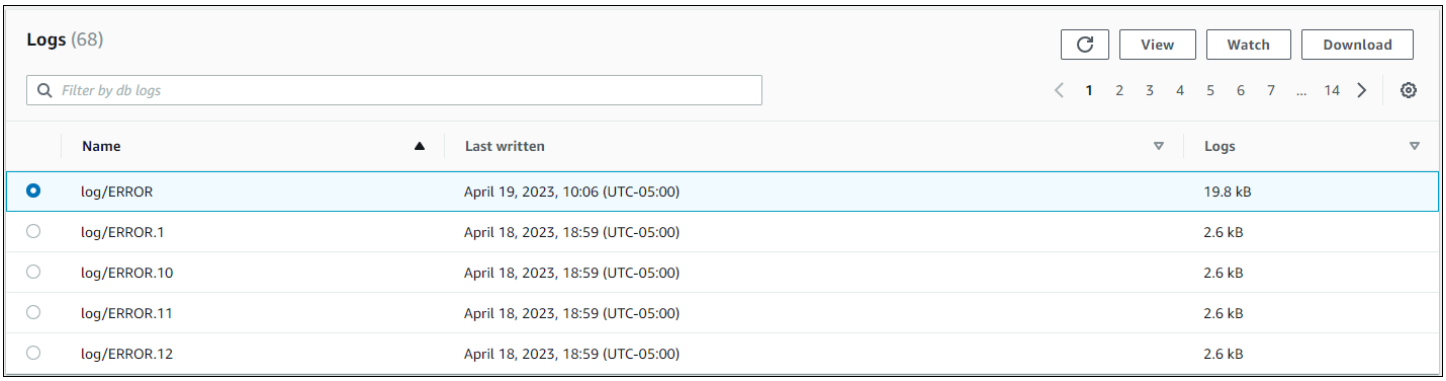
Arbeiten mit Amazon RDS for Microsoft SQL Server Server-Logs

Sie können die SQL-Server-Agent-Protokolle, Microsoft-SQL-Server-Fehlerprotokolle und SQL Server Reporting Services (SSRS)-Protokolle mit der Amazon-RDS-Konsole anzeigen, beobachten und herunterladen.

Beobachten von Protokolldateien

Wenn Sie in der Amazon RDS-Konsole ein Protokoll anzeigen, entsprechen dessen Inhalte diesem Moment. Das Beobachten eines Protokolls in der Konsole öffnet es in einem dynamischen Status, so dass die Aktualisierungen nahezu in Echtzeit angezeigt werden.

Lediglich das aktuelle Protokoll ist zur Beobachtung aktiv. Nehmen wir an, es werden die folgenden Protokolle angezeigt:



Nur Protokoll/FEHLER, da das aktuelle Protokoll aktiv aktualisiert wird. Sie können auch andere beobachten, aber diese sind statisch und werden nicht aktualisiert.

Archivieren von Protokolldateien

In der Amazon RDS-Konsole werden die Protokolle für die letzte Woche bis zum heutigen Tag angezeigt. Die Protokolle können zur Referenzzwecken für einen späteren Zeitpunkt heruntergeladen und archiviert werden. Eine Möglichkeit, Protokolle zu archivieren, besteht darin, sie in einen Amazon S3-Bucket zu laden. Anweisungen zum Einrichten eines Amazon S3-Buckets und zum Hochladen einer Datei finden Sie unter Amazon S3-Grundlagen im Handbuch "Erste Schritte" zu Amazon Simple Storage Service unter Erste Schritte.

Anzeigen von Fehler- und Agent-Protokollen

Um Microsoft SQL Server-Fehler- und Agent-Protokolle anzuzeigen, verwenden Sie die in Amazon RDS gespeicherte Prozedur `rds_read_error_log` mit den folgenden Parametern:

- **`@index`** – die Version des abzurufenden Protokolls. Der Standardwert ist 0, der das aktuelle Fehlerprotokoll abruft. Legen Sie 1 fest, um das vorherige Protokoll abzurufen, legen Sie 2 fest, um das Protokoll davor abzurufen usw.
- **`@type`** – die Art des abzurufenden Protokolls. Legen Sie 1 fest, um ein Fehlerprotokoll abzurufen. Legen Sie 2 fest, um ein Agent-Protokoll abzurufen.

Example

Das folgende Beispiel fordert das aktuelle Fehlerprotokoll an.

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

Weitere Informationen zu SQL Server-Fehlern finden Sie unter Fehler der Datenbank-Engine in der Microsoft-Dokumentation.

Arbeiten mit Trace- und Dump-Dateien für Amazon RDS for Server SQL

In diesem Abschnitt wird die Arbeit mit Trace-Dateien und Dump-Dateien für Ihre RDS Amazon-DB-Instances beschrieben, auf denen Microsoft SQL Server ausgeführt wird.

Generieren einer Trace-Abfrage SQL

```
declare @rc int
declare @TraceID int
declare @maxfilesize bigint

set @maxfilesize = 5

exec @rc = sp_trace_create @TraceID output,  0, N'D:\rdsdbdata\log\rdstest',
 @maxfilesize, NULL
```

Anzeigen eines offenen Trace

```
select * from ::fn_trace_getinfo(default)
```

Anzeigen der Trace-Inhalte

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

Festlegen des Aufbewahrungszeitraums für Trace- und Dump-Dateien

Trace- und Dump-Dateien können sich ansammeln und viel Festplattenspeicher belegen. Standardmäßig löscht Amazon RDS Trace- und Dump-Dateien, die älter als sieben Tage sind.

Um den aktuellen Aufbewahrungszeitraum für Trace- und Dump-Dateien anzuzeigen, verwenden Sie die Prozedur `rds_show_configuration`, wie im folgenden Beispiel zu sehen ist.

```
exec rdsadmin..rds_show_configuration;
```

Um den Aufbewahrungszeitraum für Trace-Dateien zu ändern, verwenden Sie die Prozedur `rds_set_configuration` und legen `tracefile retention` in Minuten fest. Beim folgenden Beispiel wird der Aufbewahrungszeitraum für Trace-Dateien auf 24 Stunden festgelegt:

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

Um den Aufbewahrungszeitraum für Dump-Dateien zu ändern, verwenden Sie die Prozedur `rds_set_configuration` und legen `dumpfile retention` in Minuten fest. Beim folgenden Beispiel wird der Aufbewahrungszeitraum für Dump-Dateien auf 3 Tage festgelegt:

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

Aus Sicherheitsgründen können Sie eine bestimmte Trace- oder Dump-Datei auf einer SQL Server-DB-Instance nicht löschen. Damit alle ungenutzten Trace- oder Dump-Dateien gelöscht werden, legen sie den Aufbewahrungszeitraum für die Dateien auf 0 fest.

Amazon RDS für mich SQL

Amazon RDS unterstützt mehrere Versionen von My SQL for DB-Instances. Vollständige Informationen zu den unterstützten Versionen finden Sie unterMySQL in Amazon RDS-Versionen.

Verwenden Sie die RDS Verwaltungstools oder Schnittstellen von Amazon, um eine Amazon RDS for My SQL DB-Instance zu erstellen. Sie können dann Folgendes durchführen:

- Ändern der Größe Ihrer DB-Instance
- Autorisieren von Verbindungen mit Ihrer DB-Instance
- Erstellen und Wiederherstellen aus Backups oder Snapshots
- Erstellen von sekundären Multi-AZ-Instances
- Erstellen von Read Replicas
- Überwachen der Leistung Ihrer DB-Instance

Um die Daten in Ihrer DB-Instance zu speichern und darauf zuzugreifen, verwenden Sie die SQL Standard-Dienstprogramme und -anwendungen von My.

Amazon RDS for My SQL entspricht vielen Industriestandards. Beispielsweise können Sie RDS for My SQL databases verwenden, um HIPAA -konforme Anwendungen zu erstellen. Sie können „Meine SQL Datenbanken“ verwendenRDS, um gesundheitsbezogene Informationen zu speichern, einschließlich geschützter Gesundheitsinformationen (PHI) im Rahmen einer Geschäftspartnervereinbarung (BAA) mit AWS. Amazon RDS for My erfüllt SQL auch die Sicherheitsanforderungen des Federal Risk and Authorization Management Program (FedRAMP). Darüber hinaus SQL hat Amazon RDS for My von der Fed RAMP Joint Authorization Board (JAB) eine vorläufige Genehmigung zur Geschäftstätigkeit (P-ATO) an der Basis der RAMP HIGH Fed in den AWS GovCloud (US) Regionen erhalten. Weitere Informationen über unterstützte Compliance-Standards finden Sie unter AWS Cloud-Compliance.

Informationen zu den Funktionen der einzelnen Versionen von My SQL finden Sie unter Die wichtigsten Funktionen von My SQL in der SQL Dokumentation zu My.

Bevor Sie eine DB-Instance erstellen, führen Sie die Schritte in Einrichtung Ihrer Amazon RDS-Umgebung aus. Wenn Sie eine DB-Instance erstellen, erhält der RDS Masterbenutzer mit einigen Einschränkungen DBA Rechte. Verwenden Sie dieses Konto für administrative Aufgaben wie das Erstellen zusätzlicher Datenbankkonten.

Sie können das folgende erstellen:

- DB-Instances
- DB-Snapshots
- P oint-in-time stellt wieder her
- Automatische Backups
- Manuelle Backups

Sie können DB-Instances verwenden, auf denen My SQL in einer auf Amazon basierenden Virtual Private Cloud (VPC) ausgeführt wirdVPC. Sie können Ihrer My SQL DB-Instance auch Funktionen hinzufügen, indem Sie verschiedene Optionen aktivieren. Amazon RDS unterstützt Multi-AZ-Bereitstellungen für My SQL als hochverfügbare Failover-Lösung.

> **Important**
>
> Um ein Managed-Serviceerlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind. Sie können mit SQL Standard-Clients wie dem MySQL-Client auf Ihre Datenbank zugreifen. Sie können jedoch nicht direkt mit Telnet oder Secure Shell (SSH) auf den Host zugreifen.

Themen

- Unterstützung von MySQL-Funktionen in Amazon RDS
- MySQL in Amazon RDS-Versionen
- Verbindung zu Ihrer MySQL-DB-Instance herstellen
- Sichern von MySQL-DB-Instance-Verbindungen
- Verbesserung der Abfrageleistung RDS für My SQL mit Amazon RDS Optimized Reads
- Verbesserung der Schreibleistung mit RDS Optimized Writes for My SQL
- Upgrades der DB-Engine RDS für MySQL
- Aktualisierung einer MySQL-DB-Snapshot-Engine-Version
- Daten in eine Amazon RDS for MySQL MySQL-DB-Instance importieren
- Arbeiten mit MySQL-Replikation in Amazon RDS
- Konfiguration von Active-Active-Clustern für My RDS SQL

- Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation
- Optionen für Meine SQL DB-Instances
- Parameter für My SQL
- Allgemeine DBA Aufgaben für My SQL DB-Instances
- Lokale Zeitzone für My SQL DB-Instances
- Bekannte Probleme und Einschränkungen für Amazon RDS for MySQL
- RDSals Referenz für meine SQL gespeicherte Prozedur

Unterstützung von MySQL-Funktionen in Amazon RDS

RDS for MySQL unterstützt die meisten Funktionen von MySQL. Einige Funktionen werden möglicherweise nur begrenzt unterstützt oder haben eingeschränkte Berechtigungen.

Sie können neue Amazon RDS Funktionen auf der Was ist neu mit Datenbank?-Seite filtern. Wählen Sie für den Filter Produkte Amazon RDS aus. Suchen Sie dann mit Schlüsselwörtern wie **`MySQL 2022`**.

> **Note**
>
> Die folgenden Listen sind nicht vollständig.

Themen

- Unterstützung von MySQL-Funktionen auf Amazon RDS for MySQL MySQL-Hauptversionen
- Unterstützte Speicher-Engines für RDS for MySQL
- Verwenden von memcached und anderen Optionen mit MySQL in Amazon RDS
- InnoDB-Cache-Warming für MySQL in Amazon RDS
- Inklusive Sprachänderungen für RDS for MySQL 8.4
- MySQL-Funktionen, die nicht von Amazon RDS unterstützt werden

Unterstützung von MySQL-Funktionen auf Amazon RDS for MySQL MySQL-Hauptversionen

In den folgenden Abschnitten finden Sie Informationen zur Unterstützung von MySQL-Funktionen in den Hauptversionen von Amazon RDS für MySQL:

Themen

- MySQL 8.4-Unterstützung auf Amazon RDS

Informationen zu unterstützten Nebenversionen von Amazon RDS for MySQL finden Sie unterUnterstützte MySQL-Nebenversionen in Amazon RDS.

MySQL 8.4-Unterstützung auf Amazon RDS

Amazon RDS unterstützt die folgenden neuen Funktionen für Ihre DB-Instances, auf denen MySQL Version 8.4 oder höher ausgeführt wird.

- Kryptografische Bibliothek — RDS for MySQL ersetzte OpenSSL durch AWS Libcrypto (AWS-LC), das FIPS 140-3-zertifiziert ist. Weitere Informationen finden Sie im Repository unter. AWS-LC GitHub https://github.com/aws/aws-lc
- Änderungen an TLS — RDS for MySQL unterstützt nur TLS 1.2 und TLS 1.3. Weitere Informationen finden Sie unter the section called “SSL/TLSUnterstützung mit My SQL”.
- Memcached-Unterstützung — Die Memcached-Schnittstelle ist in MySQL 8.4 nicht mehr verfügbar. Weitere Informationen finden Sie unter Unterstützung für MySQL-memcached.
- Standard-Authentifizierungs-Plugin — Das Standard-Authentifizierungs-Plugin ist. `caching_sha2_password` Weitere Informationen finden Sie unter the section called “MySQL-Standardauthentifizierungs-Plugin”.
- **`mysqlpump`**Client-Hilfsprogramm — Das `mysqlpump` Client-Hilfsprogramm ist in MySQL 8.4 nicht mehr verfügbar. Weitere Informationen finden Sie unter mysqldump Rollenbasiertes Privilegienmodell für for My RDS SQL und mysqlpump in der Prescriptive Guidance.AWS
- Gespeicherte Prozeduren für verwaltete Replikation — Wenn Sie gespeicherte Prozeduren verwenden, um die Replikation mit einem mit konfigurierten Replikationsbenutzer zu verwalten, müssen Sie TLS konfigurieren`caching_sha2_password`, indem Sie Folgendes angeben. `SOURCE_SSL=1` `caching_sha2_password`dist das Standard-Authentifizierungs-Plugin für RDS für MySQL 8.4.
- Änderungen des Parameterverhaltens — Die folgenden Parameter wurden für MySQL 8.4 geändert.
  - `innodb_dedicated_server`— Dieser Parameter ist jetzt standardmäßig aktiviert. Weitere Informationen finden Sie unter Konfiguration der Pufferpoolgröße und der Redo-Log-Kapazität in My 8.4 SQL.
  - `innodb_buffer_pool`— Die Datenbank-Engine berechnet diesen Parameter jetzt, aber Sie können diese Einstellung überschreiben. Weitere Informationen finden Sie unter Konfiguration der Pufferpoolgröße und der Redo-Log-Kapazität in My 8.4 SQL.
  - `innodb_redo_log_capacity`— Dieser Parameter steuert jetzt die Größe der Redo-Log-Dateien. Die Datenbank-Engine berechnet diesen Parameter jetzt, aber Sie können diese Einstellung überschreiben. Weitere Informationen finden Sie unter Konfiguration der Pufferpoolgröße und der Redo-Log-Kapazität in My 8.4 SQL.

- Veraltete oder entfernte Parameter — RDS for MySQL hat die folgenden Parameter aus Parametergruppen für MySQL 8.4-DB-Instances entfernt. Der `innodb_redo_log_capacity` Parameter steuert jetzt die Größe der Redo-Log-Dateien.
  - `innodb_log_file_size`
  - `innodb_log_files_in_group`
- Neue Standardwerte für Parameter — Die folgenden Parameter haben neue Standardwerte für MySQL 8.4-DB-Instances:
  - Verschiedene Parameter der MySQL-Community im Zusammenhang mit der Leistung wurden geändert. Weitere Informationen finden Sie unter Was ist neu in MySQL 8.4 seit MySQL 8.0.

    Wir empfehlen Ihnen, die Leistung Ihrer Anwendung auf RDS for MySQL 8.4 zu testen, bevor Sie eine Produktionsinstanz migrieren.
  - `innodb_purge_threads`— Der Standardwert ist auf die Formel gesetzt, um `LEAST({DBInstanceVCPU/2},4)` zu verhindern, dass die Länge der InnoDB-Verlaufsliste zu groß wird.
  - `group_replication_exit_state_action`— Der Standardwert ist`OFFLINE_MODE`, was dem Standard in der MySQL Community entspricht. Weitere Informationen finden Sie unter Überlegungen und bewährte Methoden RDS für Meine Active-Active-Cluster SQL.
  - `binlog_format`— Der Standardwert ist`ROW`, was dem Standard in der MySQL Community entspricht. Sie können den Parameter für Single-AZ-DB-Instances oder Multi-AZ-DB-Instances ändern, aber nicht für Multi-AZ-DB-Cluster. Multi-AZ-DB-Cluster verwenden halbsynchrone Replikation, und wenn `binlog_format` auf oder gesetzt ist, schlägt die Replikation fehl. `MIXED` `STATEMENT`
- Inklusive Sprachänderungen — RDS for MySQL 8.4 beinhaltet Änderungen gegenüber RDS for MySQL 8.0 in Bezug auf Schlüsselwörter und Systemschemas für inklusive Sprache. Weitere Informationen finden Sie unter Inklusive Sprachänderungen für RDS for MySQL 8.4.

Eine Liste aller Funktionen und Änderungen von MySQL 8.4 finden Sie unter Was ist neu in MySQL 8.4 seit MySQL 8.0 in der MySQL-Dokumentation.

Eine Liste der nicht unterstützten Funktionen finden Sie unter the section called “Nicht unterstützte Features”.

Unterstützte Speicher-Engines für RDS for MySQL

Während MySQL mehrere Speicher-Engines mit unterschiedlichen Fähigkeiten unterstützt, sind nicht alle von ihnen für die Wiederherstellung und Langlebigkeit von Daten optimiert. Amazon RDS unterstützt die InnoDB-Speicher-Engine für MySQL-DB-Instances vollständig. Amazon RDS-Funktionen wie Point-In-Time Wiederherstellung und Snapshot-Wiederherstellung erfordern eine wiederherstellbare Speicher-Engine und werden nur für die InnoDB-Speicher-Engine unterstützt. Weitere Informationen finden Sie unter Unterstützung für MySQL-memcached.

Die verbündete Speicher-Engine wird aktuell von Amazon RDS for MySQL nicht unterstützt.

Bei von Benutzern erstellten Schemas unterstützt die MyISAM-Speicher-Engine keine zuverlässige Wiederherstellung. Dies kann dazu führen, dass Daten verloren gehen oder beschädigt werden, wenn MySQL nach einer Wiederherstellung neu gestartet wird, wodurch verhindert wird, dass die Wiederherstellung oder Point-In-Time Snapshot-Wiederherstellung wie vorgesehen funktioniert. Wenn Sie trotzdem MyISAM mit Amazon RDS-Snapshots verwenden möchten, kann dies unter Umständen hilfreich sein.

> **Note**
>
> Systemtabellen im `mysql`-Schema können sich im MyISAM-Speicher befinden.

Wenn Sie vorhandene MyISAM-Tabellen in InnoDB-Tabellen konvertieren möchten, können Sie den Befehl `ALTER TABLE` verwenden (z. B. `alter table TABLE_NAME engine=innodb;`). Vergessen Sie nicht, dass MyISAM und InnoDB verschiedene Stärken und Schwächen haben, also sollten Sie eine vollständige Bewertung der Auswirkungen vornehmen, bevor Sie Ihre Anwendungen umstellen.

MySQL 5.1, 5.5 und 5.6 werden in Amazon RDS nicht mehr unterstützt. Sie können jedoch bestehende MySQL 5.1-, 5.5- und 5.6-Snapshots bei Bedarf wiederherstellen. Wenn Sie einen MySQL-5.1-, 5.5- oder 5.6-Snapshot wiederherstellen, wird die DB-Instance automatisch auf MySQL 5.7 aktualisiert.

Verwenden von memcached und anderen Optionen mit MySQL in Amazon RDS

Die meisten Amazon RDS-DB-Engines unterstützen Optionsgruppen, die Ihnen ermöglichen, zusätzliche Funktionen für Ihre DB-Instance auszuwählen. DB-Instances von RDS for MySQL

unterstützen die Option `memcached`, einen einfachen schlüsselbasierten Cache. Weitere Informationen über `memcached` und andere Optionen finden Sie unter Optionen für Meine SQL DB-Instances. Weitere Informationen über das Arbeiten mit Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

InnoDB-Cache-Warming für MySQL in Amazon RDS

Die InnoDB-Cache-Warnung kann zu Leistungssteigerungen Ihrer MySQL-DB-Instance führen, indem der aktuelle Zustand des Bufferpools gespeichert wird, wenn die DB-Instance heruntergefahren wird, und beim Hochfahren der DB-Instance der Bufferpool erneut aus den gespeicherten Informationen geladen wird. Dies überbrückt die ansonsten notwendige "Aufwärmphase" des Bufferpools bei normaler Verwendung von Datenbanken und lädt den Bufferpool vorab mit den Seiten für bereits bekannte Anfragen. Die Datei, die den gespeicherten Bufferpool abspeichert, speichert nur Metadaten für die Seite ab, die sich in dem Bufferpool befinden, nicht die Seiten selbst. Dadurch benötigt die Datei nur wenig Speicherplatz. Die Dateigröße beträgt nur 0,2 % der Cachegröße. So beträgt beispielsweise bei einem 64 GiB großen Cache die Größe der Cache-Initialisierungsdatei 128 MiB. Weitere Informationen zur InnoDB-Cache-Initialisierung finden Sie unter Saving and Restoring the Buffer Pool State in der MySQL-Dokumentation.

DB-Instances von RDS for MySQL unterstützen InnoDB-Cache-Warming. Um InnoDB-Cache-Initialisierung zu aktivieren, setzen Sie die Parameter `innodb_buffer_pool_dump_at_shutdown` und `innodb_buffer_pool_load_at_startup` in der Parametergruppe für Ihre DB-Instance auf 1. Die Änderung dieser Parameterwerte in einer Parametergruppe wirkt sich auf alle MySQL-DB-Instances aus, die diese Parametergruppe verwenden. Sie müssen möglicherweise eine neue Parametergruppe für diese Instances erstellen, um InnoDB-Cache-Initialisierung für bestimmte MySQL-DB-Instances zu aktivieren. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

InnoDB-Cache-Initialisierung bringt vor allem einen Leistungsvorteil für DB-Instances, die den Standardspeicher verwenden. Wenn Sie PIOPS-Speicher verwenden, findet in der Regel keine signifikante Leistungssteigerung statt.

> **Important**
>
> Wenn Ihre MySQL-DB-Instance nicht ordnungsgemäß herunterfährt, wie beispielsweise während eines Failovers, dann wird der Bufferpool nicht auf der Festplatte gespeichert. In diesem Fall lädt MySQL eine verfügbare Bufferpool-Datei, wenn die DB-Instance neu gestartet wurde. Das ist nicht schlimm, aber der wiederhergestellte Zwischenspeicher-Pool

spiegelt möglicherweise nicht den aktuellsten Stand des Zwischenspeicher-Pools vor dem Neustart dar. Wir empfehlen Ihnen, Ihren Bufferpool in regelmäßigen Abschnitten in Ihrem Interesse zu verwerfen, um sicherzustellen, dass Sie immer den aktuellsten Zustand in Ihrem Bufferpool für die Initialisierung des Cache beim Starten von InnoDB haben.
Sie können ein Ereignis erstellen, das den Bufferpool in regelmäßigen Abständen automatisch verwirft. Beispielsweise erstellt das folgende Statement ein Ereignis mit dem Namen `periodic_buffer_pool_dump`, das den Bufferpool stündlich verwirft.

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Weitere Informationen zu MySQL-Ereignissen finden Sie unter Event Syntax in der MySQL-Dokumentation.

Entladen und Laden des Zwischenspeicher-Pools auf Abruf

Sie können den InnoDB-Cache „nach Bedarf“ speichern und laden.

- Rufen Sie die gespeicherte Prozedur mysql.rds_innodb_buffer_pool_dump_now auf, um den aktuellen Zustand des Bufferpools auf der Festplatte zu verwerfen.
- Rufen Sie die gespeicherte Prozedur mysql.rds_innodb_buffer_pool_load_now auf, um den Zustand des Bufferpools auf der Festplatte zu laden oder zu speichern.
- Rufen Sie die gespeicherte Prozedur mysql.rds_innodb_buffer_pool_load_abort auf, um eine ladende Operation abzubrechen.

Inklusive Sprachänderungen für RDS for MySQL 8.4

RDS for MySQL 8.4 enthält Änderungen gegenüber der MySQL 8.4 Community Edition in Bezug auf Schlüsselwörter und Systemschemas für eine inklusive Sprache. Zum Beispiel ersetzt `SHOW SLAVE STATUS` der `SHOW REPLICA STATUS` Befehl.

Themen

- Der Name der Konfigurationsparameter ändert sich
- Der Name der gespeicherten Prozedur ändert sich

Der Name der Konfigurationsparameter ändert sich

Die folgenden Konfigurationsparameter haben in RDS für MySQL 8.4 neue Namen.

Aus Kompatibilitätsgründen können Sie die Parameternamen im `mysql` Client überprüfen, indem Sie einen der beiden Namen verwenden. Sie können die neuen Namen nur verwenden, wenn Sie Werte in einer benutzerdefinierten MySQL 8.4-Parametergruppe ändern. Weitere Informationen finden Sie unter Standard- und benutzerdefinierte Parametergruppen.

Name, der entfernt werden soll	Neuer oder bevorzugter Name
`init_slave`	`init_replica`
`log_slave_updates`	`log_replica_updates`
`log_slow_slave_statements`	`log_slow_replica_statements`
`rpl_stop_slave_timeout`	`rpl_stop_replica_timeout`
`skip_slave_start`	`skip_replica_start`
`slave_checkpoint_group`	`replica_checkpoint_group`
`slave_checkpoint_period`	`replica_checkpoint_period`
`slave_compressed_protocol`	`replica_compressed_protocol`
`slave_exec_mode`	`replica_exec_mode`
`slave_load_tmpdir`	`replica_load_tmpdir`
`slave_max_allowed_packet`	`replica_max_allowed_packet`
`slave_net_timeout`	`replica_net_timeout`

Name, der entfernt werden soll	Neuer oder bevorzugter Name
slave_parallel_type	replica_parallel_type
slave_parallel_workers	replica_parallel_workers
slave_pending_jobs_size_max	replica_pending_jobs_size_max
slave_preserve_commit_order	replica_preserve_commit_order
slave_skip_errors	replica_skip_errors
slave_sql_verify_checksum	replica_sql_verify_checksum
slave_transaction_retries	replica_transaction_retries
slave_type_conversions	replica_type_conversions
sql_slave_skip_counter	sql_replica_skip_counter

> **Note**
>
> Der Parameter `replica_allow_batching` ist nicht verfügbar, da Amazon RDS keine NDB-Cluster unterstützt.

Der Name der gespeicherten Prozedur ändert sich

Die folgenden gespeicherten Prozeduren haben in RDS für MySQL 8.4 neue Namen.

Aus Kompatibilitätsgründen können Sie beide Namen in der ersten Version von RDS for MySQL 8.4 verwenden. Die alten Prozedurnamen sollen in einer zukünftigen Version entfernt werden. Weitere Informationen finden Sie unter Konfiguration, Starten und Stoppen der Binärprotokollreplikation (Binlog).

Name, der entfernt werden soll	Neuer oder bevorzugter Name
`mysql.rds_next_master_log`	`mysql.rds_next_source_log`
`mysql.rds_reset_external_master`	`mysql.rds_reset_external_source`
`mysql.rds_set_external_master`	`mysql.rds_set_external_source`
`mysql.rds_set_external_master_with_auto_position`	`mysql.rds_set_external_source_with_auto_position`
`mysql.rds_set_external_master_with_delay`	`mysql.rds_set_external_source_with_delay`
`mysql.rds_set_master_auto_position`	`mysql.rds_set_source_auto_position`

MySQL-Funktionen, die nicht von Amazon RDS unterstützt werden

Amazon RDS bietet zurzeit keine Unterstützung für die folgenden MySQL-Funktionen:

- Authentifizierungs-Plugin
- Fehlerprotokollierung im Systemprotokoll
- InnoDB-Tablespace-Verschlüsselung
- NDB-Cluster
- Passwortstärke-Plugin
- Persistente Systemvariablen

- Plugin für das Umschreiben von Abfragen
- Halbsynchrone Replikation, außer bei Multi-AZ-DB-Clustern
- Transportierbarer Tablespace
- X-Plugin

Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Shell-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind. Amazon RDS unterstützt den Zugriff auf Datenbanken in einer DB-Instance mit jeder beliebigen Standard-SQL-Client-Anwendung. Amazon RDS erlaubt keinen direkten Hostzugriff auf eine DB-Instance über Telnet, Secure Shell (SSH) oder Windows Remote Desktop Connection. Wenn Sie eine DB-Instance erstellen, wird Ihnen für alle Datenbanken auf dieser Instance der Status db_owner zugewiesen, und Sie verfügen über alle Berechtigungen auf Datenbankebene, mit Ausnahme der für Backups verwendeten Berechtigungen. Amazon RDS verwaltet Backups für Sie.

MySQL in Amazon RDS-Versionen

In MySQL werden Versionsnummern als Version = X.Y.Z organisiert. In der Amazon RDS-Terminologie bezeichnet X.Y die Hauptversion und Z ist die Nummer der Unterversion. Bei Amazon RDS-Implementierungen gilt ein Versionswechsel als wesentlich, wenn sich die Hauptversionsnummer ändert — z. B. von Version 5.7 auf 8.0. Eine Versionsänderung gilt als geringfügig, wenn sich nur die geringfügige Versionsnummer ändert, z. B. wenn von Version 8.0.32 auf 8.0.34 umgestellt wird.

Themen

- Unterstützte MySQL-Nebenversionen in Amazon RDS
- Unterstützte MySQL-Hauptversionen in Amazon RDS
- Amazon RDS-Versionen mit erweitertem Support für RDS für MySQL
- Arbeiten mit der Datenbank-Vorschauumgebung
- MySQL Version 9.3 in der Database Preview-Umgebung
- MySQL Version 9.2 in der Database Preview-Umgebung
- MySQL Version 9.1 in der Database Preview-Umgebung
- Veraltete Versionen für Amazon RDS for MySQL

Unterstützte MySQL-Nebenversionen in Amazon RDS

Amazon RDS unterstützt derzeit die folgenden MySQL-Nebenversionen.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.
> Amazon RDS Extended Support ist für Nebenversionen nicht verfügbar.

Die folgende Tabelle zeigt die Nebenversionen von MySQL 8.4, die Amazon RDS derzeit unterstützt.

MySQL-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
8.4.6	22. Juli 2025	1. August 2025	1. August 2026
8.4.5	15. April 2025	29. April 2025	September 2026
8.4.4	21. Januar 2025	19. Februar 2025	März 2026
8.4.3	15. Oktober 2024	21. November 2024	März 2026

Die folgende Tabelle zeigt die Nebenversionen von MySQL 8.0, die Amazon RDS derzeit unterstützt.

> **Note**
>
> Nebenversionen können das Ende der Standardunterstützung erreichen, bevor dies bei Hauptversionen der Fall ist. Beispielsweise hat die Nebenversion 8.0.28 das Ende des Standardsupports am 28. März 2024 erreicht, während die Hauptversion 8.0 dieses Datum am 31. Juli 2026 erreicht. RDS wird weitere 8.0.*-Nebenversionen unterstützen, die die MySQL-Community zwischen diesen Daten veröffentlicht. Wir empfehlen, dass Sie für alle Hauptversionen so oft wie möglich auf die neueste verfügbare Nebenversion aktualisieren.

MySQL-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
8.0.43	22. Juli 2025	1. August 2025	1. August 2026
8.0.42	15. April 2025	29. April 2025	Juli 2026
8.0.41	21. Januar 2025	19. Februar 2025	März 2026
8.0.40	15. Oktober 2024	13. November 2024	März 2026
8.0.39	23. Juli 2024	12. August 2024	Oktober 2025

MySQL-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS-Ende des Standard-Supportdatums
8.0.37	30. April 2024	18. Juni 2024	Oktober 2025

Die folgende Tabelle zeigt die Nebenversionen von MySQL 5.7, die unter Amazon RDS Extended Support verfügbar sind.

> **Note**
>
> Bei Nebenversionen kann der erweiterte Support vor den Hauptversionen auslaufen. Beispielsweise erreicht die Nebenversion 5.7.44-RDS.20240529 das Ende des erweiterten Support im September 2025, während die Hauptversion 5.7 dieses Datum am 28. Februar 2027 erreicht. RDS wird zwischen diesen Daten weitere 5.7.44-RDS.xxyyzz Nebenversionen generieren und veröffentlichen. Wir empfehlen, für alle Hauptversionen so oft wie möglich auf die neueste verfügbare Nebenversion zu aktualisieren.

MySQL-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS: Ende des Extended Support
5.7.44-RDS.20250508*	Nicht zutreffend	20. Mai 2025	September 2026
5.7.44-RDS.20250213*	Nicht zutreffend	12. März 2025	September 2026
5.7.44-RDS.20250103*	Nicht zutreffend	13. Februar 2025	März 2026
5.7.44-RDS.20240808*	Nicht zutreffend	28. August 2024	Oktober 2025
5.7.44-RDS.20240529*	Nicht zutreffend	25. Juni 2024	Oktober 2025

MySQL-Engine-Version	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	RDS: Ende des Extended Support
5.7.44-RDS.20240408*	Nicht zutreffend	17. Mai 2024	Oktober 2025

* Die MySQL Community hat die Hauptversion 5.7 eingestellt und wird keine neuen Nebenversionen veröffentlichen. Dies ist eine Nebenversion, die Amazon RDS mit wichtigen Sicherheitspatches und Bugfixes für MySQL 5.7-Datenbanken veröffentlicht hat, die unter RDS Extended Support fallen. Weitere Informationen zu diesen Nebenversionen finden Sie unterthe section called “Versionen mit erweitertem RDS-Support”. Weitere Informationen zu RDS Extended Support finden Sie unterErweiterter Support für Amazon RDS mit Amazon RDS.

Sie können eine beliebige aktuell unterstützte MySQL-Version festlegen, wenn Sie eine DB-Instance erstellen. Sie können die Hauptversionen (wie z. B. MySQL 5.7) sowie eine beliebige unterstützte Unterversion für die festgelegte Hauptversion festlegen. Wenn keine Version angegeben wird, verwendet Amazon RDS standardmäßig eine unterstützte Version - in der Regel die aktuelle Version. Wenn die Hauptversion, jedoch nicht die Unterversion, festgelegt ist, verwendet Amazon RDS standardmäßig den letzten Release der Hauptversion, die Sie festgelegt haben. Führen Sie den `describe-db-engine-versions` AWS CLI Befehl aus, um eine Liste der unterstützten Versionen sowie die Standardeinstellungen für neu erstellte DB-Instances anzuzeigen.

Um beispielsweise die unterstützten Engine-Versionen für RDS for MySQL aufzulisten, führen Sie den folgenden CLI-Befehl aus:

```
aws rds describe-db-engine-versions --engine mysql --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Die standardmäßige MySQL-Version kann je nach AWS-Region variieren. Um eine DB-Instance mit einer bestimmten Unterversion zu erstellen, geben Sie die Unterversion bei der Erstellung der DB-Instance an. Sie können die standardmäßige Nebenversion für eine ermitteln, AWS-Region indem Sie den folgenden AWS CLI Befehl ausführen:

```
aws rds describe-db-engine-versions --default-only --engine mysql
 --engine-version major_engine_version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

*major_engine_version*Ersetzen Sie durch die Hauptversion der Engine und *region* ersetzen Sie sie durch die AWS-Region. Der folgende AWS CLI Befehl gibt beispielsweise die Standardversion der MySQL-Nebenengine für die Hauptversion 5.7 und die USA West (Oregon) AWS-Region (us-west-2) zurück:

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version 5.7
 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output
 text
```

Mit Amazon RDS können Sie steuern, wann Ihre MySQL-Instance auf eine neue Amazon RDS-unterstützte Hauptversion aktualisiert wird. Sie können die Kompatibilität mit bestimmten MySQL-Versionen aufrechterhalten, neue Versionen mit Ihrer Anwendung testen, bevor Sie diese für die Produktion bereitstellen, und Hauptversions-Upgrades zu ausgewählten Zeiten durchführen lassen.

Wenn das automatische Upgrade der Nebenversion aktiviert ist, wird Ihre DB-Instance automatisch auf neue MySQL-Nebenversionen aktualisiert, da diese von Amazon RDS unterstützt werden. Dieser Patch tritt während Ihres geplanten Wartungsfensters auf. Sie können eine DB-Instance ändern, um automatische Upgrades der Nebenversion zu aktivieren oder zu deaktivieren.

Wenn Sie sich von automatisch geplanten Upgrades abmelden, können Sie ein manuelles Upgrade auf eine der unterstützten Unterversionen durchführen, indem Sie die selben Schritte befolgen, wie bei einem Update auf eine Hauptversion. Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Amazon RDS unterstützt derzeit die folgenden Upgrades für Hauptversionen der MySQL-Datenbank-Engine:

- MySQL 5.7 auf MySQL 8.0
- MySQL 8.0 auf MySQL 8.4

Da Hauptversionsaktualisierungen Kompatibilitätsrisiken bergen, werden sie nicht automatisch ausgeführt. Sie müssen für das Ändern der DB-Instance eine Anfrage stellen. Sie sollten alle Upgrades gründlich testen, bevor Sie Ihre Produktions-Instances aktualisieren. Weitere Informationen über das Upgraden einer MySQL-DB-Instance finden Sie unter Upgrades der DB-Engine RDS für MySQL.

Sie können vor dem Aktualisieren eine DB-Instance gegen eine neue Version testen, indem Sie einen DB-Snapshot Ihrer bestehenden DB-Instance erstellen, von diesem DB-Snapshot eine neue DB-

Instance wiederherstellen und dann eine Versions-Upgrade für die neue DB-Instance durchführen. Sie können dann sicher mit dem aktualisierten Klon Ihrer DB-Instance experimentieren, bevor Sie entscheiden, Ihre originale DB-Instance zu aktualisieren.

MySQL-Nebenversionen auf Amazon RDS

Die Änderungen, die die MySQL-Community an den Nebenversionen vorgenommen hat, finden Sie unter Critical Patch Updates, Security Alerts and Bulletins auf der Oracle-Website. Wählen Sie unter Critical Patch Update den Monat aus, in dem Oracle die Nebenversion veröffentlicht hat. Wählen Sie dann unter Betroffene Produkte und Versionen die MySQL-Nebenversion aus.

Nebenversionen

- MySQL versie 8.4.6
- MySQL versie 8.4.5
- MySQL versie 8.4.4
- MySQL versie 8.0.43
- MySQL versie 8.0.42
- MySQL versie 8.0.41
- MySQL versie 8.0.40
- MySQL versie 8.0.39
- MySQL versie 8.0.37

MySQL versie 8.4.6

MySQL Version 8.4.6 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- `rds_security_audit`Datenbank-Plugin hinzugefügt, ein internes RDS für MySQL-Plugin, das sicherheitsrelevante Metriken sammelt. Es ist kein Eingreifen des Kunden erforderlich.

MySQL versie 8.4.5

MySQL Version 8.4.5 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren `tzdata2025b`.

MySQL versie 8.4.4

MySQL Version 8.4.4 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren `tzdata2025a`.
- Es wurde ein Fehler behoben, der zu einem Kollationsfehler bei der Ausführung der gespeicherten Amazon RDS-Prozeduren `mysql.rds_set_configuration` und `mysql.rds_kill` führte.

MySQL versie 8.0.43

MySQL Version 8.0.43 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

MySQL versie 8.0.42

MySQL Version 8.0.42 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren `tzdata2025b`.

MySQL versie 8.0.41

MySQL Version 8.0.41 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Die Zeitzoneninformationen wurden aktualisiert, sodass sie darauf basieren `tzdata2025a`.
- Es wurde ein Fehler behoben, der zu einem Kollationsfehler bei der Ausführung der gespeicherten Amazon RDS-Prozeduren `mysql.rds_set_configuration` und `mysql.rds_kill` führte.

MySQL versie 8.0.40

MySQL Version 8.0.40 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Es wurde ein Fehler behoben, der dazu führte, dass Zeichensätze bei Datenbank-Upgrades nicht übereinstimmten.

MySQL versie 8.0.39

MySQL Version 8.0.39 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Es wurde ein Fehler behoben, `sql_log_off` durch den das `SESSION_VARIABLES_ADMIN` Privileg nicht richtig funktionierte.
- Es wurde ein Fehler behoben, der verhinderte, dass der Hauptbenutzer das `SESSION_VARIABLE_ADMIN` Recht anderen Datenbankbenutzern gewähren konnte.
- Es wurde ein Fehler behoben, der bei der Ausführung von von RDS bereitgestellten gespeicherten Prozeduren zu einer illegalen Mischung von Kollatierungen führte.

MySQL versie 8.0.37

MySQL Version 8.0.37 ist jetzt auf Amazon RDS verfügbar. Diese Version enthält Korrekturen und Verbesserungen, die von der MySQL-Community und Amazon RDS hinzugefügt wurden.

Neue Funktionen und Verbesserungen

- Es wurde ein Fehler bei der Ausführung einer Instant-DDL-Anweisung, gefolgt von einem UPDATE, behoben, der zu einem Assertion-Fehler führte.

Unterstützte MySQL-Hauptversionen in Amazon RDS

Die Hauptversionen von RDS für MySQL stehen mindestens bis zum Ende des Lebenszyklus der Community für die entsprechende Community-Version unter Standard-Support zur Verfügung. Gegen eine Gebühr können Sie eine Hauptversion auch nach Ablauf des Standard-Supports von RDS weiter

ausführen. Weitere Informationen finden Sie unter Erweiterter Support für Amazon RDS mit Amazon RDS und Amazon RDS für MySQL – Preise.

Sie können die folgenden Daten verwenden, um Ihre Test- und Upgrade-Zyklen zu planen.

> **Note**
>
> Daten mit nur einem Monat und einem Jahr sind ungefähre Angaben und werden mit einem genauen Datum aktualisiert, wenn es bekannt ist.
> Mithilfe der AWS CLI oder der RDS-API können Sie sich auch Informationen zu den Support-Daten für wichtige Engine-Versionen anzeigen lassen. Weitere Informationen finden Sie unter Support-Daten für Engine-Versionen in Amazon RDS Extended Support anzeigen.

MySQL Hauptversion	Datum der Community-Veröffentlichung	Datum der Veröffentlichung von RDS	Datum des Lebensendes der Gemeinschaft	RDS-Ende des Standard-Supportdatums	RDS: Beginn des Extended Support, Jahr 1, Preisdatum	RDS: Beginn des Extended Support, Jahr 3, Preisdatum	RDS: Ende des Extended Support
MySQL 8.4	30. April 2024	21. November 2024	April 2029	31. Juli 2029	1. August 2029	1. August 2031	31. Juli 2032
MySQL 8.0	19. April 20	23. Oktober 2018	April 2026	31. Juli 2026	1. August 2026	1. August 2028	31. Juli 2029
MySQL 5.7*	21. Oktober 2015	22. Februar 2016	Oktober 20	29. Februar 2024	1. März 2024	1. März 2026	28. Februar 2027

* MySQL 5.7 ist jetzt nur noch unter RDS Extended Support verfügbar. Weitere Informationen finden Sie unter Erweiterter Support für Amazon RDS mit Amazon RDS.

Amazon RDS-Versionen mit erweitertem Support für RDS für MySQL

Der folgende Inhalt listet alle Versionen von RDS Extended Support for RDS for MySQL auf.

Versionen

- Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20250508
- Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20250213
- Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20250103
- Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20240808
- Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20240529
- Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20240408

Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20250508

RDS Extended Support für RDS for MySQL Version 5.7.44-RDS.20250508 ist verfügbar.

Behobene Fehler:

- Es wurde behoben, dass der virtuelle Index nach einem Rollback instabil war, wenn er größer als max `index_id` `uint32` ist.
- Behoben: Tests schlagen aufgrund eines Speicherproblems fehl.
- Behoben `<COMMAND_CLASS>` ist leer für`<NAME>Execute</NAME>`.
- Fehler beim Kompilieren von MySQL mit GCC 14 [noclose 5.7] behoben.

CVEs behoben:

- CVE-2025-30682
- CVE-2025-30687
- CVE-2025-30688
- CVE-2025-21581
- CVE-2025-21585
- CVE-2025-30689
- CVE-2025-30722
- CVE-2025-21577

- CVE-2025-30693
- CVE-2025-30695
- CVE-2025-30703

Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20250213

RDS Extended Support für RDS for MySQL Version 5.7.44-RDS.20250213 ist verfügbar.

Behobene Fehler:

- Fehlgeschlagene `result != FTS_INVALID` InnoDB-Assertion behoben.
- Der Absturz und die weit verbreitete Beschädigung von räumlichen Indizes, nachdem der `ALTER TABLE` Vorgang die InnoDB-Tabelle mithilfe des Algorithmus neu erstellt hat, wurde behoben. `INPLACE`
- Das Problem wurde behoben, da die `ON DELETE CASCADE` generierte Spalte in abstürzt. `innobase_get_computed_value`
- Ein Assertionsfehler in `row_MySQL_pad_col` wurde behoben.
- Es wurde ein Problem behoben, bei dem Online-DDL den folgenden Fehler verursacht:. `ERROR 1712 (HY000): Index PRIMARY is corrupted`
- Abstürze bei `Item_rollup_sum_switcher::current_arg` wurden behoben.
- Es wurde ein Problem behoben, bei dem der Datenbank-Cache nicht geleert wurde. `DROP USER`
- Ein Pufferüberlauf wurde behoben. `my_print_help`
- Es wurde ein InnoDB-Problem behoben, bei dem der `FULLTEXT` Index `FTS_DOC_ID` auf den maximalen 32-Bit-Wert ohne Vorzeichen beschränkt war.

CVEs behoben:

- CVE-2025-21497
- CVE-2025-21555
- CVE-2025-21559
- CVE-2025-21490
- CVE-2025-21491
- CVE-2025-21500
- CVE-2025-21501

- CVE-2025-21540
- CVE-2025-21543
- CVE-2025-21520

Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20250103

RDS Extended Support für RDS for MySQL Version 5.7.44-RDS.20250103 ist verfügbar.

Behobene Fehler:

- Das FTS-Bereinigungsproblem beim Löschen und Hinzufügen eines `FULLTEXT` Indexes in derselben Transaktion wurde behoben.
- Das Timing der Speicherzuweisung im MySQL-Client wurde optimiert, um mögliche Datenlecks zu verhindern.
- Die Kürzung der Ergebnisse bei Verwendung des `UNION` Operators auf 34 Byte wurde behoben.
- out-of-boundsDer potenzielle Zugriff aufgrund `ulong bitmask` des Autorisierungscodes wurde behoben.

CVEs behoben:

- CVE-2024-21230
- CVE-2024-21201
- CVE-2024-21241
- CVE-2024-21203

Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20240808

RDS Extended Support für RDS for MySQL Version 5.7.44-RDS.20240808 ist verfügbar.

Behobene Fehler:

- Ein Assertionsfehler im Zusammenhang mit dem Index der Wörterbuchspalte wurde behoben.
- Problem mit der `is_binlog_cache_empty()` Funktion behoben.
- `heap-use-after-free`Fehler in `sql/item.cc` Dateien behoben.
- Es wurden mehrere Probleme mit dem räumlichen Index behoben, indem sie für `index-only` Lesevorgänge deaktiviert wurden.

- Das Instrumentierungsproblem mit dem `LOCK_ORDER: CONNECTION_CONTROL` Plugin wurde behoben.
- Es wurde behoben, dass Threads mit dem `CONNECTION_CONTROL` Plugin hängen blieben.
- Es wurde behoben`PSI_THREAD_INFO`, dass für nicht aktualisiert wurde`PREPARED STATEMENTS`.
- Die doppelte Verarbeitung von FTS-Indexwörtern mit wurde behoben`innodb_optimize_fulltext_only`.

CVEs behoben:

- CVE-2024-21177

Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20240529

RDS Extended Support für RDS for MySQL Version 5.7.44-RDS.20240529 ist verfügbar.

Behobene Fehler:

- Ein `field.cc` Assertion-Fehler wurde durch die Implementierung `fix_after_pullout` behoben.
- Ein Nullzeigerfehler bei der Rückgabe von Metadaten an den Client für bestimmte SQL-Abfragen wurde behoben. Diese Abfragen enthielten dynamische Parameter und Unterabfragen in `SELECT` Klauseln.
- Es wurden falsche Ergebnisse bei der Verwendung `GROUP BY` für Scans mit losem Index oder für Scans von nicht zusammenhängenden Indexbereichen behoben.
- Der Verlust von GTID-Informationen beim Absturz von MySQL während der Persistenz wurde behoben.
- Es wurde eine Race-Bedingung behoben, die dazu führen konnte, dass eine InnoDB-Transaktion auf unbestimmte Zeit hängen blieb.
- Bei der Bereinigung von Zertifizierungsinformationen der Gruppenreplikation wurde ein Fehler behoben, bei dem es zu einem Wettlauf kam.
- Es wurde ein Problem beim Rückwärtsscannen des Index bei gleichzeitigen Seitenoperationen behoben.
- Ein Problem mit dem inkonsistenten Status der Volltextsuche (FTS) in gleichzeitigen Szenarien wurde behoben.
- Ein Assertion-Problem mit dem Änderungspuffer beim Löschen von Tabellen wurde behoben.

- Einheitliches Verhalten beim Aufrufen von `deinit` Funktionen für alle Plugin-Typen.

CVEs behoben:

- CVE-2024-20963
- CVE-2024-20993
- CVE-2024-20998
- CVE-2024-21009
- CVE-2024-21054
- CVE-2024-21055
- CVE-2024-21057
- CVE-2024-21062
- CVE-2024-21008
- CVE-2024-21013
- CVE-2024-21047
- CVE-2024-21087
- CVE-2024-21096

Erweiterte RDS-Unterstützung für RDS für MySQL Version 5.7.44-RDS.20240408

RDS Extended Support für RDS for MySQL Version 5.7.44-RDS.20240408 ist verfügbar.

Diese Version enthält Patches für Folgendes: CVEs

- CVE-2024-20963

Arbeiten mit der Datenbank-Vorschauumgebung

Im Juli 2023 hat Oracle ein neues Release-Modell für MySQL angekündigt. Dieses Modell umfasst zwei Arten von Releases: Innovations-Releases und LTS-Releases. Amazon RDS stellt MySQL-Innovations-Releases in der RDS-Vorschauumgebung zur Verfügung. Weitere Informationen zu den MySQL-Innovations-Releases finden Sie unter Introducing MySQL Innovation and Long-Term Support (LTS) versions.

DB-Instances von RDS für MySQL in der Datenbank-Vorschauumgebung funktionieren ähnlich wie andere DB-Instances von RDS für MySQL. Sie können die Datenbank-Vorschauumgebung jedoch nicht für Produktions-Workloads nutzen.

Für Vorschauumgebungen gelten folgende Einschränkungen:

- Amazon RDS löscht alle DB-Instances 60 Tage nach Erstellung zusammen mit allen Backups und Snapshots.
- Sie können nur Allzweck-SSD und bereitgestellte IOPS-SSD als Speicher verwenden.
- Support Bei DB-Instances erhalten Sie keine Hilfe. Stattdessen können Sie Ihre Fragen in der von uns AWS verwalteten Q&A-Community re:POST stellen.AWS
- Sie können einen Snapshot einer DB-Instance nicht in eine Produktionsumgebung kopieren.

Die folgenden Optionen werden von der Vorschauversion unterstützt.

- Sie können DB-Instances mithilfe von db.m6i-, db.r6i-, db.m6g-, db.m5-, db.t3-, db.r6g- und db.r5-DB-Instance-Klassen erstellen. Weitere Informationen zu RDS-Instance-Klassen erhalten Sie unter .
- Sie können Single-AZ- und Multi-AZ-Bereitstellungen verwenden.
- Sie können die standardmäßigen MySQL-Dump- und -Ladefunktionen verwenden, um Datenbanken aus der Datenbank-Vorschauumgebung zu exportieren oder in diese zu importieren.

Nicht in der Datenbank-Vorschauumgebung unterstützte Features

Die folgenden Features sind in der Datenbank-Vorschauumgebung nicht verfügbar:

- Regionsübergreifende Snapshot-Kopie
- Regionsübergreifende Lesereplikate
- RDS-Proxy

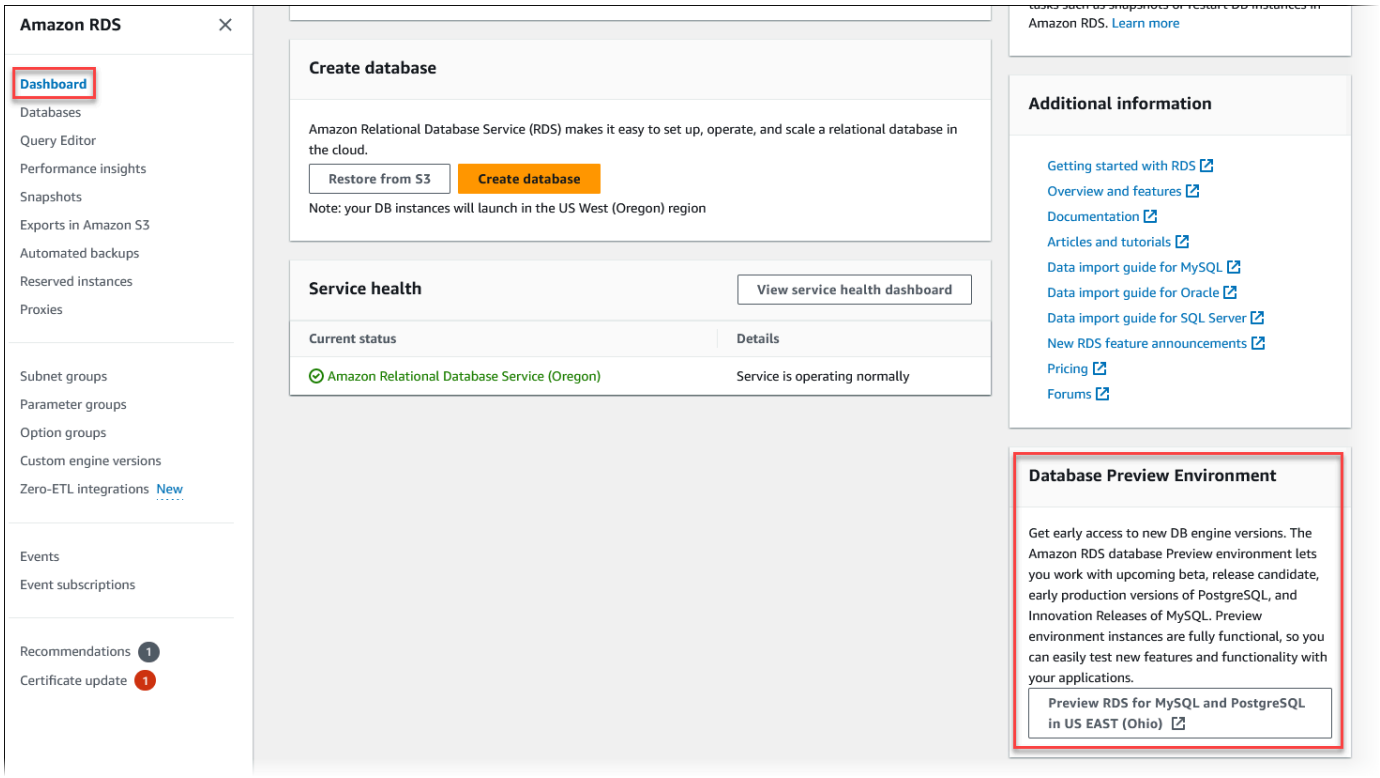
Erstellen einer neuen DB-Instance in der Datenbank-Vorschauumgebung

Sie können eine DB-Instance in der Database Preview-Umgebung mithilfe der AWS Management Console AWS CLI, oder RDS-API erstellen.

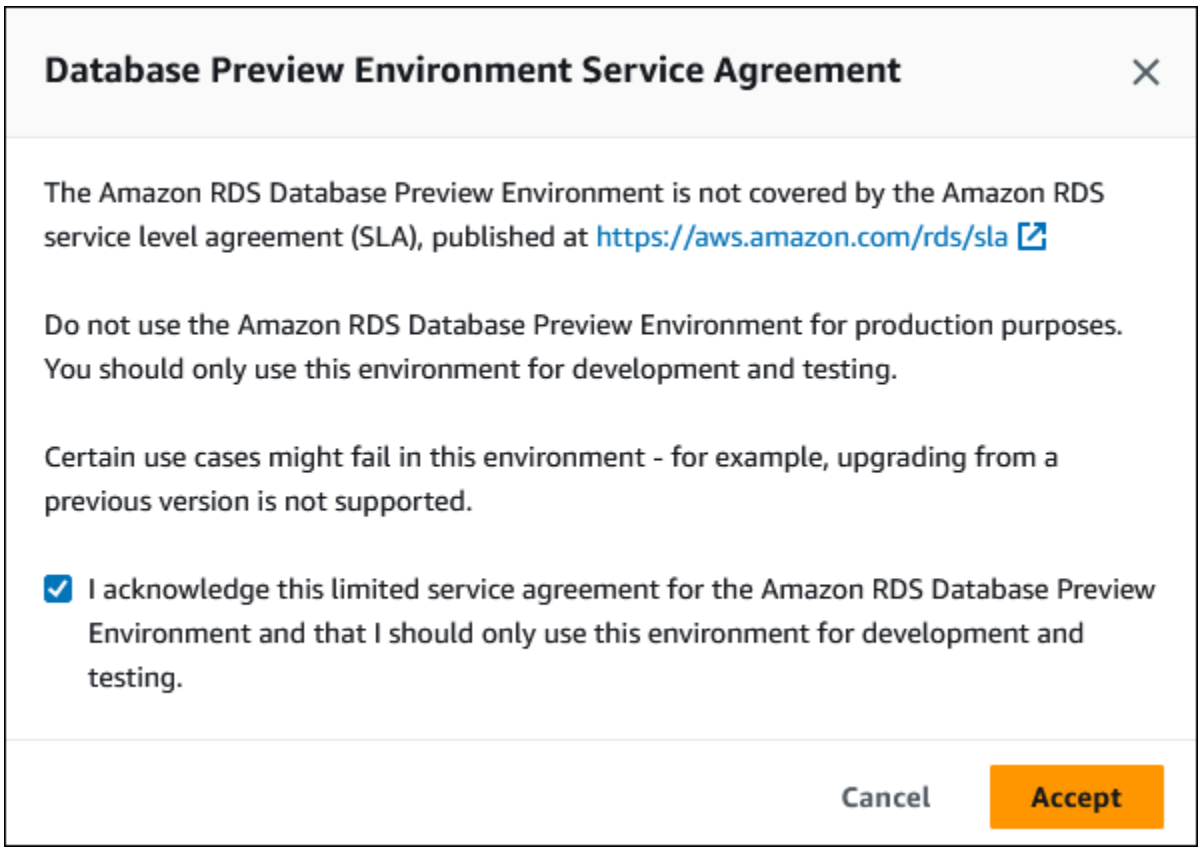
Konsole

So erstellen Sie eine DB-Instance in der Datenbank-Vorschauumgebung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Dashboard aus.
3. Suchen Sie auf der Seite Dashboard nach Datenbank-Preview-Umgebung, wie in der folgenden Abbildung gezeigt.



Sie können auch direkt zu Datenbank-Preview-Umgebung navigieren. Bevor Sie fortfahren können, müssen Sie die Einschränkungen bestätigen und akzeptieren.



4. Wenn Sie die DB-Instance von RDS für MySQL erstellen möchten, gehen Sie genauso vor wie bei der Erstellung einer DB-Instance von Amazon RDS. Weitere Informationen finden Sie im Verfahren Konsole unter Erstellen einer DB-Instance.

AWS CLI

Verwenden Sie den folgenden Endpunkt, um eine DB-Instance in der Datenbank-Vorschauumgebung über die AWS CLI zu erstellen.

```
rds-preview.us-east-2.amazonaws.com
```

Wenn Sie die DB-Instance von RDS für MySQL erstellen möchten, gehen Sie genauso vor wie bei der Erstellung einer DB-Instance von Amazon RDS. Weitere Informationen finden Sie im Verfahren AWS CLI unter Erstellen einer DB-Instance.

RDS-API

Verwenden Sie den folgenden Endpunkt, um eine DB-Instance in der Datenbank-Vorschauumgebung über die RDS-API zu erstellen.

```
rds-preview.us-east-2.amazonaws.com
```

Wenn Sie die DB-Instance von RDS für MySQL erstellen möchten, gehen Sie genauso vor wie bei der Erstellung einer DB-Instance von Amazon RDS. Weitere Informationen finden Sie im Verfahren RDS-API unter Erstellen einer DB-Instance.

MySQL Version 9.3 in der Database Preview-Umgebung

MySQL Version 9.3 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. MySQL Version 9.3 enthält mehrere Verbesserungen, die unter Änderungen in MySQL 9.3.0 beschrieben sind.

Weitere Informationen zur Database-Vorschauumgebung finden Sie unter the section called “Umgebung mit Datenbankvorschau”. Um von der Konsole aus auf die Vorschau-Umgebung zuzugreifen, wählen Sie https://console.aws.amazon.com/rds-preview/.

MySQL Version 9.2 in der Database Preview-Umgebung

MySQL Version 9.2 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. MySQL Version 9.2 enthält mehrere Verbesserungen, die unter Änderungen in MySQL 9.2.0 beschrieben sind.

Weitere Informationen zur Database-Vorschauumgebung finden Sie unter the section called “Umgebung mit Datenbankvorschau”. Um von der Konsole aus auf die Vorschau-Umgebung zuzugreifen, wählen Sie https://console.aws.amazon.com/rds-preview/.

MySQL Version 9.1 in der Database Preview-Umgebung

MySQL Version 9.1 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. MySQL Version 9.1 enthält mehrere Verbesserungen, die unter Änderungen in MySQL 9.1.0 beschrieben sind.

Weitere Informationen zur Database-Vorschauumgebung finden Sie unter the section called “Umgebung mit Datenbankvorschau”. Um von der Konsole aus auf die Vorschau-Umgebung zuzugreifen, wählen Sie https://console.aws.amazon.com/rds-preview/.

Veraltete Versionen für Amazon RDS for MySQL

Amazon RDS for MySQL Version 5.1, 5.5 und 5.6 sind veraltet.

Informationen zur Amazon RDS-Richtlinie für MySQL zur Deprecation finden Sie unter Amazon RDS. FAQs

Verbindung zu Ihrer MySQL-DB-Instance herstellen

Bevor Sie eine Verbindung zu einer DB-Instance auf einer MySQL-Datenbank-Engine herstellen können, müssen Sie eine DB-Instance erstellen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance. Nachdem Amazon RDS Ihre DB-Instance bereitgestellt hat, können Sie jede beliebige Standard-MySQL-Client-Anwendung und jedes -Hilfsprogramm verwenden, um sich mit der Instance zu verbinden. Geben Sie in der Verbindungszeichenfolge die DNS-Adresse aus dem primären DB-Instance-Endpunkt als Host-Parameter an sowie die Portnummer vom Instance-Endpunkt als Port-Parameter.

Um sich bei Ihrer RDS-DB-Instance zu authentifizieren, können Sie eine der Authentifizierungsmethoden für MySQL und AWS Identity and Access Management (IAM) - Datenbankauthentifizierung verwenden:

- Weitere Informationen über das Authentifizieren gegenüber MySQL mithilfe einer der Authentifizierungsmethoden für MySQL finden Sie unter Authentication Method in der MySQL-Dokumentation.
- Weitere Informationen über das Authentifizieren in MySQL mithilfe von IAM-Datenbank-Authentifizierung finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

Sie können sich mit einer MySQL-DB-Instance verbinden, indem Sie die MySQL-Befehlszeilenfunktion verwenden. Weitere Informationen zur Verwendung des MySQL-Clients finden Sie unter mysql – The MySQL Command-Line Client in der MySQL-Dokumentation. Eine GUI-basierte Anwendung, die Sie zum Herstellen einer Verbindung verwenden können, ist MySQL Workbench. Weitere Informationen finden Sie auf der Seite Download MySQL Workbench. Weitere Informationen zum Installieren von MySQL (einschließlich des MySQL-Clients) finden Sie unter Installation und Aktualisierung von MySQL.

Um eine Verbindung zu einer DB-Instance von außerhalb ihrer Amazon VPC herzustellen, muss die DB-Instance öffentlich zugänglich sein und der Zugriff muss unter Anwendung der Regeln für den eingehenden Datenverkehr der Sicherheitsgruppe der DB-Instance gewährt werden. Darüber hinaus müssen weitere Anforderungen erfüllt werden. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Sie können Secure Socket Layer (SSL) oder Transport Layer Security (TLS) für die Verschlüsselung von Verbindungen mit einer MySQL-DB-Instance verwenden. Weitere Informationen finden Sie

unter SSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS. Wenn Sie die AWS Identity and Access Management (IAM-) Datenbankauthentifizierung verwenden, stellen Sie sicher, dass Sie eine SSL/TLS-Verbindung verwenden. Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

Sie können auch eine Verbindung zu einer DB-Instance von einem Webserver herstellen. Weitere Informationen finden Sie unter Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance.

> **Note**
>
> Weitere Informationen zum Herstellen einer Verbindung mit einer MariaDB-DB-Instance finden Sie unter Verbindung zu Ihrer MariaDB-DB-Instance herstellen.

Informationen zum Suchen und Herstellen einer Verbindung zu einer RDS for MySQL-DB-Instance finden Sie in den folgenden Themen.

Themen

- Suchen der Verbindungsinformationen für eine RDS for My SQL DB-Instance
- Installation des SQL My-Befehlszeilenclients
- Verbindung über den SQL My-Befehlszeilenclient herstellen (unverschlüsselt)
- Von My SQL Workbench aus verbinden
- Herstellen einer Verbindung zu RDS for My SQL AWS JDBC with the Driver, AWS Python Driver und AWS ODBC Driver for My SQL
- Problembehandlung bei Verbindungen zu Ihrer My SQL DB-Instance

Suchen der Verbindungsinformationen für eine RDS for My SQL DB-Instance

Die Verbindungsinformationen für eine DB-Instance umfassen ihren Endpunkt, ihren Port und einen gültigen Datenbankbenutzer, z. B. den Masterbenutzer. Nehmen wir zum Beispiel an, dass ein Endpunktwert laute `mydb.123456789012.us-east-1.rds.amazonaws.com`. In diesem Fall ist `3306` der Port-Wert und der Datenbankbenutzer ist `admin`. Angesichts dieser Informationen geben Sie die folgenden Werte in einer Verbindungszeichenfolge an:

- Geben Sie für Host- oder DNS Hostname oder -namen an`mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Als Port `3306`.
- Geben Sie für Benutzer a `admin`.

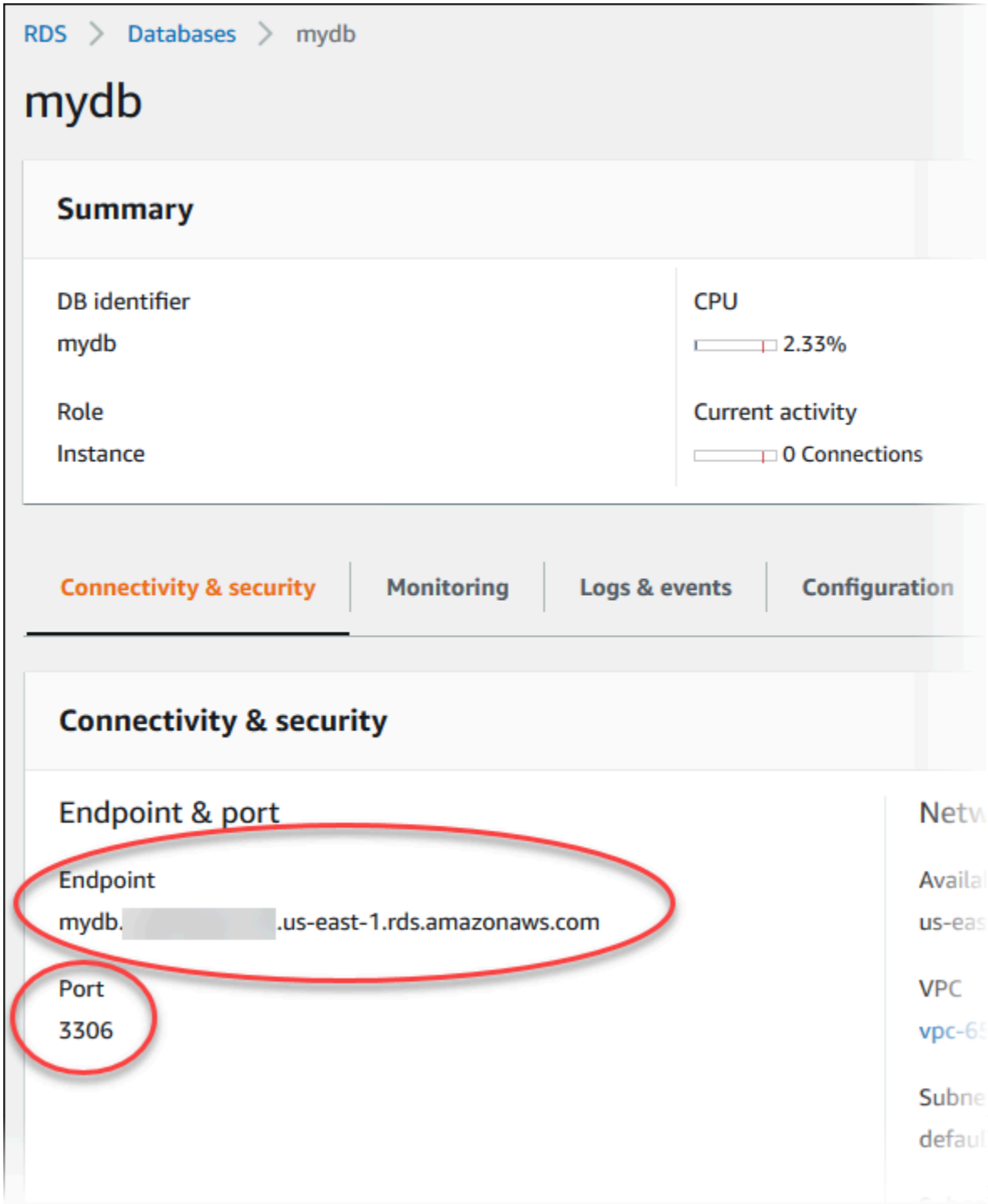
Um eine Verbindung zu einer DB-Instance herzustellen, verwenden Sie einen beliebigen Client für die My SQL DB-Engine. Sie können beispielsweise den SQL Befehlszeilenclient My oder My SQL Workbench verwenden.

Um die Verbindungsinformationen für eine DB-Instance zu finden, können Sie den AWS Management Console, den AWS CLI describe-db-instancesBefehl oder die Amazon RDS API escribeDBInstancesD-Operation verwenden, um deren Details aufzulisten.

Konsole

Um die Verbindungsinformationen für eine DB-Instance zu finden, finden Sie im AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Klicken Sie im Navigationsbereich auf Datenbanken, um eine Liste Ihrer DB-Instances anzuzeigen.
3. Wählen Sie den Namen der My SQL DB-Instance, um deren Details anzuzeigen.
4. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



5. Wenn Sie den Masterbenutzernamen finden müssen, wählen Sie die Registerkarte Konfiguration und den Wert für den Masterbenutzernamen an.

AWS CLI

Um die Verbindungsinformationen für eine My SQL DB-Instance mithilfe von zu finden AWS CLI, führen Sie den describe-db-instancesBefehl aus. Fragen Sie beim Aufruf die DB-Instance-ID, den Endpunkt, den Port und den Masterbenutzernamen ab.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mysql" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mysql" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Die Ausgabe sollte in etwa wie folgt aussehen.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

RDS API

Rufen Sie die escribeDBInstances Operation D auf, um die Verbindungsinformationen für eine DB-Instance mithilfe von Amazon RDS API zu ermitteln. Suchen Sie in der Ausgabe die Werte für die Endpunktadresse, den Endpunktport und den Masterbenutzernamen.

Installation des SQL My-Befehlszeilenclients

Die meisten Linux-Distributionen enthalten den MariaDB-Client anstelle des Oracle My-Clients. SQL Führen Sie den folgenden Befehl aus, um den SQL Befehlszeilenclient My auf Amazon Linux 2023 zu installieren:

```
sudo dnf install mariadb105
```

Führen Sie den folgenden Befehl aus, um den SQL My-Befehlszeilenclient auf Amazon Linux 2 zu installieren:

```
sudo yum install mariadb
```

Führen Sie den folgenden Befehl aus, um den SQL My-Befehlszeilenclient auf den meisten DEB Linux-Distributionen zu installieren:

```
apt-get install mariadb-client
```

Führen Sie den folgenden Befehl aus, um die Version Ihres SQL My-Befehlszeilenclients zu überprüfen:

```
mysql --version
```

Führen Sie den folgenden Befehl aus, um die SQL Dokumentation von My für Ihre aktuelle Client-Version zu lesen:

```
man mysql
```

Verbindung über den SQL My-Befehlszeilenclient herstellen (unverschlüsselt)

> **Important**
>
> Verwenden Sie eine unverschlüsselte SQL My-Verbindung nur, wenn sich Client und Server im selben Netzwerk befinden VPC und das Netzwerk vertrauenswürdig ist. Weitere IInformationen zur Verwendung verschlüsselter Verbindungen finden Sie unter Herstellen

einer Verbindung zu Ihrer My SQL DB-Instance auf Amazon RDS SSL TLS mit/über den SQL My-Befehlszeilenclient (verschlüsselt).

Um über den SQL Befehlszeilenclient My eine Verbindung zu einer DB-Instance herzustellen, geben Sie an der Befehlszeile den folgenden Befehl ein. Ersetzen Sie für den Parameter -h den DNS Namen (Endpunkt) für Ihre DB-Instance. Ersetzen Sie den Parameter -P durch den Port Ihrer DB-Instance. Ersetzen Sie für den Parameter -u den Benutzernamen eines gültigen Datenbankbenutzers, z. B. des Masterbenutzers. Geben Sie bei Aufforderung das Passwort für den Masterbenutzer ein.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -
u mymasteruser -p
```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, sollte eine Ausgabe wie die folgende angezeigt werden.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Von My SQL Workbench aus verbinden

Um von My SQL Workbench aus eine Verbindung herzustellen

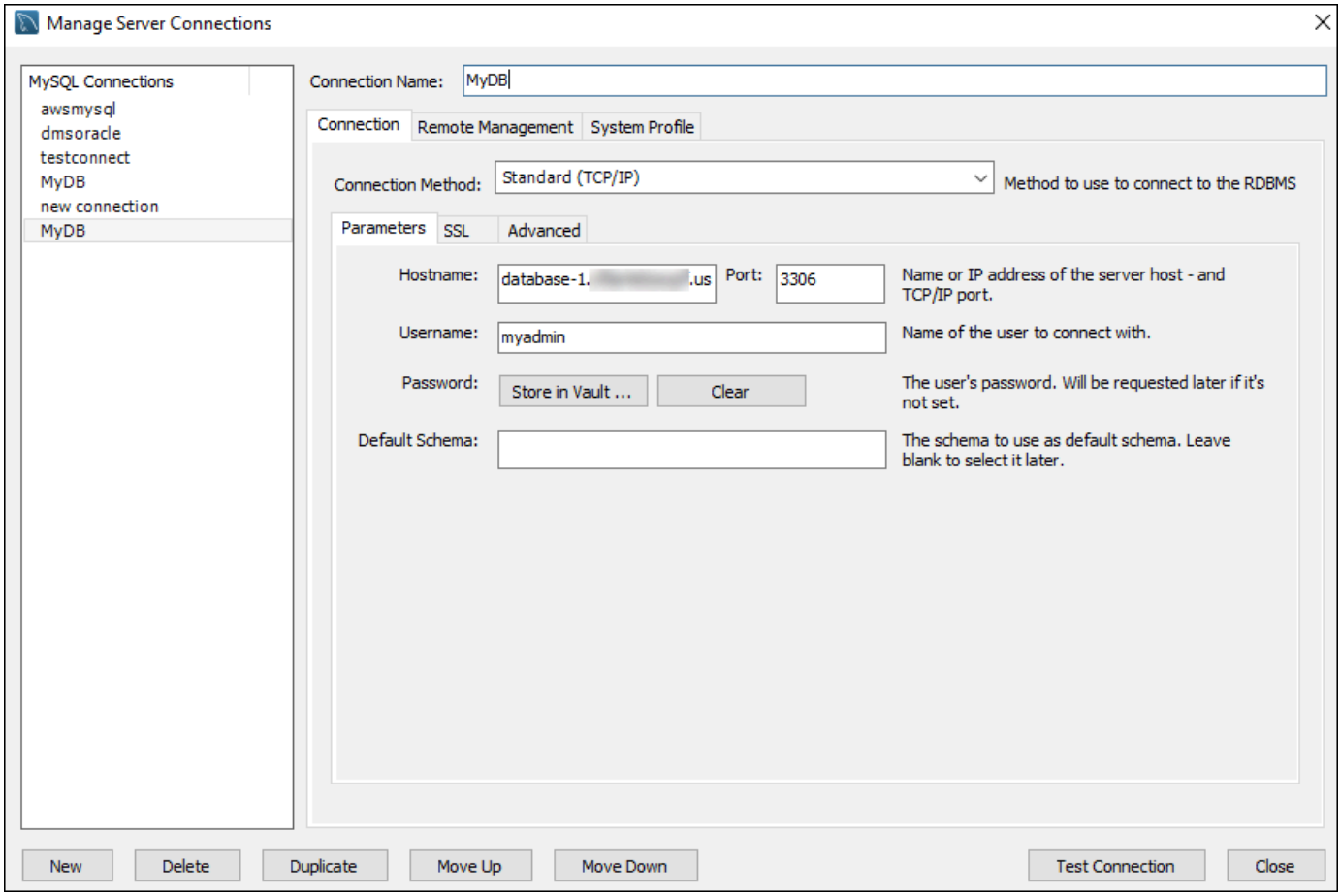
1. Laden Sie My Workbench unter Download My SQL Workbench herunter und installieren Sie es. SQL
2. Öffnen Sie My WorkbenchSQL.



3. Wählen Sie unter Database die Option Manage Connections aus.
4. Wählen Sie im Fenster Manage Server Connections New aus.
5. Geben Sie im Fenster Connect to Database die folgenden Informationen ein:

   - Stored Connection – Geben Sie einen Namen für die Verbindung ein, beispielsweise **MyDB**.
   - Hostname – Geben Sie den Endpunkt der DB-Instance ein.
   - Port – Geben Sie den von der DB-Instance verwendeten Port ein.
   - Benutzername – Geben Sie den Benutzernamen eines gültigen Datenbankbenutzers ein, beispielsweise den des Masterbenutzers.
   - Password – Wählen Sie optional Store in Vault (In Vault speichern) aus, geben Sie das Passwort des Benutzers ein und speichern Sie dieses.

   Das Fenster sieht in etwa wie folgt aus:



Sie können die Funktionen von My SQL Workbench verwenden, um Verbindungen anzupassen. Sie können die SSLRegisterkarte beispielsweise verwenden, um SSL TLS /-Verbindungen zu konfigurieren. Informationen zur Verwendung von My SQL Workbench finden Sie in der My SQL Workbench-Dokumentation. Verschlüsseln von Client-Verbindungen zu My SQL DB-Instances SSL mit/TLS, siehe. Verschlüsselung von Client-Verbindungen SSL TLS mit/zu My SQL DB-Instances auf Amazon RDS

6. Wählen Sie optional Test Connection aus, um zu prüfen, ob die Verbindung zur DB-Instance erfolgreich hergestellt wurde.
7. Klicken Sie auf Close (Schließen).
8. Wählen Sie unter Database die Option Connect to Database aus.
9. Wählen Sie für Stored Connection Ihre Verbindung aus.
10. Wählen Sie OK aus.

Herstellen einer Verbindung zu RDS for My SQL AWS JDBC with the Driver, AWS Python Driver und AWS ODBC Driver for My SQL

Stellen Sie RDS mit dem AWS JDBC Driver, dem AWS Python-Treiber und dem Driver for My eine Connect zu My SQL DB-Instances herSQL. AWS ODBC Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- SQLMit dem RDS Amazon Web Services (AWS) JDBC -Treiber eine Verbindung zu for My herstellen
- SQLMit dem RDS Python-Treiber von Amazon Web Services (AWS) eine Verbindung zu for My herstellen
- Herstellen einer Verbindung zu RDS for My SQL mit dem Amazon Web Services (AWS) ODBC -Treiber für My SQL

SQLMit dem RDS Amazon Web Services (AWS) JDBC -Treiber eine Verbindung zu for My herstellen

Der Amazon Web Services (AWS) JDBC -Treiber ist als erweiterter JDBC Wrapper konzipiert. Dieser Wrapper ergänzt einen vorhandenen JDBC Treiber und erweitert dessen Funktionalität. Der Treiber ist Drop-In-kompatibel mit dem Community-Treiber My SQL Connector/J und dem Community-Treiber MariaDB Connector/J.

Um den Treiber zu installieren, hängen Sie die AWS JDBC Treiber-.jar-Datei an (befindet sich in der Anwendung`CLASSPATH`) und behalten Sie die Verweise auf den jeweiligen AWS JDBC Community-Treiber bei. Aktualisieren Sie das jeweilige URL Verbindungspräfix wie folgt:

- `jdbc:mysql://` auf `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` auf `jdbc:aws-wrapper:mariadb://`

Weitere Informationen zum AWS JDBC Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im JDBC GitHub Treiber-Repository von Amazon Web Services (AWS).

SQLMit dem RDS Python-Treiber von Amazon Web Services (AWS) eine Verbindung zu for My herstellen

Der Amazon Web Services (AWS) Python-Treiber ist als fortschrittlicher Python-Wrapper konzipiert. Dieser Wrapper ergänzt den Open-Source-Treiber Psycopg und erweitert dessen Funktionalität. Der AWS Python-Treiber unterstützt Python-Versionen 3.8 und höher. Sie können das `aws-advanced-python-wrapper` Paket zusammen mit den `psycopg` Open-Source-Paketen mit dem `pip` Befehl installieren.

Weitere Informationen zum AWS Python-Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im GitHub Python-Treiber-Repository von Amazon Web Services (AWS).

Herstellen einer Verbindung zu RDS for My SQL mit dem Amazon Web Services (AWS) ODBC -Treiber für My SQL

Der AWS ODBC Driver for My SQL ist ein Client-Treiber, der für die hohe Verfügbarkeit von RDS for My entwickelt wurdeSQL. Der Treiber kann zusammen mit dem My SQL ODBC Connector/-Treiber vorhanden sein und ist mit denselben Workflows kompatibel.

Weitere Informationen zum AWS ODBC Driver for My SQL und vollständige Anweisungen zu dessen Installation und Verwendung finden Sie im Amazon Web Services (AWS) ODBC Driver for My SQL GitHub Repository.

Problembehandlung bei Verbindungen zu Ihrer My SQL DB-Instance

Zwei häufige Ursachen für Verbindungsfehler mit einer neuen DB-Instance sind folgende:

- Die DB-Instance wurde mithilfe einer Sicherheitsgruppe erstellt, die keine Verbindungen von dem Gerät oder der EC2 Amazon-Instance autorisiert, auf dem die Anwendung oder das Hilfsprogramm Meine SQL Anwendung oder das Hilfsprogramm ausgeführt wird. Die DB-Instance muss über eine VPC Sicherheitsgruppe verfügen, die die Verbindungen autorisiert. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS.

  Sie können eine Regel für eingehenden Datenverkehr in der Sicherheitsgruppe hinzufügen oder ändern. Wählen Sie für Source (Quelle) die Option My IP (Meine IP) aus. Dies ermöglicht Zugriff auf die DB-Instance von der IP-Adresse, die in Ihrem Browser erkannt wird.
- Die DB-Instance wurde mithilfe des Standard-Port 3306 erstellt; die Firewall Ihres Unternehmens blockiert jedoch Verbindungen zu diesem Port von Geräten aus Ihrem Unternehmensnetzwerk. Erstellen Sie die Instance erneut mit einem andern Port, um diesen Fehler zu beheben.

Weitere Informationen zu Verbindungsproblemen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Sichern von MySQL-DB-Instance-Verbindungen

Sie können robuste Sicherheitsmaßnahmen implementieren, um MySQL-DB-Instances vor unbefugtem Zugriff und potenziellen Bedrohungen zu schützen. Sicherheitsgruppen, SSL/TLS-Verschlüsselung und IAM-Datenbankauthentifizierung arbeiten zusammen, um mehrere Ebenen der Verbindungssicherheit für Ihre MySQL-DB-Instances zu schaffen. Diese Sicherheitskontrollen helfen Ihnen dabei, Compliance-Anforderungen zu erfüllen, Datenschutzverletzungen zu verhindern und sichere Kommunikationskanäle zwischen Anwendungen und Datenbanken aufrechtzuerhalten. Sie können Ihre MySQL-DB-Instances schützen, indem Sie Daten während der Übertragung verschlüsseln, den Zugriff auf bestimmte IP-Bereiche einschränken und die Benutzerauthentifizierung über IAM-Rollen statt über Datenbankkennwörter verwalten.

Sicherheit für MySQL-DB-Instances wird auf drei Ebenen verwaltet:

- AWS Identity and Access Management steuert, wer Amazon RDS-Managementaktionen auf DB-Instances ausführen kann. Wenn Sie AWS mithilfe von IAM-Anmeldeinformationen eine Verbindung herstellen, muss Ihr IAM-Konto über IAM-Richtlinien verfügen, die die für die Durchführung von Amazon RDS-Verwaltungsvorgängen erforderlichen Berechtigungen gewähren. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.
- Wenn Sie eine DB-Instance erstellen, verwenden Sie eine VPC-Sicherheitsgruppe, um zu steuern, welche Geräte und EC2 Amazon-Instances Verbindungen zum Endpunkt und Port der DB-Instance öffnen können. Diese Verbindungen können mit Secure Socket Layer (SSL) und Transport Layer Security (TLS) hergestellt werden. Zusätzlich können Firewall-Regeln in Ihrem Unternehmen steuern, ob Geräte in Ihrem Unternehmen bestehende Verbindungen zur DB-Instance öffnen können.
- Um die Anmeldung und die Berechtigungen für eine MySQL-DB-Instance zu authentifizieren, können Sie einen der folgenden Ansätze oder eine Kombination davon wählen:
  - Sie können denselben Ansatz wie mit einer eigenständigen Instance in MySQL auswählen. Befehle wie `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` und `SET PASSWORD` funktionieren genau wie auf lokalen Datenbanken, so wie auch das direkte Ändern von Datenbank-Schema-Tabellen. Das direkte Ändern der Datenbankschematabellen ist jedoch keine bewährte Methode, und ab RDS for MySQL Version 8.0.36 wird es nicht unterstützt. Weitere Informationen finden Sie unter Access Control and Account Management in der MySQL-Dokumentation.
  - Sie können auch die IAM-Datenbank-Authentifizierung verwenden. Mit IAM-Datenbank-Authentifizierung, können Sie mithilfe eines IAM-Benutzers, einer IAM-Rolle oder eines Authentifizierungstokens Ihre DB-Instance bestätigen. Ein Authentifizierungstoken ist ein

eindeutiger Wert, der mithilfe des Signatur-Version 4-Signiervorgangs erstellt wird. Durch das Verwenden der IAM-Datenbank-Authentifizierung können Sie dieselben Anmeldeinformationen verwenden, um den Zugang zu Ihren AWS -Ressourcen und Ihrer Datenbank zu steuern. Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

- Eine weitere Option ist die Kerberos-Authentifizierung für RDS for MySQL. Die DB-Instance arbeitet mit AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD), um die Kerberos-Authentifizierung zu aktivieren. Wenn Benutzer sich mit einer MySQL-DB-Instance authentifizieren, die mit der vertrauenswürdigen Domäne verbunden ist, werden Authentifizierungsanfragen weitergeleitet. Weitergeleitete Anfragen werden in das Domänenverzeichnis geleitet, mit dem Sie sie erstellen. AWS Directory Service Weitere Informationen finden Sie unter KerberosAuthentifizierung für Amazon RDS for MySQL verwenden.

Wenn Sie eine Amazon RDS DB-Instance erstellen, hat der Master-Benutzer standardmäßig folgende Berechtigungen:

Engine-Version	Systemberechtigung	Datenbankrolle
RDS für MySQL Version 8.4.3 und höher	`GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES ,INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE , REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE ROLE, DROP ROLE, APPLICATION_PASSWORD_ADMIN , FLUSH_OPTIMIZER_COSTS , FLUSH_PRIVILEGES , FLUSH_STATUS , FLUSH_TABLES , FLUSH_USER_RESOURCES , ROLE_ADMIN , SENSITIVE_VARIABLES_OBSERVER , SESSION_VARIABLES_ADMIN , SET_ANY_DEFINER , SHOW_ROUTINE , XA_RECOVER_ADMIN`	`rds_superuser_role` Mehr über `rds_superuser_role` erfahren Sie unter Rollenbasiertes Privilegienmodell für for My RDS SQL.

Engine-Version	Systemberechtigung	Datenbankrolle
RDS für MySQL Version 8.0.36 und höher	`SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE ROLE, DROP ROLE, APPLICATION_PASSWORD_ADMIN , ROLE_ADMIN , SET_USER_ID , XA_RECOVER_ADMIN`	`rds_superuser_role` Mehr über `rds_superuser_role` erfahren Sie unter Rollenbasiertes Privilegienmodell für for My RDS SQL.
RDS für MySQL-Versionen unter 8.0.36	`SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, REPLICATION SLAVE`	Keine

> **Note**
>
> Obwohl es möglich ist, den Masterbenutzer in der DB-Instance zu löschen, wird dies nicht empfohlen. Um den Master-Benutzer neu zu erstellen, verwenden Sie den Vorgang Modify DBInstance RDS API oder führen Sie den modify-db-instance AWS CLI Befehl aus und geben Sie ein neues Masterbenutzerkennwort mit dem entsprechenden Parameter an. Wenn der Masterbenutzer nicht bereits in der Instance vorhanden ist, wird der Masterbenutzer mit dem angegebenen Passwort erstellt.

Um Verwaltungsdienste für jede DB-Instance bereitzustellen, wird der `rdsadmin`-Benutzer erstellt, wenn die DB-Instance erstellt wird. Der Versuch, das Passwort zu verwerfen, umzubenennen oder zu ändern, oder die Sonderrechte für das `rdsadmin`-Konto zu ändern, wird fehlschlagen.

Um die Verwaltung der DB-Instance zu erlauben, wurden die Befehle `kill` und `kill_query` beschränkt. Die Amazon RDS-Befehle `rds_kill` und `rds_kill_query` werden bereitgestellt, um Ihnen das Beenden von Benutzersitzungen oder Abfragen in DB-Instances zu ermöglichen.

Passwortvalidierung für RDS für My SQL

My SQL bietet das `validate_password` Plugin für verbesserte Sicherheit. Das Plugin erzwingt Kennwortrichtlinien mithilfe von Parametern in der DB-Parametergruppe für Ihre My SQL DB-Instance. Das Plugin wird für DB-Instances unterstützt, auf denen My SQL Version 5.7, 8.0 und 8.4 ausgeführt wird. Weitere Informationen zum `validate_password` Plugin finden Sie unter Das Password Validation Plugin in der SQL Dokumentation „Meine“.

Um das **`validate_password`** Plugin für eine My SQL DB-Instance zu aktivieren

1. Connect zu Ihrer My SQL DB-Instance her und führen Sie den folgenden Befehl aus.

   ```
   INSTALL PLUGIN validate_password SONAME 'validate_password.so';
   ```

2. Konfigurieren Sie die Parameter für das Plugin in der DB-Parametergruppe, die von der DB-Instance verwendet wird.

   Weitere Informationen zu den Parametern finden Sie in der SQL Dokumentation Meine Dokumentation unter Optionen und Variablen des Passwortvalidierungs-Plugins.

   Weitere Informationen zum Ändern von DB-Instance-Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

3. Starten Sie die DB-Instance neu.

Nachdem Sie das `validate_password` Plugin aktiviert haben, setzen Sie die vorhandenen Passwörter zurück, damit sie Ihren neuen Validierungsrichtlinien entsprechen.

Amazon validiert RDS keine Passwörter. Die My SQL DB-Instance führt eine Passwortvalidierung durch. Wenn Sie mit dem AWS Management Console, dem `modify-db-instance` AWS CLI Befehl oder der `ModifyDBInstance` RDS API Operation ein Benutzerkennwort festlegen, kann die Änderung erfolgreich sein, auch wenn das neue Passwort nicht Ihren Passwortrichtlinien entspricht. Ein neues Passwort wird jedoch nur dann in der My SQL DB-Instance festgelegt, wenn es den Passwortrichtlinien entspricht. In diesem Fall RDS zeichnet Amazon das folgende Ereignis auf.

```
   "RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has
failed.
```

Weitere Informationen zu RDS Amazon-Veranstaltungen finden Sie unterMit RDS Amazon-Event-Benachrichtigungen arbeiten.

Verschlüsselung von Client-Verbindungen SSL TLS mit/zu My SQL DB-Instances auf Amazon RDS

Secure Sockets Layer (SSL) ist ein Industriestandardprotokoll zur Sicherung von Netzwerkverbindungen zwischen Client und Server. Nach SSL Version 3.0 wurde der Name in Transport Layer Security (TLS) geändert. Amazon RDS unterstützt SSL TLS /Verschlüsselung für My SQL DB-Instances. MitSSL/TLSkönnen Sie eine Verbindung zwischen Ihrem Anwendungsclient und Ihrer My SQL DB-Instance verschlüsseln. SSL/TLSSupport ist in allen Bereichen AWS-Regionen für My SQL verfügbar.

Mit Amazon können Sie Daten während der Übertragung sichernRDS, indem Sie Client-Verbindungen zu My SQL DB-Instances mitSSL/verschlüsselnTLS, TLS für alle Verbindungen zu einer My SQL DB-InstanceSSL/verlangen und vom SQL My-Befehlszeilenclient aus eine Verbindung mitSSL/TLS(verschlüsselt) herstellen. Die folgenden Abschnitte enthalten Anleitungen zur Konfiguration und SSL TLS Nutzung/Verschlüsselung für My SQL DB-Instances auf AmazonRDS.

Themen

- SSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS
- SSL//TLSfür bestimmte Benutzerkonten für eine My SQL DB-Instance bei Amazon erforderlich RDS
- SSL//TLSfür alle Verbindungen zu einer My SQL DB-Instance bei Amazon erforderlich RDS
- Herstellen einer Verbindung zu Ihrer My SQL DB-Instance auf Amazon RDS SSL TLS mit/über den SQL My-Befehlszeilenclient (verschlüsselt)

SSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS

Amazon RDS erstellt ein SSL/TLS certificate and installs the certificate on the DB instance when Amazon RDS provisions the instance. These certificates are signed by a certificate authority. The

SSL/TLS certificate includes the DB instance endpoint as the Common Name (CN) for the SSL/TLS Zertifikat zum Schutz vor Spoofing-Angriffen.

Ein von Amazon erstelltesSSL/TLS-Zertifikat RDS ist die vertrauenswürdige Root-Entität und sollte in den meisten Fällen funktionieren, kann aber fehlschlagen, wenn Ihre Anwendung keine Zertifikatsketten akzeptiert. Wenn Ihre Anwendung keine Zertifikatsketten akzeptiert, versuchen Sie, ein Zwischenzertifikat zu verwenden, um eine Verbindung zu Ihrem herzustellen AWS-Region. Sie müssen beispielsweise ein Zwischenzertifikat verwenden, um eine Verbindung zu den AWS GovCloud (US) Regionen mitSSL/herzustellenTLS.

Informationen zum Herunterladen von Zertifikaten finden Sie unter . Weitere Hinweise zur Verwendung vonSSL/TLSmit My SQL finden Sie unterAktualisierung von Anwendungen für die Verbindung zu My SQL DB-Instances mithilfe neuer SSL TLS /-Zertifikate.

Für My SQL Version 8.0 und niedriger SQL verwendet Amazon RDS for My Open SSL für sichere Verbindungen. Für My SQL Version 8.4 und höher SQL verwendet Amazon RDS for My AWS-LC. TLSDie Unterstützung hängt von der SQL My-Version ab. Die folgende Tabelle zeigt die TLS Unterstützung für Meine SQL Versionen.

Meine SQL Version	TLS 1.0	TLS1.1	TLS1.2	TLS1.3
Mein SQL 8.4	Nicht unterstützt	Nicht unterstützt	Unterstützt	Unterstützt
Mein SQL 8.0	Nicht unterstützt	Nicht unterstützt	Unterstützt	Unterstützt
Mein SQL 5.7	Unterstützt	Unterstützt	Unterstützt	Nicht unterstützt

SSL//TLSfür bestimmte Benutzerkonten für eine My SQL DB-Instance bei Amazon erforderlich RDS

Sie können für bestimmte Benutzerkontoverbindungen zu Ihren My SQL DB-Instances bei Amazon eine SSL TLS /Verschlüsselung verlangenRDS. Der Schutz vertraulicher Informationen vor unberechtigtem Zugriff oder Abfangen ist entscheidend für die Durchsetzung von Sicherheitsrichtlinien, wenn es um die Vertraulichkeit von Daten geht.

UmSSL/TLS-Verbindungen für bestimmte Benutzerkonten vorzuschreiben, verwenden Sie je nach Ihrer SQL My-Version eine der folgenden Anweisungen, umSSL/TLS-Verbindungen für das Benutzerkonto `encrypted_user` vorzuschreiben.

Verwenden Sie dazu die folgende Anweisung.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Weitere Informationen zu SSL TLS /-Verbindungen mit My SQL finden Sie unter Verschlüsselte Verbindungen verwenden in der SQL Dokumentation Meine.

SSL//TLSfür alle Verbindungen zu einer My SQL DB-Instance bei Amazon erforderlich RDS

Verwenden Sie den `require_secure_transport` Parameter, um zu verlangen, dass alle Benutzerverbindungen zu Ihrer My SQL DB-InstanceSSL/verwendenTLS. Standardmäßig ist der `require_secure_transport`-Parameter auf `OFF` festgelegt. Sie können den `require_secure_transport` Parameter auf `ON` requireSSL/TLSfür Verbindungen zu Ihrer DB-Instance setzen.

Sie können den Parameterwert `require_secure_transport` festlegen, indem Sie die DB-Parametergruppe für Ihre DB-Instance aktualisieren. Sie müssen Ihre DB-Instance nicht neu starten, damit die Änderung wirksam wird.

Wenn der `require_secure_transport`-Parameter auf `ON` für eine DB-Instance festgelegt ist, kann ein Datenbank-Client eine Verbindung zu ihr herstellen, wenn er eine verschlüsselte Verbindung aufbauen kann. Andernfalls wird eine Fehlermeldung ähnlich der folgenden an den Client zurückgegeben:

```
MySQL Error 3159 (HY000): Connections using insecure transport are prohibited while --
require_secure_transport=ON.
```

Weitere Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Weitere Informationen zu dem `require_secure_transport` Parameter finden Sie in der SQLDokumentation Meine Daten.

Herstellen einer Verbindung zu Ihrer My SQL DB-Instance auf Amazon RDS SSL TLS mit/über den SQL My-Befehlszeilenclient (verschlüsselt)

Die Parameter des `mysql` Client-Programms unterscheiden sich geringfügig, je nachdem, welche Version von My SQL oder MariaDB Sie verwenden.

Um herauszufinden, welche Version Sie haben, führen Siemysql-Befehl mit --version-Option aus. Im folgenden Beispiel zeigt die Ausgabe, dass das Client-Programm von MariaDB stammt.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

Die meisten Linux-Distributionen wie Amazon Linux, CentOS und Debian haben My durch MariaDB SQL ersetzt, und die darin enthaltene mysql Version stammt von MariaDB. SUSE

Gehen Sie wie folgt vor, um mit/eine Verbindung zu Ihrer DB-Instance herzustellenSSL: TLS

So stellen Sie mit dem SQL Befehlszeilenclient My eine Verbindung zu einer DB-Instance TLS mitSSL/her

1. Laden Sie ein Stammzertifikat herunter, das für alle funktioniert. AWS-Regionen

   Informationen zum Herunterladen von Zertifikaten finden Sie unter .
2. Verwenden Sie einen SQL My-Befehlszeilenclient, um eine Verbindung zu einer DB-Instance mit SSL TLS /-Verschlüsselung herzustellen. Ersetzen Sie für den -h Parameter den DNS Namen (Endpunkt) für Ihre DB-Instance. Ersetzen Sie für den --ssl-ca Parameter den Namen der TLS ZertifikatsdateiSSL//. Ersetzen Sie für den -P-Parameter den Port für Ihre DB-Instance. Ersetzen Sie für den -u-Parameter den Benutzernamen eines gültigen Datenbankbenutzers, z. B. des Masterbenutzers. Geben Sie bei Aufforderung das Passwort für den Masterbenutzer ein.

   Das folgende Beispiel zeigt, wie der Client mithilfe des --ssl-ca Parameters mit dem My SQL 5.7-Client oder höher gestartet wird:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
   ```

   Geben Sie den folgenden Befehl ein, um das SSL/TLS connection verifies the DB instance endpoint against the endpoint in the SSL/TLS Zertifikat anzufordern:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=VERIFY_IDENTITY -P 3306 -u myadmin -p
   ```

   Im folgenden Beispiel sehen Sie für MariaDB, wie der Client mit dem Parameter --ssl-ca gestartet wird.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
```

3. Geben Sie bei Aufforderung das Passwort für den Masterbenutzer ein.

Die Ausgabe entspricht weitgehend der Folgenden.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Aktualisierung von Anwendungen für die Verbindung zu My SQL DB-Instances mithilfe neuer SSL TLS /-Zertifikate

Seit dem 13. Januar 2023 RDS hat Amazon neue Certificate Authority (CA) -Zertifikate für die Verbindung zu Ihren RDS DB-Instances über Secure Socket Layer oder Transport Layer Security (SSL/TLS) veröffentlicht. Im Folgenden finden Sie Informationen dazu, wie Sie Ihre Anwendungen aktualisieren, um die neuen Zertifikate verwenden zu können.

Anhand dieses Themas können Sie feststellen, ob Client-AnwendungenSSL/verwenden, TLS um eine Verbindung zu Ihren DB-Instances herzustellen. Wenn dies der Fall ist, können Sie weiter überprüfen, ob diese Anwendungen zur Herstellung von Verbindungen Zertifikatverifizierungen erfordern.

> **Note**
>
> Einige Anwendungen sind so konfiguriert, dass sie nur dann eine Verbindung zu My SQL DB-Instances herstellen, wenn sie das Zertifikat auf dem Server erfolgreich verifizieren können. Für solche Anwendungen müssen Sie die Trust Stores Ihrer Client-Anwendung aktualisieren, damit diese die neuen CA-Zertifikate enthalten.
> Sie können die folgenden SSL Modi angeben: `disabled`, `preferred`, und`required`. Wenn Sie den `preferred` SSL Modus verwenden und das CA-Zertifikat nicht existiert oder nicht aktuell ist, wird die Verbindung wieder nicht verwendet SSL und die Verbindung wird ohne Verschlüsselung hergestellt.

Wir empfehlenden `preferred`-Modus. Wenn die Verbindung im `preferred`-Modus auf ein ungültiges Zertifikat stößt, wird die Verschlüsselung beendet und unverschlüsselt fortgesetzt.

Nach der Aktualisierung der CA-Zertifikate in den Trust Stores Ihrer Client-Anwendung können Sie die Zertifikate auf Ihren DB-Instances rotieren. Es wird nachdrücklich empfohlen, diese Verfahren vor der Implementierung in Produktionsumgebungen in einer Entwicklungs- oder Testumgebung zu testen.

Weitere Informationen zur Zertifikatrotation finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate. Weitere Informationen zum Herunterladen von Zertifikaten finden Sie unter . Hinweise zur Verwendung vonSSL/TLSmit My SQL DB-Instances finden Sie unterSSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS.

Themen

- Feststellen, ob Anwendungen eine Verbindung zu Ihrer My SQL DB-Instance herstellen SSL
- Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert
- Aktualisieren des Trust Stores Ihrer Anwendung
- Beispiel für Java-Code zum Herstellen von SSL Verbindungen

Feststellen, ob Anwendungen eine Verbindung zu Ihrer My SQL DB-Instance herstellen SSL

Wenn Sie Amazon RDS for My SQL Version 5.7, 8.0 oder 8.4 verwenden und das Leistungsschema aktiviert ist, führen Sie die folgende Abfrage aus, um zu überprüfen, ob VerbindungenSSL/ verwendenTLS. Informationen zur Aktivierung des Leistungsschemas finden Sie in der SQL Dokumentation Meine Dokumentation unter Leistungsschema — Schnellstart.

```
mysql> SELECT id, user, host, connection_type
       FROM performance_schema.threads pst
       INNER JOIN information_schema.processlist isp
       ON pst.processlist_id = isp.id;
```

In dieser Beispielausgabe können Sie sowohl Ihre eigene Sitzung (`admin`) als auch eine Anwendung sehen, bei der Sie angemeldet `webapp1` sindSSL.

```
+----+-----------------+------------------+-----------------+
| id | user            | host             | connection_type |
+----+-----------------+------------------+-----------------+
|  8 | admin           | 10.0.4.249:42590 | SSL/TLS         |
|  4 | event_scheduler | localhost        | NULL            |
| 10 | webapp1         | 159.28.1.1:42189 | SSL/TLS         |
+----+-----------------+------------------+-----------------+
3 rows in set (0.00 sec)
```

Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert

Sie können überprüfen, ob für JDBC Clients und Meine SQL Clients eine Zertifikatsüberprüfung erforderlich ist, um eine Verbindung herzustellen.

JDBC

Das folgende Beispiel mit My SQL Connector/J 8.0 zeigt eine Möglichkeit, die JDBC Verbindungseigenschaften einer Anwendung zu überprüfen, um festzustellen, ob für erfolgreiche Verbindungen ein gültiges Zertifikat erforderlich ist. Weitere Informationen zu allen JDBC Verbindungsoptionen für My SQL finden Sie unter Konfigurationseigenschaften in der Dokumentation My. SQL

Wenn Sie My SQL Connector/J 8.0 verwenden, muss eine SSL Verbindung anhand des DB-Serverzertifikats überprüft werden, wenn Ihre Verbindungseigenschaften auf `VERIFY_CA` oder `sslMode` eingestellt sind`VERIFY_IDENTITY`, wie im folgenden Beispiel.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Wenn Sie entweder My SQL Java Connector v5.1.38 oder höher oder My SQL Java Connector v8.0.9 oder höher verwenden, um eine Verbindung zu Ihren Datenbanken

herzustellen, führen sie, auch wenn Sie Ihre Anwendungen nicht explizit für die Verwendung konfiguriert haben, eine teilweise Zertifikatsüberprüfung durch und es kann keine Verbindung hergestellt werdenSSL/TLS when connecting to your databases, these client drivers default to using SSL/TLS. In addition, when using SSL/TLS, wenn das Datenbankserverzertifikat abgelaufen ist.

Mein SQL

Die folgenden Beispiele mit My SQL Client zeigen zwei Möglichkeiten, die Option Meine SQL Verbindung eines Skripts dahingehend zu überprüfen, ob für erfolgreiche Verbindungen ein gültiges Zertifikat erforderlich ist. Weitere Informationen zu allen Verbindungsoptionen mit My SQL Client finden Sie in der Dokumentation My unter Clientseitige Konfiguration für verschlüsselte Verbindungen. SQL

Wenn Sie My SQL Client Version 5.7 und höher verwenden, muss eine SSL Verbindung anhand des CA-Zertifikats des Servers verifiziert werden, wenn Sie für die `--ssl-mode` Option `VERIFY_CA` oder angeben`VERIFY_IDENTITY`, wie im folgenden Beispiel.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Aktualisieren des Trust Stores Ihrer Anwendung

Informationen zur Aktualisierung des Vertrauensspeichers für Meine SQL Anwendungen finden Sie unter SSLZertifikate installieren in der SQL Dokumentation Meine Dateien.

Informationen zum Herunterladen des Stammverzeichnisses finden Sie unter .

Beispiele für Skripte, die Zertifikate importieren, finden Sie unter Beispielskript für den Import von Zertifikaten in Ihren Trust Store.

> **Note**
>
> Wenn Sie den Trust Store aktualisieren, können Sie ältere Zertifikate beibehalten und die neuen Zertifikate einfach hinzufügen.

Wenn Sie den JDBC MySQL-Treiber in einer Anwendung verwenden, legen Sie die folgenden Eigenschaften in der Anwendung fest.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Legen Sie während des Startens der Anwendung die folgenden Eigenschaften fest.

```
java -Djavax.net.ssl.trustStore=/path_to_trust_store/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_trust_store_password com.companyName.MyApplication
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Beispiel für Java-Code zum Herstellen von SSL Verbindungen

Das folgende Codebeispiel zeigt, wie Sie die SSL Verbindung einrichten, mit JDBC der das Serverzertifikat validiert wird.

```
public class MySQLSSLTest {

        private static final String DB_USER = "username";
        private static final String DB_PASSWORD = "password";
        // This trust store has only the prod root ca.
        private static final String TRUST_STORE_FILE_PATH = "file-path-to-trust-store";
        private static final String TRUST_STORE_PASS = "trust-store-password";

        public static void test(String[] args) throws Exception {
            Class.forName("com.mysql.jdbc.Driver");

            System.setProperty("javax.net.ssl.trustStore", TRUST_STORE_FILE_PATH);
            System.setProperty("javax.net.ssl.trustStorePassword", TRUST_STORE_PASS);
```

```
            Properties properties = new Properties();
            properties.setProperty("sslMode", "VERIFY_IDENTITY");
            properties.put("user", DB_USER);
            properties.put("password", DB_PASSWORD);


            Connection connection = null;
            Statement stmt = null;
            ResultSet rs = null;
            try {
                connection =
 DriverManager.getConnection("jdbc:mysql://mydatabase.123456789012.us-
east-1.rds.amazonaws.com:3306",properties);
                stmt = connection.createStatement();
                rs=stmt.executeQuery("SELECT 1 from dual");
            } finally {
                if (rs != null) {
                    try {
                        rs.close();
                    } catch (SQLException e) {
                    }
                }
                if (stmt != null) {
                   try {
                        stmt.close();
                    } catch (SQLException e) {
                   }
                }
                if (connection != null) {
                    try {
                        connection.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            }
            return;
        }
    }
```

> **Important**
>
> Nachdem Sie festgestellt haben, dass Ihre DatenbankverbindungenSSL/verwenden, TLS und Ihren Application Trust Store aktualisiert haben, können Sie Ihre Datenbank so aktualisieren, dass sie die rds-ca-rsa 2048-g1-Zertifikate verwendet. Anweisungen hierzu finden Sie in Schritt 3 unter Aktualisierung Ihres CA-Zertifikats durch Änderung Ihrer DB-Instance oder Ihres Clusters.
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

KerberosAuthentifizierung für Amazon RDS for MySQL verwenden

Sie können die Kerberos-Authentifizierung verwenden, um Benutzer zu authentifizieren, wenn sie sich mit Ihrer MySQL-DB-Instance verbinden. Die DB-Instance arbeitet mit AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD), um die Kerberos-Authentifizierung zu aktivieren. Wenn Benutzer sich mit einer MySQL-DB-Instance authentifizieren, die mit der vertrauenswürdigen Domäne verbunden ist, werden Authentifizierungsanfragen weitergeleitet. Weitergeleitete Anfragen werden in das Domänenverzeichnis geleitet, mit dem Sie sie erstellen. AWS Directory Service

Wenn Sie alle Ihre Anmeldeinformationen im selben Verzeichnis aufbewahren, können Sie Zeit und Mühe sparen. Mit diesem Ansatz haben Sie einen zentralen Ort für die Speicherung und Verwaltung von Anmeldedaten für mehrere DB-Instances. Die Verwendung eines Verzeichnisses kann auch Ihr allgemeines Sicherheitsprofil verbessern.

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Versions- und Regionsverfügbarkeit von Amazon RDS mit Kerberos-Authentifizierung finden Sie unter Unterstützte Regionen und DB-Engines für die Kerberos-Authentifizierung in Amazon RDS.

Übersicht über das Einrichten der Kerberos-Authentifizierung für MySQL-DB-Instances

Um die Kerberos-Authentifizierung für eine MySQL-DB-Instance einzurichten, führen Sie die folgenden allgemeinen Schritte aus, die später näher beschrieben werden:

1. Wird verwendet AWS Managed Microsoft AD , um ein AWS Managed Microsoft AD Verzeichnis zu erstellen. Sie können das AWS Management Console, das oder das verwenden AWS CLI,

AWS Directory Service um das Verzeichnis zu erstellen. Einzelheiten dazu finden Sie unter AWS Managed Microsoft AD Verzeichnis erstellen im AWS Directory Service Administratorhandbuch.

2. Erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle, die die verwaltete IAM-Richtlinie verwendet. `AmazonRDSDirectoryServiceAccess` Die Rolle erlaubt, dass Amazon RDS Aufrufe an Ihr Verzeichnis schickt.

   Damit die Rolle Zugriff gewährt, muss der Endpunkt AWS Security Token Service (AWS STS) im AWS-Region für Ihr AWS Konto aktiviert sein. AWS STS Endpunkte sind standardmäßig in allen aktiv AWS-Regionen, und Sie können sie ohne weitere Aktionen verwenden. Weitere Informationen finden Sie unter Aktivierung und Deaktivierung AWS STSAWS-Region im IAM-Benutzerhandbuch.

3. Erstellen und konfigurieren Sie Benutzer im AWS Managed Microsoft AD Verzeichnis mithilfe der Microsoft Active Directory-Tools. Weitere Informationen zum Erstellen von Benutzern in Ihrem Active Directory finden Sie unter Verwalten von Benutzern und Gruppen in AWS verwaltetem Microsoft AD im AWS Directory Service Administratorhandbuch.

4. Erstellen oder ändern Sie eine MySQL DB-Instance. Wenn Sie entweder die CLI oder die RDS-API für die Erstellungsanforderung verwenden, geben Sie eine Domänen-ID mit dem Parameter `Domain` an. Verwenden Sie die `d-*`-ID, die bei der Erstellung Ihres Verzeichnisses generiert wurde, und den Namen der von Ihnen erstellten Rolle.

   Wenn Sie eine vorhandene MySQL-DB-Instance so ändern, dass sie die Kerberos-Authentifizierung verwendet, legen Sie die Parameter für die Domäne und die IAM-Rolle für die DB-Instance fest. Suchen Sie die DB-Instance in derselben VPC wie das Domänenverzeichnis.

5. Verwenden Sie die Amazon RDS-Hauptbenutzer-Anmeldeinformationen, um sich mit der MySQL-DB-Instance zu verbinden. Erstellen Sie den Benutzer unter Verwendung der `CREATE USER`-Klausel `IDENTIFIED WITH 'auth_pam'` in MySQL. Benutzer, die Sie auf diese Weise anlegen, können sich mit der Kerberos-Authentifizierung an der MySQL-DB-Instance anmelden.

Einrichten der Kerberos-Authentifizierung für MySQL-DB-Instances

Sie verwenden AWS Managed Microsoft AD , um die Kerberos-Authentifizierung für eine MySQL-DB-Instance einzurichten. Um die Kerberos-Authentifizierung einzurichten, führen Sie die folgenden Schritte durch.

Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD

AWS Directory Service erstellt ein vollständig verwaltetes Active Directory in der AWS Cloud. Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service erstellt in Ihrem Namen zwei Domänencontroller und DNS-Server (Domain Name System). Die Verzeichnisserver werden in verschiedenen Subnetzen in einer VPC erstellt. Diese Redundanz trägt dazu bei, dass Ihr Verzeichnis auch im Fehlerfall erreichbar bleibt.

Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service führt er in Ihrem Namen die folgenden Aufgaben aus:

- Einrichten eines Active Directory innerhalb der VPC.
- Erstellt ein Verzeichnisadministratorkonto mit dem Benutzernamen Admin und dem angegebenen Passwort. Mit diesem Konto verwalten Sie das Verzeichnis.

  > **Note**
  >
  > Achten Sie darauf, dieses Passwort zu speichern. AWS Directory Service speichert es nicht. Sie können es zurücksetzen, aber Sie können es nicht abrufen.

- Erstellt eine Sicherheitsgruppe für die Verzeichniscontroller.

Wenn Sie eine starten AWS Managed Microsoft AD, AWS erstellt eine Organisationseinheit (OU), die alle Objekte Ihres Verzeichnisses enthält. Diese OU hat den NetBIOS-Namen, den Sie bei der Erstellung Ihres Verzeichnisses angegeben haben. Sie befindet sich im Domänenstamm. Der Domänenstamm gehört und wird von diesem verwaltet AWS.

Das Administratorkonto, das mit Ihrem AWS Managed Microsoft AD Verzeichnis erstellt wurde, verfügt über Berechtigungen für die gängigsten Verwaltungsaktivitäten Ihrer Organisationseinheit:

- Erstellen, Aktualisieren oder Löschen von Benutzern
- Hinzufügen von Ressourcen zu Ihrer Domäne, etwa Datei- oder Druckserver, und anschließendes Gewähren der zugehörigen Ressourcenberechtigungen für Benutzer in der OU
- Zusätzliche OUs Container erstellen
- Delegieren von Befugnissen
- Wiederherstellen von gelöschten Objekten aus dem Active Directory-Papierkorb
- Führen Sie AD- und PowerShell DNS-Windows-Module im Active Directory-Webdienst aus

Das Admin-Konto hat außerdem die Rechte zur Durchführung der folgenden domänenweiten Aktivitäten:

- Verwalten von DNS-Konfigurationen (Hinzufügen, Entfernen oder Aktualisieren von Datensätzen, Zonen und Weiterleitungen)
- Aufrufen von DNS-Ereignisprotokollen
- Anzeigen von Sicherheitsereignisprotokollen

Um ein Verzeichnis zu erstellen mit AWS Managed Microsoft AD

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Directory Service Konsole unter https://console.aws.amazon.com/directoryservicev2/.
2. Wählen Sie im Navigationsbereich Directories (Verzeichnisse) aus. Wählen Sie denn Set up Directory (Verzeichnis einrichten) aus.
3. Wählen Sie AWS Managed Microsoft AD. AWS Managed Microsoft AD ist die einzige Option, die Sie derzeit mit Amazon RDS verwenden können.
4. Geben Sie die folgenden Informationen ein:

   DNS-Name des Verzeichnisses

   Den vollständig qualifizierten Namen für das Verzeichnis, z. B. **`corp.example.com`**.

   NetBIOS-Name des Verzeichnisses

   Die kurzen Namen für das Verzeichnis, z. B. **`CORP`**.

   Verzeichnisbeschreibung

   (Optional) Eine Beschreibung für das Verzeichnis.

   Administratorpasswort

   Das Passwort für den Verzeichnisadministrator. Während des Verzeichniserstellungsprozesses wird ein Administratorkonto mit dem Benutzernamen Admin und diesem Passwort angelegt.

   Das Passwort für den Verzeichnisadministrator das nicht das Wort "admin" enthalten. Beachten Sie beim Passwort die Groß- und Kleinschreibung und es muss 8 bis 64 Zeichen lang sein. Zudem muss es mindestens ein Zeichen aus dreien der vier folgenden Kategorien enthalten:

- Kleinbuchstaben (a–z)
- Großbuchstaben (A–Z)
- Zahlen (0–9)
- Nicht-alphanumerische Zeichen (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Passwort bestätigen

Das Administratorpasswort, das erneut eingegeben wurde.

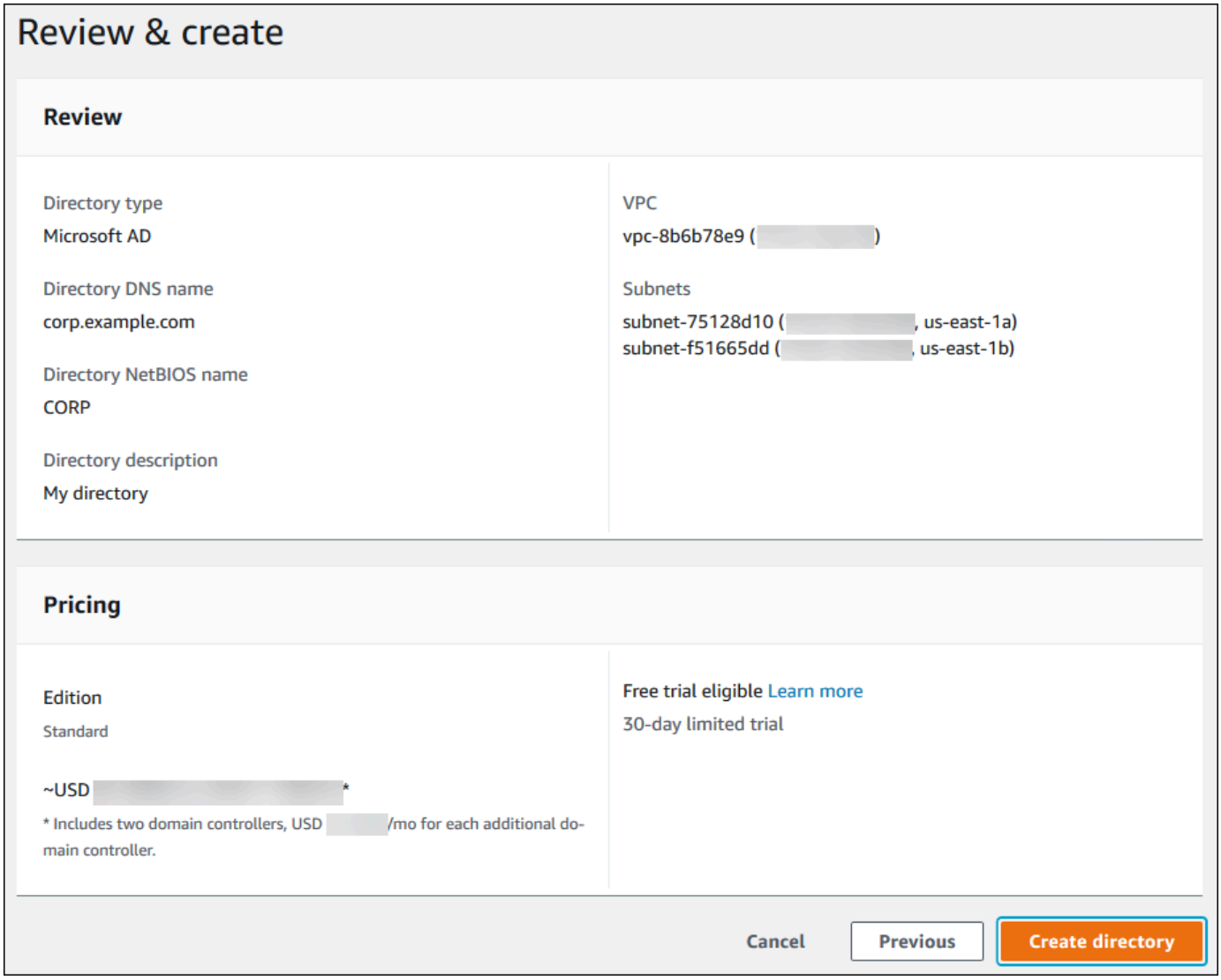
5. Wählen Sie Weiter aus.
6. Geben Sie die folgenden Informationen in den Abschnitt Networking ein. Wählen Sie dann Next (Weiter) aus:

   VPC

   Die VPC für das Verzeichnis. Erstellen Sie die MySQL-DB-Instance in derselben VPC.

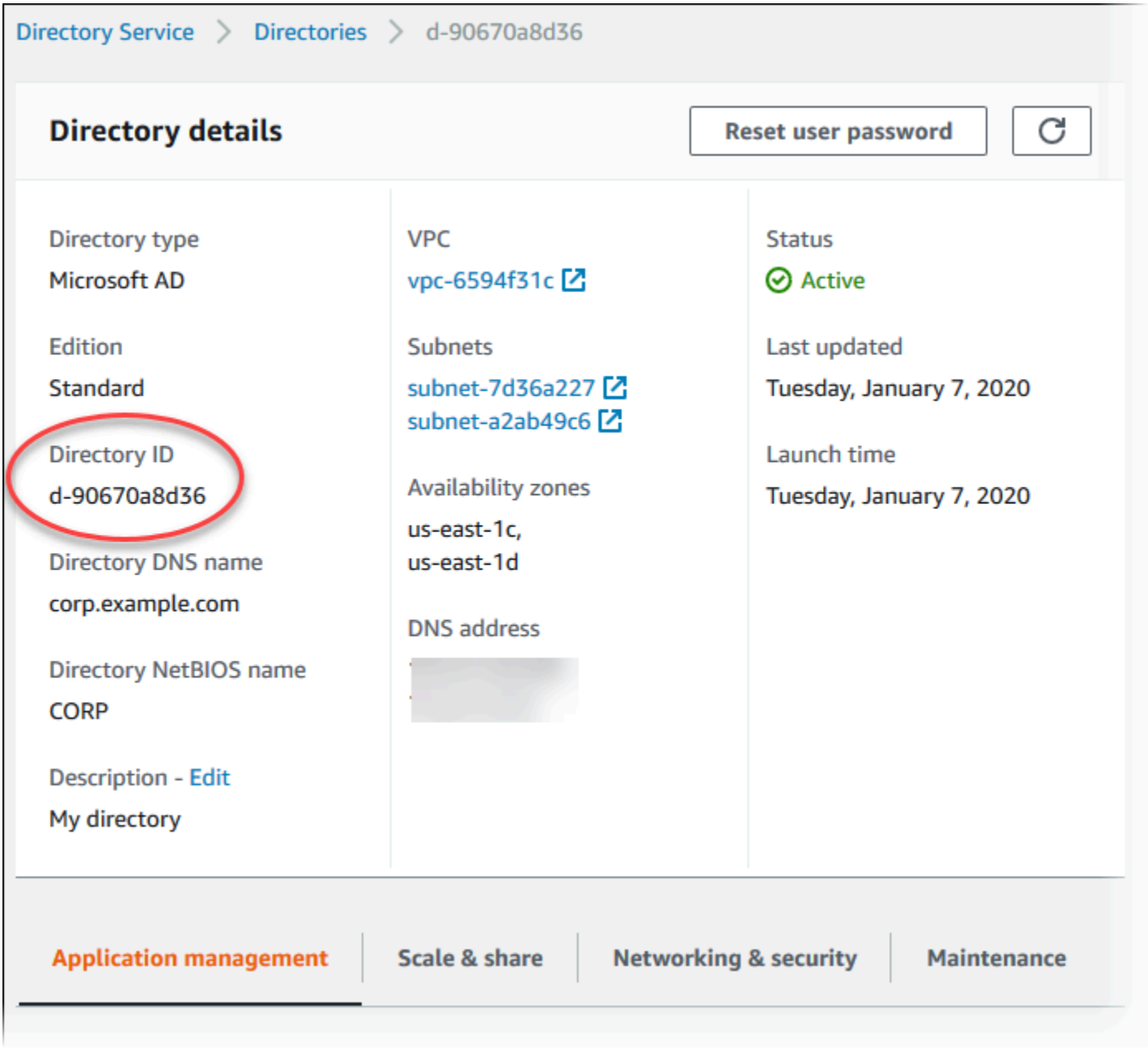
   Subnetze

   Subnetze für die Verzeichnisserver. Die beiden Subnetze müssen zu verschiedenen Availability-Zonen gehören.
7. Prüfen Sie die Verzeichnisinformationen und nehmen Sie ggf. Änderungen vor. Wenn die Informationen richtig sind, wählen Sie Create directory (Verzeichnis erstellen).



Es dauert einige Minuten, bis das Verzeichnis erstellt wurde. Wenn es erfolgreich erstellt wurde, ändert sich der Wert Status in Active (Aktiv).

Um Informationen über Ihr Verzeichnis anzuzeigen, wählen Sie den Verzeichnisnamen in der Verzeichnisliste aus. Beachten Sie den Wert Directory ID (Verzeichnis-ID). Sie benötigen diesen Wert, wenn Sie Ihre MySQL-DB-Instance erstellen oder ändern.



Schritt 2: Erstellen der IAM-Rolle für die Verwendung durch Amazon RDS

Damit Amazon RDS AWS Directory Service für Sie anrufen kann, ist eine IAM-Rolle erforderlich, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet. Diese Rolle ermöglicht es Amazon RDS, Aufrufe von AWS Directory Service durchzuführen.

Wenn eine DB-Instance mit dem erstellt wird AWS Management Console und der Konsolenbenutzer über die `iam:CreateRole` entsprechende Berechtigung verfügt, erstellt die Konsole diese Rolle automatisch. In diesem Fall lautet der Rollenname `rds-directoryservice-kerberos-access-role`. Andernfalls müssen Sie die IAM-Rolle manuell erstellen. Wenn Sie diese IAM-Rolle

erstellen`Directory Service`, wählen Sie die AWS verwaltete Richtlinie aus und hängen Sie sie `AmazonRDSDirectoryServiceAccess` an.

Weitere Informationen zum Erstellen von IAM-Rollen für einen Dienst finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

> **Note**
>
> Die für die Windows-Authentifizierung für RDS for SQL Server verwendete IAM-Rolle kann nicht für RDS for MySQL verwendet werden.

Optional können Sie Richtlinien mit den erforderlichen Berechtigungen erstellen, anstatt die verwaltete IAM-Richtlinie zu verwende `AmazonRDSDirectoryServiceAccess`. In diesem Fall muss die IAM-Rolle die folgende IAM-Vertrauensrichtlinie haben.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Die Rolle muss auch über die folgende IAM-Rollenrichtlinie verfügen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Schritt 3: Anlegen und konfigurieren von Benutzern

Sie können Benutzer mit dem Tool "Active Directory-Benutzer und -Computer" erstellen. Dieses Tool ist Teil der Tools Active Directory Domain Services und Active Directory Lightweight Directory Services. „Benutzer“ sind Einzelpersonen oder Entitäten, die Zugriff auf Ihr Verzeichnis haben.

Um Benutzer in einem AWS Directory Service Verzeichnis zu erstellen, müssen Sie mit einer EC2 Amazon-Instance verbunden sein, die auf Microsoft Windows basiert. Diese Instance muss Mitglied des AWS Directory Service Verzeichnisses sein und als Benutzer angemeldet sein, der über die Rechte zum Erstellen von Benutzern verfügt. Weitere Informationen finden Sie unter Verwalten von Benutzern und Gruppen AWS Managed Microsoft AD im AWS Directory-Service-Administrationshandbuch.

Schritt 4: Erstellen oder Ändern einer MySQL-DB-Instance

Erstellen oder ändern Sie eine MySQL-DB-Instance zur Verwendung mit Ihrem Verzeichnis. Sie können die Konsole, CLI oder RDS-API verwenden, um eine DB-Instance einem Verzeichnis zuzuordnen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Erstellen Sie eine neue MySQL-DB-Instance mit der Konsole, dem create-db-instanceCLI-Befehl oder dem Vorgang Create DBInstance RDS API.

Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

- Ändern Sie eine bestehende MySQL-DB-Instance mithilfe der Konsole, des modify-db-instanceCLI-Befehls oder der Operation Modify DBInstance RDS API.

  Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.

- Stellen Sie mithilfe der Konsole, des CLI-Befehls -db-snapshot oder des API-Vorgangs Restore From DBSnapshot RDS eine restore-db-instance-fromMySQL-DB-Instance DBInstance aus einem DB-Snapshot wieder her.

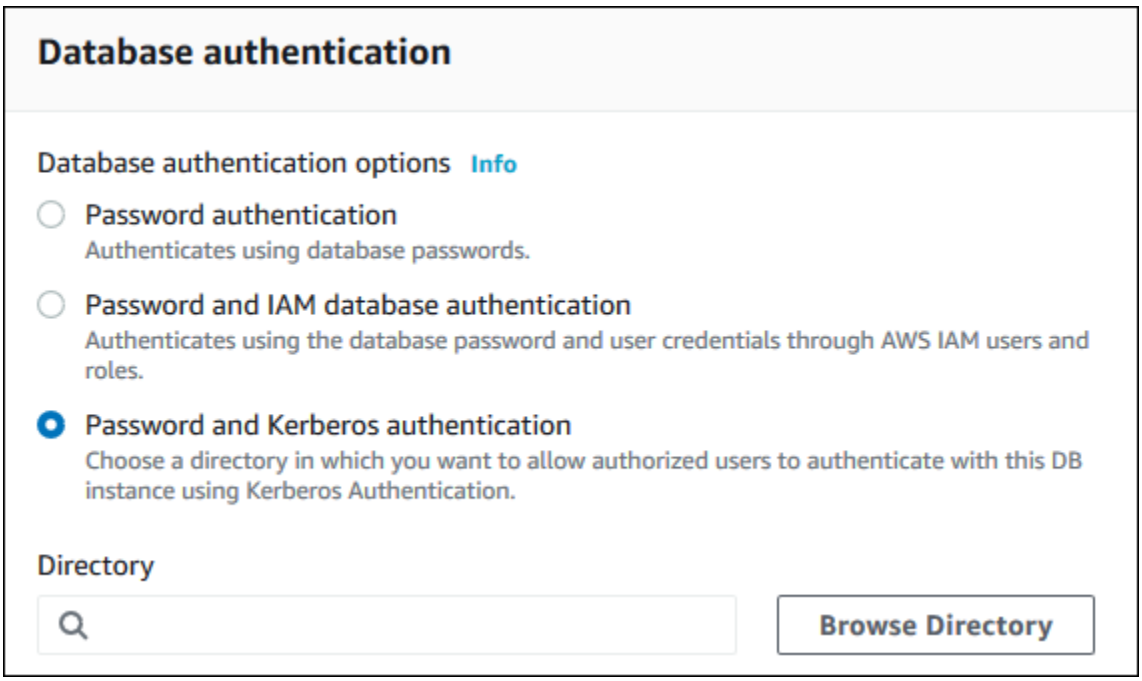
  Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.

- Stellen Sie eine MySQL-DB-Instance point-in-time mithilfe der Konsole, des Befehls restore-db-instance-to- point-in-time CLI oder der Operation Restore DBInstance ToPointInTime RDS API in a wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Die Kerberos-Authentifizierung wird nur für MySQL-DB-Instances in einer VPC unterstützt. Die DB-Instance kann sich in derselben VPC wie das Verzeichnis oder in einer anderen VPC befinden. Die DB-Instance muss eine Sicherheitsgruppe verwenden, die ausgehenden Datenverkehr innerhalb der VPC des Verzeichnisses ermöglicht, damit die DB-Instance mit dem Verzeichnis kommunizieren kann.

Wenn Sie die Konsole verwenden, ändern oder wiederherstellen, um eine DB-Instance zu erstellen, wählen Sie im Abschnitt Datenbankauthentifizierung die Option Passwort- und Kerberos-Authentifizierung aus. Wählen Sie Verzeichnis durchsuchen und dann das Verzeichnis aus, oder klicken Sie auf Neues Verzeichnis erstellen.



Wenn Sie die AWS CLI oder RDS-API verwenden, ordnen Sie eine DB-Instance einem Verzeichnis zu. Die folgenden Parameter sind erforderlich, damit die DB-Instance das von Ihnen erstellte Domain-Verzeichnis verwendet:

- Für den `--domain`-Parameter verwenden Sie den Domänenbezeichner („d-*“-Bezeichner), der beim Erstellen des Verzeichnisses generiert wurde.
- Verwenden Sie für den `--domain-iam-role-name`-Parameter die von Ihnen erstellte Rolle, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet.

Beispielsweise ändert der folgende CLI-Befehl eine DB-Instance zur Verwendung eines Verzeichnisses.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Windows:

```
aws rds modify-db-instance ^
```

```
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Wenn Sie eine DB-Instance zur Aktivierung der Kerberos-Authentifizierung ändern, starten Sie die DB-Instance neu, nachdem Sie die Änderung vorgenommen haben.

Schritt 5: Erstellen von MySQL-Anmeldeinformationen für die Kerberos-Authentifizierung

Verwenden Sie die Amazon RDS-Hauptbenutzer-Anmeldeinformationen, um sich mit der MySQL-DB-Instance wie mit jeder anderen DB-Instance zu verbinden. Die DB-Instance ist mit der AWS Managed Microsoft AD Domain verbunden. Auf diese Weise können Sie MySQL-Anmeldenamen und -Benutzer aus den Active Directory-Benutzern Ihrer Domäne bereitstellen. Die Datenbankberechtigungen werden durch Standard-MySQL-Berechtigungen verwaltet, die diesen Anmeldeinformationen gewährt und entzogen werden.

Sie können einem Active Directory-Benutzer erlauben, sich bei MySQL zu authentifizieren. Dazu verwenden Sie zunächst die Amazon RDS-Hauptbenutzer-Anmeldeinformationen, um sich mit der MySQL-DB-Instance wie mit jeder anderen DB-Instance zu verbinden. Nachdem Sie angemeldet sind, erstellen Sie einen extern authentifizierten Benutzer mit PAM (Pluggable Authentication Modules) in MySQL, indem Sie den folgenden Befehl ausführen. Ersetzen Sie *testuser* durch den Benutzernamen.

```
CREATE USER 'testuser'@'%' IDENTIFIED WITH 'auth_pam';
```

Benutzer (sowohl Menschen als auch Anwendungen) aus Ihrer Domäne können sich nun von einem mit der Domäne verbundenen Client-Rechner aus mit Hilfe der Kerberos-Authentifizierung mit der DB-Instance verbinden.

> **Important**
>
> Wir empfehlen dringend, dass Clients SSL/TLS Verbindungen verwenden, wenn sie die PAM-Authentifizierung verwenden. Wenn sie keine SSL/TLS Verbindungen verwenden, wird das Passwort in einigen Fällen möglicherweise als Klartext gesendet. Um eine SSL/TLS

verschlüsselte Verbindung für Ihren AD-Benutzer zu verlangen, führen Sie den folgenden Befehl aus und *testuser* ersetzen Sie ihn durch den Benutzernamen:

```
ALTER USER 'testuser'@'%' REQUIRE SSL;
```

Weitere Informationen finden Sie unter SSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS.

Verwalten einer DB-Instance in einer Domäne

Sie können die CLI oder die RDS-API verwenden, um Ihre DB-Instance und ihre Beziehung zu Ihrem verwalteten Active Directory zu verwalten. Sie können z. B. ein Active Directory für die Kerberos-Authentifizierung zuordnen und ein Active Directory trennen, um die Kerberos-Authentifizierung zu deaktivieren. Sie können auch eine DB-Instance, die extern von einem Active Directory authentifiziert werden soll, in ein anderes Active Directory verschieben.

Sie können z. B. mithilfe der Amazon RDS-API Folgendes tun:

- Um erneut zu versuchen, die Kerberos-Authentifizierung für eine fehlgeschlagene Mitgliedschaft zu aktivieren, verwenden Sie die `ModifyDBInstance` API-Operation und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft an.
- Um den IAM-Rollennamen für die Mitgliedschaft zu aktualisieren, verwenden Sie die `ModifyDBInstance`-API-Operation und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft und die neue IAM-Rolle an.
- Um die Kerberos-Authentifizierung in einer DB-Instance zu deaktivieren, verwenden Sie die `ModifyDBInstance` API-Operation. Geben Sie `none` als Domänenparameter an.
- Um eine DB-Instance von einer Domäne in eine andere zu verschieben, verwenden Sie die `ModifyDBInstance` API-Operation. Geben Sie die Domänen-ID der neuen Domäne als Domänenparameter an.
- Um die Mitgliedschaft für jede DB-Instance aufzulisten, verwenden Sie die `DescribeDBInstances` API-Operation.

Grundlegendes zur Domänenmitgliedschaft

Nachdem Sie Ihre DB-Instance erstellt oder geändert haben, wird sie Mitglied der Domäne. Sie können den Status der Domänenmitgliedschaft für die DB-Instance anzeigen, indem Sie den

describe-db-instancesCLI-Befehl ausführen. Der Status der DB-Instance kann einer der folgenden sein:

- `kerberos-enabled` – Für die DB-Instance ist die Kerberos-Authentifizierung aktiviert.
- `enabling-kerberos`— AWS ist dabei, die Kerberos-Authentifizierung für diese DB-Instance zu aktivieren.
- `pending-enable-kerberos` – Die Aktivierung der Kerberos-Authentifizierung in dieser DB-Instance steht aus.
- `pending-maintenance-enable-kerberos`— AWS wird versuchen, die Kerberos-Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu aktivieren.
- `pending-disable-kerberos` – Die Deaktivierung der Kerberos-Authentifizierung in dieser DB-Instance steht aus.
- `pending-maintenance-disable-kerberos`— AWS wird versuchen, die Kerberos-Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu deaktivieren.
- `enable-kerberos-failed` – Ein Konfigurationsproblem hat AWS daran gehindert, die Kerberos-Authentifizierung auf der DB-Instance zu aktivieren. Überprüfen und korrigieren Sie Ihre Konfiguration, bevor Sie den Befehl zu Änderung der DB-Instance erneut ausführen.
- `disabling-kerberos`— AWS ist dabei, die Kerberos-Authentifizierung auf dieser DB-Instance zu deaktivieren.

Eine Anfrage zur Aktivierung der Kerberos-Authentifizierung kann wegen eines Netzwerkverbindungsproblems oder einer falschen IAM-Rolle fehlschlagen. Angenommen, Sie erstellen eine DB-Instance oder ändern eine vorhandene DB-Instance und der Versuch, die Kerberos-Authentifizierung zu aktivieren, schlägt fehl. Wenn dies geschieht, führen Sie den Ändern-Befehl erneut aus oder ändern Sie die neu erstellte DB-Instance, um der Domäne beizutreten.

Verbindung zu MySQL mit Kerberos-Authentifizierung

Um eine Verbindung zu MySQL mit Kerberos-Authentifizierung herzustellen, müssen Sie sich mit dem Kerberos-Authentifizierungstyp anmelden.

Um einen Datenbankbenutzer zu erstellen, zu dem Sie eine Verbindung mit der Kerberos-Authentifizierung herstellen können, verwenden Sie eine `IDENTIFIED WITH`-Klausel in der `CREATE`

USER-Anweisung. Detaillierte Anweisungen finden Sie unter Schritt 5: Erstellen von MySQL-Anmeldeinformationen für die Kerberos-Authentifizierung.

Um Fehler zu vermeiden, sollten Sie den MariaDB-`mysql`-Client verwenden. Sie können die MariaDB-Software unter https://downloads.mariadb.org/ herunterladen.

Stellen Sie über die Eingabeaufforderung eine Verbindung zu einem der Endpunkte her, die mit Ihrer MySQL-DB-Instance verbunden sind. Befolgen Sie die allgemeinen Verfahren in Verbindung zu Ihrer MySQL-DB-Instance herstellen. Wenn Sie zur Eingabe des Passworts aufgefordert werden, geben Sie es mit diesem Benutzernamen verknüpfte Kerberos-Passwort ein.

Wiederherstellen einer MySQL-DB-Instance und Hinzufügen zu einer Domäne

Sie können einen DB-Snapshot point-in-time wiederherstellen oder eine Wiederherstellung für eine MySQL-DB-Instance abschließen und sie dann einer Domain hinzufügen. Nachdem die DB-Instance wiederhergestellt wurde, modifizieren Sie die DB-Instance mit dem in Schritt 4: Erstellen oder Ändern einer MySQL-DB-Instance erklärten Prozess, um die DB-Instance zu einer Domäne hinzuzufügen.

MySQL-Einschränkungen bei der Kerberos-Authentifizierung

Die folgenden Einschränkungen gelten für die Kerberos-Authentifizierung für MySQL:

- Es AWS Managed Microsoft AD wird nur eine Verbindung unterstützt. Sie können jedoch DB-Instances von RDS for MySQL zu gemeinsam genutzten verwalteten Microsoft-AD-Domänen zusammenführen, die verschiedene Konten in derselben AWS-Region gehören.
- Sie müssen die DB-Instance neu starten, nachdem Sie die Funktion aktiviert haben.
- Die Länge des Domänennamens darf nicht länger als 61 Zeichen sein.
- Sie können nicht gleichzeitig die Kerberos-Authentifizierung und die IAM-Authentifizierung aktivieren. Wählen Sie die eine oder die andere Authentifizierungsmethode für Ihre MySQL-DB-Instance aus.
- Ändern Sie den DB-Instance-Port nicht, nachdem Sie die Funktion aktiviert haben.
- Verwenden Sie keine Kerberos-Authentifizierung mit Lesereplikaten.
- Wenn Sie das automatische Minor-Versions-Upgrade für eine MySQL DB-Instance aktiviert haben, die die Kerberos-Authentifizierung verwendet, müssen Sie die Kerberos-Authentifizierung deaktivieren und nach einem automatischen Upgrade wieder einschalten. Weitere Informationen zu kleineren automatischen Versionsaktualisierungen finden Sie unter Automatische Upgrades für kleinere Versionen für RDS für My SQL.

- Um eine DB-Instance bei aktivierter Funktion zu löschen, deaktivieren Sie zuerst die Funktion. Führen Sie dazu den `modify-db-instance` CLI-Befehl für die DB-Instance aus und geben Sie `none` den `--domain` Parameter an.

  Wenn Sie die CLI oder die RDS-API verwenden, um eine DB-Instance bei aktivierter Funktion zu löschen, müssen Sie mit einer Verzögerung rechnen.
- RDS for MySQL unterstützt keine Kerberos-Authentifizierung in einer Gesamtstruktur zwischen Ihrem lokalen oder selbst gehosteten AD und dem. AWS Managed Microsoft AD

Verbesserung der Abfrageleistung RDS für My SQL mit Amazon RDS Optimized Reads

Mit Amazon RDS Optimized Reads können Sie eine schnellere Abfrageverarbeitung RDS für for My SQL erreichen. Eine RDS SQL For-My-DB-Instance oder ein Multi-AZ-DB-Cluster, der RDS Optimized Reads verwendet, kann eine bis zu zweimal schnellere Abfrageverarbeitung erreichen als eine DB-Instance oder ein DB-Cluster, die Optimized Reads nicht verwenden.

Themen

- Überblick über RDS optimierte Lesevorgänge
- Anwendungsfälle für RDS optimierte Lesevorgänge
- Bewährte Methoden für RDS optimierte Lesevorgänge
- RDSOptimierte Lesevorgänge verwenden
- Überwachung von DB-Instances, die RDS Optimized Reads verwenden
- Einschränkungen für RDS optimierte Lesevorgänge

Überblick über RDS optimierte Lesevorgänge

Wenn Sie eine RDS For My SQL DB-Instance oder einen Multi-AZ-DB-Cluster verwenden, bei dem RDS Optimized Reads aktiviert sind, wird durch die Verwendung eines Instance-Speichers eine schnellere Abfrageleistung erzielt. Ein Instance-Speicher stellt für Ihre DB-Instance bzw. Ihren Multi-AZ-DB-Cluster temporären Speicher auf Blockebene bereit. Der Speicher befindet sich auf Non-Volatile Memory Express (NVMe) Solid-State-Laufwerken (SSDs), die physisch mit dem Host-Server verbunden sind. Dieser Speicher ist für niedrige Latenzen, eine hohe Random-I/O-Leistung und einen hohen sequentiellen Lesedurchsatz optimiert.

RDSOptimized Reads ist standardmäßig aktiviert, wenn eine DB-Instance oder ein Multi-AZ-DB-Cluster eine DB-Instance-Klasse mit einem Instance-Speicher verwendet, z. B. db.m5d oder db.m6gd. Bei RDS Optimized Reads werden einige temporäre Objekte im Instance-Speicher gespeichert. Zu diesen temporären Objekten gehören interne temporäre Dateien, interne temporäre Tabellen auf der Festplatte, Speicherzuordnungsdateien und Binärprotokolldateien (binlog) im Cache. Weitere Informationen zum Instance-Store finden Sie unter Amazon EC2 Instance Store im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Die Workloads, die temporäre Objekte in My SQL für die Abfrageverarbeitung generieren, können den Instance-Speicher für eine schnellere Abfrageverarbeitung nutzen. Zu dieser Art

von Arbeitslast gehören Abfragen, die Sortierungen, Hash-Aggregationen, Verknüpfungen mit hoher Auslastung, allgemeine Tabellenausdrücke (CTEs) und Abfragen für Spalten ohne Index beinhalten. Diese Instance-Speicher-Volumes bieten eine höhere IOPS Leistung, unabhängig von den Speicherkonfigurationen, die für persistenten EBS Amazon-Speicher verwendet werden. Da RDS Optimized Reads Operationen an temporären Objekten in den Instance-Speicher auslagert, können die Eingabe-/Ausgabeoperationen pro Sekunde (IOPS) oder der Durchsatz des persistenten Speichers (AmazonEBS) jetzt für Operationen mit persistenten Objekten verwendet werden. Zu diesen Vorgängen gehören reguläre Lese- und Schreibvorgänge von Datendateien sowie Engine-Operationen im Hintergrund, wie das Leeren und Zusammenführen von Puffern zum Einfügen.

> **Note**
>
> Sowohl manuelle als auch automatisierte RDS Snapshots enthalten nur Engine-Dateien für persistente Objekte. Die im Instanzspeicher erstellten temporären Objekte sind nicht in RDS Snapshots enthalten.

Anwendungsfälle für RDS optimierte Lesevorgänge

Wenn Sie Workloads haben, die für die Ausführung von Abfragen stark auf temporäre Objekte wie interne Tabellen oder Dateien angewiesen sind, können Sie von der Aktivierung RDS optimierter Lesevorgänge profitieren. Die folgenden Anwendungsfälle kommen für RDS optimierte Lesevorgänge in Frage:

- Anwendungen, die analytische Abfragen mit komplexen allgemeinen Tabellenausdrücken (CTEs), abgeleiteten Tabellen und Gruppierungsoperationen ausführen
- Lesereplikate, die einen hohen Leseverkehr mit nicht optimierten Abfragen bewältigen
- Anwendungen, die auf Abruf laufen oder dynamische Berichtsabfragen ausführen, die komplexe Operationen beinhalten, z. B. Abfragen mit `GROUP BY`- und `ORDER BY`-Klauseln
- Workloads, die interne temporäre Tabellen für die Abfrageverarbeitung verwenden

  Sie können die Engine-Statusvariable `created_tmp_disk_tables` überwachen, um die Anzahl der festplattenbasierten temporären Tabellen zu ermitteln, die auf Ihrer DB-Instance erstellt wurden.
- Anwendungen, die direkt oder in Prozeduren große temporäre Tabellen zum Speichern von Zwischenergebnissen erstellen
- Datenbankabfragen, die das Gruppieren oder Sortieren von nicht indizierten Spalten durchführen

Bewährte Methoden für RDS optimierte Lesevorgänge

Verwenden Sie die folgenden bewährten Methoden für RDS optimierte Lesevorgänge:

- Fügen Sie Wiederholungslogik für schreibgeschützte Abfragen hinzu, falls diese aufgrund eines Fehlers wegen eines vollen Instance-Speichers während der Ausführung fehlschlagen.
- Überwachen Sie anhand der CloudWatch Metrik den im Instance-Speicher verfügbaren Speicherplatz`FreeLocalStorage`. Wenn der Instance-Speicher aufgrund der Workload der DB-Instance sein Limit erreicht, ändern Sie die DB-Instance, um eine größere DB-Instance-Klasse zu verwenden.
- Wenn Ihre DB-Instance bzw. Ihr Multi-AZ-DB-Cluster über ausreichend Speicher verfügt, aber immer noch das Speicherlimit für den Instance-Speicher erreicht, erhöhen Sie den `binlog_cache_size`-Wert, um die sitzungsspezifischen Binlog-Einträge im Speicher zu behalten. Diese Konfiguration verhindert, dass die Binlog-Einträge in temporäre Binlog-Cache-Dateien auf der Festplatte geschrieben werden.

  Der `binlog_cache_size`-Parameter ist sitzungsspezifisch. Sie können den Wert für jede neue Sitzung ändern. Die Einstellung für diesen Parameter kann die Speicherauslastung der DB-Instance bei Spitzenauslastung erhöhen. Erwägen Sie daher, den Parameterwert auf der Grundlage des Workload-Musters Ihrer Anwendung und des verfügbaren Speichers in der DB-Instance zu erhöhen.
- Verwenden Sie SQL für My 8.0-Versionen und frühere Versionen den Standardwert von `MIXED` für den `binlog_format` Parameter. Abhängig von der Größe der Transaktionen kann die Einstellung von `binlog_format` auf `ROW` zu großen Binlog-Cache-Dateien im Instance-Speicher führen. Verwenden Sie für My SQL 8.4 und höher den Standardwert von `ROW` für den `binlog_format` Parameter.
- Stellen Sie den Parameter internal_tmp_mem_storage_engine auf `TempTable` ein und legen Sie den Parameter temptable_max_mmap so fest, dass er der Größe des verfügbaren Speichers im Instance-Speicher entspricht.
- Vermeiden Sie es, Massenänderungen in einer einzigen Transaktion durchzuführen. Diese Arten von Transaktionen können große Binlog-Cache-Dateien im Instance-Speicher generieren und Probleme verursachen, wenn der Instance-Speicher voll ist. Erwägen Sie, Schreibvorgänge in mehrere kleine Transaktionen aufzuteilen, um den Speicherverbrauch für Binlog-Cache-Dateien zu minimieren.

- Verwenden Sie den Standardwert `ABORT_SERVER` für den `binlog_error_action`-Parameter. Dadurch werden Probleme mit der binären Protokollierung auf DB-Instances mit aktivierten Backups vermieden.

RDSOptimierte Lesevorgänge verwenden

Wenn Sie eine RDS for My SQL DB-Instance mit einer der folgenden DB-Instance-Klassen in einer Single-AZ-DB-Instance-Bereitstellung, einer Multi-AZ-DB-Instance-Bereitstellung oder einer Multi-AZ-DB-Cluster-Bereitstellung bereitstellen, verwendet RDS die DB-Instance automatisch Optimized Reads.

Gehen Sie wie folgt vor, um RDS Optimized Reads zu aktivieren:

- Erstellen Sie mithilfe einer dieser SQL DB-Instance-Klassen eine RDS For My DB-Instance oder einen Multi-AZ-DB-Cluster. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine bestehende RDS für My SQL DB-Instance oder einen Multi-AZ-DB-Cluster, um eine dieser DB-Instance-Klassen zu verwenden. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

RDSOptimized Reads ist überall verfügbar AWS-Regionen RDS, wo eine oder mehrere DB-Instance-Klassen mit lokalem NVMe SSD Speicher unterstützt werden. Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”.

Die Verfügbarkeit von DB-Instance-Klassen unterscheidet sich für AWS-Regionen. Informationen darüber, ob eine DB-Instance-Klasse in einer bestimmten Klasse unterstützt wird AWS-Region, finden Sie unterthe section called “Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen”.

Wenn Sie RDS Optimized Reads nicht verwenden möchten, ändern Sie Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster so, dass keine DB-Instance-Klasse verwendet wird, die diese Funktion unterstützt.

Überwachung von DB-Instances, die RDS Optimized Reads verwenden

Sie können DB-Instances, die RDS Optimized Reads verwenden, mit den folgenden CloudWatch Metriken überwachen:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Diese Metriken liefern Daten über verfügbaren Instance-SpeicherIOPS, Speicherplatz und Durchsatz. Weitere Informationen zu diesen Metriken finden Sie unter Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS.

Einschränkungen für RDS optimierte Lesevorgänge

Die folgenden Einschränkungen gelten für RDS optimierte Lesevorgänge:

- RDSOptimized Reads wird für die folgenden Versionen unterstützt:
  - RDSfür meine SQL Version 8.0.28 und höhere Haupt- und Nebenversionen

  Informationen zu RDS for My SQL versions finden Sie unter. MySQL in Amazon RDS-Versionen
- Sie können den Speicherort temporärer Objekte in den DB-Instance-Klassen, die RDS optimierte Lesevorgänge unterstützen, nicht in persistenten Speicher (AmazonEBS) ändern.
- Wenn die binäre Protokollierung auf einer DB-Instance aktiviert ist, ist die maximale Transaktionsgröße durch die Größe des Instance-Speichers begrenzt. In My `binlog_cache_size` schreibt jede SitzungSQL, die mehr Speicherplatz als den Wert von benötigt, Transaktionsänderungen in temporäre Binlog-Cache-Dateien, die im Instance-Speicher erstellt werden.
- Transaktionen können fehlschlagen, wenn der Instance-Speicher voll ist.

Verbesserung der Schreibleistung mit RDS Optimized Writes for My SQL

Mit RDS Optimized Writes for My können Sie die Leistung von Schreibtransaktionen verbessernSQL. Wenn Ihre RDS For My SQL -Datenbank RDS Optimized Writes verwendet, kann sie einen bis zu zweimal höheren Durchsatz bei Schreibtransaktionen erreichen.

Themen

- Überblick über RDS optimierte Schreibvorgänge
- RDSOptimierte Schreibvorgänge verwenden
- RDSOptimierte Schreibvorgänge in einer vorhandenen Datenbank aktivieren
- Einschränkungen für RDS optimierte Schreibvorgänge

Überblick über RDS optimierte Schreibvorgänge

Wenn Sie RDS Optimierte Schreibvorgänge aktivieren, schreiben Ihre Datenbanken RDS für Meine SQL Datenbanken nur einmal, wenn sie Daten in einen dauerhaften Speicher speichern, ohne dass der DoubleWrite-Puffer erforderlich ist. Die Datenbanken bieten weiterhin ACID Eigentumsschutz für zuverlässige Datenbanktransaktionen und verbessern gleichzeitig die Leistung.

Relationale Datenbanken wie My bieten die ACIDEigenschaften AtomaritätSQL, Konsistenz, Isolation und Beständigkeit, die für zuverlässige Datenbanktransaktionen erforderlich sind. Um diese Eigenschaften bereitzustellen, SQL verwendet My einen Datenspeicherbereich, den sogenannten Doublewrite-Puffer, der partielle Seitenschreibfehler verhindert. Diese Fehler treten bei einem Hardwarefehler auf, während die Datenbank eine Seite aktualisiert, z. B. bei einem Stromausfall. A Meine SQL Datenbank kann partielle Seitenschreibvorgänge erkennen und mit einer Kopie der Seite im Doublewrite-Puffer wiederherstellen. Diese Technik bietet zwar Schutz, führt aber auch zu zusätzlichen Schreiboperationen. Weitere Informationen zum My SQL Doublewrite-Buffer finden Sie unter Doublewrite-Buffer in der Dokumentation Meine Daten. SQL

Wenn RDS Optimierte Schreibvorgänge aktiviert sind, wird RDS für Meine SQL Datenbanken nur einmal geschrieben, wenn Daten in einen dauerhaften Speicher geleert werden, ohne den Doublewrite-Puffer zu verwenden. RDS Optimierte Schreibvorgänge sind nützlich, wenn Sie schreibintensive Workloads in Ihren RDS for My Datenbanken ausführen. SQL Zu den Datenbanken mit schreibintensiven Workloads gehören Datenbanken, die digitale Zahlungen, Finanzhandel und Spieleanwendungen unterstützen.

Diese Datenbanken laufen auf DB-Instance-Klassen, die das Nitro-System verwenden. AWS Aufgrund der Hardwarekonfiguration in diesen Systemen kann die Datenbank in einem Schritt zuverlässig und dauerhaft Seiten mit 16 KiB direkt in Datendateien schreiben. Das AWS Nitro-System ermöglicht RDS optimierte Schreibvorgänge.

Sie können den neuen Datenbankparameter festlegen`rds.optimized_writes`, um die Funktion RDS Optimierte Schreibvorgänge RDS für Meine SQL Datenbanken zu steuern. Greifen Sie in den DB-Parametergruppen RDS für Meine SQL Version 8.0 und RDS für Meine SQL Version 8.4 auf diesen Parameter zu. Legen Sie den Parameter anhand der folgenden Werte fest:

- `AUTO`— Aktiviert RDS Optimized Writes, wenn die Datenbank dies unterstützt. Schalten Sie RDS Optimierte Schreibvorgänge aus, wenn die Datenbank dies nicht unterstützt. Dies ist die Standardeinstellung.
- `OFF`— Schalten Sie RDS optimierte Schreibvorgänge aus, auch wenn die Datenbank dies unterstützt.

Wenn Sie über eine bestehende Datenbank mit einer Engine-Version, DB-Instance-Klasse und and/or file system format that doesn't support RDS Optimized Writes, you can enable the feature by creating a blue/green Bereitstellung verfügen. Weitere Informationen finden Sie unter the section called “Aktivieren in einer vorhandenen Datenbank”.

Wenn Sie eine RDS SQL For-My-Datenbank, die für die Verwendung RDS optimierter Schreibvorgänge konfiguriert ist, zu einer DB-Instance-Klasse migrieren, die diese Funktion nicht unterstützt, RDS werden RDS Optimierte Schreibvorgänge für die Datenbank automatisch deaktiviert.

Wenn RDS Optimized Writes deaktiviert ist, verwendet die Datenbank den SQL Doublewrite-Puffer My.

Um festzustellen, ob eine „RDSFor My“ SQL -Datenbank RDS optimierte Schreibvorgänge verwendet, zeigen Sie den aktuellen Wert des `innodb_doublewrite` Parameters für die Datenbank an. Wenn die Datenbank RDS Optimized Writes verwendet, ist dieser Parameter auf `FALSE` (`0`) gesetzt.

RDSOptimierte Schreibvorgänge verwenden

Sie können RDS optimierte Schreibvorgänge aktivieren, wenn Sie eine Datenbank RDS für Meine SQL Datenbank mit der RDS Konsole AWS CLI, dem oder dem erstellen RDSAPI. RDS Optimierte Schreibvorgänge werden automatisch aktiviert, wenn bei der Datenbankerstellung beide der folgenden Bedingungen zutreffen:

- Sie geben eine DB-Engine-Version und eine DB-Instance-Klasse an, die RDS optimierte Schreibvorgänge unterstützen.
  - RDSOptimized Writes wird RDS für My SQL Version 8.0.30 und höher unterstützt. Informationen zu RDS for My SQL versions finden Sie unter. MySQL in Amazon RDS-Versionen
  - RDSOptimized Writes wird RDS für SQL Datenbanken vom Typ My unterstützt, die die folgenden DB-Instance-Klassen verwenden:
    - db.m7i
    - db.m7g
    - db.m6g
    - db.m6gd
    - db.m6i
    - db.m5
    - db.m5d
    - db.r7i
    - db.r7g
    - db.r6g
    - db.r6gd
    - db.r6i
    - db.r5
    - db.r5b
    - db.r5d
    - db.x2idn
    - db.x2iedn

    Weitere Informationen zu DB-Instance-Klassen finden Sie unter the section called “DB-Instance-Klassen”.

    Die Verfügbarkeit der DB-Instance-Klassen unterscheidet sich für. AWS-Regionen Informationen darüber, ob eine DB-Instance-Klasse in einer bestimmten Klasse unterstützt wird AWS-Region, finden Sie unterthe section called “Ermitteln der Unterstützung für DB-Instance-Klassen in AWS-Regionen”.

Um Ihre Datenbank auf eine DB-Instance-Klasse zu aktualisieren, die RDS optimierte Schreibvorgänge unterstützt, können Sie eine blaue/grüne Bereitstellung erstellen. Weitere Informationen finden Sie unter the section called “Aktivieren in einer vorhandenen Datenbank”.

- In der Parametergruppe, die der Datenbank zugeordnet ist, ist der `rds.optimized_writes`-Parameter auf `AUTO` eingestellt. In Standardparametergruppen ist dieser Parameter immer auf `AUTO` festgelegt.

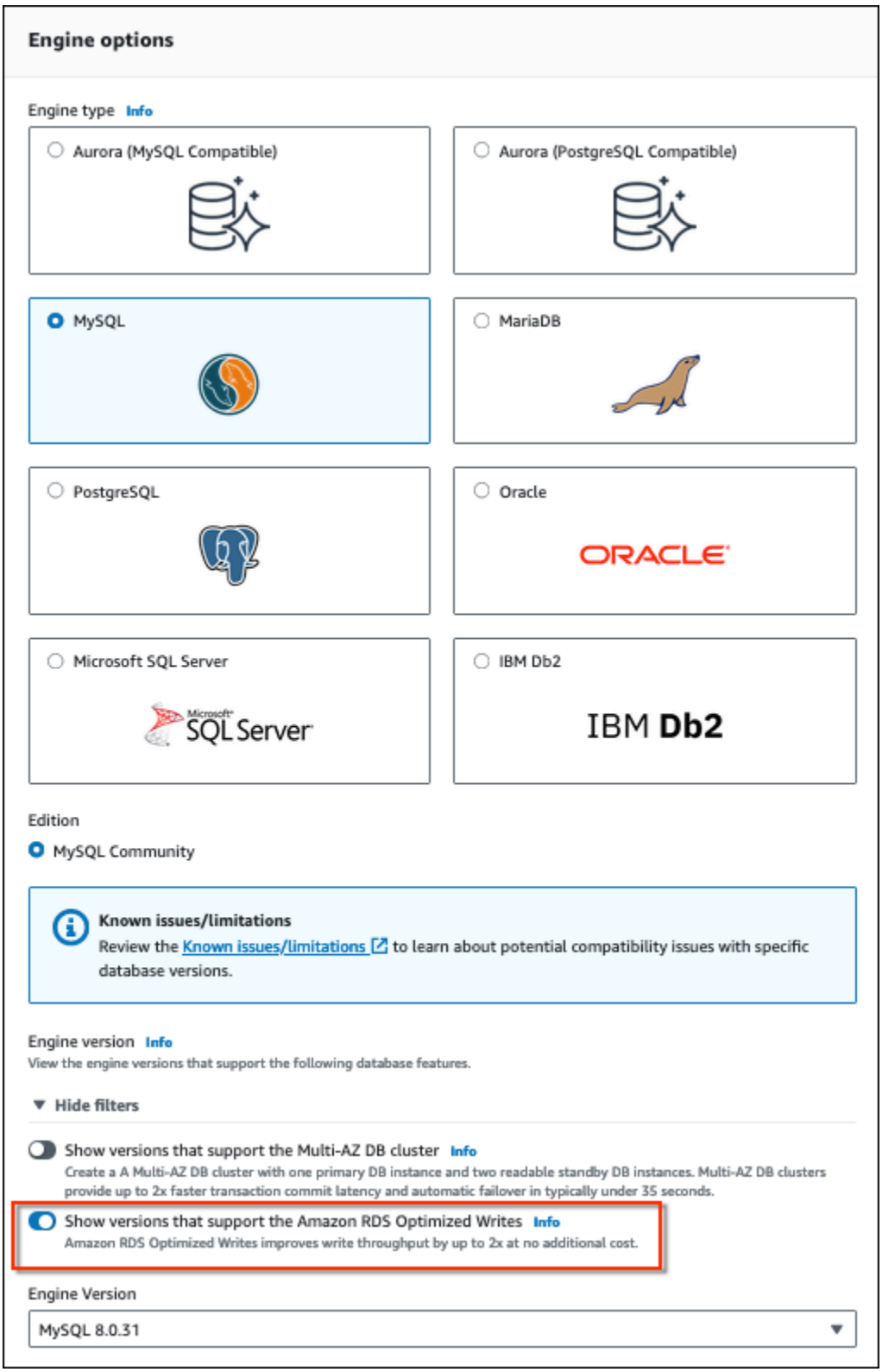
Wenn Sie eine DB-Engine-Version und eine DB-Instance-Klasse verwenden möchten, die RDS optimierte Schreibvorgänge unterstützen, diese Funktion jedoch nicht verwenden möchten, geben Sie beim Erstellen der Datenbank eine benutzerdefinierte Parametergruppe an. Legen Sie den Parameter `rds.optimized_writes` in dieser Parametergruppe auf `OFF` fest. Wenn Sie möchten, dass die Datenbank später RDS optimierte Schreibvorgänge verwendet, können Sie den Parameter auf setzen`AUTO`, um sie zu aktivieren. Weitere Informationen über das Erstellen von benutzerdefinierten DB-Parametergruppen und das Festlegen von Parametern finden Sie unter Parametergruppen für Amazon RDS.

Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

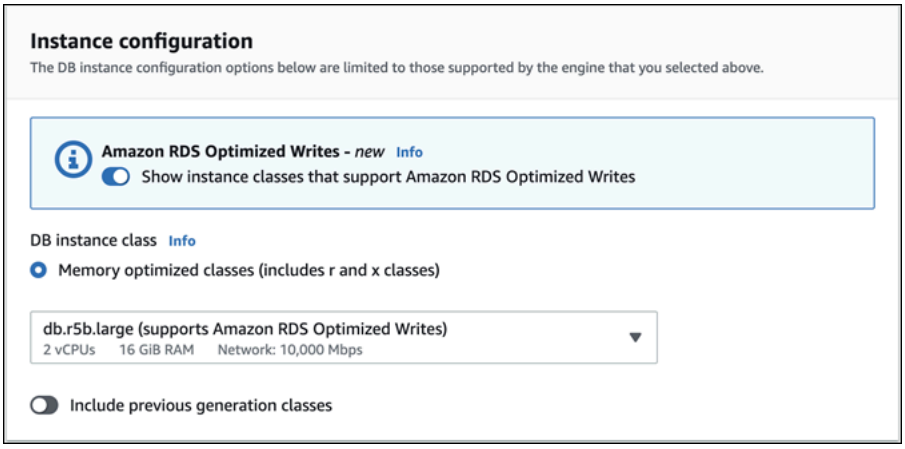
Konsole

Wenn Sie die RDS Konsole verwenden, um eine Datenbank RDS für Meine SQL Datenbank zu erstellen, können Sie nach den DB-Engine-Versionen und DB-Instance-Klassen filtern, die RDS optimierte Schreibvorgänge unterstützen. Nachdem Sie die Filter aktiviert haben, können Sie aus den verfügbaren DB-Engine-Versionen und DB-Instance-Klassen auswählen.

Um eine DB-Engine-Version auszuwählen, die RDS optimierte Schreibvorgänge unterstützt, filtern Sie in der Engine-Version nach den Versionen RDS für Meine SQL DB-Engine, die diese unterstützen, und wählen Sie dann eine Version aus.



Filtern Sie im Abschnitt Instance-Konfiguration nach den DB-Instance-Klassen, die RDS Optimized Writes unterstützen, und wählen Sie dann eine DB-Instance-Klasse aus.



Nachdem Sie diese Auswahl getroffen haben, können Sie andere Einstellungen wählen, die Ihren Anforderungen entsprechen, und die Erstellung der Datenbank RDS für Meine SQL Datenbank mit der Konsole abschließen.

AWS CLI

Um eine DB-Instance mit dem zu erstellen AWS CLI, führen Sie den create-db-instanceBefehl aus. Stellen Sie sicher, dass die `--db-instance-class` Werte `--engine-version` und RDS optimierte Schreibvorgänge unterstützen. Stellen Sie außerdem sicher, dass der `rds.optimized_writes`-Parameter für die Parametergruppe, die der DB-Instance zugeordnet ist, auf AUT0 festgelegt ist. Im folgenden Beispiel wird die Standardparametergruppe mit der DB-Instance verknüpft.

Example Eine DB-Instance erstellen, die RDS optimierte Schreibvorgänge verwendet

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.30 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
```

```
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
    --engine-version 8.0.30 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

RDS API

Sie können eine DB-Instance mithilfe der reateDBInstanceC-Operation erstellen. Wenn Sie diesen Vorgang verwenden, stellen Sie sicher, dass die `DBInstanceClass` Werte `EngineVersion` und RDS optimierte Schreibvorgänge unterstützen. Stellen Sie außerdem sicher, dass der `rds.optimized_writes`-Parameter für die Parametergruppe, die der DB-Instance zugeordnet ist, auf `AUTO` festgelegt ist.

RDSOptimierte Schreibvorgänge in einer vorhandenen Datenbank aktivieren

Um eine bestehende RDS SQL Datenbank so zu ändern, dass RDS optimierte Schreibvorgänge aktiviert werden, muss die Datenbank mit einer unterstützten DB-Engine-Version und DB-Instance-Klasse erstellt worden sein. Darüber hinaus muss die Datenbank nach der Veröffentlichung von RDS Optimized Writes am 27. November 2022 erstellt worden sein, da die erforderliche zugrunde liegende Dateisystemkonfiguration nicht mit der von Datenbanken kompatibel ist, die vor der Veröffentlichung erstellt wurden. Wenn diese Bedingungen erfüllt sind, können Sie RDS Optimized Writes aktivieren, indem Sie den `rds.optimized_writes` Parameter auf setzen`AUTO`.

Wenn Ihre Datenbank nicht mit einer unterstützten Engine-Version, Instanzklasse oder Dateisystemkonfiguration erstellt wurde, können Sie RDS Blue/Green Deployments to migrate to a supported configuration. While creating the blue/green Deployment verwenden. Gehen Sie dazu wie folgt vor:

- Wählen Sie Optimierte Schreibvorgänge für grüne Datenbank aktivieren aus und geben Sie dann eine Engine-Version und eine DB-Instance-Klasse an, die RDS optimierte Schreibvorgänge unterstützt. Eine Liste der unterstützten Engine-Versionen und Instance-Klassen finden Sie unter RDSOptimierte Schreibvorgänge verwenden.
- Wählen Sie unter Speicher die Option Speicherdatei-Systemkonfiguration aktualisieren aus. Mit dieser Option wird die Datenbank auf eine kompatible zugrunde liegende Dateisystemkonfiguration aktualisiert.

Wenn Sie die blaue/grüne Bereitstellung erstellen und der `rds.optimized_writes` Parameter auf gesetzt ist`AUTO`, werden RDS optimierte Schreibvorgänge in der grünen Umgebung automatisch aktiviert. Sie können dann die Blau/Grün-Bereitstellung umstellen, wodurch die grüne Umgebung zur neuen Produktionsumgebung hochgestuft wird.

Weitere Informationen finden Sie unter the section called “Erstellen einer Blau/Grün-Bereitstellung”.

Einschränkungen für RDS optimierte Schreibvorgänge

Wenn Sie eine Datenbank RDS für Meine SQL Datenbank aus einem Snapshot wiederherstellen, können Sie RDS Optimierte Schreibvorgänge für die Datenbank nur aktivieren, wenn alle der folgenden Bedingungen zutreffen:

- Der Snapshot wurde aus einer Datenbank erstellt, die RDS optimierte Schreibvorgänge unterstützt.
- Der Snapshot wurde aus einer Datenbank erstellt, die nach der Veröffentlichung von RDS Optimized Writes erstellt wurde.
- Der Snapshot wird in einer Datenbank wiederhergestellt, die RDS optimierte Schreibvorgänge unterstützt.
- Die wiederhergestellte Datenbank ist einer Parametergruppe zugeordnet, deren `rds.optimized_writes`-Parameter auf `AUTO` eingestellt ist.

Upgrades der DB-Engine RDS für MySQL

Sofern Amazon RDS eine neue Version der Datenbank-Engine unterstützt, können Sie Ihre DB-Instances auf die neue Version aktualisieren. Es gibt zwei Arten von Upgrades für MySQL-Datenbanken: Hauptversions-Upgrades und Nebenversions-Upgrades.

Hauptversions-Upgrades

Hauptversions-Upgrades können Datenbankänderungen enthalten, die nicht mit vorhandenen Anwendungen rückwärts kompatibel sind. Daher müssen Sie Hauptversions-Upgrades Ihrer DB-Instances manuell durchführen. Sie können ein Hauptversions-Upgrade starten, indem Sie Ihre DB-Instance ändern. Bevor Sie ein Upgrade auf eine Hauptversion durchführen, empfehlen wir Ihnen, die Anweisungen unter zu befolgenHauptversions-Upgrades für RDS for MySQL.

Bei Hauptversions-Upgrades von Multi-AZ-DB-Instance-Bereitstellungen aktualisiert Amazon RDS gleichzeitig die Primär- und Standby-Replikate. Ihre DB-Instance ist erst verfügbar, wenn das Upgrade abgeschlossen ist. Bei Hauptversions-Upgrades von Multi-AZ-DB-Cluster-Bereitstellungen aktualisiert Amazon RDS die Cluster-Mitgliedsinstanzen nacheinander.

> **Tip**
>
> Sie können die Ausfallzeit, die für ein Upgrade einer Hauptversion erforderlich ist, minimieren, indem Sie eine blaue/grüne Bereitstellung verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Unterversion-Upgrades

Upgrades kleinerer Versionen beinhalten nur Änderungen, die mit vorhandenen Anwendungen abwärtskompatibel sind. Sie können ein Nebenversions-Upgrade manuell starten, indem Sie Ihre DB-Instance ändern. Sie können auch die Option Automatisches Upgrade der Nebenversion aktivieren, wenn Sie eine DB-Instance erstellen oder ändern. Das bedeutet, dass Amazon RDS Ihre DB-Instance automatisch aktualisiert, nachdem Sie die neue Version getestet und genehmigt haben. Weitere Informationen zum Ausführen eines Upgrades finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Wenn Sie ein kleineres Versions-Upgrade eines Multi-AZ-DB-Clusters durchführen, aktualisiert Amazon RDS die Reader-DB-Instances nacheinander. Dann wird eine der Reader-DB-Instances

zur neuen Writer-DB-Instance. Amazon RDS aktualisiert dann die alte Writer-Instance (die jetzt eine Reader-Instance ist).

> **Note**
>
> Die Ausfallzeit bei einem kleineren Versions-Upgrade einer Multi-AZ-DB-Instance-Bereitstellung kann mehrere Minuten dauern. Multi-AZ-DB-Cluster reduzieren die Ausfallzeit bei Upgrades kleinerer Versionen in der Regel auf etwa 35 Sekunden. Bei Verwendung mit RDS Proxy können Sie die Ausfallzeit weiter auf eine Sekunde oder weniger reduzieren. Weitere Informationen finden Sie unter Amazon-RDS-Proxy. Alternativ können Sie einen Open-Source-Datenbank-Proxy wie ProxySQL oder den AWS Advanced JDBC PgBouncerWrapper Driver verwenden.

Wenn Ihre MySQL-DB-Instance Read Replicas verwendet, müssen Sie alle Read Replicas aktualisieren, bevor Sie die Quell-Instance aktualisieren.

Themen

- Überlegungen zu MySQL-Upgrades
- Gültige Upgrade-Ziele finden
- MySQL-Versionsnummern
- RDS-Versionsnummern in RDS für MySQL
- Hauptversions-Upgrades für RDS for MySQL
- Ein SQL Upgrade RDS für mein Gerät testen
- Upgraden einer MySQL-DB-Instance
- Automatische Upgrades für kleinere Versionen für RDS für My SQL
- Verwendung einer Read Replica zur Reduzierung von Ausfallzeiten beim Upgrade einer RDS for MySQL-Datenbank

Überlegungen zu MySQL-Upgrades

Amazon RDS erstellt zwei oder mehr DB-Snapshots während des Upgrades. Amazon RDS erstellt bis zu zwei Snapshots der DB-Instance, bevor Upgrade-Änderungen vorgenommen werden. Wenn das Upgrade bei Ihren Datenbanken nicht funktioniert, können Sie einen dieser Snapshots wiederherstellen, um eine DB-Instance zu erstellen, auf der die alte Version ausgeführt wird. Amazon

RDS erstellt einen weiteren Snapshot der DB-Instance, wenn das Upgrade abgeschlossen ist. Amazon RDS erstellt diese Snapshots unabhängig davon, ob die Backups für die DB-Instance AWS Backup verwaltet werden.

> **Note**
>
> Amazon RDS nimmt nur DB-Snapshots auf, wenn Sie den Sicherungsaufbewahrungszeitraum für Ihre DB-Instance auf eine Zahl größer als 0 festgelegt haben. Informationen über das Ändern Ihres Aufbewahrungszeitraums für Backups finden Sie unter Ändern einer Amazon RDS DB-Instance.

Nachdem das Upgrade abgeschlossen ist, können Sie nicht zur vorherigen Version der Datenbank-Engine zurückkehren. Wenn Sie zur vorherigen Version zurückkehren möchten, stellen Sie den ersten DB-Snapshot wieder her, um eine neue DB-Instance zu erstellen.

Sie steuern, wann Ihre DB-Instance auf eine neue Version aktualisiert werden soll, die von Amazon RDS unterstützt wird. Diese Kontrollebene hilft Ihnen, die Kompatibilität mit bestimmten Datenbankversionen aufrechtzuerhalten und neue Versionen mit Ihrer Anwendung zu testen, bevor sie produktiv bereitgestellt werden. Wenn Sie bereit sind, können Sie Versions-Upgrades zu den Zeiten durchführen, die am besten zu Ihrem Zeitplan passen.

Wenn Ihre DB-Instance die Lesereplikation verwendet, müssen Sie alle Read Replicas aktualisieren, bevor Sie die Quell-Instance aktualisieren.

Gültige Upgrade-Ziele finden

Wenn Sie den verwenden, AWS Management Console um eine DB-Instance zu aktualisieren, werden die gültigen Upgrade-Ziele für die DB-Instance angezeigt. Sie können auch den folgenden AWS CLI Befehl ausführen, um die gültigen Upgrade-Ziele für eine DB-Instance zu identifizieren:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version version_number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version version_number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Um beispielsweise die gültigen Upgrade-Ziele für eine MySQL-DB-Instance der Version 8.0.28 zu identifizieren, führen Sie den folgenden AWS CLI Befehl aus:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version 8.0.28 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version 8.0.28 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

MySQL-Versionsnummern

Die Versionsnummerierungssequenz für die RDS for MySQL-Datenbank-Engine hat entweder die Form major.minor.patch.yyyymmDD oder major.minor.patch, zum Beispiel 8.0.33.R2.20231201 oder 5.7.44. Das verwendete Format hängt von der Version der MySQL-Engine ab. Informationen zur Versionsnummerierung von RDS Extended Support finden Sie unterVersionsbenennung für Amazon RDS Extended Support.

wichtig

Die Hauptversionsnummer ist sowohl die Ganzzahl als auch der erste Bruchteil der Versionsnummer, z. B. 8.0. Ein Upgrade der Hauptversion erhöht den Hauptteil der Versionsnummer. Ein Upgrade von 5.7 .44 auf 8.0.33 ist beispielsweise ein Hauptversionsupgrade, wobei 5.7 und 8.0 die Hauptversionsnummern sind.

geringfügig

Die Nebenversionsnummer ist der dritte Teil der Versionsnummer, zum Beispiel die 33 in 8.0.33.

Patch

Der Patch ist der vierte Teil der Versionsnummer, zum Beispiel R2 in 8.0.33.R2. Eine RDS-Patch-Version enthält wichtige Korrekturen, die einer Nebenversion nach ihrer Veröffentlichung hinzugefügt werden.

YYYYMMDD

Das Datum ist der fünfte Teil der Versionsnummer, zum Beispiel 20231201 in 8.0.33.R2.20231201. Eine RDS-Datumsversion ist ein Sicherheitspatch, der wichtige Sicherheitsupdates enthält, die einer Nebenversion nach ihrer Veröffentlichung hinzugefügt wurden. Es enthält keine Korrekturen, die das Verhalten einer Engine ändern könnten.

In der folgenden Tabelle wird das Benennungsschema für RDS for MySQL Version 8.4 erklärt.

8.4 Nebenversion	Benennungsschema
≥ 3	Neue DB-Instances verwenden Major.Minor.Patch.YYMMDD, zum Beispiel 8.4.3.R2.20241201. Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 8.4.3.R2, bis Sie Ihre nächste Haupt- oder Nebenversion aktualisieren.

In der folgenden Tabelle wird das Benennungsschema für RDS for MySQL Version 8.0 erklärt.

8.0 Nebenversion	Benennungsschema
≥ 33	Neue DB-Instances verwenden Major.Minor.Patch.YYMMDD, zum Beispiel 8.0.33.R2.20231201. Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 8.0.33.R2, bis Sie Ihre nächste Haupt- oder Nebenversion aktualisieren.

8.0 Nebenversion	Benennungsschema
< 33	Bestehende DB-Instances verwenden major.minor.patch, beispielsweise 8.0.32.R2.

In der folgenden Tabelle wird das Benennungsschema für RDS for MySQL Version 5.7 erklärt.

5.7 Nebenversion	Benennungsschema
≥ 42	Neue DB-Instances verwenden Major.Minor.Patch.YYMMDD, zum Beispiel 5.7.42.R2.20231201. Bestehende DB-Instances verwenden möglicherweise major.minor.patch, z. B. 5.7.42.R2, bis Sie Ihre nächste Haupt- oder Nebenversion aktualisieren.

RDS-Versionsnummern in RDS für MySQL

RDS-Versionsnummern verwenden entweder das *major*.*minor*.*patch* oder das *major*.*minor*.*patch*.*YYYYMMDD* Benennungsschema. Die Versionen von Amazon RDS Extended Support verwenden das Benennungsschema für *minor-RDS*.*YYYYMMDD* Nebenversionen.

Eine RDS-Patch-Version enthält wichtige Korrekturen, die einer Nebenversion nach ihrer Veröffentlichung hinzugefügt werden. Eine RDS-Datumsversion (*YYYYMMDD*) ist ein Sicherheitspatch. Ein Sicherheitspatch enthält keine Fixes, die das Verhalten der Engine ändern könnten. Informationen zur Versionsnummerierung von RDS Extended Support finden Sie unterVersionsbenennung für Amazon RDS Extended Support.

Sie können die RDS-Versionsnummer Ihrer RDS for MySQL-Datenbank mit der folgenden SQL-Abfrage herausfinden:

```
mysql> select mysql.rds_version();
```

Wenn Sie beispielsweise eine RDS-Datenbank für MySQL 8.0.34 abfragen, wird die folgende Ausgabe zurückgegeben:

```
+---------------------+
```

```
| mysql.rds_version() |
+---------------------+
| 8.0.34.R2.20231201  |
+---------------------+
1 row in set (0.01 sec)
```

Hauptversions-Upgrades für RDS for MySQL

Amazon RDS unterstützt die folgenden direkten Upgrades für Hauptversionen des MySQL-Datenbank-Engine:

- MySQL 5.7 auf MySQL 8.0
- MySQL 8.0 auf MySQL 8.4

> **Note**
>
> Sie können nur DB-Instances der Versionen 5.7, 8.0 und 8.4 mit DB-Instance-Klassen der neuesten Generation und der aktuellen Generation erstellen.
> In einigen Fällen möchten Sie eine DB-Instance, die auf einer DB-Instance-Klasse der vorherigen Generation ausgeführt wird, auf eine DB-Instance mit einer höheren MySQL-Engine-Version aktualisieren. In einem solchen Fall ändern Sie zunächst die DB-Instance so, dass diese eine DB-Instance-Klasse der neuesten oder aktuellen Generation verwendet. Nachdem Sie dies getan haben, können Sie die DB-Instance so ändern, dass sie die höhere Version der MySQL-Datenbank-Engine verwendet. Weiterführende Informationen zu Amazon RDS-DB-Instance-Klassen finden Sie unter .

Themen

- Übersicht über Upgrades von MySQL-Hauptversionen
- Prüft vorab, ob Upgrades verfügbar sind
- Rollback nach einem fehlgeschlagenen Upgrade

Übersicht über Upgrades von MySQL-Hauptversionen

Hauptversions-Upgrades können Datenbankänderungen enthalten, die nicht mit vorhandenen Anwendungen rückwärts kompatibel sind. Folglich werden Hauptversion-Upgrades in Amazon RDS

nicht automatisch ausgeführt, sondern Sie müssen Ihre DB-Instance manuell ändern. Sie sollten jedes Upgrade gründlich testen, bevor Sie es auf Ihre Produktions-Instances anwenden.

Um ein Upgrade einer Hauptversion durchzuführen, führen Sie zunächst alle verfügbaren Betriebssystemupdates durch. Führen Sie nach Abschluss der Betriebssystemupdates ein Upgrade auf jede Hauptversion durch, z. B. 5.7 auf 8.0 und dann 8.0 auf 8.4. Informationen zum Upgrade eines RDS for MySQL Multi-AZ-DB-Clusters finden Sie unterAktualisierung der Engine-Version eines Multi-AZ-DB-Clusters für Amazon RDS. Vor dem 24. April 2014 erstellte MySQL-DB-Instances zeigen verfügbare Updates für das Betriebssystem an, bis es angewendet wurde. Weitere Informationen zu Betriebssystem-Updates finden Sie unter Updates auf einen anwenden.

Während des Upgrades einer Hauptversion von MySQL führt Amazon RDS die MySQL-Binärdatei `mysql_upgrade` aus, um die Tabellen zu aktualisieren, falls erforderlich. Außerdem leert Amazon RDS während des Upgrades einer Hauptversion die Tabellen `slow_log` und `general_log`. Speichern Sie die Protokollinhalte vor dem Upgrade einer Hauptversion, um die Protokollinformationen zu erhalten.

MySQL-Hauptversions-Upgrades sind normalerweise in etwa 10 Minuten abgeschlossen. Einige Aktualisierungen können aufgrund der Klassengröße der DB-Instance länger dauern, oder weil die Instance bestimmten Richtlinien in nicht entsprich Bewährte Methoden für Amazon RDS. Wenn Sie eine DB-Instance von der Amazon RDS-Konsole aktualisieren, zeigt der Status der DB-Instance an, wann das Upgrade abgeschlossen ist. Wenn Sie ein Upgrade mit AWS Command Line Interface (AWS CLI) durchführen, verwenden Sie den describe-db-instancesBefehl und überprüfen Sie den `Status` Wert.

Prüft vorab, ob Upgrades verfügbar sind

Amazon RDS führt vor dem Upgrade Vorabprüfungen durch, um nach Inkompatibilitäten zu suchen. Diese Inkompatibilitäten variieren je nach der MySQL-Version, auf die aktualisiert wird.

Die Vorabprüfungen enthalten einige Prüfungen, die in MySQL enthalten sind, und einige spezifische Prüfungen, die vom Amazon RDS-Team erstellt wurden. Informationen zu den von MySQL bereitgestellten Vorabprüfungen finden Sie unter Upgrade Checker-Dienstprogramm.

Die Vorabprüfungen werden ausgeführt, bevor die DB-Instance aufgrund des Upgrades angehalten wird. Sie verursachen also keine Ausfallzeiten. Wird während der Vorabprüfungen eine Inkompatibilität entdeckt, bricht Amazon RDS automatisch das Upgrade ab, ehe die DB-Instance angehalten wird. Amazon RDS generiert auch ein Ereignis für die Inkompatibilität. Weitere Informationen über Amazon RDS-Ereignisse finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.

Amazon RDS zeichnet detaillierte Informationen zu allen Inkompatibilitäten in der Protokolldatei au `PrePatchCompatibility.log`. In den meisten Fällen enthalten die Protokolleinträge einen Link zur MySQL-Dokumentation mit Informationen zur Lösung des Inkompatibilitätsproblems. Weitere Informationen zum Anzeigen von Protokolldateien finden Sie unter Anzeigen und Auflisten von Datenbank-Protokolldateien.

Aufgrund der Art der Vorabprüfungen analysieren sie die Objekte in Ihrer Datenbank. Diese Analyse verbraucht Ressourcen und verlängert die Zeit, die für die Durchführung des Upgrades benötigt wird.

Themen

- Vorabprüfungen für Upgrades von MySQL 8.0 auf 8.4
- Vorabprüfung bei Upgrades von MySQL 5.7 auf 8.0

Vorabprüfungen für Upgrades von MySQL 8.0 auf 8.4

MySQL 8.4 beinhaltet eine Reihe von Inkompatibilitäten mit MySQL 8.0. Diese Inkompatibilitäten können bei einem Upgrade von MySQL 8.0 auf MySQL 8.4 zu Problemen führen. Damit das Upgrade erfolgreich durchgeführt werden kann, sind einige Vorbereitungsmaßnahmen auf Ihrer Datenbank durchzuführen. Im Folgenden finden Sie eine allgemeine Liste dieser Inkompatibilitäten:

- Es darf keine Tabellen geben, die veraltete Datentypen oder Funktionen verwenden.
- Es darf keine Auslöser mit fehlenden oder leeren Definern oder ungültigen Erstellungskontexten geben.
- Es darf keine Verletzungen von Schlüsselwörtern oder reservierten Wörtern geben. Einige Schlüsselwörter sind möglicherweise in MySQL 8.4 reserviert, die zuvor nicht reserviert waren.

  Weitere Informationen finden Sie unter Schlüsselwörter und reservierte Wörter in der MySQL-Dokumentation.
- In der MySQL `mysql` 8.0-Systemdatenbank dürfen keine Tabellen vorhanden sein, die denselben Namen haben wie eine Tabelle, die vom MySQL 8.4-Datenwörterbuch verwendet wird.
- Es dürfen keine veralteten SQL-Modi in Ihrer `sql_mode`-Systemvariableneinstellung definiert sein.
- Es darf keine Tabellen oder gespeicherte Prozeduren mit einzelnen `ENUM`- oder `SET`-Spaltenelementen geben, deren Länge 255 Zeichen oder 1020 Bytes überschreitet.
- Ihre MySQL 8.0-Installation darf keine Funktionen verwenden, die in MySQL 8.4 nicht unterstützt werden.

Weitere Informationen finden Sie unter In MySQL 8.4 entfernte Funktionen in der MySQL-Dokumentation.

- Es darf keine Namen für Fremdschlüsseleinschränkungen mit mehr als 64 Zeichen geben.
- Informationen zur verbesserten Unicode-Unterstützung finden Sie in den folgenden Informationen:
  - Erwägen Sie, Objekte, die den `utf8mb3` Zeichensatz verwenden, in den `utf8mb4` Zeichensatz umzuwandeln. Der `utf8mb3`-Zeichensatz ist veraltet.
  - Erwägen Sie `utf8mb4` die Verwendung von für Zeichensatzreferenzen`utf8`, da dies derzeit ein Alias für den `utf8mb3` Zeichensatz `utf8` ist. Wenn möglich, wechseln Sie `utf8mb4` zuerst `utf8` zu und aktualisieren Sie dann Ihre Datenbank.
  - Da bei älteren Clients möglicherweise ein unbekannter Zeichensatzfehler für angezeigt wird`utf8mb3`, aktualisieren Sie Ihre Datenbankclients, bevor Sie Ihre Datenbank aktualisieren.

  Weitere Informationen finden Sie unter Der utf8mb3-Zeichensatz (UTF-8-Unicode-Kodierung mit 3 Bytes) in der MySQL-Dokumentation.

  Um die Zeichensätze zu ändern, können Sie manuell eine Sicherung, Wiederherstellung und Replikation Ihrer Datenbank durchführen. Oder Sie können Amazon RDS Blue/Green Deployments verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Wenn Sie ein Upgrade von MySQL 8.0 auf 8.4 starten, führt Amazon RDS automatisch Vorabprüfungen durch, um diese Inkompatibilitäten zu erkennen. Informationen zum Upgrade auf MySQL 8.4 finden Sie unter Upgrading MySQL in der MySQL-Dokumentation.

Diese Vorabprüfungen müssen durchgeführt werden. Sie können nicht ausgelassen werden. Die Vorabprüfungen bieten folgende Vorteile:

- Sie können ungeplante Ausfallzeiten während des Upgrades vermeiden.
- Wenn es Inkompatibilitäten gibt, verhindert Amazon RDS das Upgrade und stellt Ihnen ein Protokoll mit Informationen zu den Inkompatibilitäten bereit. Anschließend können Sie das Protokoll verwenden, um Ihre Datenbank auf das Upgrade auf MySQL 8.4 vorzubereiten, indem Sie die Inkompatibilitäten reduzieren. Ausführliche Informationen zum Entfernen von Inkompatibilitäten finden Sie unter Vorbereiten Ihrer Installation für das Upgrade in der MySQL-Dokumentation.

Vorabprüfung bei Upgrades von MySQL 5.7 auf 8.0

MySQL 8.0 ist in vielen Punkten nicht mit MySQL 5.7 kompatibel. Diese Inkompatibilitäten können bei einem Upgrade von MySQL 5.7 auf MySQL 8.0 Probleme verursachen. Damit das Upgrade erfolgreich durchgeführt werden kann, sind einige Vorbereitungsmaßnahmen auf Ihrer Datenbank durchzuführen. Im Folgenden finden Sie eine allgemeine Liste dieser Inkompatibilitäten:

- Es darf keine Tabellen geben, die veraltete Datentypen oder Funktionen verwenden.
- Es darf keine verwaisten FRM-Dateien geben.
- Es darf keine Auslöser mit fehlenden oder leeren Definern oder ungültigen Erstellungskontexten geben.
- Es darf keine partitionierte Tabelle mit einer Speicher-Engine geben, für die es keine native Partitionierungsunterstützung gibt.
- Es darf keine Verletzungen von Schlüsselwörtern oder reservierten Wörtern geben. Einige Schlüsselwörter sind in MySQL 8.0 möglicherweise reserviert, die zuvor nicht reserviert waren.

  Weitere Informationen finden Sie unter Schlüsselwörter und reservierte Wörter in der MySQL-Dokumentation.
- Es darf keine Tabellen in der MySQL 5.7 `mysql`-Systemdatenbank geben, die denselben Namen wie eine Tabelle haben, die vom MySQL 8.0-Daten-Dictionary verwendet wird.
- Es dürfen keine veralteten SQL-Modi in Ihrer `sql_mode`-Systemvariableneinstellung definiert sein.
- Es darf keine Tabellen oder gespeicherte Prozeduren mit einzelnen `ENUM`- oder `SET`-Spaltenelementen geben, deren Länge 255 Zeichen oder 1020 Bytes überschreitet.
- Vor dem Upgrade auf MySQL 8.0.13 oder höher darf es keine Tabellenpartitionen innerhalb von freigegebenen InnoDB-Tabellenräumen geben.
- Es darf keine Abfragen oder gespeicherten Programmdefinitionen aus MySQL 8.0.12 oder früher geben, die `ASC` oder `DESC`-Qualifizierer für `GROUP BY`-Klauseln verwenden.
- Ihre MySQL 5.7-Installation darf keine Funktionen verwenden, die in MySQL 8.0 nicht unterstützt werden.

  Weitere Informationen finden Sie unter In MySQL 8.0 entfernte Funktionen in der MySQL-Dokumentation.
- Es darf keine Namen für Fremdschlüsseleinschränkungen mit mehr als 64 Zeichen geben.
- Informationen zur verbesserten Unicode-Unterstützung finden Sie in den folgenden Informationen:

- Erwägen Sie, Objekte, die den `utf8mb3` Zeichensatz verwenden, in den `utf8mb4` Zeichensatz umzuwandeln. Der `utf8mb3`-Zeichensatz ist veraltet.
- Erwägen Sie `utf8mb4` die Verwendung von für Zeichensatzreferenzen`utf8`, da dies derzeit ein Alias für den `utf8mb3` Zeichensatz `utf8` ist. Wenn möglich, wechseln Sie `utf8mb4` zuerst `utf8` zu und aktualisieren Sie dann Ihre Datenbank.
- Da bei älteren Clients möglicherweise ein unbekannter Zeichensatzfehler für angezeigt wird`utf8mb3`, aktualisieren Sie Ihre Datenbankclients, bevor Sie Ihre Datenbank aktualisieren.

Weitere Informationen finden Sie unter Der utf8mb3-Zeichensatz (UTF-8-Unicode-Kodierung mit 3 Bytes) in der MySQL-Dokumentation.

Um die Zeichensätze zu ändern, können Sie manuell eine Sicherung, Wiederherstellung und Replikation Ihrer Datenbank durchführen. Oder Sie können Amazon RDS Blue/Green Deployments verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Wenn Sie ein Upgrade von MySQL 5.7 auf 8.0 starten, führt Amazon RDS Vorabprüfungen durch, um eventuelle Inkompatibilitäten zu entdecken. Informationen zum Ausführen von Upgrades auf MySQL 8.0 finden Sie unter Ausführen von MySQL Upgrades in der MySQL-Dokumentation.

Diese Vorabprüfungen müssen durchgeführt werden. Sie können nicht ausgelassen werden. Die Vorabprüfungen bieten folgende Vorteile:

- Sie können ungeplante Ausfallzeiten während des Upgrades vermeiden.
- Wenn es Inkompatibilitäten gibt, verhindert Amazon RDS das Upgrade und stellt Ihnen ein Protokoll mit Informationen zu den Inkompatibilitäten bereit. Sie können das Protokoll für die Vorbereitung Ihrer Datenbank auf das Upgrade auf MySQL 8.0 verwenden, indem Sie die Inkompatibilitäten reduzieren. Detaillierte Informationen zum Entfernen von Inkompatibilitäten finden Sie unter Vorbereiten Ihrer Installation auf ein Upgrade in der MySQL-Dokumentation und unter Upgrade auf MySQL 8.0? Dies müssen Sie wissen ... im MySQL Server Blog.

Rollback nach einem fehlgeschlagenen Upgrade

Wenn Sie eine DB-Instance von MySQL Version 5.7 auf MySQL Version 8.0 oder von MySQL Version 8.0 auf 8.4 aktualisieren, kann das Upgrade fehlschlagen. Insbesondere kann es scheitern, wenn das Datenwörterbuch Inkompatibilitäten enthält, die von den Vorprüfungen nicht erfasst wurden. In diesem Fall kann die Datenbank in der neuen MySQL-Version 8.0 oder 8.4 nicht

erfolgreich gestartet werden. Zu diesem Zeitpunkt macht Amazon RDS die für das Upgrade durchgeführten Änderungen rückgängig. Nach dem Rollback läuft auf der MySQL-DB-Instance die Originalversion:

- MySQL Version 8.0 (für ein Rollback von MySQL 8.4)
- MySQL Version 5.7 (für ein Rollback von MySQL 8.0)

Wenn ein Upgrade fehlschlägt und rückgängig gemacht wird, generiert Amazon RDS ein Ereignis mit der Ereignis-ID RDS-EVENT-0188.

In der Regel schlägt ein Upgrade fehl, da es Inkompatibilitäten in den Metadaten zwischen den Datenbanken in Ihrer DB-Instance und der Ziel-MySQL-Version gibt. Wenn ein Upgrade fehlschlägt, können Sie die Details zu diesen Inkompatibilitäten in der `upgradeFailure.log`-Datei einsehen. Beheben Sie die Inkompatibilitäten, bevor Sie erneut versuchen, ein Upgrade durchzuführen.

Während eines erfolglosen Upgrade-Versuchs und Rollbacks wird Ihre DB-Instance neu gestartet. Alle ausstehenden Parameteränderungen werden während des Neustarts angewendet und bleiben nach dem Rollback bestehen.

Weitere Informationen zum Upgrade auf MySQL 8.0 finden Sie in den folgenden Themen der MySQL-Dokumentation:

- Vorbereiten Ihrer Installation für das Upgrade
- Upgrade auf MySQL 8.0? Hier ist was Sie wissen müssen…

Weitere Informationen zum Upgrade auf MySQL 8.4 finden Sie unter Vorbereiten Ihrer Installation für das Upgrade in der MySQL-Dokumentation.

Ein SQL Upgrade RDS für mein Gerät testen

Ehe Sie ein Upgrade einer Hauptversion auf Ihrer DB-Instance durchführen, sollten Sie sorgfältig prüfen, ob Ihre Datenbank mit der neuen Version kompatibel ist. Darüber hinaus sollten Sie die Kompatibilität aller Anwendungen mit der neuen Version testen, die auf die Datenbank zugreifen. Wir empfehlen Ihnen folgendes Vorgehen.

Um ein Hauptversions-Upgrade zu testen

1. Informieren Sie sich in der Upgrade-Dokumentation von Oracle über die neue Version der Datenbank-Engine, um zu prüfen, ob es Kompatibilitätsprobleme geben könnte, die sich auf Ihre Datenbank oder Anwendungen auswirken könnten:
   - Änderungen in My SQL 5.7
   - Änderungen in My SQL 8.0
   - Änderungen in My SQL 8.4
2. Wenn Ihre DB-Instance Mitglied einer benutzerdefinierten DB-Parametergruppe ist, müssen Sie eine neue DB-Parametergruppe mit Ihren vorhandenen Einstellungen erstellen, die mit der neuen Hauptversion kompatibel ist. Geben Sie die neue DB-Parametergruppe an, wenn Sie Ihre Test-Instance aktualisieren, damit Ihr Upgrade-Test sicherstellt, dass sie ordnungsgemäß funktioniert. Weitere Informationen über das Erstellen einer Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.
3. Erstellen Sie einen DB-Snapshot der zu aktualisierenden DB-Instance. Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
4. Stellen Sie den DB-Snapshot wieder her, um eine neue Test-DB-Instance zu erstellen. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.
5. Ändern Sie diese neue Test-DB-Instance, um sie auf die neue Version upzugraden. Verwenden Sie dazu eine der folgenden Methoden. Wenn Sie in Schritt 2 eine neue Parametergruppe erstellt haben, geben Sie diese Parametergruppe an.
6. Beurteilen Sie den Speicherplatz, den die upgegradete Instance verwendet, um zu bestimmen, ob das Upgrade zusätzlichen Speicherplatz benötigt.
7. Führen Sie so viele Qualitätssicherungstests mit der upgegradeten DB-Instance durch, wie nötig, um sicherzustellen, dass Ihre Datenbank und Anwendung mit der neuen Version korrekt ausgeführt werden. Führen Sie alle nötigen neuen Tests aus, um die Auswirkungen von Kompatibilitätsproblemen zu bewerten, die Sie in Schritt 1 bestimmt haben. Testen Sie alle gespeicherten Prozeduren und Funktionen. Leiten Sie Testversionen Ihrer Anwendungen an die aktualisierte DB-Instance weiter.
8. Wenn alle Tests erfolgreich sind, führen Sie das Upgrade für Ihre Produktions-DB-Instance durch. Wir empfehlen, dass Sie keine Schreiboperationen auf der DB-Instance zulassen, bis Sie bestätigen können, dass alles richtig ausgeführt wird.

Upgraden einer MySQL-DB-Instance

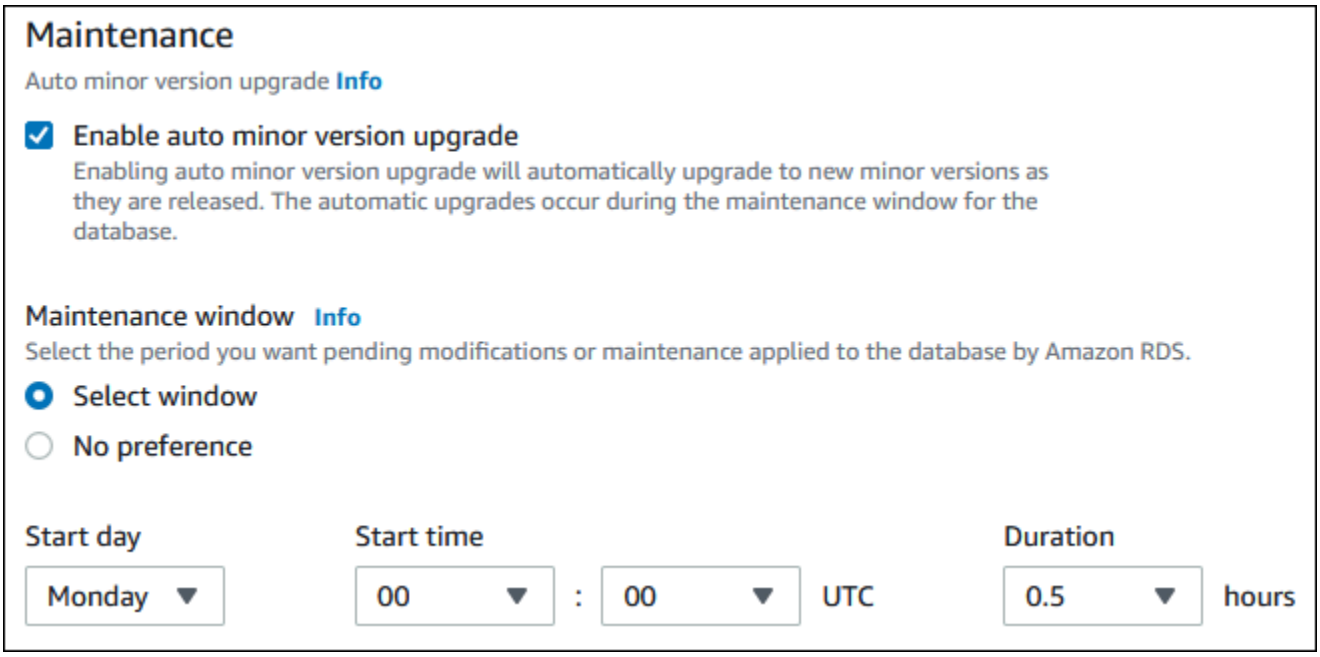
Informationen über das manuelle oder automatische Upgraden einer MySQL-DB-Instance finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Automatische Upgrades für kleinere Versionen für RDS für My SQL

Wenn Sie beim Erstellen oder Ändern einer DB-Instance die folgenden Einstellungen angeben, können Sie Ihre DB-Instance automatisch aktualisieren lassen.

- Die Einstellung Automatisches Upgrade der Nebenversion ist aktiviert.
- Die Einstellung Aufbewahrungszeitraum für Sicherungen beträgt mehr als 0.

In der befinden AWS Management Console sich diese Einstellungen unter Zusätzliche Konfiguration. Die folgende Abbildung zeigt die Auto minor version upgrade (Upgrade einer Unterversion automatisch durchführen)-Einstellung.



Weitere Informationen zu diesen Einstellungen finden Sie unter Einstellungen für DB-Instances.

RDSBei einigen Versionen von My SQL Major wird in einigen AWS-Regionen Fällen eine Nebenversion RDS als automatische Upgrade-Version bezeichnet. Nachdem eine Nebenversion von Amazon getestet und genehmigt wurdeRDS, erfolgt das Upgrade der Nebenversion automatisch während Ihres Wartungsfensters. RDSlegt nicht automatisch neuere veröffentlichte Nebenversionen

als automatische Upgrade-Version fest. Bevor RDS eine neuere automatische Upgrade-Version bestimmt, werden mehrere Kriterien berücksichtigt, wie beispielsweise die folgenden:

- Bekannte Sicherheitsprobleme
- Fehler in der Version „Meine SQL Community“
- Gesamtflottenstabilität seit Erscheinen der Minor-Version

Sie können den folgenden AWS CLI Befehl ausführen, um die aktuelle Zielversion für ein automatisches kleines Upgrade für eine bestimmte SQL Nebenversion von My in einer bestimmten Version zu ermitteln AWS-Region.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version minor_version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version minor_version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Der folgende AWS CLI Befehl bestimmt beispielsweise das automatische kleinere Upgrade-Ziel für Meine SQL Nebenversion 8.0.11 in den USA Ost (Ohio) AWS-Region (us-east-2).

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
```

```
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Ihre Ausgabe sieht Folgendem ähnlich.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

In diesem Beispiel ist der `AutoUpgrade` Wert `True` für Meine SQL Version 8.0.23. Das automatische kleinere Upgrade-Ziel ist also Meine SQL Version 8.0.23, was in der Ausgabe hervorgehoben ist.

Eine My SQL DB-Instance wird während Ihres Wartungsfensters automatisch aktualisiert, wenn die folgenden Kriterien erfüllt sind:

- Die Einstellung Automatisches Upgrade der Nebenversion ist aktiviert.
- Die Einstellung Aufbewahrungszeitraum für Sicherungen beträgt mehr als 0.

- Die DB-Instance führt eine Minor-Version der DB-Engine aus, die niedriger ist als die aktuelle Minor-Version des automatischen Upgrades.

Weitere Informationen finden Sie unter Automatisches Upgraden der Engine-Unterversion.

Verwendung einer Read Replica zur Reduzierung von Ausfallzeiten beim Upgrade einer RDS for MySQL-Datenbank

In den meisten Fällen ist eine Blau/Grün-Bereitstellung die beste Option, um Ausfallzeiten beim Upgrade einer MySQL-DB-Instance zu reduzieren. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.

Wenn Sie keine Blau/Grün-Bereitstellung verwenden können und Ihre MySQL-DB-Instance aktuell von einer Produktionsanwendung genutzt wird, können Sie mit dem folgenden Verfahren die Datenbankversion Ihrer DB-Instance aktualisieren. Dieses Verfahren kann die Ausfallzeiten Ihrer Anwendung reduzieren.

Mithilfe einer Read Replica können Sie die meisten Wartungsschritte im Voraus durchführen und die erforderlichen Änderungen während des tatsächlichen Ausfalls minimieren. Mit dieser Technik können Sie die neue DB-Instance testen und vorbereiten, ohne Änderungen an Ihrer bestehenden DB-Instance vorzunehmen.

Im Folgenden wird ein Beispiel für ein Upgrade der MySQL Version 5.7 auf MySQL Version 8.0 gezeigt. Sie können die gleichen allgemeinen Schritte für Upgrades auf andere Hauptversionen durchführen. Sie können die gleichen allgemeinen Schritte für Upgrades auf andere Hauptversionen durchführen.

> **Note**
>
> Wenn Sie von MySQL Version 5.7 auf MySQL Version 8.0 oder von MySQL Version 8.0 auf MySQL Version 8.4 aktualisieren, führen Sie die Vorprüfungen durch, bevor Sie das Upgrade durchführen. Weitere Informationen erhalten Sie unter Vorabprüfung bei Upgrades von MySQL 5.7 auf 8.0 und Vorabprüfungen für Upgrades von MySQL 8.0 auf 8.4.

So führen Sie ein Upgrade einer MySQL-Datenbank durch, während eine DB-Instance verwendet wird

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Erstellen Sie eine Read Replica Ihrer MySQL 5.7-DB-Instance. Dieser Prozess erstellt eine aktualisierbare Kopie Ihrer Datenbank. Andere Read Replicas der DB-Instance könnten ebenfalls vorhanden sein.

   a. Wählen Sie in der Konsole Datenbanken und dann die DB-Instance aus, die Sie upgraden möchten.

   b. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.

   c. Geben Sie für die Read Replica einen Wert im Feld DB-Instance-Kennung ein und stellen Sie sicher, dass der Eintrag unter DB-Instance-Klasse) und die anderen Einstellungen mit Ihrer MySQL 5.7-DB-Instance übereinstimmen.

   d. Wählen Sie Read Replica erstellen aus.
3. (Optional) Wenn die Read Replica erstellt wurde und der Status Verfügbar anzeigt, konvertieren Sie die Read Replica in eine Multi-AZ-Bereitstellung und aktivieren Sie Sicherungen.

   Standardmäßig wird eine Read Replica mit deaktivierten Backups erstellt. Da die Read Replica letztendlich zur Produktions-DB-Instance wird, empfiehlt es sich, eine Multi-AZ-Bereitstellung zu konfigurieren und Backups zu aktivieren.

   a. Wählen Sie in der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade erstellt haben.

   b. Wählen Sie Ändern aus.

   c. Für die Multi-AZ-Bereitstellungwählen Sie Standby-Instance erstellen.

   d. Wählen Sie unter Backup Retention Period (Aufbewahrungszeitraum für Backups) einen positiven Wert größer als null aus, z. B. 3 Tage. Klicken Sie anschließend auf Continue (Weiter).

   e. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.

   f. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.
4. Wenn der Read Replica-Status Verfügbar anzeigt, aktualisieren Sie die Read Replica auf MySQL 8.0:

a. Wählen Sie in der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade erstellt haben.

b. Wählen Sie Ändern aus.

c. Wählen Sie im Feld DB-Engine-Version die gewünschte Version von MySQL 8.0 für das Upgrade aus und klicken Sie auf Weiter.

d. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.

e. Wählen Sie Modify DB instance (DB-Instance ändern) aus, um das Upgrade zu starten.

5. Wenn das Upgrade abgeschlossen ist und der Status Verfügbar anzeigt, stellen Sie sicher, dass sich die aktualisierte Read Replica up-to-date mit der Quell-DB-Instance von MySQL 5.7 befindet. Stellen Sie zur Überprüfung eine Verbindung mit dem Lesereplikat her und führen Sie den Befehl `SHOW REPLICA STATUS` aus. Wenn das `Seconds_Behind_Master` Feld lautet`0`, ist die Replikation. up-to-date

> **Note**
>
> Frühere Versionen von MySQL verwenden `SHOW SLAVE STATUS` anstelle von `SHOW REPLICA STATUS`. Wenn Sie vor 8.0.23 eine MySQL-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

6. (Optional) Erstellen Sie eine Read Replica Ihrer Read Replica.

   Wenn Sie möchten, dass die DB-Instance eine Read Replica hat, nachdem sie auf eine eigenständige DB-Instance hochgestuft wurde, können Sie jetzt die Read Replica erstellen.

   a. Wählen Sie auf der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade aktualisiert haben.

   b. Wählen Sie unter Aktionen Create read replica (Read Replica erstellen) aus.

   c. Geben Sie für die Read Replica einen Wert im Feld DB-Instance-Kennung ein und stellen Sie sicher, dass der Eintrag unter DB-Instance-Klasse) und die anderen Einstellungen mit Ihrer MySQL 5.7-DB-Instance übereinstimmen.

   d. Wählen Sie Read Replica erstellen aus.

7. (Optional) Konfigurieren Sie eine benutzerdefinierte DB-Parametergruppe für die Read Replica.

Wenn Sie möchten, dass die DB-Instance eine benutzerdefinierte Parametergruppe verwendet, nachdem sie zu einer eigenständigen DB-Instance hochgestuft wurde, können Sie die DB-Parametergruppe erstellen und sie jetzt dem Lesereplikat zuordnen kann.

a. Erstellen Sie eine benutzerdefinierte DB-Parametergruppe für MySQL 8.0. Detaillierte Anweisungen finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS.

b. Ändern Sie die Parameter, die Sie in der gerade erstellten DB-Parametergruppe ändern möchten. Detaillierte Anweisungen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

c. Wählen Sie in der Konsole Datenbankenund dann die Read Replica aus.

d. Wählen Sie Ändern aus.

e. Wählen Sie für die DB-Parametergruppe die soeben erstellte MySQL 8.0 DB-Parametergruppe aus, und klicken Sie dann auf Weiter.

f. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.

g. Wählen Sie Modify DB instance (DB-Instance ändern) aus, um das Upgrade zu starten.

8. Machen Sie Ihre MySQL 8.0 Read Replica zu einer eigenständigen DB-Instance.

> **Important**
>
> Wenn Sie Ihr Lesereplikat von MySQL 8.0 auf eine eigenständige DB-Instance hochstufen, handelt es sich nicht mehr um ein Replikat der DB-Instance von MySQL 5.7. Wir empfehlen, dass Sie Ihre MySQL 8.0 Read Replica während eines Wartungsfensters hochstufen, wenn sich Ihre MySQL-5.7 Quell-DB-Instance im schreibgeschützten Modus befindet und alle Schreiboperationen ausgesetzt sind. Wenn die Aktion abgeschlossen ist, können Sie Ihre Schreiboperationen an die aktualisierte MySQL 8.0 DB-Instance weiterleiten, um sicherzustellen, dass keine Schreiboperationen verloren gehen. Zusätzlich empfehlen wir, dass Sie, bevor Sie die MySQL 8.0 Read Replica hochstufen, alle erforderlichen Data Definition Language (DDL)-Operationen auf der MySQL 8.0 Read Replica ausführen. Ein Beispiel hierfür ist das Erstellen von Indizes. Dieser Ansatz vermeidet negative Auswirkungen auf die Leistung der MySQL 8.0 Read Replica, nachdem sie hochgestuft wurde. Gehen Sie folgendermaßen vor, um ein Lesereplikat hochzustufen.

a. Wählen Sie auf der Konsole Datenbanken und dann die Read Replica aus, die Sie gerade aktualisiert haben.
b. Wählen Sie für Actions (Aktionen) Promote (Hochstufen) aus.
c. Klicken Sie auf Yes (Ja), um automatische Sicherungen für die Lesereplikat-Instance zu aktivieren. Weitere Informationen finden Sie unter Einführung in Backups.
d. Klicken Sie auf Continue (Fortfahren).
e. Klicken Sie auf Read Replica hochstufen.

9. Sie haben jetzt eine upgegradete Version Ihrer MySQL-Datenbank. An dieser Stelle können Sie Ihre Anwendungen auf die neue MySQL 8.0 DB-Instance verweisen.

Aktualisierung einer MySQL-DB-Snapshot-Engine-Version

Mit Amazon RDS können Sie einen DB-Snapshot für das Speicher-Volume Ihrer MySQL-DB-Instance erstellen. Wenn Sie einen DB-Snapshot erstellen, basiert der Snapshot auf der Engine-Version, die von Ihrer DB-Instance verwendet wird. Sie können die Engine-Version für Ihre DB-Snapshots aktualisieren.

Für RDS for MySQL können Sie einen Snapshot der Version 5.7 auf Version 8.0 oder einen Snapshot der Version 8.0 auf Version 8.4 aktualisieren. Sie können verschlüsselte oder unverschlüsselte DB-Snapshots aktualisieren.

Verwenden Sie das folgende AWS CLI Beispiel, um die verfügbaren Engine-Versionen für Ihren RDS MySQL MySQL-DB-Snapshot anzuzeigen.

```
aws rds describe-db-engine-versions --engine mysql --include-all --engine-
version example-engine-version --query "DBEngineVersions[*].ValidUpgradeTarget[*].
{EngineVersion:EngineVersion}" --output text
```

Wenn Sie keine Ergebnisse für Ihren Snapshot sehen, ist Ihre Engine-Version möglicherweise veraltet. Wenn Ihre Engine-Version veraltet ist, empfehlen wir Ihnen, ein Upgrade auf das neueste Upgrade-Ziel für die Hauptversion oder auf eines der anderen verfügbaren Upgrade-Ziele für diese Version durchzuführen. Weitere Informationen finden Sie unter Upgrade-Optionen für DB-Snapshots mit nicht unterstützten Engine-Versionen für RDS for MySQL.

Wenn Sie einen DB-Snapshot wiederherstellen, der auf eine neue Engine-Version aktualisiert wurde, sollten Sie prüfen, ob das Upgrade erfolgreich durchgeführt wurde. Weitere Informationen zu größeren Versionsaktualisierungen finden Sie unter the section called “Upgrades der MySQL-DB-Engine”. Informationen zum Wiederherstellen eines DB-Snapshots finden Sie unter the section called “Wiederherstellung auf einer DB-Instance”.

> **Note**
>
> Automatisierte DB-Snapshots, die während des automatisierten Backup-Vorgangs erstellt wurden, können nicht aktualisiert werden.

Sie können einen DB-Snapshot mithilfe der AWS Management Console AWS CLI, oder RDS-API aktualisieren.

Console

Gehen Sie wie folgt vor AWS Management Console, um eine DB-Snapshot-Engine-Version mithilfe von zu aktualisieren.

So führen Sie ein Upgrade eines DB-Snapshots durch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den Snapshot für die Aktualisierung aus.
4. Wählen Sie unter Actions (Aktionen) die Option Upgrade Snapshot (Snapshot aktualisieren). Die Seite Upgrade snapshot (Snapshot aktualisieren) erscheint.
5. Wählen Sie zum Aktualisieren New engine version (Neue Engine-Version).
6. Wählen Sie Save changes (Änderungen speichern), um den Snapshot zu aktualisieren.

   Während des Upgrades werden alle Snapshot-Aktionen für diesen DB-Snapshot deaktiviert. Außerdem ändert sich der Status des DB-Snapshots von „Verfügbar“ zu „Aktualisierung“ und nach Abschluss des Vorgangs zu „Aktiv“. Wenn der DB-Snapshot aufgrund von Problemen mit dem Snapshot nicht aktualisiert werden kann, ändert sich der Status in Nicht verfügbar. Sie können den Snapshot aus diesem Zustand nicht wiederherstellen.

   > **Note**
   >
   > Wenn die Aktualisierung des DB-Snapshots fehlschlägt, wird der Snapshot wieder in seinen ursprünglichen Zustand zurückgebracht.

AWS CLI

Um einen DB-Snapshot auf eine neue Version der Datenbank-Engine zu aktualisieren, führen Sie den AWS CLI modify-db-snapshotBefehl aus.

Optionen

- `--db-snapshot-identifier`: die Kennung des DB-Snapshots, für den das Upgrade durchgeführt werden soll Die Kennung muss ein eindeutiger Amazon-Ressourcenname (ARN) sein. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.

- `--engine-version`: Die Engine-Version, auf die das Upgrade des DB-Snapshots durchgeführt werden soll

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot \

    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

Amazon RDS API

Um einen DB-Snapshot auf eine neue Version der Datenbank-Engine zu aktualisieren, rufen Sie den DBSnapshot RDS-API-Vorgang Modify auf.

Parameter

- `DBSnapshotIdentifier`: die Kennung des DB-Snapshots, für den das Upgrade durchgeführt werden soll Die Kennung muss ein eindeutiger Amazon-Ressourcenname (ARN) sein. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.
- `EngineVersion`: Die Engine-Version, auf die das Upgrade des DB-Snapshots durchgeführt werden soll

Upgrade-Optionen für DB-Snapshots mit nicht unterstützten Engine-Versionen für RDS for MySQL

Die folgende Tabelle zeigt, auf welche Engine-Versionen Sie von einer nicht unterstützten Engine-Version für RDS for MySQL-DB-Snapshots aktualisieren können.

> **Note**
>
> Möglicherweise müssen Sie Ihren DB-Snapshot mehrmals aktualisieren, um auf die von Ihnen gewählte Engine-Version zu aktualisieren.

Version der DB-Snapshot-Engine	Engine-Versionen sind für ein Upgrade verfügbar
5.5.8	5,5,62, 5,6,51
5.5.12	5,5,62, 5,6,51
5,5,20	5,5,62, 5,6,51
5.5.23	5,5,62, 5,6,51
5,5,25a	5.5.62, 5.6,51
5,5,27	5,5,62, 5,6,51
5,5,31	5,5,62, 5,6,51
5,5,33	5,5,62, 5,6,51
5,5,37	5,5,62, 5,6,51
5,5,38	5,5,62, 5,6,51
5,5,40	5,5,62, 5,6,51
5,5,40a	5.5.62, 5.6,51
5,5,40 b	5.5.62, 5.6,51
5,5,41	5,5,62, 5,6,51
5,5,42	5,5,62, 5,6,51
5.5.59	5,5,62, 5,6,51

Version der DB-Snapshot-Engine	Engine-Versionen sind für ein Upgrade verfügbar
5.6.12	5.6.51, 5,7,44
5.6.13	5.6.51, 5,7,44
5.6.17	5,6,51, 5,7,44
5.6,19	5,6,51, 5,7,44
5.6.19a	5.6.51, 5.7.44
5.6.19b	5.6.51, 5.7.44
5.6.21	5,6,51, 5,7,44
5.6.21b	5.6.51, 5.7.44
5.6.22	5,6,51, 5,7,44
5.6,23	5,6,51, 5,7,44
5.6,27	5,6,51, 5,7,44
5.6.27a	5.6.51, 5.7.44
5.7.10	5.7.44, 5.7.44-Rds.20240408, 5.7.44-Rds.20240529, 5.7.44-Rds.20250103, 5.7.44-Rds.20250213, 8.0.32, 8.0.33, 8.0.34, 8.0.35, 8.0.36, 8.0.37, 8.0.39, 8.0.40, 8.0.41
5.7.11	5.7.44, 5.7.44-Rds.20240408, 5.7.44-Rds.20240529, 5.7.44-Rds.20250103, 5.7.44-Rds.20250213, 8.0.32, 8.0.33, 8.0.34, 8.0.35, 8.0.36, 8.0.37, 8.0.39, 8.0.40, 8.0.41
5.7.12	5.7.44, 5.7.44-Rds.20240408, 5.7.44-Rds.20240529, 5.7.44-Rds.20250103, 5.7.44-Rds.20250213, 8.0.32, 8.0.33, 8.0.34, 8.0.35, 8.0.36, 8.0.37, 8.0.39, 8.0.40, 8.0.41

Daten in eine Amazon RDS for MySQL MySQL-DB-Instance importieren

Für den Import von Daten in eine DB-Instance von RDS for MySQL stehen verschiedene Techniken zur Verfügung. Der beste Ansatz hängt von einer Reihe von Faktoren ab:

- Quelle der Daten
- Datenmenge
- Einmaliger Import oder fortlaufend
- Umfang der Ausfallzeit

Wenn Sie auch eine Anwendung mit den Daten migrieren, ist es wichtig, die Höhe der Ausfallzeit zu berücksichtigen.

In der folgenden Tabelle sind Techniken zum Importieren von Daten in eine RDS for MySQL-DB-Instance aufgeführt:

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Bestehende MySQL-Datenbank vor Ort oder bei Amazon EC2	Any	Einmalig	Etwas	Erstellen Sie ein Backup der Datenbank vor Ort, speichern Sie es auf Amazon S3 und stellen Sie die Sicherungsdatei anschließend auf einer neuen Amazon RDS-DB-Instance wieder her, auf der MySQL ausgeführt wird.	Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Bestehend MySQL-Datenbank vor Ort oder bei Amazon EC2	Any	Kontinuierlich	Minimal	Konfigurieren Sie die Replikation mit einer vorhandenen MySQL-Datenbank als Replikationsquelle.	Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Alle vorhandenen Datenbanken	Alle	Einmalig oder kontinuierlich	Minimal	Wird verwendet AWS Database Migration Service , um die Datenbank mit minimalen Ausfallzeiten zu migrieren und bei vielen Datenbank-DB-Engines die fortlaufende Replikation fortzusetzen.	Was ist AWS Database Migration Service und Verwenden einer MySQL-kompatiblen Datenbank als Ziel für AWS DMS im AWS Database Migration Service -Benutzerhandbuch
Bestehende MySQL-DB-Instance	Alle	Einmalig oder kontinuierlich	Minimal	Erstellen Sie eine Read Replica für die laufende Replikation. Stufen Sie die Read Replica für die einmalige Erstellung einer neuen DB-Instance hoch.	Arbeiten mit DB-Instance-Lesereplikaten

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Bestehende MySQL-Datenbank	Small	Einmalig	Etwas	Kopieren Sie die Daten mit einem Befehlszeilen-Dienstprogramm direkt in die MySQL-DB-Instance.	Daten aus einer externen MySQL-Datenbank in eine Amazon RDS for MySQL MySQL-DB-Instance importieren

Quelle	Datenmenge	Einmalig oder kontinuierlich	Ausfallzeit der Anwendur	Technik	Weitere Informationen
Nicht in einer vorhandenen Datenbank gespeicherte Daten	Medium	Einmalig	Etwas	Erstellen Sie Flatfiles und importieren Sie sie mithilfe von `LOAD DATA LOCAL INFILE` MySQL-Anweisungen.	Daten aus einer beliebigen Quelle in eine Amazon RDS for MySQL MySQL-DB-Instance importieren

> **Note**
>
> Die Systemdatenbank `mysql` enthält Authentifizierungs- und Autorisierungsinformationen, die für die Anmeldung bei der DB-Instance und für den Zugriff auf die Daten erforderlich sind. Das Verwerfen, Verändern, Umbenennen oder Verkürzen von Tabellen, Daten oder anderen Inhalten der `mysql`-Datenbank in der DB-Instance kann zu Fehlern führen und den Zugriff auf die DB-Instance und die Daten verhindern. In diesem Fall können Sie die DB-Instance aus einem Snapshot wiederherstellen, indem Sie den AWS CLI restore-db-instance-from-db-snapshotBefehl. Sie können die DB-Instance wiederherstellen mit dem AWS CLI restore-db-instance-to-point-in-timeBefehl.

Überlegungen zum Import von Daten für MySQL

Der folgende Inhalt enthält technische Informationen zum Laden von Daten in MySQL. Dieser Inhalt richtet sich an Benutzer, die mit der MySQL-Serverarchitektur vertraut sind.

Binäre Protokollierung

Die Aktivierung der binären Protokollierung reduziert die Leistung beim Laden von Daten und erfordert bis zu viermal mehr Festplattenspeicher als bei deaktivierter Protokollierung. Die Transaktionsgröße, die zum Laden der Daten verwendet wird, wirkt sich direkt auf die Systemleistung und den Speicherplatzbedarf aus. Größere Transaktionen erfordern mehr Ressourcen.

Transaktionsgröße

Die Transaktionsgröße beeinflusst die folgenden Aspekte des Ladens von MySQL-Daten:

- Ressourcenverbrauch
- Nutzung des Festplattenspeichers
- Vorgang fortsetzen
- Zeit, sich zu erholen
- Eingabeformat (Flatfiles oder SQL)

In diesem Abschnitt wird beschrieben, wie sich die Transaktionsgröße auf die binäre Protokollierung auswirkt, und verdeutlicht, welche Vorteile das Deaktivieren der binären Protokollierung bei umfangreichen Datenladevorgängen mit sich bringen kann. Sie können die binäre Protokollierung aktivieren und deaktivieren, indem Sie den Aufbewahrungszeitraum für automatische Amazon RDS-Backups festlegen. Nicht-Null-Werte aktivieren die binäre Protokollierung und Null deaktiviert diese. Weitere Informationen finden Sie unter Aufbewahrungszeitraum für Backups.

In diesem Abschnitt werden auch die Auswirkungen großer Transaktionen auf InnoDB beschrieben und warum es wichtig ist, die Transaktionsgrößen klein zu halten.

Kleine Transaktionen

Bei kleinen Transaktionen sorgt die binäre Protokollierung für eine Verdopplung der Schreibvorgänge auf der Festplatte, die für das Laden der Daten erforderlich sind. Dieser Effekt kann die Leistung für andere Datenbanksitzungen signifikant herabsetzen und die zum Laden der Daten erforderliche Zeit erhöhen. Die eingetretene Verschlechterung hängt teilweise von den folgenden Faktoren ab:

- Upload-Rate
- Andere Datenbankaktivitäten, die während des Ladevorgangs stattfinden
- Kapazität Ihrer Amazon RDS-DB-Instance

Die Binärprotokolle verbrauchen außerdem Festplattenspeicher, der in etwa der Menge der geladenen Daten entspricht, bis die Protokolle gesichert und entfernt werden. Amazon RDS minimiert dies, indem binäre Protokolle häufig gesichert und entfernt werden.

Große Transaktionen

Bei großen Transaktionen verdreifacht die binäre Protokollierung die IOPS und die Festplattennutzung aus den folgenden Gründen:

- Der Binärprotokollcache speichert Transaktionsdaten vorübergehend auf der Festplatte.
- Dieser Cache wächst mit der Transaktionsgröße, wodurch Speicherplatz beansprucht wird.
- Wenn die Transaktion (Commit oder Rollback) abgeschlossen ist, kopiert das System den Cache in das Binärprotokoll.

Bei diesem Vorgang werden drei Kopien der Daten erstellt:

- Die Originaldaten
- Der Cache auf der Festplatte
- Der letzte binäre Protokolleintrag

Jeder Schreibvorgang verursacht zusätzliche I/O, was sich weiter auf die Leistung auswirkt.

Aus diesem Grund benötigt die binäre Protokollierung dreimal so viel Festplattenspeicher wie die deaktivierte Protokollierung. Wenn Sie beispielsweise 10 GiB an Daten als eine einzelne Transaktion laden, werden drei Kopien erstellt:

- 10 GiB für die Tabellendaten
- 10 GiB für den binären Log-Cache
- 10 GiB für die binäre Logdatei

Der gesamte benötigte temporäre Festplattenspeicher beträgt 30 GiB.

Wichtige Überlegungen zum Festplattenspeicher:

- Die Cache-Datei bleibt bestehen, bis entweder die Sitzung endet oder eine neue Transaktion einen weiteren Cache erstellt.
- Das Binärprotokoll bleibt so lange bestehen, bis es gesichert wird, und kann möglicherweise 20 GiB (Cache und Protokoll) für einen längeren Zeitraum enthalten.

Wenn Sie die Daten `LOAD DATA LOCAL INFILE` zum Laden verwenden, erstellt Data Recovery eine vierte Kopie für den Fall, dass die Datenbank aus einer vor dem Ladevorgang erstellten Sicherung wiederhergestellt werden muss. Während der Wiederherstellung extrahiert MySQL die Daten aus dem Binärprotokoll in eine Flat-File. MySQL läuft dann`LOAD DATA LOCAL INFILE`. Aufbauend auf dem vorherigen Beispiel erfordert diese Wiederherstellung insgesamt einen temporären Festplattenspeicher von 40 GiB oder jeweils 10 GiB für Tabelle, Cache, Protokoll und lokale Datei. Ohne mindestens 40 GiB freien Festplattenspeicher schlägt die Wiederherstellung fehl.

Optimierung großer Datenlasten

Deaktivieren Sie bei großen Datenlasten die binäre Protokollierung, um den Overhead und den Speicherplatzbedarf zu reduzieren. Sie können die binäre Protokollierung deaktivieren, indem Sie den Aufbewahrungszeitraum für Backups auf 0 setzen. Stellen Sie nach Abschluss des Ladevorgangs den Aufbewahrungszeitraum für Backups auf den entsprechenden Wert ungleich Null zurück. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance und Aufbewahrungszeitraum für Backup in der Einstellungstabelle.

> **Note**
>
> Wenn es sich bei der DB-Instance um eine Quell-DB-Instance für Read Replicas handelt, können Sie den Aufbewahrungszeitraum für Backups nicht auf 0 setzen.

Wir empfehlen Ihnen, vor dem Laden der Daten einen DB-Snapshot zu erstellen. Weitere Informationen finden Sie unter Verwaltung manueller Backups.

InnoDB

Die folgenden Informationen zu Undo-Logging- und Wiederherstellungsoptionen unterstützen dabei, InnoDB-Transaktionen klein zu halten, um die Datenbankleistung zu optimieren.

InnoDB-Undo-Logging verstehen

Undo ist ein Protokollierungsmechanismus, der das Rollback von Transaktionen ermöglicht und Multi-Version Concurrency Control (MVCC) unterstützt.

Für MySQL 5.7 und niedrigere Versionen werden Undo-Logs im InnoDB-System-Tablespace (normalerweise ibdata1) gespeichert und so lange aufbewahrt, bis der Purge-Thread sie entfernt. Infolgedessen können umfangreiche Dataload-Transaktionen dazu führen, dass der System-Tablespace ziemlich groß wird und Festplattenspeicher beansprucht, den Sie nicht zurückgewinnen können, es sei denn, Sie erstellen die Datenbank neu.

Bei allen MySQL-Versionen muss der Purge-Thread warten, bis alle Undo-Logs gelöscht werden, bis die älteste aktive Transaktion entweder festgeschrieben oder zurückgesetzt wird. Wenn die Datenbank während des Ladevorgangs andere Transaktionen verarbeitet, sammeln sich auch deren Undo-Logs an und können nicht entfernt werden, selbst wenn die Transaktionen festgeschrieben werden und keine andere Transaktion die Undo-Logs für MVCC benötigt. In dieser Situation werden alle Transaktionen — einschließlich schreibgeschützter Transaktionen — langsamer. Diese Verlangsamung ist darauf zurückzuführen, dass alle Transaktionen auf alle Zeilen zugreifen, die von jeder Transaktion — nicht nur der Ladetransaktion — geändert werden. Tatsächlich müssen Transaktionen Undo-Logs durchsuchen, deren Löschung durch lang andauernde Ladetransaktionen während einer Undo-Log-Bereinigung verhindert wurde. Dies beeinträchtigt die Leistung aller Operationen, die auf geänderte Zeilen zugreifen.

Optionen zur Wiederherstellung von InnoDB-Transaktionen

Obwohl InnoDB Commit-Operationen optimiert, sind große Transaktions-Rollbacks langsam. Führen Sie für eine schnellere Wiederherstellung eine Wiederherstellung durch oder stellen Sie einen point-in-time DB-Snapshot wieder her. Weitere Informationen erhalten Sie unter Point-in-time Wiederherstellung und Wiederherstellung auf einer DB-Instance.

Formate für den Datenimport

MySQL unterstützt zwei Datenimportformate: Flatfiles und SQL. Lesen Sie die Informationen zu den einzelnen Formaten, um die beste Option für Ihre Bedürfnisse zu ermitteln.

Flache Dateien

Laden Sie für kleine Transaktionen Flatfiles mit`LOAD DATA LOCAL INFILE`. Dieses Datenimportformat bietet gegenüber der Verwendung von SQL die folgenden Vorteile:

- Weniger Netzwerkverkehr
- Niedrigere Datenübertragungskosten
- Geringerer Aufwand für die Datenbankverarbeitung
- Schnellere Verarbeitung

`LOAD DATA LOCAL INFILE`lädt die gesamte Flat-Datei als eine Transaktion. Halten Sie die Größe der einzelnen Dateien klein, um die folgenden Vorteile zu erzielen:

- Wiederaufnahmefunktion — Sie können verfolgen, welche Dateien geladen wurden. Wenn beim Laden ein Problem auftritt, können Sie dort weitermachen, wo Sie aufgehört haben. Möglicherweise müssen Sie einige Daten erneut an Amazon RDS übertragen, aber bei kleinen Dateien ist die Menge der erneuten Übertragung minimal.
- Paralleles Laden von Daten — Wenn Sie über ausreichend IOPS und Netzwerkbandbreite für das Laden einer einzelnen Datei verfügen, kann das parallele Laden Zeit sparen.
- Steuerung der Laderate — Wenn sich Ihre Datenlast negativ auf andere Prozesse auswirkt, können Sie die Laderate steuern, indem Sie das Intervall zwischen den Dateien verlängern.

Große Transaktionen verringern die Vorteile, die sich aus `LOAD DATA LOCAL INFILE` dem Import von Daten ergeben. Wenn Sie eine große Datenmenge nicht in kleinere Dateien aufteilen können, sollten Sie die Verwendung von SQL in Betracht ziehen.

SQL

SQL hat einen Hauptvorteil gegenüber flachen Dateien: Sie können die Transaktionsgröße leicht klein halten. Das Laden von SQL kann jedoch erheblich länger dauern als das Laden von Flatfiles. Außerdem kann es nach einem Fehler schwierig sein, zu bestimmen, wo der Vorgang fortgesetzt werden soll — Sie können mysqldump-Dateien nicht neu starten. Wenn beim Laden einer Mysqldump-Datei ein Fehler auftritt, müssen Sie die Datei ändern oder ersetzen, bevor der Ladevorgang fortgesetzt werden kann. Alternativ können Sie, nachdem Sie die Ursache des Fehlers behoben haben, den Zeitpunkt vor dem Laden wiederherstellen und die Datei erneut senden. Weitere Informationen finden Sie unter Point-in-time Wiederherstellung.

Verwenden von Amazon RDS-DB-Snapshots für Datenbank-Checkpoints

Wenn Sie Daten über lange Zeiträume (z. B. Stunden oder Tage) ohne binäre Protokollierung laden, verwenden Sie DB-Snapshots, um die Datensicherheit regelmäßig zu überprüfen. Jeder

DB-Snapshot erstellt eine konsistente Kopie Ihrer Datenbank-Instance, die bei Systemausfällen oder Datenbeschädigungen als Wiederherstellungspunkt dient. Da DB-Snapshots schnell sind, hat häufiges Checkpointing nur minimale Auswirkungen auf die Ladeleistung. Sie können frühere DB-Snapshots löschen, ohne die Haltbarkeit der Datenbank oder die Wiederherstellungsmöglichkeiten zu beeinträchtigen. Weitere Hinweise zu DB-Snapshots finden Sie unter. Verwaltung manueller Backups

Verkürzung der Ladezeiten der Datenbank

Die folgenden Punkte sind zusätzliche Tipps zur Verkürzung der Ladezeiten:

- Erstellen Sie alle sekundären Indizes, bevor Sie Daten in MySQL-Datenbanken laden. Im Gegensatz zu anderen Datenbanksystemen erstellt MySQL die gesamte Tabelle neu, wenn Sekundärindizes hinzugefügt oder geändert werden. Dieser Prozess erstellt eine neue Tabelle mit Indexänderungen, kopiert alle Daten und löscht die Originaltabelle.
- Lädt Daten in der Reihenfolge der Primärschlüssel. Für InnoDB-Tabellen kann dies die Ladezeiten um 75 bis 80% und die Datendateigröße um 50% reduzieren.
- Deaktivieren Sie Fremdschlüsseleinschränkungen, indem Sie `foreign_key_checks` auf `0` setzen. Dies ist häufig für Flatfiles erforderlich, die mit geladen werden`LOAD DATA LOCAL INFILE`. Bei jedem Ladevorgang beschleunigt die Deaktivierung der Fremdschlüsselprüfungen das Laden der Daten. Aktivieren Sie nach Abschluss des Ladevorgangs die Einschränkungen erneut, indem Sie die Einstellungen `foreign_key_checks` auf einstellen `1` und die Daten überprüfen.
- Laden Sie Daten parallel, sofern Sie sich nicht einem Ressourcenlimit nähern. Verwenden Sie gegebenenfalls partitionierte Tabellen, um das gleichzeitige Laden über mehrere Tabellensegmente hinweg zu ermöglichen.
- Um den Aufwand bei der SQL-Ausführung zu reduzieren, kombinieren Sie mehrere `INSERT` Anweisungen zu einzelnen Operationen mit mehreren `INSERT` Werten. `mysqldump`implementiert diese Optimierung automatisch.
- Reduzieren Sie InnoDB-Protokoll-IO-Operationen, indem Sie `innodb_flush_log_at_trx_commit` auf `0` setzen. Stellen Sie nach Abschluss des Ladevorgangs `innodb_flush_log_at_trx_commit` auf `1` wieder her.

> **Warning**
>
> Die Einstellung `innodb_flush_log_at_trx_commit` auf `0` bewirkt, dass InnoDB seine Logs jede Sekunde leert, nicht bei jedem Commit. Diese Einstellung erhöht die Leistung, kann jedoch zu Transaktionsverlusten bei Systemausfällen führen.

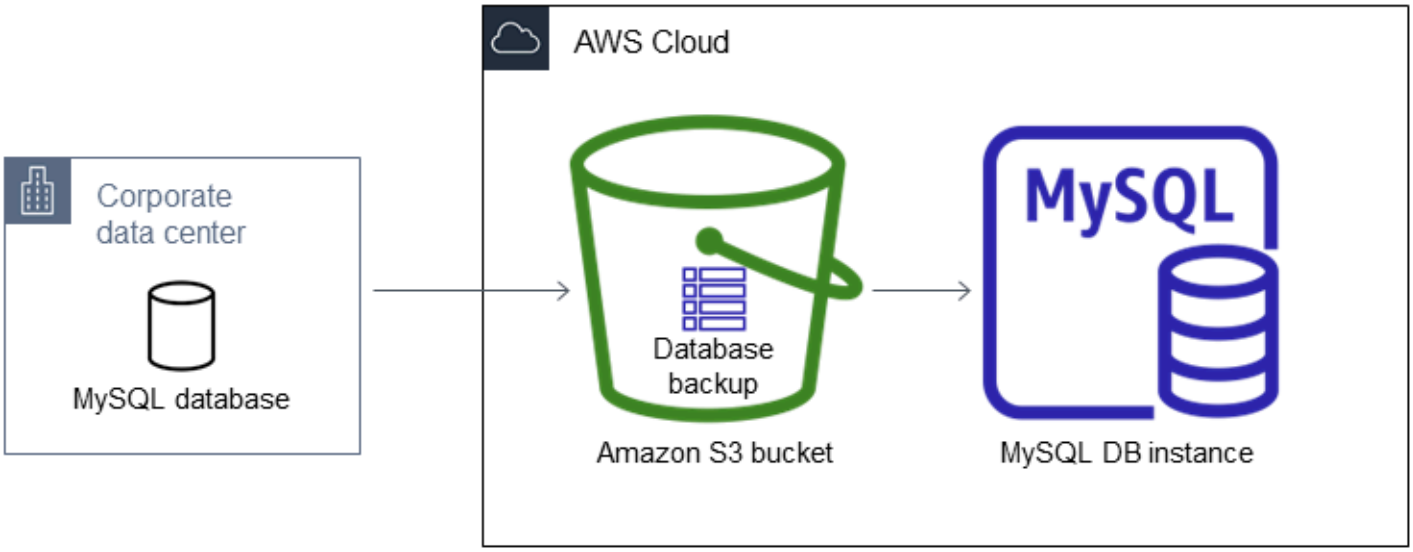
- Wenn Sie Daten in eine DB-Instance laden, die keine Read Replicas hat, setzen Sie den Wert `sync_binlog` auf`0`. Stellen Sie nach Abschluss des Ladevorgangs `sync_binlog parameter` auf 1 wieder her.
- Laden Sie Daten in eine Single-AZ-DB-Instance, bevor Sie die DB-Instance in eine Multi-AZ-Bereitstellung konvertieren. Wenn die DB-Instance bereits eine Multi-AZ-Bereitstellung verwendet, empfehlen wir nicht, zum Laden von Daten zu einer Single-AZ-Bereitstellung zu wechseln. Dies führt nur zu geringfügigen Verbesserungen.

Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance

Amazon RDS unterstützt den Import von MySQL-Datenbanken mit Backup-Dateien. Sie können eine Sicherungskopie Ihrer Datenbank erstellen, die Sicherungsdatei auf Amazon S3 speichern und dann die Sicherungsdatei auf einer neuen Amazon RDS-DB-Instance wiederherstellen, auf der MySQL ausgeführt wird. Amazon RDS unterstützt insgesamt den Import von Backup-Dateien aus Amazon S3 AWS-Regionen.

Im in diesem Abschnitt beschriebenen Szenario wird ein Backup einer On-Premise-Datenbank wiederhergestellt. Solange auf die Datenbank zugegriffen werden kann, können Sie diese Technik für Datenbanken an anderen Standorten wie Amazon EC2 oder anderen Cloud-Diensten verwenden.

Das folgende Diagramm veranschaulicht das unterstützte Szenario.



Wenn Ihre lokale Datenbank offline sein kann, während Sie Sicherungsdateien erstellen, kopieren und wiederherstellen, empfehlen wir Ihnen, Sicherungsdateien zu verwenden, um Ihre Datenbank

in Amazon RDS zu importieren. Wenn Ihre Datenbank nicht offline sein kann, können Sie eine der folgenden Methoden verwenden:

- Binärprotokolle — Importieren Sie zunächst Sicherungsdateien von Amazon S3 und Amazon RDS, wie in diesem Thema erklärt. Verwenden Sie dann die Replikation von Binärprotokollen (Binlog), um Ihre Datenbank zu aktualisieren. Weitere Informationen finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.
- AWS Database Migration Service— Wird verwendet AWS Database Migration Service , um Ihre Datenbank zu Amazon RDS zu migrieren. Weitere Informationen finden Sie unter Was ist AWS Database Migration Service?

Überblick über die Einrichtung für den Import von Backup-Dateien von Amazon S3 nach Amazon RDS

Um Backup-Dateien von Amazon S3 nach Amazon RDS zu importieren, benötigen Sie die folgenden Komponenten:

- Einen Amazon S3-Bucket zum Speichern Ihrer Sicherungsdateien

  Wenn Sie bereits einen Amazon S3 S3-Bucket haben, können Sie diesen Bucket verwenden. Wenn Sie keinen Amazon S3 S3-Bucket haben, erstellen Sie einen neuen. Weitere Informationen finden Sie unter Bucket erstellen.
- Ein von XtraBackup Percona erstelltes Backup Ihrer lokalen Datenbank.

  Weitere Informationen finden Sie unter Erstellen Ihrer Datenbanksicherung.
- Eine AWS Identity and Access Management (IAM-) Rolle, die Amazon RDS den Zugriff auf den S3-Bucket ermöglicht.

  Wenn Sie bereits über eine IAM-Rolle verfügen, können Sie diese Rolle verwenden und ihr Vertrauens- und Berechtigungsrichtlinien zuordnen. Weitere Informationen finden Sie unter Manuelles Erstellen einer IAM-Rolle.

  Wenn Sie keine IAM-Rolle haben, haben Sie zwei Möglichkeiten:

  - Sie können manuell eine neue IAM-Rolle erstellen. Weitere Informationen finden Sie unter Manuelles Erstellen einer IAM-Rolle.
  - Sie können wählen, dass Amazon RDS eine neue IAM-Rolle für Sie erstellt. Wenn Sie möchten, dass Amazon RDS eine neue IAM-Rolle für Sie erstellt, folgen Sie dem Verfahren, das den

Importieren von Daten aus Amazon S3 in eine neue MySQL-DB-Instance Abschnitt AWS Management Console in verwendet.

Erstellen Ihrer Datenbanksicherung

Verwenden Sie die XtraBackup Percona-Software, um Ihr Backup zu erstellen. Wir empfehlen Ihnen, die neueste Version von XtraBackup Percona zu verwenden. Sie können Percona XtraBackup über Software-Downloads auf der Percona-Website installieren.

> **Warning**
>
> Beim Erstellen einer Datenbanksicherung werden XtraBackup möglicherweise Anmeldeinformationen in der Datei xtrabackup_info gespeichert. Stellen Sie sicher, dass die `tool_command` Einstellung in der Datei xtrabackup_info keine vertraulichen Informationen enthält.

Welche XtraBackup Percona-Version Sie verwenden, hängt von der MySQL-Version ab, die Sie sichern.

- MySQL 8.4 — Verwenden Sie Percona XtraBackup Version 8.4.
- MySQL 8.0 — Verwenden Sie Percona XtraBackup Version 8.0.

  > **Note**
  >
  > Percona XtraBackup 8.0.12 und höhere Versionen unterstützen die Migration aller Versionen von MySQL 8.0. Wenn Sie zu RDS für MySQL 8.0.32 oder höher migrieren, müssen Sie Percona XtraBackup 8.0.12 oder höher verwenden.

- MySQL 5.7 — Verwenden Sie Percona XtraBackup Version 2.4.

Sie können Percona verwenden XtraBackup , um eine vollständige Sicherung Ihrer MySQL-Datenbankdateien zu erstellen. Wenn Sie Percona bereits XtraBackup zum Sichern Ihrer MySQL-Datenbankdateien verwenden, können Sie alternativ Ihre vorhandenen vollständigen und inkrementellen Backup-Verzeichnisse und -Dateien hochladen.

Weitere Informationen zur Sicherung Ihrer Datenbank mit Percona finden Sie unter Percona XtraBackup XtraBackup — Dokumentation auf der Percona-Website.

Erstellen Sie ein vollständiges Backup mit Percona XtraBackup

Verwenden Sie das XtraBackup Percona-Hilfsprogramm (`xtrabackup`), um eine vollständige Sicherung Ihrer MySQL-Datenbankdateien zu erstellen, die Amazon RDS aus Amazon S3 wiederherstellen kann.

Mit dem folgenden Befehl können Sie beispielsweise ein Backup einer MySQL-Datenbank erstellen und die Dateien im Ordner `/on-premises/s3-restore/backup` speichern.

```
xtrabackup --backup --user=myuser --password=password --target-dir=/on-premises/s3-
restore/backup
```

Wenn Sie Ihr Backup in eine einzige Datei komprimieren möchten, die Sie später bei Bedarf in mehrere Dateien aufteilen können, können Sie Ihr Backup je nach Ihrer MySQL-Version in einem der folgenden Formate speichern:

- Gzip (.gz) — Für MySQL 5.7 und niedrigere Versionen
- tar (.tar) — Für MySQL 5.7 und niedrigere Versionen
- Percona xbstream (.xbstream) — Für alle MySQL-Versionen

> **Note**
>
> Percona XtraBackup 8.0 und höher unterstützt nur Percona xbstream zur Komprimierung.

MySQL 5.7 und niedrigere Versionen

Mit dem folgenden Befehl wird ein Backup einer MySQL-Datenbank erstellt und in mehreren Gzip-Dateien gespeichert. Ersetzen Sie Werte durch Ihre eigenen Informationen.

```
xtrabackup --backup --user=my_user --password=password --stream=tar \
   --target-dir=/on-premises/s3-restore/backup | gzip - | split -d --bytes=500MB \
   - /on-premises/s3-restore/backup/backup.tar.gz
```

MySQL 5.7 und niedrigere Versionen

Mit dem folgenden Befehl wird ein Backup einer MySQL-Datenbank erstellt und in mehreren tar-Dateien gespeichert. Ersetzen Sie Werte durch Ihre eigenen Informationen.

```
xtrabackup --backup --user=my_user --password=password --stream=tar \
   --target-dir=/on-premises/s3-restore/backup | split -d --bytes=500MB \
   - /on-premises/s3-restore/backup/backup.tar
```

Alle MySQL-Versionen

Mit dem folgenden Befehl wird ein Backup einer MySQL-Datenbank erstellt und in mehreren xbstream-Dateien gespeichert. Ersetzen Sie Werte durch Ihre eigenen Informationen.

```
xtrabackup --backup --user=myuser --password=password --stream=xbstream \
   --target-dir=/on-premises/s3-restore/backup | split -d --bytes=500MB \
   - /on-premises/s3-restore/backup/backup.xbstream
```

> **Note**
>
> Wenn Ihnen der folgende Fehler angezeigt wird, liegt das möglicherweise daran, dass Sie in Ihrem Befehl Dateiformate gemischt haben:
>
> ```
> ERROR:/bin/tar: This does not look like a tar archive
> ```

Verwendung inkrementeller Backups mit Percona XtraBackup

Wenn Sie Percona bereits verwenden XtraBackup , um vollständige und inkrementelle Backups Ihrer MySQL-Datenbankdateien durchzuführen, müssen Sie kein vollständiges Backup erstellen und die Sicherungsdateien auf Amazon S3 hochladen. Um Zeit zu sparen, kopieren Sie stattdessen Ihre vorhandenen Backup-Verzeichnisse und -Dateien in Ihren Amazon S3 S3-Bucket. Weitere Informationen zum Erstellen inkrementeller Backups mit Percona XtraBackup finden Sie unter Erstellen eines inkrementellen Backups auf der Percona-Website.

Wenn Sie die Dateien der vollständigen und inkrementellen Backups in einen Amazon S3-Bucket hochladen, müssen Sie den Inhalt des Basisverzeichnisses rekursiv kopieren. Diese Inhalte umfassen sowohl das vollständige Backup als auch alle inkrementellen Backup-Verzeichnisse und -Dateien. Diese Kopie muss die Verzeichnisstruktur im Amazon-S3-Bucket beibehalten. Amazon RDS durchläuft alle Dateien und Verzeichnisse. Amazon RDS verwendet die `xtrabackup-checkpoints` Datei, die in jedem inkrementellen Backup enthalten ist, um das Basisverzeichnis zu identifizieren und inkrementelle Backups nach Log-Sequenznummer (LSN) -Bereich zu ordnen.

Überlegungen zum Backup für Percona XtraBackup

Amazon RDS verarbeitet Sicherungsdateien auf Basis des Dateinamens. Benennen Sie Ihre Backup-Dateien je nach Dateiformat mit der entsprechenden Dateierweiterung. Verwenden Sie dies beispielsweise `.xbstream` für Dateien, die im Percona-Xbstream-Format gespeichert wurden.

Amazon RDS verarbeitet Sicherungsdateien in alphanumerischer Reihenfolge. Um sicherzustellen, dass Ihre Sicherungsdateien in der richtigen Reihenfolge geschrieben und benannt werden, verwenden Sie die `split` Option, wenn Sie den Befehl ausführen. `xtrabackup`

Amazon RDS unterstützt keine Teilsicherungen, die mit Percona XtraBackup erstellt wurden. Sie können die folgenden Optionen nicht verwenden, um eine Teilsicherung zu erstellen, wenn Sie die Quelldateien für Ihre Datenbank sichern:

- `--tables`
- `--tables-exclude`
- `--tables-file`
- `--databases`
- `--databases-exclude`
- `--databases-file`

Manuelles Erstellen einer IAM-Rolle

Wenn Sie noch keine IAM-Rolle besitzen, können Sie manuell eine neue erstellen. Wenn Sie die Datenbank jedoch mithilfe von wiederherstellen AWS Management Console, empfehlen wir Ihnen, Amazon RDS diese neue IAM-Rolle für Sie erstellen zu lassen. Damit Amazon RDS diese Rolle für Sie erstellen kann, gehen Sie wie im Importieren von Daten aus Amazon S3 in eine neue MySQL-DB-Instance Abschnitt beschrieben vor.

Um manuell eine neue IAM-Rolle für den Import Ihrer Datenbank aus Amazon S3 zu erstellen, erstellen Sie eine Rolle, um Berechtigungen von Amazon RDS an Ihren Amazon S3 S3-Bucket zu delegieren. Beim Anlegen einer IAM-Rolle geben Sie Vertrauens- und Berechtigungsrichtlinien an. Verwenden Sie Vertrauens- und Berechtigungsrichtlinien, die den folgenden Beispielen ähneln, um Ihre Backup-Dateien aus Amazon S3 zu importieren. Weitere Informationen zum Erstellen der Rolle finden Sie unter Eine Rolle erstellen, um Berechtigungen an einen AWS Dienst zu delegieren.

Für die Vertrauens- und Berechtigungsrichtlinien müssen Sie einen Amazon-Ressourcennamen (ARN) angeben. Weitere Informationen zur ARN-Formatierung finden Sie unter Amazon Resource Names (ARNs) und AWS Service-Namespaces.

Example Vertrauensrichtlinie für den Import aus Amazon S3

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AssumeRoleForBackup",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Example Berechtigungsrichtlinie für den Import aus Amazon S3 — IAM-Benutzerberechtigungen

Ersetzen Sie im folgenden Beispiel *iam_user_id* durch Ihren eigenen Wert.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowS3AccessRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::111122223333:role/S3Access"
        }
    ]
```

```
}
```

Example Berechtigungsrichtlinie für den Import aus Amazon S3 — Rollenberechtigungen

Ersetzen Sie im folgenden Beispiel *amzn-s3-demo-bucket* und *prefix* durch Ihre eigenen Werte.

JSON

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObject"
            ],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix*"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "kms:Decrypt"
            ],
        "Resource": [
            "arn:aws:kms:us-east-1:111122223333:key/key_id*"
            ]
        }
    ]
}
```

> **Note**
>
> Wenn Sie ein Dateinamen-Präfix einfügen, fügen Sie das Sternchen (*) nach dem Präfix ein. Wenn Sie kein Präfix verwenden möchten, geben Sie nur ein Sternchen ein.

Importieren von Daten aus Amazon S3 in eine neue MySQL-DB-Instance

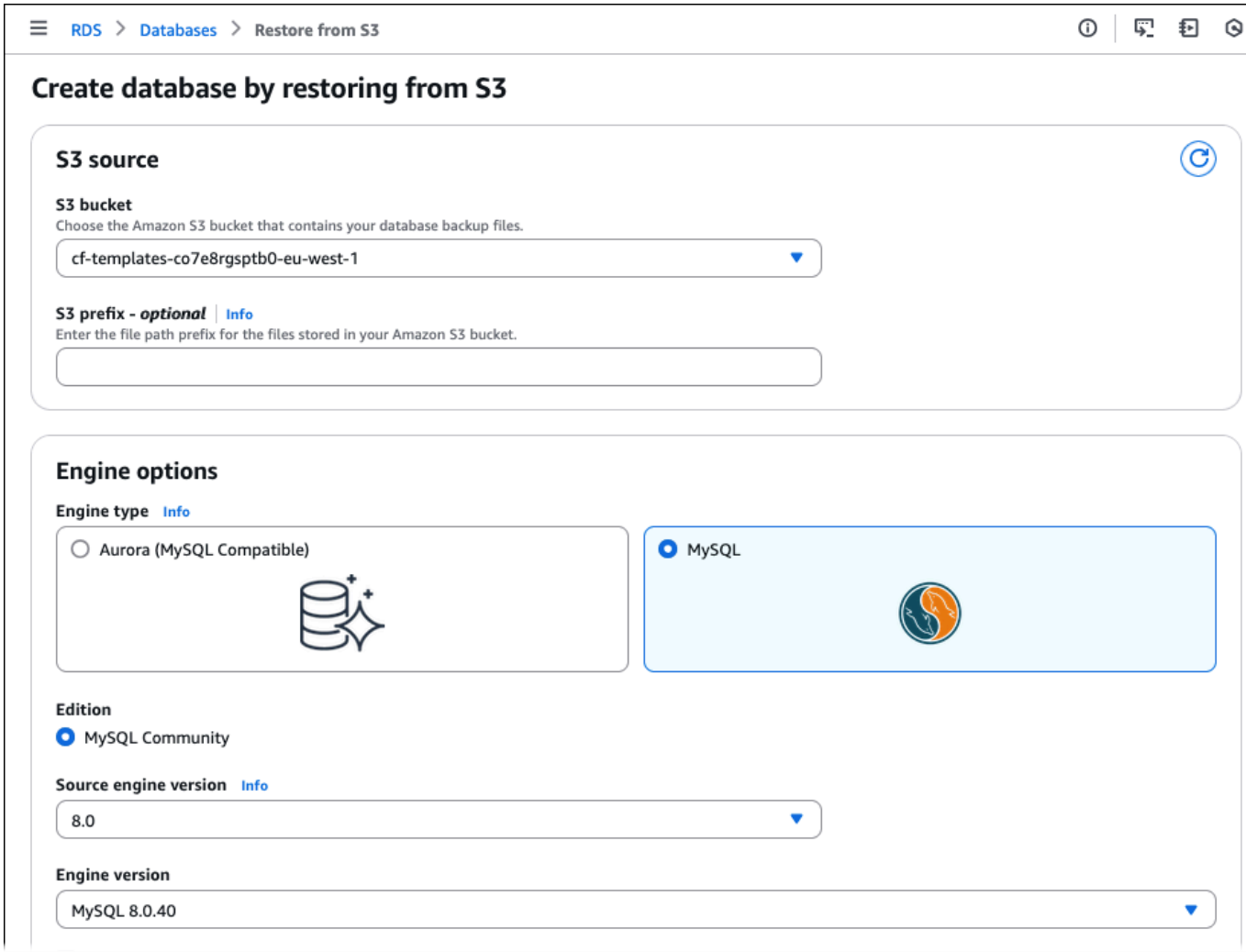
Mit der, oder RDS-API können Sie Daten aus Amazon S3 in eine neue MySQL-DB-Instance importieren. AWS Management Console AWS CLI

Konsole

Um Daten aus Amazon S3 in eine neue MySQL DB-Instance zu importieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole den Ort aus, AWS-Region an dem Sie Ihre DB-Instance erstellen möchten. Wählen Sie dasselbe AWS-Region wie für den Amazon S3 S3-Bucket, der Ihr Datenbank-Backup enthält.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Von S3 wiederherstellen.

   Die Seite Datenbank durch Wiederherstellen von S3 erstellen wird angezeigt.



5. Unter S3-Quelle:

   a. Wählen Sie den S3 bucket (S3-Bucket) aus, in dem sich das Backup befindet.

   b. (Optional) Geben Sie für das S3-Präfix das Dateipfadpräfix für die in Ihrem Amazon S3 S3-Bucket gespeicherten Dateien ein.

      Wenn Sie kein Präfix angeben, erstellt Amazon RDS Ihre DB-Instance mithilfe aller Dateien und Ordner im Stammordner des S3-Buckets. Wenn Sie ein Präfix angeben, erstellt Amazon RDS Ihre DB-Instance anhand der Dateien und Ordner im S3-Bucket, wobei der Pfad für die Datei mit dem angegebenen Präfix beginnt.

      Beispiel: Sie speichern Ihre Sicherungsdateien auf S3 in einem Unterordner mit dem Namen backups, und Sie haben mehrere Sätze von Sicherungsdateien, jede in einem eigenen Verzeichnis (gzip_backup1, gzip_backup2 usw.). In diesem Fall geben Sie das

Präfix backups/gzip_backup1 an, um Daten aus den Dateien im Ordner gzip_backup1 wiederherzustellen.

6. Unter Engine-Optionen:

   a. Wählen Sie in Engine-Typ die Option MySQL aus.

   b. Wählen Sie für Quellen-Engine-Version die MySQL-Hauptversion Ihrer Quelldatenbank.

   c. Wählen Sie für Engine-Version die Standard-Nebenversion Ihrer MySQL-Hauptversion in Ihrem AWS-Region.

      In der AWS Management Console ist nur die Standard-Nebenversion verfügbar. Nachdem Sie den Import abgeschlossen haben, können Sie Ihre DB-Instance aktualisieren.

7. Erstellen oder wählen Sie für die IAM-Rolle eine IAM-Rolle mit den erforderlichen Vertrauens- und Berechtigungsrichtlinien, die Amazon RDS den Zugriff auf Ihren Amazon S3 S3-Bucket ermöglichen. Durchführen einer der folgenden Aktionen:

   - (Empfohlen) Wählen Sie Neue Rolle erstellen und geben Sie den Namen der IAM-Rolle ein. Mit dieser Option erstellt Amazon RDS automatisch die Rolle mit der Vertrauensrichtlinie und der Berechtigungsrichtlinie für Sie.
   - Wählen Sie eine bestehende IAM-Rolle aus. Stellen Sie sicher, dass diese Rolle alle Kriterien in the section called “Manuelles Erstellen einer IAM-Rolle” erfüllt.

8. Geben Sie Ihre DB-Instance-Informationen an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

   > **Note**
   >
   > Stellen Sie sicher, dass Sie Ihrer neuen DB-Instance genügend Speicherplatz zuweisen, damit der Wiederherstellungsvorgang erfolgreich sein kann.
   > Um zukünftiges Wachstum automatisch zu ermöglichen, wählen Sie unter Zusätzliche Speicherkonfiguration die Option Automatische Speicherskalierung aktivieren aus.

9. Wählen Sie zusätzliche Einstellungen wie erforderlich aus.

10. Wählen Sie Create database (Datenbank erstellen) aus.

AWS CLI

Um Daten mit dem aus Amazon S3 in eine neue MySQL-DB-Instance zu importieren AWS CLI, führen Sie den Befehl restore-db-instance-from-s3 mit den folgenden Optionen aus. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

> **Note**
>
> Stellen Sie sicher, dass Sie Ihrer neuen DB-Instance genügend Speicherplatz zuweisen, damit der Wiederherstellungsvorgang erfolgreich sein kann.
> Verwenden Sie die `--max-allocated-storage` Option, um die automatische Speicherskalierung zu aktivieren und future Wachstum automatisch zu ermöglichen.

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--manage-master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`
- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

Für LinuxmacOS, oderUnix:

```
aws rds restore-db-instance-from-s3 \
    --allocated-storage 250 \
    --db-instance-identifier my_identifier \
    --db-instance-class db.m5.large \
    --engine mysql \
    --master-username admin \
```

```
    --manage-master-user-password \
    --s3-bucket-name amzn-s3-demo-bucket \
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \
    --s3-prefix bucket_prefix \
    --source-engine my_sql \
    --source-engine-version 8.0.32 \
    --max-allocated-storage 1000
```

Windows:

```
aws rds restore-db-instance-from-s3 ^
    --allocated-storage 250 ^
    --db-instance-identifier my_identifier ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --master-username admin ^
    --manage-master-user-password ^
    --s3-bucket-name amzn-s3-demo-bucket ^
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^
    --s3-prefix bucket_prefix ^
    --source-engine mysql ^
    --source-engine-version 8.0.32 ^
    --max-allocated-storage 1000
```

RDS-API

Um Daten mithilfe der Amazon RDS-API aus Amazon S3 in eine neue MySQL-DB-Instance zu importieren, rufen Sie den Vorgang Restore DBInstance FromS3 auf.

Einschränkungen und Überlegungen für den Import von Backup-Dateien von Amazon S3 nach Amazon RDS

Die folgenden Einschränkungen und Überlegungen gelten für den Import von Backup-Dateien von Amazon S3 in eine RDS for MySQL-DB-Instance:

- Sie können Ihre Daten nur auf eine neue DB-Instance migrieren, nicht auf eine bestehende DB-Instance.
- Sie müssen Percona verwenden XtraBackup , um Ihre Daten auf Amazon S3 zu sichern. Weitere Informationen finden Sie unter Erstellen Ihrer Datenbanksicherung.
- Der Amazon S3 S3-Bucket und die RDS for MySQL-DB-Instance müssen sich im selben System befinden AWS-Region.

- Aus den folgenden Quellen können Sie keine Wiederherstellung durchführen:
  - Ein DB-Instance-Snapshot-Export nach Amazon S3. Sie können auch keine Daten aus einem DB-Instance-Snapshot-Export in Ihren Amazon S3 S3-Bucket migrieren.
  - Eine verschlüsselte Quelldatenbank. Sie können die migrierten Daten jedoch verschlüsseln. Sie können die Daten während des Migrationsprozesses auch unverschlüsselt lassen.
  - Eine MySQL 5.5- oder 5.6-Datenbank.
- RDS for MySQL unterstützt Percona Server for MySQL nicht als Quelldatenbank, da es `compression_dictionary*` Tabellen in der `mysql schema` enthalten kann.
- RDS for MySQL unterstützt keine Rückwärtsmigration für Haupt- oder Nebenversionen. Sie können beispielsweise nicht von MySQL Version 8.0 zu RDS für MySQL 5.7 migrieren, und Sie können nicht von MySQL Version 8.0.32 zu RDS für MySQL Version 8.0.26 migrieren.
- Amazon RDS unterstützt den Import in der DB-Instance-Klasse db.t2.micro aus Amazon S3 nicht. Sie können jedoch eine Wiederherstellung zu einer anderen DB-Instance-Klasse durchführen und die DB-Instance-Klasse später ändern. Weitere Informationen zu Instance-Klassen finden Sie unter Hardware-Spezifikationen für DB-Instance-Klassen .
- Amazon S3 begrenzt die Größe einer Datei, die in einen Amazon S3-Bucket hochgeladen werden kann, auf 5 TB. Wenn eine Sicherungsdatei größer als 5 TB ist, müssen Sie die Sicherungsdatei in kleinere Dateien aufteilen.
- Amazon RDS begrenzt die Anzahl der Dateien, die in ein Amazon S3-Bucket hochgeladen werden können, auf 1 Million. Wenn es mehr als mehr als 1 Million Sicherungsdateien für Ihre Datenbank gibt (einschließlich aller vollständigen und inkrementellen Backups), speichern Sie diese mittels Gzip (.gz), tar (.tar.gz) oder Percona xbstream (.xbstream) im Amazon S3-Bucket. Percona XtraBackup 8.0 unterstützt nur Percona xbstream für die Komprimierung.
- Um Verwaltungsdienste für jede DB-Instance bereitzustellen, erstellt Amazon RDS den `rdsadmin` Benutzer, wenn es die DB-Instance erstellt. Da `rdsamin` es sich um einen reservierten Benutzer in Amazon RDS handelt, gelten die folgenden Einschränkungen:
  - Amazon RDS importiert keine Funktionen, Prozeduren, Ansichten, Ereignisse und Trigger mit dem `'rdsadmin'@'localhost'` Definer. Weitere Informationen erhalten Sie unter Gespeicherte Objekte mit 'rdsamin'@'localhost' als Definierer und Berechtigungen von Hauptbenutzerkonten.
  - Bei der Erstellung der DB-Instance erstellt Amazon RDS einen Master-Benutzer mit den maximal unterstützten Rechten. Bei der Wiederherstellung aus einem Backup entfernt Amazon RDS automatisch alle nicht unterstützten Rechte, die den importierten Benutzern zugewiesen wurden.

Informationen zu Benutzern, die davon betroffen sein könnten, finden Sie unter Benutzerkonten mit nicht unterstützten Rechten. Weitere Informationen zu den unterstützten Rechten in RDS for MySQL finden Sie unterRollenbasiertes Privilegienmodell für for My RDS SQL.

- Amazon RDS migriert keine vom Benutzer erstellten Tabellen im `mysql` Schema.
- Sie müssen den `innodb_data_file_path` Parameter mit nur einer Datendatei konfigurieren, die den Standard-Datendateinamen `ibdata1:12M:autoextend` verwendet. Mit dieser Methode können Sie Datenbanken mit zwei Datendateien oder mit einer Datendatei mit einem anderen Namen migrieren.

  Die folgenden Beispiele sind Dateinamen, die Amazon RDS nicht zulässt:

  - `innodb_data_file_path=ibdata1:50M`
  - `ibdata2:50M:autoextend`
  - `innodb_data_file_path=ibdata01:50M:autoextend`
- Sie können nicht von einer Quelldatenbank migrieren, deren Tabellen außerhalb des standardmäßigen MySQL-Datenverzeichnisses definiert sind.
- Die maximal unterstützte Größe für unkomprimierte Backups mit dieser Methode ist auf 64 TiB begrenzt. Bei komprimierten Backups ist dieser Grenzwert niedriger, um dem Speicherbedarf bei der Dekomprimierung Rechnung zu tragen. In solchen Fällen beträgt die maximal unterstützte Backup-Größe. `64 TiB - compressed backup size`

  Hinweise zur maximal unterstützten Datenbankgröße, die RDS for MySQL unterstützt, finden Sie unter Allzweck-SSD-Speicher undBereitgestellter IOPS SSD-Speicher.
- Amazon RDS unterstützt den Import von MySQL und anderen externen Komponenten und Plugins nicht.
- Amazon RDS stellt nicht alles aus Ihrer Datenbank wieder her. Wir empfehlen Ihnen, das Datenbankschema und die Werte für die folgenden Elemente aus Ihrer MySQL-Quellsystemdatenbank zu speichern und sie dann zu Ihrer wiederhergestellten RDS for MySQL-DB-Instance hinzuzufügen, nachdem sie erstellt wurde:
  - Benutzerkonten
  - Funktionen
  - Gespeicherte Prozeduren
  - Zeitzoneninformation. Zeitzoneninformationen werden aus dem lokalen Betriebssystem Ihrer RDS for MySQL-DB-Instance geladen. Weitere Informationen finden Sie unter Lokale Zeitzone für My SQL DB-Instances.

Gespeicherte Objekte mit 'rdsamin'@'localhost' als Definierer

Amazon RDS importiert keine Funktionen, Prozeduren, Ansichten, Ereignisse und Trigger `'rdsadmin'@'localhost'` als Definierer.

Mit dem folgenden SQL-Skript können die in Ihrer MySQL-Datenbank gespeicherten Objekte mit nicht unterstützten Definierern auflisten.

```
-- This SQL query lists routines with `rdsadmin`@`localhost` as the definer.

SELECT
    ROUTINE_SCHEMA,
    ROUTINE_NAME
FROM
    information_schema.routines
WHERE
    definer = 'rdsadmin@localhost';

-- This SQL query lists triggers with `rdsadmin`@`localhost` as the definer.

SELECT
    TRIGGER_SCHEMA,
    TRIGGER_NAME,
    DEFINER
FROM
    information_schema.triggers
WHERE
    DEFINER = 'rdsadmin@localhost';

-- This SQL query lists events with `rdsadmin`@`localhost` as the definer.

SELECT
    EVENT_SCHEMA,
    EVENT_NAME
FROM
    information_schema.events
WHERE
    DEFINER = 'rdsadmin@localhost';

-- This SQL query lists views with `rdsadmin`@`localhost` as the definer.
SELECT
    TABLE_SCHEMA,
    TABLE_NAME
```

```
FROM
    information_schema.views
WHERE
    DEFINER = 'rdsadmin@localhost';
```

Benutzerkonten mit nicht unterstützten Rechten

Benutzerkonten mit Rechten, die RDS for MySQL nicht unterstützt, werden ohne die nicht unterstützten Rechte importiert. Die Liste der unterstützten Rechte finden Sie unter Rollenbasiertes Privilegienmodell für for My RDS SQL.

Sie können die folgende SQL-Abfrage in Ihrer Quelldatenbank ausführen, um die Benutzerkonten aufzulisten, die nicht unterstützte Rechte haben.

```
SELECT
    user,
    host
FROM
    mysql.user
WHERE
    Shutdown_priv = 'y'
    OR File_priv = 'y'
    OR Super_priv = 'y'
    OR Create_tablespace_priv = 'y';
```

Daten aus einer externen MySQL-Datenbank in eine Amazon RDS for MySQL MySQL-DB-Instance importieren

Sie können Daten aus einer vorhandenen MariaDB- oder MySQL-Datenbank in eine RDS for MariaDB- oder RDS for MySQL-DB-Instance importieren. Dazu kopieren Sie die Datenbank mit mysqldump und leiten sie direkt an die DB-Instance RDS for MariaDB oder RDS for MySQL weiter. Das Befehlszeilen-Hilfsprogramm `mysqldump` wird üblicherweise verwendet, um Backups zu erstellen und Daten aus einem MariaDB- oder MySQL-Server in einen anderen zu übertragen. Es ist in der MySQL- und MariaDB-Client-Software enthalten.

In MariaDB 10.5 heißt der Client mariadb-dump. Ab MariaDB 11.0.1 müssen Sie anstelle von verwenden`mariadb-dump`. `mysqldump`

> **Note**
>
> Wenn Sie große Datenmengen mit einer MySQL-DB-Instance importieren oder exportieren, ist es zuverlässiger und schneller, Daten mithilfe von `xtrabackup` Sicherungsdateien und Amazon S3 in und aus Amazon RDS zu verschieben. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance. Amazon RDS unterstützt `xtrabackup` MariaDB nicht oder. `mariabackup` Amazon RDS unterstützt nur den Import aus Amazon S3 für MySQL.

Ein typischer `mysqldump` Befehl zum Verschieben von Daten aus einer externen Datenbank in eine Amazon RDS-DB-Instance sieht dem folgenden Beispiel ähnlich. Ersetzen Sie Werte durch Ihre eigenen Informationen. Ersetzen Sie für MariaDB 11.0.1 und höhere Versionen durch und `mysqldump` durch`mariadb-dump`. `mysql` `mariadb`

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Lassen Sie ein Leerzeichen zwischen der Option `-p` und dem eingegeben Passwort.
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die in diesem Beispiel gezeigten Eingabeaufforderungen angeben.

Stellen Sie sicher, dass Sie die folgenden Empfehlungen und Überlegungen kennen:

- Schließen Sie die folgenden Schemas aus der Dump-Datei aus:
  - `sys`

  - `performance_schema`
  - `information_schema`

  Das `mariadb-dump` Hilfsprogramm `mysqldump` and schließt diese Schemas standardmäßig aus.
- Wenn Sie Benutzer und Rechte migrieren müssen, sollten Sie in Erwägung ziehen, ein Tool zu verwenden, das die Data Control Language (DCL) für deren Neuerstellung generiert, z. B. das Hilfsprogramm. pt-show-grants
- Um den Import durchzuführen, stellen Sie sicher, dass der Benutzer Zugriff auf die DB-Instance hat. Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Die folgenden Parameter werden verwendet:

- `-u` *`local_user`* – für die Angabe eines Benutzernamens. Geben Sie bei der ersten Verwendung dieses Parameters den Namen eines Benutzerkontos in der lokalen MariaDB- oder MySQL-Datenbank an, das Sie mit dem `--databases` Parameter identifizieren.
- `--databases` *`database_name`* – für die Angabe des Datenbanknamens in der lokalen MariaDB- oder MySQL-Instance, die Sie in Amazon RDS importieren möchten.
- `--single-transaction`: zur Sicherstellung, dass alle aus der lokalen Datenbank geladenen Daten mit einem einzelnen Zeitpunkt übereinstimmen Wenn andere Prozesse die Daten ändern, während diese von `mysqldump` gelesen werden, kann durch die Verwendung dieses Parameters die Datenintegrität erhalten bleiben.
- `--compress`: für die Reduzierung des Verbrauchs der Netzwerkbandbreite, indem Daten vor dem Sendevorgang aus der lokalen Datenbank an Amazon RDS komprimiert werden.
- `--order-by-primary`: für die Reduzierung der Ladezeit durch Sortieren der Daten jeder Tabelle nach entsprechendem Primärschlüssel
- `--routines`— Verwenden Sie diese Option, wenn Routinen wie gespeicherte Prozeduren oder Funktionen in der Datenbank vorhanden sind, die Sie kopieren. Setzen Sie den Parameter auf`0`, wodurch die Routinen während des Importvorgangs ausgeschlossen werden. Erstellen Sie die Routinen später manuell in der Amazon RDS-Datenbank neu.
- `--triggers`— Verwenden Sie diese Option, wenn in der Datenbank, die Sie kopieren, Trigger vorhanden sind. Setzen Sie den Parameter auf`0`, wodurch die Trigger während des Importvorgangs ausgeschlossen werden. Erstellen Sie die Trigger später manuell in der Amazon RDS-Datenbank neu.

- `--events`— Verwenden Sie diese Option, wenn Ereignisse in der Datenbank vorhanden sind, die Sie kopieren. Setzen Sie den Parameter auf`0`, wodurch die Ereignisse während des Importvorgangs ausgeschlossen werden. Erstellen Sie die Ereignisse später manuell in der Amazon RDS-Datenbank neu.
- `-p`*`local_password`* – für die Angabe eines Passworts. Geben Sie bei der ersten Verwendung dieses Parameters das Passwort für das Benutzerkonto an, das Sie mit dem ersten `-u` Parameter identifizieren.
- `-u` *`RDS_user`* – für die Angabe eines Benutzernamens. Bei der zweiten Verwendung dieses Parameters geben Sie den Namen eines Benutzerkontos in der Standarddatenbank für die MariaDB- oder MySQL-DB-Instance an, die Sie mit dem `--host` Parameter identifizieren.
- `--port` *`port_number`* – für die Angabe des Ports für Ihre MariaDB- oder MySQL-DB-Instance. Standardmäßig ist dies 3306, sofern Sie den Wert nicht beim Erstellen der DB-Instance geändert haben.
- `--host` *`host_name`* – für die Angabe des Domain-Name-System(DNS)-Namens aus dem Endpunkt der Amazon-RDS-DB-Instance, zum Beispiel, `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Sie finden den Endpunktwert in den DB-Instance-Details in der Amazon RDS-Konsole.
- `-p`*`RDS_password`* – für die Angabe eines Passworts. Beim zweiten Gebrauch dieses Parameters geben Sie den Namen eines Passworts einer lokalen MySQL- oder MariaDB-Datenbank an, die durch den zweiten Parameter `-u` bezeichnet wird.

Stellen Sie sicher, dass Sie alle gespeicherten Prozeduren, Auslöser, Funktionen oder Ereignisse manuell in Ihrer Amazon-RDS-Datenbank erstellen. Wenn Sie eines dieser Objekte in der Datenbank haben, die Sie kopieren, schließen Sie es aus, wenn Sie `mysqldump` oder ausführen`mariadb-dump`. Fügen Sie dazu Ihrem `mariadb-dump` Befehl `mysqldump` oder die folgenden Parameter hinzu:

- `--routines=0`
- `--triggers=0`
- `--events=0`

Beispiel

Im folgenden Beispiel wird die `world` Beispieldatenbank auf dem lokalen Host in eine RDS for MySQL-DB-Instance kopiert. Ersetzen Sie Werte durch Ihre eigenen Informationen. Um die

Beispieldatenbank in eine RDS for MariaDB-DB-Instance zu kopieren, `mysqldump` ersetzen Sie sie durch `mariadb-dump` und `mysql` durch. `mariadb`

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sudo mysqldump -u local_user \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocal_password | mysql -u rds_user \
        --port=3306 \
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com \
        -pRDS_password
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

Führen Sie den folgenden Befehl in einer Befehlszeile aus, die geöffnet wurde, indem Sie im Windows-Programmmenü mit der rechten Maustaste auf Eingabeaufforderung klicken und Als Administrator ausführen wählen. Ersetzen Sie Werte durch Ihre eigenen Informationen. Um die Beispieldatenbank in eine RDS for MariaDB-DB-Instance zu kopieren, `mysqldump` ersetzen Sie sie durch `mariadb-dump` und `mysql` durch. `mariadb`

```
mysqldump -u local_user ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocal_password | mysql -u RDS_user ^
        --port=3306 ^
        --host=my_instance.123456789012.us-east-1.rds.amazonaws.com ^
        -pRDS_password
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.

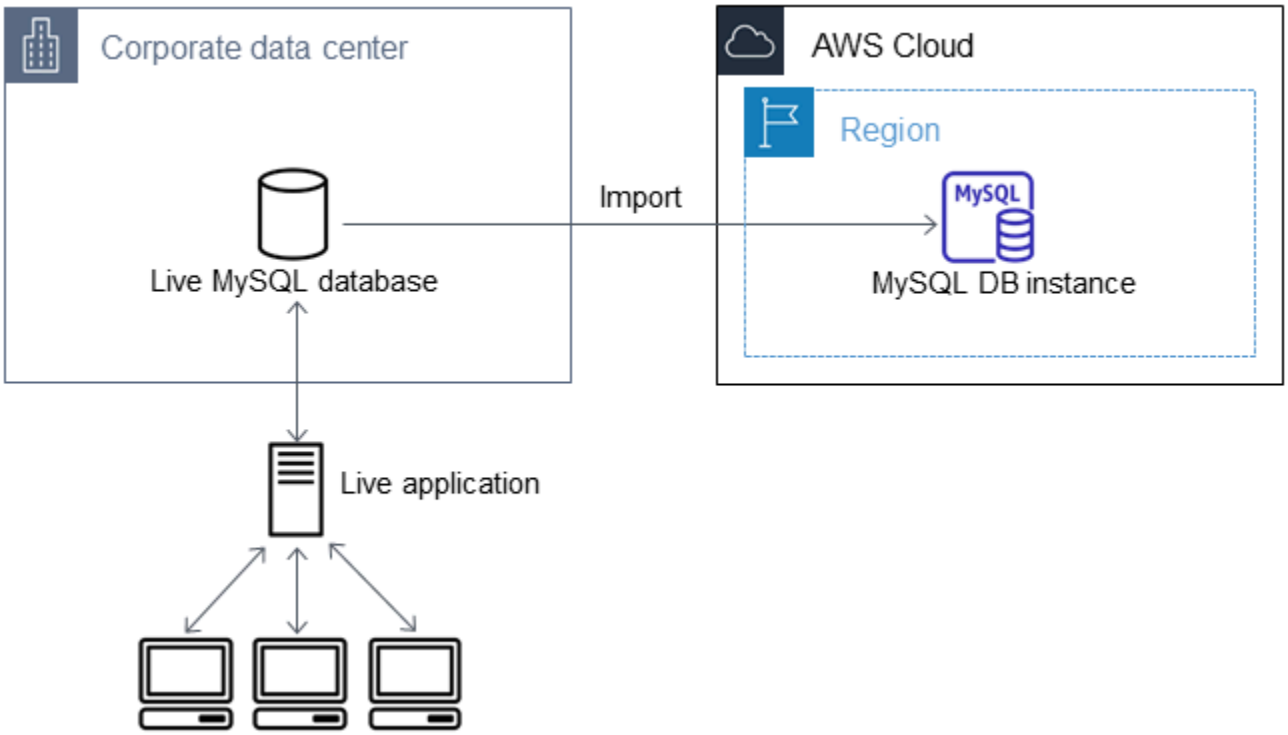
Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit

In einigen Fällen müssen Sie möglicherweise Daten aus einer externen MariaDB- oder MySQL-Datenbank, die eine Live-Anwendung unterstützt, in eine RDS for MariaDB-DB-Instance, eine RDS for MySQL-DB-Instance oder einen RDS for MySQL Multi-AZ-DB-Cluster importieren. Nutzen Sie das folgende Verfahren, um die Auswirkungen auf die Verfügbarkeit der Anwendung zu minimieren. Dieses Verfahren kann außerdem hilfreich sein, wenn Sie mit einer sehr großen Datenbank arbeiten. Mit diesem Verfahren können Sie die Importkosten senken, indem Sie die Datenmenge reduzieren, die über das Netzwerk übertragen wird. AWS

In diesem Verfahren übertragen Sie eine Kopie Ihrer Datenbankdaten auf eine EC2 Amazon-Instance und importieren die Daten in eine neue Amazon RDS-Datenbank. Anschließend verwenden Sie die Replikation, um die Amazon RDS-Datenbank up-to-date mit Ihrer externen Live-Instance zu verknüpfen, bevor Sie Ihre Anwendung auf die Amazon RDS-Datenbank umleiten. Wenn die externe Instance MariaDB 10.0.24 oder höher ist und die Zielinstanz RDS for MariaDB ist, konfigurieren Sie die MariaDB-Replikation basierend auf globalen Transaktions-Identifikatoren (). GTIDs Andernfalls konfigurieren Sie die Replikation basierend auf den Binärprotokollkoordinaten. Wir empfehlen die GTID-basierte Replikation, wenn Ihre externe Datenbank diese unterstützt, da diese Methode zuverlässiger ist. Weitere Informationen finden Sie unter Global Transaction ID in der MariaDB-Dokumentation.

> **Note**
>
> Wenn Sie Daten in eine RDS for MySQL-DB-Instance importieren möchten und Ihr Szenario dies unterstützt, empfehlen wir, Daten mithilfe von Sicherungsdateien und Amazon S3 in und aus Amazon RDS zu verschieben. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance.

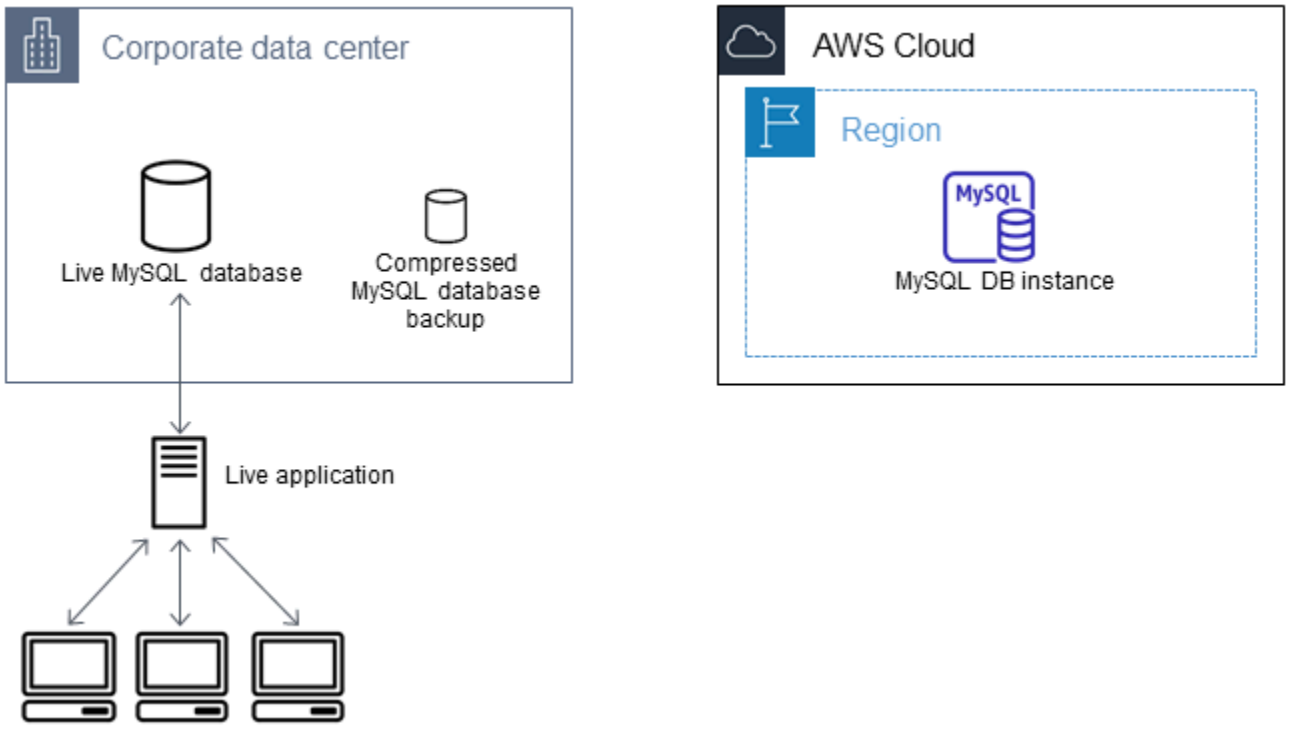
Das folgende Diagramm zeigt den Import einer externen MySQL-Datenbank in eine MySQL-Datenbank auf Amazon RDS.



Aufgabe 1: Erstellen Sie eine Kopie Ihrer vorhandenen Datenbank

Der erste Schritt bei der Migration von großen Datenmengen in eine Datenbank von RDS für MariaDB oder RDS für MySQL mit minimaler Ausfallzeit ist das Erstellen einer Kopie der Quelldaten.

Das folgende Diagramm zeigt die Erstellung einer Sicherungskopie der MySQL-Datenbank.



Sie können das Hilfsprogramm `mysqldump` verwenden, um ein Datenbank-Backup im SQL-Format oder im separierten Textformat zu erstellen. In MariaDB 10.5 heißt der Client mariadb-dump. Ab MariaDB 11.0.1 müssen Sie anstelle von verwenden`mariadb-dump`. `mysqldump` Es wird empfohlen, mit jedem Format in einer Nichtproduktionsumgebung einen Testlauf durchzuführen, um zu sehen, welche Methode die Ausführungsdauer von mysqldump minimiert.

Wir empfehlen außerdem, die `mysqldump` Leistung gegen den Vorteil abzuwägen, der sich aus der Verwendung des Textformats mit Trennzeichen beim Laden ergibt. Ein Backup, das ein separiertes Textformat verwendet, erstellt eine tabulatorseparierte Textdatei für jede verworfene Tabelle. Um den Zeitaufwand für den Import Ihrer Datenbank zu reduzieren, können Sie diese Dateien mit dem Befehl `LOAD DATA LOCAL INFILE` parallel laden. Weitere Informationen finden Sie unter Daten für MariaDB laden oder Daten für MySQL laden.

Bevor Sie mit dem Sicherungsvorgang beginnen, müssen Sie die Optionen für die Replikation in der nach Amazon RDS zu kopierenden MariaDB- oder MySQL-Datenbank einstellen. Die Optionen für die Replikation schließen die Aktivierung der Binärprotokollierung und das Einstellen einer eindeutigen Server-ID mit ein. Das Einstellen dieser Optionen veranlasst den Server, mit der Protokollierung Ihrer Datenbanktransaktionen zu beginnen, und bereitet ihn darauf vor, später im Vorgang als Quellreplikationsinstance zu agieren.

Stellen Sie sicher, dass Sie die folgenden Empfehlungen und Überlegungen kennen:

- Verwenden Sie die Option `--single-transaction` mit `mysqldump`, da sie einen einheitlichen Zustand der Datenbank speichert. Um eine gültige Dump-Datei sicherzustellen, führen Sie beim Ausführen von `mysqldump` keine DDL-Anweisungen (Data Definition Language) aus. Sie können ein Wartungsfenster für diese Abläufe planen.
- Schließen Sie die folgenden Schemas aus der Dump-Datei aus:
  - `sys`
  - `performance_schema`
  - `information_schema`

  Das `mariadb-dump` Hilfsprogramm `mysqldump` and schließt diese Schemas standardmäßig aus.
- Wenn Sie Benutzer und Rechte migrieren müssen, sollten Sie in Erwägung ziehen, ein Tool zu verwenden, das die Data Control Language (DCL) für deren Neuerstellung generiert, z. B. das Hilfsprogramm. pt-show-grants

So stellen Sie Optionen für die Replikation ein:

1. Bearbeiten Sie die `my.cnf`-Datei. Diese Datei befindet sich normalerweise unter. `/etc`

   ```
   sudo vi /etc/my.cnf
   ```

   Fügen Sie die Optionen `log_bin` und `server_id` zum Abschnitt `[mysqld]` hinzu. Die Option `log_bin` bietet eine Dateinamenkennung für Binärprotokolldateien. Die Option `server_id` stellt eine eindeutige Kennung für den Server für Quelle-Replica-Beziehungen bereit.

   Das folgende Beispiel zeigt den aktualisierten `[mysqld]` Abschnitt einer `my.cnf` Datei:

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Weitere Informationen finden Sie unter Einstellung der Konfiguration der Replikationsquelle in der MySQL-Dokumentation.
2. Für die Replikation mit einem Multi-AZ-DB-Cluster setzen Sie den `ENFORCE_GTID_CONSISTENCY` und den `GTID_MODE` Parameter auf`ON`.

   ```
   mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
   ```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

Diese Einstellungen sind für die Replikation mit einer DB-Instance nicht erforderlich.

3. Den Service `mysql` neu starten.

```
sudo service mysqld restart
```

So erstellen Sie eine Sicherungskopie für Ihre bestehende Datenbank:

1. Erstellen Sie mit dem `mysqldump` Hilfsprogramm oder dem Hilfsprogramm eine Sicherungskopie Ihrer Daten und geben Sie dabei entweder das `mariadb-dump` SQL-Format oder das Textformat mit Trennzeichen an.

   Geben Sie für MySQL 8.0.25 und niedrigere Versionen `--master-data=2` an, dass eine Sicherungsdatei erstellt werden soll, mit der die Replikation zwischen Servern gestartet werden kann. Geben Sie für MySQL 8.0.26 und höhere Versionen `--source-data=2` an, dass eine Sicherungsdatei erstellt werden soll, mit der die Replikation zwischen Servern gestartet werden kann. Weitere Informationen finden Sie unter mysqldump — A Database Backup Program in der MySQL-Dokumentation.

   Um die Leistung zu verbessern und die Datenintegrität sicherzustellen, verwenden Sie die Optionen `--order-by-primary` und `--single-transaction` für. `mysqldump`

   Um zu vermeiden, dass die MySQL-Systemdatenbank in das Backup aufgenommen wird, verwenden Sie die `--all-databases` Option mit nicht`mysqldump`. Weitere Informationen finden Sie unter Erstellen eines Datensnapshots mit mysqldump in der MySQL-Dokumentation.

   Stellen Sie `chmod` bei Bedarf sicher, dass das Verzeichnis, in dem die Sicherungsdatei erstellt wird, beschreibbar ist.

   > **Important**
   >
   > Führen Sie unter Windows die Eingabeaufforderung als Administrator aus.

   - Verwenden Sie den folgenden Befehl, um eine SQL-Ausgabe zu erzeugen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
    --order-by-primary \
    -r backup.sql \
    -u local_user \
    -ppassword
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -ppassword
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.

- Verwenden Sie den folgenden Befehl, um eine Ausgabe mit begrenztem Text zu erzeugen:

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sudo mysqldump \
    --tab=target_directory \
```

```
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -ppassword
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -ppassword
```

> **Note**
>
> Aus Sicherheitsgründen sollten Sie andere Anmeldeinformationen als die im Beispiel gezeigten Eingabeaufforderungen angeben.
> Stellen Sie sicher, dass Sie alle gespeicherten Prozeduren, Auslöser, Funktionen oder Ereignisse manuell in Ihrer Amazon-RDS-Datenbank erstellen. Falls Sie eines dieser Objekte in der Datenbank haben, die Sie kopieren, schließen Sie sie aus, wenn Sie `mysqldump` ausführen. Fügen Sie dazu Ihrem `mysqldump` Befehl die folgenden Argumente hinzu:
>
> - `--routines=0`
> - `--triggers=0`
> - `--events=0`

Bei MySQL 8.0.22 und niedrigeren Versionen und MariaDB-Versionen wird ein Kommentar zurückgegeben, wenn Sie das Format mit Trennzeichen ausführen `mysqldump` und angeben. `CHANGE MASTER TO` Dieser Kommentar beinhaltet den Namen und die Position der

Hauptprotokolldatei. Wenn Sie für MySQL 8.0.23 und höhere Versionen das Format mit getrenntem Text `mysqldump` verwenden, wird ein Kommentar zurückgegeben. `CHANGE REPLICATION SOURCE TO` Dieser Kommentar enthält den Namen und die Position der Quellprotokolldatei. Wenn es sich bei der externen Instanz um etwas anderes als MariaDB 10.0.24 und höhere Versionen oder MySQL 8.0.23 und höhere Versionen handelt, notieren Sie sich die Werte für und. `MASTER_LOG_FILE MASTER_LOG_POS` Sie benötigen diese Werte beim Einrichten der Replikation.

Die folgende Ausgabe wird für MariaDB-Versionen und für MySQL 8.0.22 und niedrigere Versionen zurückgegeben:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

Die folgende Ausgabe wird für MySQL 8.0.23 und höhere Versionen zurückgegeben:

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE SOURCE TO SOURCE_LOG_FILE='mysql-bin-changelog.000031',
 SOURCE_LOG_POS=107;
```

Für MySQL 8.0.22 und niedrigere Versionen können Sie, wenn Sie das SQL-Format verwenden, den Namen und die Position der Master-Logdatei im `CHANGE MASTER TO` Kommentar in der Backup-Datei abrufen. Für MySQL 8.0.23 und höhere Versionen können Sie, wenn Sie das SQL-Format verwenden, den Namen und die Position der Quellprotokolldatei im `CHANGE REPLICATION SOURCE TO` Kommentar in der Sicherungsdatei abrufen. Wenn die externe Instance MariaDB Version 10.0.24 oder höher ist, können Sie die GTID im nächsten Schritt abrufen.

2. Wenn es sich bei der von Ihnen verwendeten externen Instanz um MariaDB-Version 10.0.24 oder höher handelt, verwenden Sie die GTID-basierte Replikation. Führen Sie es `SHOW MASTER STATUS` auf der externen MariaDB-Instanz aus, um den Namen und die Position der binären Protokolldatei abzurufen, und konvertieren Sie sie dann in eine GTID, indem Sie sie `BINLOG_GTID_POS` auf der externen MariaDB-Instanz ausführen.

```
SELECT BINLOG_GTID_POS('binary_log_file_name', binary_log_file_position);
```

Beachten Sie die zurückgegebene GTID. Sie benötigen die GTID, um die Replikation zu konfigurieren.

3. Komprimieren Sie die kopierten Daten, um die Menge der Netzwerkressourcen zu reduzieren, die benötigt werden, um Ihre Daten in eine Amazon-RDS-Datenbank zu kopieren. Notieren Sie sich die Größe der Backup-Datei. Sie benötigen diese Informationen, um zu bestimmen, wie groß eine EC2 Amazon-Instance erstellt werden soll. Wenn Sie fertig sind, komprimieren Sie die Sicherungsdatei mithilfe von GZIP oder Ihrem bevorzugten Komprimierungsprogramm.
   - Verwenden Sie den folgenden Befehl, um die SQL-Ausgabe zu komprimieren:

     ```
     gzip backup.sql
     ```

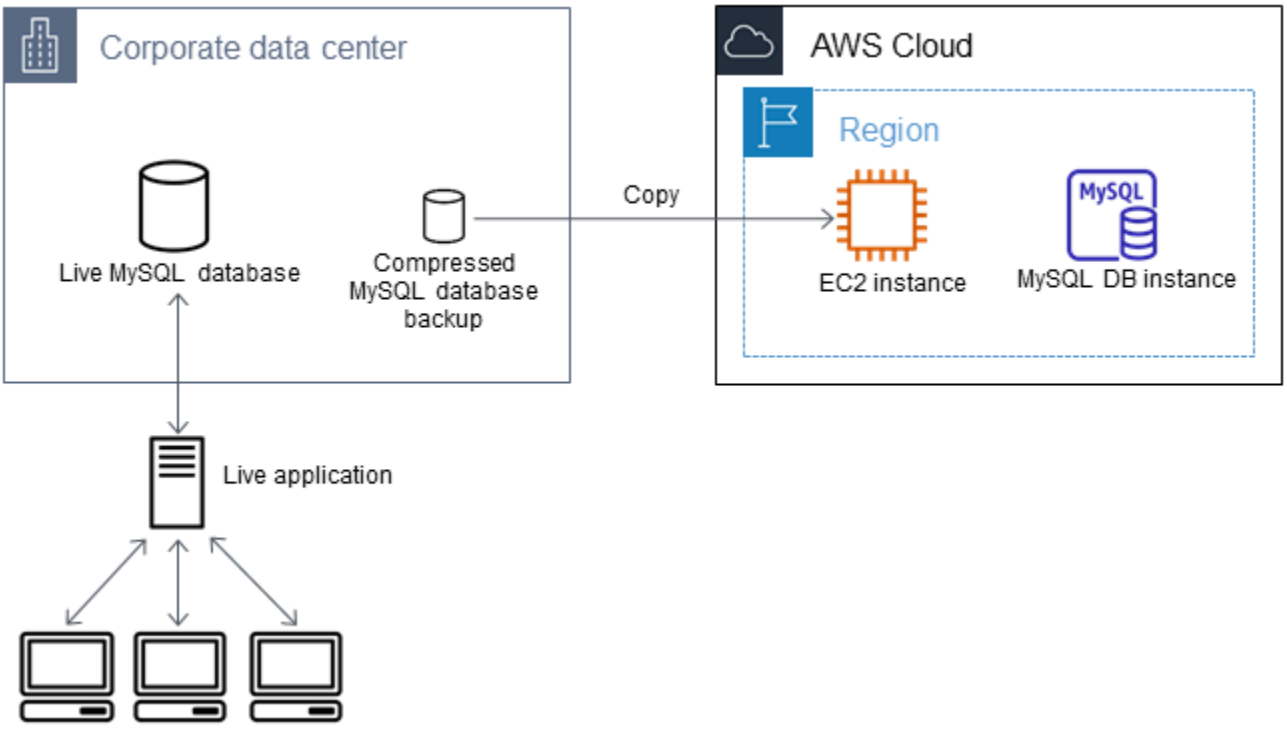
   - Verwenden Sie den folgenden Befehl, um die Ausgabe mit begrenztem Text zu komprimieren:

     ```
     tar -zcvf backup.tar.gz target_directory
     ```

Aufgabe 2: Erstellen Sie eine EC2 Amazon-Instance und kopieren Sie die komprimierte Datenbank

Das Kopieren Ihrer komprimierten Datenbank-Backup-Datei auf eine EC2 Amazon-Instance benötigt weniger Netzwerkressourcen als das direkte Kopieren unkomprimierter Daten zwischen Datenbank-Instances. Nachdem sich Ihre Daten in Amazon befinden EC2, können Sie sie von dort direkt in Ihre MariaDB- oder MySQL-Datenbank kopieren. Damit Sie Kosten für Netzwerkressourcen sparen können, muss sich Ihre EC2 Amazon-Instance in derselben Umgebung AWS-Region wie Ihre Amazon RDS-DB-Instance befinden. Wenn sich die EC2 Amazon-Instance in derselben Datenbank AWS-Region wie Ihre Amazon RDS-Datenbank befindet, verringert sich auch die Netzwerklatenz während des Imports.

Das folgende Diagramm zeigt das Kopieren der Datenbanksicherung auf eine EC2 Amazon-Instance.



Um eine EC2 Amazon-Instance zu erstellen und Ihre Daten zu kopieren

1. Erstellen Sie in AWS-Region dem Bereich, in dem Sie die Amazon RDS-Datenbank erstellen möchten, eine Virtual Private Cloud (VPC), eine VPC-Sicherheitsgruppe und ein VPC-Subnetz. Stellen Sie sicher, dass die eingehenden Regeln für Ihre VPC-Sicherheitsgruppe IP-Adressen zulassen, die für eine Verbindung Ihrer Anwendung mit erforderlich sin AWS. Sie können beispielsweise einen Bereich von IP-Adressen `203.0.113.0/24` oder eine andere VPC-Sicherheitsgruppe angeben. Sie können die Amazon VPC-Konsole verwenden VPCs, um Subnetze und Sicherheitsgruppen zu erstellen und zu verwalten. Weitere Informationen finden Sie unter Erste Schritte mit Amazon VPC im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
2. Öffnen Sie die EC2 Amazon-Konsole und wählen Sie die aus, die sowohl Ihre EC2 Amazon-Instance als auch Ihre Amazon RDS-Datenbank enthalten AWS-Region soll. Starten Sie eine EC2 Amazon-Instance mit der VPC, dem Subnetz und der Sicherheitsgruppe, die Sie in Schritt 1 erstellt haben. Stellen Sie sicher, dass Sie einen Instance-Typ mit genügend Speicherplatz für Ihre unkomprimierte Datenbank-Sicherungsdatei ausgewählt haben. Einzelheiten zu EC2 Amazon-Instances finden Sie unter Erste Schritte mit Amazon EC2 im Amazon Elastic Compute Cloud-Benutzerhandbuch.
3. Um von Ihrer EC2 Amazon-Instance aus eine Verbindung zu Ihrer Amazon RDS-Datenbank herzustellen, bearbeiten Sie Ihre VPC-Sicherheitsgruppe. Fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die die private IP-Adresse Ihrer EC2 Instance angibt. Sie finden die private

IP-Adresse auf der Registerkarte „Details“ im Bereich „Instanz“ im EC2 Konsolenfenster. Um die VPC-Sicherheitsgruppe zu bearbeiten und eine Regel für eingehenden Datenverkehr hinzuzufügen, wählen Sie im Navigationsbereich der EC2 Konsole Sicherheitsgruppen, wählen Sie Ihre Sicherheitsgruppe aus und fügen Sie dann eine eingehende Regel für MySQL oder Aurora hinzu, die die private IP-Adresse Ihrer Instance angibt. EC2 Informationen zum Hinzufügen einer Regel für eingehenden Datenverkehr zu einer VPC-Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

4. Kopieren Sie Ihre komprimierte Datenbank-Backup-Datei von Ihrem lokalen System auf Ihre EC2 Amazon-Instance. Verwenden Sie `chmod`, falls erforderlich, um sicherzustellen, dass Sie über Schreibberechtigungen für das Zielverzeichnis der EC2 Amazon-Instance verfügen. Sie können `scp` oder einen Secure-Shell(SSH)-Client verwenden, um die Datei zu kopieren. Der folgende Befehl ist ein `scp` Beispielbefehl:

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Achten Sie beim Kopieren vertraulicher Daten darauf, ein sicheres Netzwerkübertragungsprotokoll zu verwenden.

5. Stellen Sie mithilfe der folgenden Befehle Connect zu Ihrer EC2 Amazon-Instance her und installieren Sie die neuesten Updates und die MySQL-Client-Tools:

```
sudo yum update -y
sudo yum install mysql -y
```

Weitere Informationen finden Sie unter Connect Ihrer Instance für Linux-Instances herstellen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

> **Important**
>
> In diesem Beispiel wird der MySQL-Client auf einem Amazon Machine Image (AMI) für eine Amazon-Linux-Verteilung installiert. In diesem Beispiel wird der MySQL-Client nicht auf einer anderen Distribution wie Ubuntu oder Red Hat Enterprise Linux installiert. Informationen zur Installation von MySQL finden Sie unter Installation von MySQL in der MySQL-Dokumentation.

6. Während Sie mit Ihrer EC2 Amazon-Instance verbunden sind, dekomprimieren Sie Ihre Datenbank-Backup-Datei. Die folgenden Befehle sind Beispiele.
   - Verwenden Sie den folgenden Befehl, um eine SQL-Ausgabe zu extrahieren:

     ```
     gzip backup.sql.gz -d
     ```

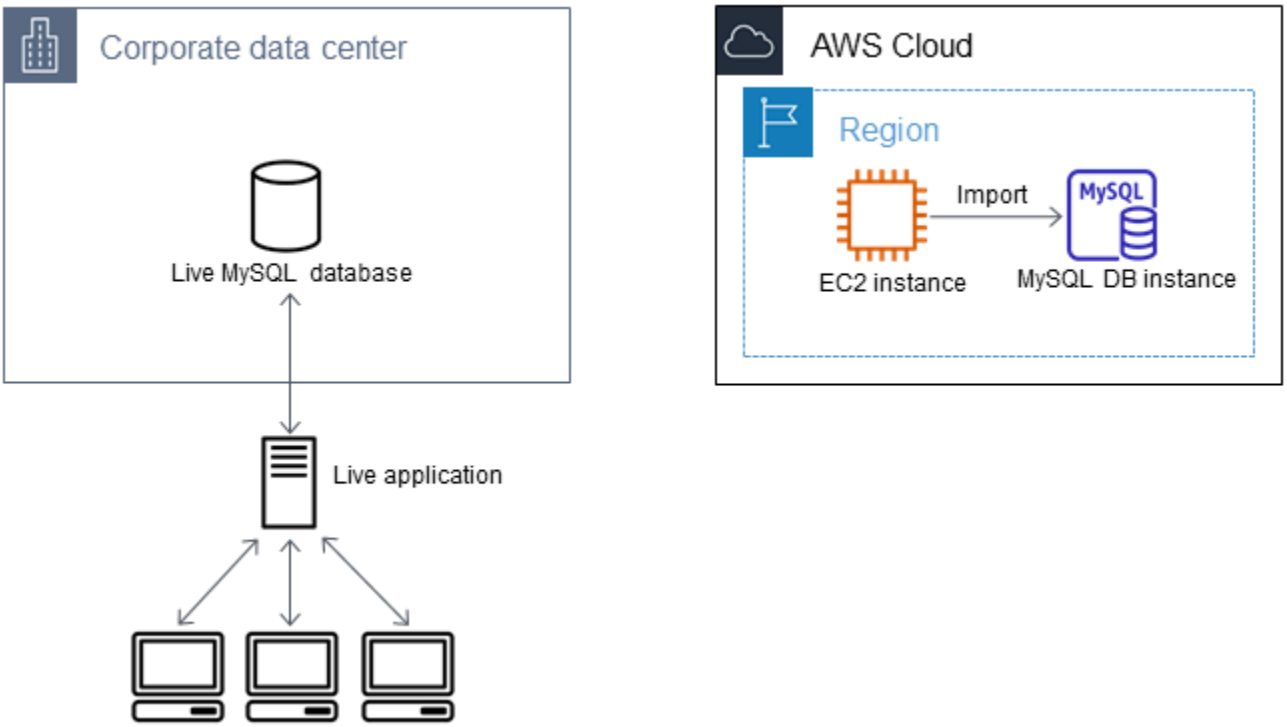
   - Verwenden Sie den folgenden Befehl, um eine separierte Textausgabe zu extrahieren:

     ```
     tar xzvf backup.tar.gz
     ```

Aufgabe 3: Erstellen Sie eine MySQL- oder MariaDB-Datenbank und importieren Sie Daten aus Ihrer Amazon-Instance EC2

Indem Sie eine RDS for MariaDB-DB-Instance, eine RDS for MySQL-DB-Instance oder einen RDS for MySQL Multi-AZ-DB-Cluster in derselben Umgebung AWS-Region wie Ihre EC2 Amazon-Instance erstellen, können Sie die Datenbank-Backup-Datei EC2 schneller als über das Internet von Amazon importieren.

Das folgende Diagramm zeigt den Import des Backups aus einer EC2 Amazon-Instance in eine MySQL-Datenbank.



So erstellen Sie eine MariaDB- oder MySQL-Datenbank und importieren Ihre Daten

1. Bestimmen Sie, welche DB-Instance-Klasse und wie viel Speicherplatz erforderlich sind, um den erwarteten Workload für diese Amazon-RDS-Datenbank unterstützen zu können. Bei diesem Vorgang sollten Sie auch entscheiden, wie viel Speicherplatz und Verarbeitungskapazität für Ihre Datenladevorgänge ausreichen. Entscheiden Sie außerdem, was zur Bewältigung der Produktionslast erforderlich ist. Sie können diese Faktoren anhand der Größe und der Ressourcen Ihrer Quell-MariaDB- oder MySQL-Datenbank einschätzen. Weitere Informationen finden Sie unter .
2. Erstellen Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster in dem AWS-Region , der Ihre EC2 Amazon-Instance enthält.

   Um einen RDS for MySQL Multi-AZ-DB-Cluster zu erstellen, folgen Sie den Anweisungen unterEinen Multi-AZ-DB-Cluster für Amazon RDS erstellen.

   Um eine RDS for MariaDB- oder RDS for MySQL-DB-Instance zu erstellen, folgen Sie den Anweisungen unter Erstellen einer Amazon RDS-DB-Instance und verwenden Sie die folgenden Richtlinien:

   - Geben Sie eine DB-Engine-Version an, die mit Ihrer Quell-DB-Instance kompatibel ist.
     - Wenn Ihre Quell-Instance MySQL 5.5.x ist, muss Ihre Amazon-RDS-DB-Instance MySQL sein.
     - Wenn Ihre Quell-Instance MySQL 5.6.x oder 5.7.x ist, muss Ihre Amazon RDS-DB-Instance MySQL oder MariaDB sein.
     - Wenn Ihre Quell-Instance MySQL 8.0.x ist, muss Ihre Amazon RDS-DB-Instance MySQL 8.0.x sein.
     - Wenn Ihre Quell-Instance MySQL 8.4.x ist, muss die Amazon RDS-DB-Instance MySQL 8.4.x sein.
     - Wenn Ihre Quell-Instance MariaDB 5.5 oder höher ist, muss Ihre Amazon-RDS-DB-Instance MariaDB sein.
   - Geben Sie dieselbe Virtual Private Cloud (VPC) und dieselbe VPC-Sicherheitsgruppe wie für Ihre EC2 Amazon-Instance an. Dieser Ansatz stellt sicher, dass Ihre EC2 Amazon-Instance und Ihre Amazon RDS-Instance im Netzwerk füreinander sichtbar sind. Stellen Sie sicher, dass Ihre DB-Instance öffentlich zugänglich ist. Um die Replikation mit Ihrer Quelldatenbank wie im folgenden Abschnitt beschrieben einzurichten, muss Ihre DB-Instance öffentlich zugänglich sein.

- Konfigurieren Sie nicht mehrere Availability Zones, Backup-Aufbewahrungen oder Lesereplikate, nachdem Sie das Datenbank-Backup importiert haben. Wenn dieser Importvorgang abgeschlossen ist, können Sie Multi-AZ und Backup-Aufbewahrung für die Produktions-Instance konfigurieren.

3. Überprüfen Sie die Optionen der Standardkonfiguration für die Amazon-RDS-Datenbank. Wenn in der Standardparametergruppe für die Datenbank die von Ihnen gewünschten Optionen nicht konfiguriert sind, wählen Sie eine andere aus, die die entsprechenden Konfigurationsoptionen enthält, oder erstellen Sie eine neue Parametergruppe. Weitere Hinweise zum Erstellen einer Parametergruppe finden Sie unterParametergruppen für Amazon RDS.
4. Stellen Sie als Hauptbenutzer eine Verbindung mit der neuen Amazon-RDS-Datenbank her. Erstellen Sie die Benutzer, die zur Unterstützung der Administratoren, Anwendungen und Dienste erforderlich sind, die auf die DB-Instance zugreifen müssen. Der Hostname für die Amazon RDS-Datenbank ist der Endpunktwert für diese DB-Instance ohne die Portnummer, zum Beispiel. `mysampledb.123456789012.us-west-2.rds.amazonaws.com` Sie finden den Endpunktwert in den Datenbankdetails in der Amazon RDS-Konsole.
5. Connect zu Ihrer EC2 Amazon-Instance her. Weitere Informationen finden Sie unter Connect Ihrer Instance für Linux-Instances herstellen im Amazon Elastic Compute Cloud-Benutzerhandbuch.
6. Stellen Sie mit dem `mysql` Befehl eine Connect zu Ihrer Amazon RDS-Datenbank als Remote-Host von Ihrer EC2 Amazon-Instance her. Der folgende Befehl ist ein Beispiel:

   ```
   mysql -h host_name -P 3306 -u db_master_user -p
   ```

   Das *`host_name`* ist der Amazon RDS-Datenbank-Endpunkt.
7. Führen Sie an der `mysql` Eingabeaufforderung den `source` Befehl aus und übergeben Sie ihm den Namen Ihrer Datenbank-Dump-Datei. Dieser Befehl lädt die Daten in die Amazon RDS-DB-Instance.

   - Verwenden Sie für das SQL-Format den folgenden Befehl:

     ```
     mysql> source backup.sql;
     ```

   - Für das Format mit begrenztem Text erstellen Sie zuerst die Datenbank, falls es sich nicht um die Standarddatenbank handelt, die Sie bei der Einrichtung der Amazon RDS-Datenbank erstellt haben.

```
mysql> create database database_name;
mysql> use database_name;
```

Erstellen Sie anschließend die Tabellen.

```
mysql> source table1.sql
mysql> source table2.sql
etc...
```

Importieren Sie dann die Daten.

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
etc...
```

Zur Verbesserung der Leistung können Sie diese Operationen parallel aus mehreren Verbindungen ausführen, damit alle Ihre Tabellen erstellt und die Daten anschließend gleichzeitig geladen werden.

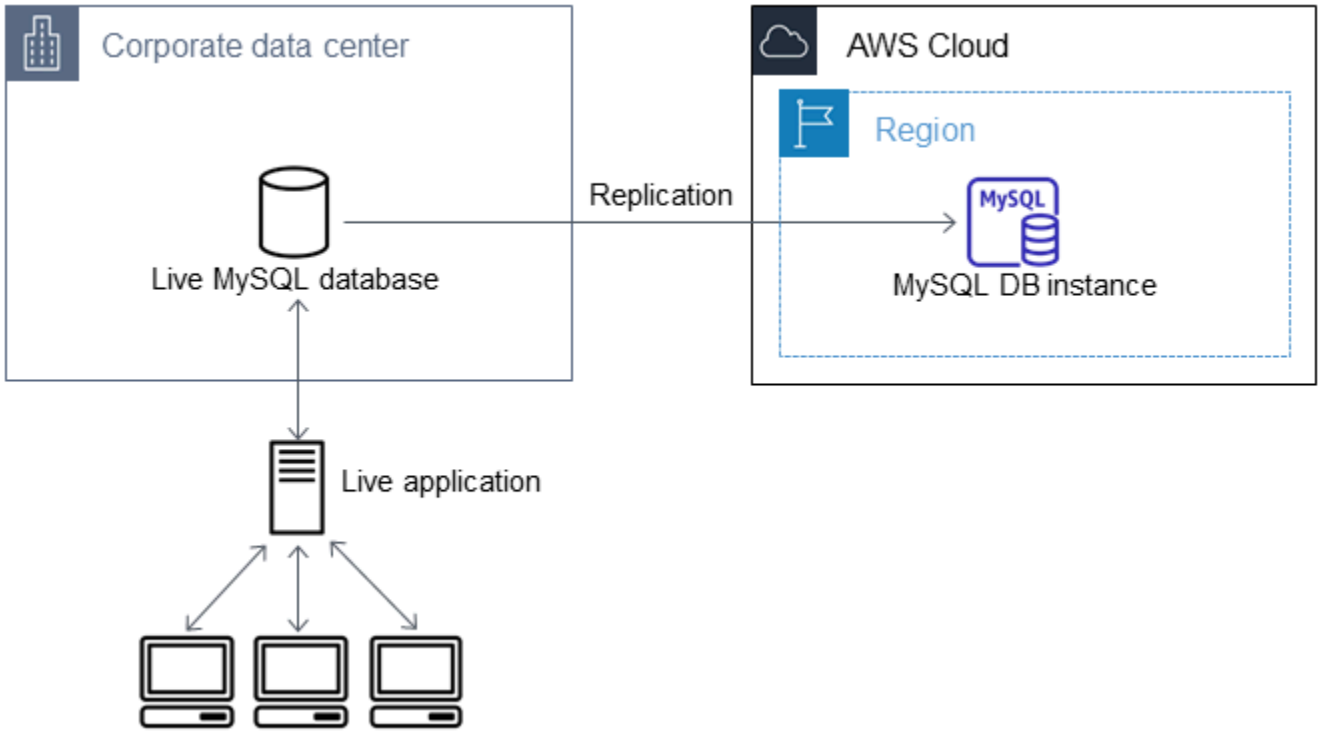
> **Note**
>
> Wenn Sie `mysqldump` beim ersten Speichern der Tabelle irgendwelche Datenformatierungsoptionen verwendet haben, stellen Sie sicher, dass Sie dieselben Optionen mit verwenden, um eine korrekte Interpretation des `LOAD DATA LOCAL INFILE` Inhalts der Datendatei sicherzustellen.

8. Führen Sie eine einfache `SELECT` Abfrage für eine oder zwei der Tabellen in der importierten Datenbank aus, um zu überprüfen, ob der Import erfolgreich war.

Wenn Sie die in diesem Verfahren verwendete EC2 Amazon-Instance nicht mehr benötigen, beenden Sie die EC2 Instance, um Ihren AWS Ressourcenverbrauch zu reduzieren. Informationen zum Beenden einer EC2 Instance finden Sie unter Kündigen einer Instance im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Aufgabe 4: Daten aus Ihrer externen Datenbank in Ihre neue Amazon RDS-Datenbank replizieren

Die Quelldatenbank wurde in der Zeit, in der die Daten in die MariaDB- oder MySQL-Datenbank kopiert und übertragen wurden, wahrscheinlich aktualisiert. Sie können also die Replikation verwenden, um die kopierte Datenbank up-to-date mit der Quelldatenbank zu verbinden.



Die zum Starten der Replikation auf einer Amazon RDS-Datenbank erforderlichen Berechtigungen sind eingeschränkt und stehen Ihrem Amazon RDS-Mastbenutzer nicht zur Verfügung. Verwenden Sie aus diesem Grund die entsprechende gespeicherte Amazon RDS-Prozedur für Ihre Haupt-Engine-Version:

-
-
- mysql.rds_set_external_master_gtidum die Replikation zu konfigurieren und mysql.rds_start_replication die Replikation zu starten

So starten Sie eine Replikation:

In Aufgabe 1 haben Sie bei der Festlegung der Replikationsoptionen die binäre Protokollierung aktiviert und eine eindeutige Server-ID für Ihre Quelldatenbank festgelegt. Jetzt können Sie Ihre

Amazon-RDS-Datenbank als Replikat mit Ihrer Live-Datenbank als Quellreplikations-Instance einrichten.

1. Fügen Sie in der Amazon RDS-Konsole die IP-Adresse des Servers, der die Quelldatenbank hostet, zur VPC-Sicherheitsgruppe für die Amazon RDS-Datenbank hinzu. Weitere Informationen zur Konfiguration einer VPC-Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln konfigurieren im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

   Möglicherweise müssen Sie auch Ihr lokales Netzwerk so konfigurieren, dass Verbindungen von der IP-Adresse Ihrer Amazon RDS-Datenbank zugelassen werden, damit diese mit Ihrer Quell-Instance kommunizieren kann. Verwenden Sie den `host` folgenden Befehl, um die IP-Adresse der Amazon RDS-Datenbank zu ermitteln:

   ```
   host host_name
   ```

   Das *`host_name`* ist zum Beispiel der DNS-Name vom Amazon RDS-Datenbank-Endpunkt`myinstance.123456789012.us-east-1.rds.amazonaws.com`. Sie finden den Endpunktwert in den DB-Instance-Details in der Amazon RDS-Konsole.

2. Verbinden Sie sich mithilfe eines Clients Ihrer Wahl mit der Quell-Instance und erstellen Sie einen Benutzer, der für die Replikation verwendet werden soll. Dieses Konto wird ausschließlich für die Replikation verwendet und muss auf Ihre Domäne beschränkt sein, um die Sicherheit zu erhöhen. Der folgende Befehl ist ein Beispiel:

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   > **Note**
   >
   > Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

3. Erteilen Sie für die Quell-Instance die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` für Ihren Replikationsbenutzer. Erteilen Sie beispielsweise die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` in allen Datenbanken für den `repl_user`-Benutzer für Ihre Domäne, mit dem folgenden Befehl:

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

4. Wenn Sie zum Erstellen Ihrer Sicherungsdatei das SQL-Format verwendet haben und die externe Instanz nicht MariaDB 10.0.24 oder höher ist, sehen Sie sich den Inhalt dieser Datei an, indem Sie den folgenden Befehl ausführen:

   ```
   cat backup.sql
   ```

   Die Datei beinhaltet einen `CHANGE MASTER TO`-Kommentar, der den Namen und die Position der Hauptprotokolldatei beinhaltet. Dieser Kommentar ist in der Sicherungsdatei enthalten, wenn Sie die Option `--master-data` mit `mysqldump` verwenden. Beachten Sie die Werte für `MASTER_LOG_FILE` und `MASTER_LOG_POS`.

   ```
   --
   -- Position to start replication or point-in-time recovery from
   --

   -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
   ```

   Wenn Sie zum Erstellen Ihrer Sicherungsdatei ein Textformat mit Trennzeichen verwendet haben und die externe Instanz nicht MariaDB 10.0.24 oder höher ist, sollten Sie bereits über binäre Protokollkoordinaten aus Schritt 1 der Prozedur So erstellen Sie eine Sicherungskopie Ihrer vorhandenen Datenbank unter Aufgabe 1 verfügen.

   Wenn es sich bei der externen Instanz um MariaDB 10.0.24 oder höher handelt, sollten Sie bereits über die GTID verfügen, von der aus Sie die Replikation starten können. Gehen Sie dazu aus Schritt 2 der Prozedur So erstellen Sie eine Sicherungskopie Ihrer vorhandenen Datenbank unter Aufgabe 1.

5. Konfigurieren Sie die Amazon-RDS-Datenbank als Replikat. Wenn die externe Instance nicht MariaDB 10.0.24 oder höher ist, stellen Sie als Masterbenutzer eine Verbindung zur Amazon RDS-Datenbank her und identifizieren Sie die Quelldatenbank als Quellreplikationsinstanz, indem Sie die entsprechende gespeicherte Amazon RDS-Prozedur verwenden:

   •

   •

   Wenn Sie über eine Sicherungsdatei im SQL-Format verfügen, verwenden Sie den Namen der Master-Log-Datei und die Master-Log-Position, die Sie in Schritt 4 festgelegt haben. Wenn Sie das Format mit Trennzeichen verwendet haben, verwenden Sie den Namen und die Position, die Sie bei der Erstellung der Sicherungsdateien festgelegt haben. Die folgenden Befehle sind Beispiele:

MySQL 8.4 und höhere Versionen

```
CALL mysql.rds_set_external_source ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

MariaDB und MySQL 8.0 und niedrigere Versionen

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

Wenn die externe Instance MariaDB 10.0.24 oder höher ausführt, stellen Sie eine Verbindung mit der Amazon-RDS-Datenbank als Hauptbenutzer her und identifizieren Sie die Quelldatenbank mithilfe des Befehls mysql.rds_set_external_master_gtid als Quellreplikations-Instance. Verwenden Sie die GTID, die Sie in Schritt 2 der Prozedur So erstellen Sie eine Sicherungskopie Ihrer vorhandenen Datenbank unter Aufgabe 1 festgelegt haben. Der folgende Befehl ist ein Beispiel:

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 1);
```

Die *source_server_ip_address* ist die IP-Adresse der Quellreplikationsinstance. Eine EC2 private DNS-Adresse wird derzeit nicht unterstützt.

> **Note**
>
> Geben Sie aus Sicherheitsgründen andere Anmeldeinformationen als hier angegeben an.

6. Führen Sie in der Amazon RDS-Datenbank den folgenden mysql.rds_start_replication Befehl aus, um die Replikation zu starten:

```
CALL mysql.rds_start_replication;
```

7. Führen Sie in der Amazon RDS-Datenbank den Befehl SHOW REPLICA STATUS aus, um zu ermitteln, wann das Replikat mit der Quellreplikationsinstanz auf dem neuesten Stand ist. Zu den Ergebnissen des `SHOW REPLICA STATUS`-Befehls gehört das Feld `Seconds_Behind_Master`. Wenn das `Seconds_Behind_Master` Feld 0 zurückgibt, entspricht das Replikat der Quellreplikationsinstanz.

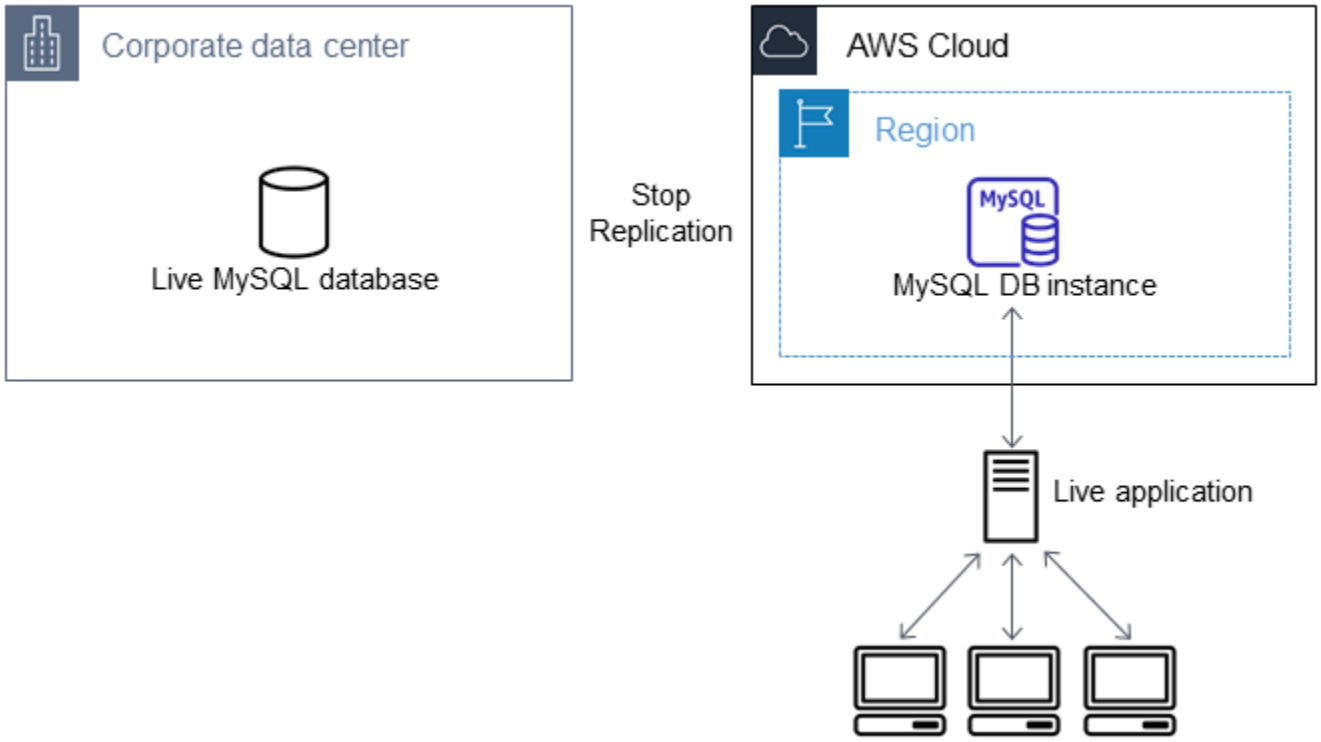
   > **Note**
   >
   > Frühere Versionen von MySQL verwenden `SHOW SLAVE STATUS` anstelle von `SHOW REPLICA STATUS`. Wenn Sie vor 8.0.23 eine MySQL-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

   Führen Sie für eine MariaDB 10.5-, 10.6- oder 10.11-DB-Instance die mysql.rds_replica_status gespeicherte Prozedur anstelle des MySQL-Befehls aus.

8. Sobald die Amazon RDS-Datenbank auf dem neuesten Stand ist, aktivieren Sie automatische Backups, damit Sie diese Datenbank bei Bedarf wiederherstellen können. Sie können automatische Backups für Ihre Amazon RDS-Datenbank mithilfe der Amazon RDS-Konsole aktivieren oder ändern. Weitere Informationen finden Sie unter Einführung in Backups.

Aufgabe 5: Leiten Sie Ihre Live-Anwendung auf Ihre Amazon RDS-Instance weiter

Nachdem die MariaDB- oder MySQL-Datenbank mit der Quellreplikationsinstanz auf dem neuesten Stand ist, können Sie jetzt Ihre Live-Anwendung aktualisieren, um die Amazon RDS-Instance zu verwenden.



So leiten Sie Ihre Live-Anwendung an die MariaDB- oder MySQL-Datenbank weiter und halten die Replikation an

1. Fügen Sie die IP-Adresse des Host-Servers der Anwendung hinzu, um die VPC-Sicherheitsgruppe für Ihre Amazon RDS-Datenbank hinzuzufügen. Weitere Informationen zum Ändern einer VPC-Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln konfigurieren im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
2. Stellen Sie sicher, dass das `Seconds_Behind_Master` Feld in den Ergebnissen des Befehls SHOW REPLICA STATUS den Wert 0 hat. Dies bedeutet, dass das Replikat mit der Quellreplikationsinstanz auf dem neuesten Stand ist.

```
SHOW REPLICA STATUS;
```

> **Note**
>
> Frühere Versionen von MySQL verwenden `SHOW SLAVE STATUS` anstelle von `SHOW REPLICA STATUS`. Wenn Sie vor 8.0.23 eine MySQL-Version verwenden, verwenden Si `SHOW SLAVE STATUS`.

Bei einer DB-Instance von MariaDB 10.5, 10.6 oder 10.11 führen Sie das Verfahren mysql.rds_replica_status anstelle des MySQL-Befehls aus.

3. Schließen Sie alle Verbindungen zur Quelle, nachdem ihre Transaktionen abgeschlossen sind.
4. Aktualisieren Sie Ihre Anwendung, um die Amazon-RDS-Datenbank zu nutzen. Dieses Update ändert die Verbindungseinstellungen, um den Hostnamen und den Port der Amazon-RDS-Datenbank, das Benutzerkonto und Passwort für die Verbindung und die zu verwendende Datenbank zu bestimmen.
5. Stellen Sie eine Verbindung mit der DB-Instance her.

   Stellen Sie für einen Multi-AZ-DB-Cluster eine Verbindung mit der Writer-DB-Instance her.
6. Beenden Sie die Replikation für die Amazon RDS-Instance mit dem folgenden mysql.rds_stop_replication Befehl:

   ```
   CALL mysql.rds_stop_replication;
   ```
7. Führen Sie die entsprechende gespeicherte Amazon RDS-Prozedur in Ihrer Amazon RDS-Datenbank aus, um die Replikationskonfiguration zurückzusetzen, sodass diese Instance nicht mehr als Replikat identifiziert wird.

   •

   •

   MySQL 8.4 und höhere Versionen

   ```
   CALL mysql.rds_reset_external_source;
   ```

   MariaDB und MySQL 8.0 und niedrigere Versionen

   ```
   CALL mysql.rds_reset_external_master;
   ```
8. Aktivieren Sie zusätzliche Amazon-RDS-Funktionen, wie Multi-AZ-Unterstützung und Lesereplikate. Weitere Informationen erhalten Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS und Arbeiten mit DB-Instance-Lesereplikaten.

Daten aus einer beliebigen Quelle in eine Amazon RDS for MySQL MySQL-DB-Instance importieren

Die folgenden Schritte führen Sie durch den Import von Daten in eine Amazon RDS-DB-Instance:

Schritt 1: Erstellen Sie flache Dateien, die die zu ladenden Daten enthalten

Verwenden Sie ein gängiges Format, z. B. kommagetrennte Werte (CSV), um die zu ladenden Daten zu speichern. Jede Tabelle muss eine eigene Datei haben — Sie können keine Daten für mehrere Tabellen in derselben Datei kombinieren. Geben Sie jeder Datei denselben Namen wie der zugehörigen Tabelle. Die Dateierweiterung können Sie benennen, wie Sie möchten. Wenn der Tabellenname beispielsweise lautet`sales`, könnte der Dateiname oder lauten`sales.csv`. `sales.txt`

Wenn möglich, ordnen Sie die Daten nach dem Primärschlüssel der Tabelle, die geladen wird. Dadurch werden die Ladezeiten drastisch verbessert und die Anforderungen an den Festplattenspeicher minimiert.

Die optimale Geschwindigkeit und Effizienz dieser Prozedur ist auf kleine Dateigrößen ausgelegt. Wenn die Größe einer einzelnen unkomprimierten Datei mehr als 1 GiB beträgt, teilen Sie diese in mehrere Dateien auf, die danach separat geladen werden können.

Verwenden Sie auf Unix-ähnlichen Systemen (einschließlich Linux) den `split`-Befehl. Der folgende Befehl teilt beispielsweise die Datei `sales.csv` in mehrere Dateien auf, die kleiner als 1 GiB sind. Die Teilung findet nur an Zeilenumbrüchen statt (-C 1024m). Die Namen der neuen Dateien enthalten aufsteigende numerische Suffixe. Der folgende Befehl erzeugt Dateien mit Namen wie und. `sales.part_00` `sales.part_01`

```
split -C 1024m -d sales.csv sales.part_
```

Ähnliche Hilfsprogramme sind auch für andere Betriebssysteme verfügbar.

Sie können die Flatfiles überall speichern. Wenn Sie die Daten in Schritt 5 laden, müssen Sie die `mysql` Shell jedoch von demselben Speicherort aufrufen, an dem sich die Dateien befinden, oder Sie müssen bei der Ausführung `LOAD DATA LOCAL INFILE` den absoluten Pfad für die Dateien verwenden.

Schritt 2: Verhindern Sie, dass alle Anwendungen auf die Ziel-DB-Instance zugreifen

Bevor Sie einen großen Ladevorgang starten, beenden Sie den Zugriff aller Anwendungsaktivitäten auf die Ziel-DB-Instance, in die Sie laden möchten. Wir empfehlen dies insbesondere, wenn andere Sitzungen die zu ladenden Tabellen oder die in diesen Tabellen referenzierten Tabellen verändern. Dadurch wird die Gefahr von Verstößen gegen Einschränkungen während des Ladens reduziert und zugleich die Ausführungsgeschwindigkeit des Ladevorgangs erhöht. Zudem wird es möglich, die DB-Instance im Zustand unmittelbar vor dem Ladevorgang wiederherzustellen, ohne die Änderungen durch Prozesse zu verlieren, die nicht am Ladevorgang beteiligt sind.

Unter Umständen ist dies jedoch nicht möglich oder nicht praktikabel. Wenn Sie vor dem Ladevorgang den Zugriff von Anwendungen auf die DB-Instance nicht stoppen können, führen Sie die erforderlichen Schritte aus, um die Verfügbarkeit und Integrität der Daten sicherzustellen. Die jeweiligen erforderlichen Schritte können sich stark unterscheiden, je nachdem welche besonderen Verwendungsfälle der Standortanforderungen vorliegen.

Schritt 3: Erstellen Sie einen DB-Snapshot

Wenn Sie Daten in eine neue DB-Instance laden wollen, die keine Daten enthält, können Sie diesen Schritt überspringen. Andernfalls empfehlen wir, vor und nach dem Laden der Daten DB-Snapshots der Amazon RDS-DB-Zielinstanz zu erstellen. Amazon RDS-DB-Snapshots sind vollständige Backups Ihrer DB-Instance, mit denen Sie Ihre DB-Instance in einen bekannten Zustand zurückversetzen können. Wenn Sie ein DB-Snapshot-I/O initiieren, werden alle Operationen in Ihrer DB-Instance augenblicklich unterbrochen, während Ihre Datenbank gesichert wird.

Wenn Sie einen DB-Snapshot unmittelbar vor dem Laden erstellen, können Sie die Datenbank bei Bedarf in ihrem Zustand vor dem Laden wiederherstellen. Ein DB-Snapshot, der sofort nach einem Ladevorgang erstellt wird, bewahrt Sie davor, die Daten bei einem Fehler erneut laden zu müssen. Sie können DB-Snapshots auch nach dem Laden verwenden, um Daten in neue Datenbank-Instances zu importieren.

Im folgenden Beispiel wird der AWS CLI create-db-snapshotBefehl ausgeführt, um einen DB-Snapshot der `AcmeRDS` Instance zu erstellen und dem DB-Snapshot die ID `"preload"` zu geben.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Sie können auch die Wiederherstellung aus der DB-Snapshot-Funktionalität verwenden, um Test-DB-Instances für Testversuche zu erstellen, oder, um die Änderungen während eines Ladevorgangs rückgängig zu machen.

Bedenken Sie, dass die Wiederherstellung einer Datenbank aus einem DB-Snapshot eine neue DB-Instance erstellt, die wie alle DB-Instances über eine eindeutige Kennung und einen Endpunkt verfügt. Um die DB-Instance wiederherzustellen, ohne den Endpunkt zu ändern, löschen Sie zuerst die DB-Instance, damit Sie den Endpunkt wiederverwenden können.

Um beispielsweise eine DB-Instance für einen Testlauf oder andere Testzwecke zu erstellen, weisen Sie der DB-Instance eine eigene Kennung zu. Im Beispiel ist AcmeRDS-2" die Kennung. Das Beispiel stellt über den Endpunkt, der AcmeRDS-2 zugeordnet ist, eine Verbindung mit der DB-Instance her. Weitere Informationen finden Sie unter restore-db-instance-from-db-snapshot.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Um einen bestehenden Endpunkt erneut zu verwenden, löschen Sie zuerst die DB-Instance und weisen Sie dann der wiederhergestellten Datenbank dieselbe Kennung zu. Weitere Informationen finden Sie unter delete-db-instance.

Im folgenden Beispiel wird auch ein letzter DB-Snapshot der DB-Instance erstellt, bevor sie gelöscht wird. Dies ist zwar optional, wird aber empfohlen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Schritt 4 (optional): Automatische Amazon RDS-Backups ausschalten

> **Warning**
>
> Schalten Sie automatische Backups nicht aus, wenn Sie eine point-in-time Wiederherstellung durchführen müssen.

Das Ausschalten automatisierter Backups dient der Leistungsoptimierung und ist für das Laden von Daten nicht erforderlich. Durch das Deaktivieren automatisierter Backups werden alle vorhandenen Backups gelöscht. Daher ist eine point-in-time Wiederherstellung nicht möglich, nachdem Sie automatische Backups deaktiviert haben. Beachten Sie, dass manuelle DB-Snapshots nicht von der Deaktivierung automatischer Backups betroffen sind. Alle bestehenden manuellen DB-Snapshots bleiben für eine Wiederherstellung verfügbar.

Das Deaktivieren von automatischen Backups reduziert die Ladezeit um 25 % und verringert zugleich den während des Ladevorgangs erforderlichen Speicherplatz. Wenn Sie Daten in eine neue DB-Instance laden wollen, die keine Daten enthält, ist das Deaktivieren von Backups eine einfache Möglichkeit, die Übertragungsgeschwindigkeit zu erhöhen und zusätzlichen Speicherverbrauch zu vermeiden. In einigen Fällen können Sie jedoch in eine DB-Instance laden, die bereits Daten enthält.

Wenn ja, sollten Sie die Vorteile der Deaktivierung von Backups gegen die Auswirkungen eines Verlusts der Leistungsfähigkeit abwägen point-in-time-recovery.

In DB-Instances ist die Funktion für automatische Backups standardmäßig aktiviert (mit einem Aufbewahrungszeitraum von 1 Tag). Setzen Sie den Wert des Aufbewahrungszeitraums für Backups auf 0, um automatische Backups zu deaktivieren. Nach dem Ladevorgang können Sie Backups erneut aktivieren, indem Sie den Aufbewahrungszeitraum für Backups auf einen Nicht-Null-Wert setzen. Um Backups ein- oder auszuschalten, fährt Amazon RDS die DB-Instance herunter und startet sie dann neu, um die MariaDB- oder MySQL-Protokollierung ein- oder auszuschalten.

Führen Sie den AWS CLI `modify-db-instance` Befehl aus, um die Backup-Aufbewahrung auf Null zu setzen und die Änderung sofort zu übernehmen. Das Setzen des Aufbewahrungszeitraums für Backups auf Null erfordert den Neustart einer DB-Instance. Warten Sie daher bitte, bis der Neustart abgeschlossen wurde, bevor Sie fortfahren. Weitere Informationen finden Sie unter modify-db-instance.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Sie können den Status Ihrer DB-Instance mit dem AWS CLI describe-db-instancesBefehl überprüfen. Im folgenden Beispiel wird der DB-Instance-Status der `AcmeRDS` DB-Instance angezeigt:

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Wenn der DB-Instance-Status lautet`available`, können Sie mit dem nächsten Schritt fortfahren.

Schritt 5: Laden Sie die Daten

Verwenden Sie die `LOAD DATA LOCAL INFILE` MySQL-Anweisung, um Zeilen aus Ihren Flatfiles in die Datenbanktabellen zu lesen.

> **Note**
>
> Sie müssen die `mysql` Shell von demselben Ort aufrufen, an dem sich Ihre Flatfiles befinden, oder bei der Ausführung `LOAD DATA LOCAL INFILE` den absoluten Pfad für die Dateien verwenden.

Das folgende Beispiel zeigt, wie Daten aus einer Datei mit dem Namen `sales.txt` in eine `Sales` in der Datenbank benannte Tabelle geladen werden:

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Weitere Informationen zu der `LOAD DATA` Anweisung finden Sie unter LOAD DATA Statement in der MySQL-Dokumentation.

Schritt 6: Automatisierte Amazon RDS-Backups wieder aktivieren

Wenn Sie in Schritt 4 automatische Amazon RDS-Backups deaktiviert haben, schalten Sie nach Abschluss des Ladevorgangs automatische Backups ein, indem Sie die Aufbewahrungsfrist für Backups wieder auf den Wert vor dem Laden setzen. Wie in Schritt 4 erwähnt, startet Amazon RDS die DB-Instance neu. Seien Sie also auf einen kurzen Ausfall vorbereitet.

Im folgenden Beispiel wird der AWS CLI modify-db-instanceBefehl ausgeführt, um automatische Backups für die `AcmeRDS` DB-Instance zu aktivieren und die Aufbewahrungsfrist auf einen Tag festzulegen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
```

```
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Arbeiten mit MySQL-Replikation in Amazon RDS

Sie verwenden Lesereplikate üblicherweise, um die Replikation zwischen Amazon RDS-DB-Instances zu konfigurieren. Allgemeine Informationen zu Lesereplikaten finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten. Spezifische Informationen zum Arbeiten mit Lesereplikaten unter Amazon RDS for MySQL finden Sie unter Mit My SQL Read Replicas arbeiten.

Sie können globale Transaktions-Identifikatoren (GTIDs) für die Replikation mit RDS for MySQL verwenden. Weitere Informationen finden Sie unter Verwenden der GTID basierten Replikation.

Sie können auch eine Replikation zwischen einer RDS for MySQL-DB-Instance und einer MariaDB- oder MySQL-Instance, die außerhalb von Amazon RDS ausgeführt wird, einrichten. Informationen zum Konfigurieren der Replikation mit einer externen Quelle finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

Für jede dieser Replikationsoptionen können Sie die Replikation vom Typ „row-based“, „statement-based“ oder „mixed“ verwenden. Die Replikation vom Typ „row-based“ repliziert nur die geänderten Zeilen, die sich aus einer SQL-Anweisung ergeben. Die Replikation vom Typ „statement-based“ repliziert die gesamte SQL-Anweisung. Die Replikation vom Typ „mixed“ verwendet nach Möglichkeit Replikation vom Typ „statement-based“, wechselt aber auf Replikation vom Typ „row-based“, wenn SQL-Anweisungen ausgeführt werden, die bei der Replikation vom Typ „statement-based“ nicht sicher sind. In den meisten Fällen wird eine Replikation vom Typ „mixed“ empfohlen. Das binäre Protokollformat der DB-Instance bestimmt, ob die Replikation vom Typ „row-based“, „statement-based“ oder „mixed“ ist. Weitere Informationen zum Einrichten des binären Protokollformats finden Sie unter Konfiguration von RDS für MySQL-Binärprotokollierung für Single-AZ-Datenbanken.

> **Note**
>
> Sie können die Replikation zum Importieren von Datenbanken aus einer MariaDB- oder MySQL-Instance, die außerhalb von Amazon RDS ausgeführt wird, oder zum Exportieren von Datenbanken in solche Instances konfigurieren. Weitere Informationen erhalten Sie unter Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit und Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation.

Nachdem Sie Ihre DB-Instance aus einem Snapshot wiederhergestellt oder eine point-in-time Wiederherstellung durchgeführt haben, können Sie die zuletzt wiederhergestellte Binlog-Position aus

der Quelldatenbank in der RDS-Konsole anzeigen. Geben Sie unter Logs & Events binlog ein. Die Binlog-Position wird unter Systemnotizen angezeigt.

Themen

- Mit My SQL Read Replicas arbeiten
- Verwenden der GTID basierten Replikation
- Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance
- Konfiguration multi-source-replication für Amazon RDS for MySQL

Mit My SQL Read Replicas arbeiten

Im Folgenden finden Sie spezifische Informationen zur Arbeit mit Read Replicas auf RDS for My. SQL Allgemeine Informationen zu Lesereplikaten und Anleitungen zu ihrer Verwendung finden Sie in Arbeiten mit DB-Instance-Lesereplikaten.

Weitere Informationen zu My SQL Read Replicas finden Sie in den folgenden Themen.

- Konfiguration von Replikationsfiltern mit My SQL
- Konfiguration der verzögerten Replikation mit My SQL
- Aktualisierung von Read Replicas mit My SQL
- Arbeiten mit Multi-AZ-Read Replica-Bereitstellungen mit My SQL
- Verwenden von kaskadierenden Read Replicas mit for My RDS SQL
- Überwachung der Replikationsverzögerung für My SQL Read Replicas
- Starten und Beenden der Replikation mit My SQL Read Replicas
- Behebung eines My SQL Read Replica-Problems

Konfiguration von Read Replicas mit My SQL

Bevor eine My SQL DB-Instance als Replikationsquelle dienen kann, stellen Sie sicher, dass automatische Backups auf der Quell-DB-Instance aktiviert sind. Hierzu legen Sie für den Aufbewahrungszeitraum für Sicherungen einen anderen Wert als 0 fest. Diese Erfordernis gilt auch für ein Lesereplikat, das die Quell-DB-Instance für ein anderes Lesereplikat ist. Automatische Backups werden für Read Replicas unterstützt, auf denen eine beliebige Version von My SQL ausgeführt wird. Sie können die Replikation auf der Grundlage binärer Protokollkoordinaten für eine My SQL DB-Instance konfigurieren.

Sie können die Replikation mithilfe globaler Transaktions-Identifikatoren (GTIDS) in den folgenden Versionen konfigurieren:

- RDSfür meine SQL Version 5.7.44 und höhere 5.7-Versionen
- RDSfür meine SQL Version 8.0.28 und höhere 8.0-Versionen
- RDSfür meine SQL Version 8.4.3 und höhere 8.4-Versionen

Weitere Informationen finden Sie unter Verwenden der GTID basierten Replikation.

Sie können bis zu 15 Lesereplikate von einer DB-Instance innerhalb derselben Region erstellen. Damit die Replikation effektiv durchgeführt werden kann, sollte jedes Lesereplikat über die selbe Menge an Rechen- und Speicherressourcen wie die Quell-DB-Instance verfügen. Wenn Sie die Quell-DB-Instance skalieren, skalieren Sie auch die Lesereplikate.

RDSfor My SQL unterstützt kaskadierende Read Replicas. Informationen zum Konfigurieren von kaskadierenden Lesereplikaten finden Sie unter Verwenden von kaskadierenden Read Replicas mit for My RDS SQL.

Sie können mehrere Erstellungs- und Löschaktionen für Lesereplikate gleichzeitig ausführen, die auf die gleiche Quell-DB-Instance verweisen. Halten Sie sich bei der Ausführung dieser Aktionen an die Grenze von 15 Lesereplikaten für jede Quell-Instance.

Eine Read Replica einer My SQL DB-Instance kann keine niedrigere DB-Engine-Version als ihre Quell-DB-Instance verwenden.

Meine SQL DB-Instances, die My verwenden, werden vorbereitet ISAM

Wenn Ihre My SQL DB-Instance eine nicht transaktionale Engine wie My verwendetISAM, müssen Sie die folgenden Schritte ausführen, um Ihre Read Replica erfolgreich einzurichten. Diese Schritte sind unbedingt notwendig, damit das Lesereplikat eine konsistente Kopie Ihrer Daten enthält. Diese Schritte sind nicht erforderlich, wenn alle Ihre Tabellen eine transaktionale Engine wie InnoDB nutzen.

1. Beenden Sie alle Operationen mit Datenmanipulationssprache (DML) und Datendefinitionssprache (DDL) für Tabellen, die keine Transaktionen sind, in der Quell-DB-Instance und warten Sie, bis sie abgeschlossen sind. SELECTAnweisungen können weiter ausgeführt werden.
2. Bereinigen und sperren Sie die Tabellen in der Quell-DB-Instance.
3. Erstellen Sie das Lesereplikat mithilfe der Methoden in den folgenden Abschnitten.

4. Überprüfen Sie den Fortschritt der Read Replica-Erstellung, indem Sie beispielsweise den `DescribeDBInstances` API Vorgang verwenden. Sobald das Lesereplikat verfügbar ist, entsperren Sie die Tabellen in der Quell-DB-Instance und fahren Sie mit den normalen Datenbank-Operationen fort.

Konfiguration von Replikationsfiltern mit My SQL

Sie können Replikationsfilter verwenden, um anzugeben, welche Datenbanken und Tabellen mit einem Lesereplikat repliziert werden. Replikationsfilter können Datenbanken und Tabellen in die Replikation einbeziehen oder sie von der Replikation ausschließen.

Im Folgenden finden Sie einige Anwendungsfälle für Replikationsfilter:

- Reduzieren der Größe eines Lesereplikats. Mit Replikationsfiltern können Sie die Datenbanken und Tabellen ausschließen, die für das Lesereplikat nicht benötigt werden.
- Ausschließen von Datenbanken und Tabellen von Lesereplikaten aus Sicherheitsgründen.
- Replizieren verschiedener Datenbanken und Tabellen für spezifische Anwendungsfälle bei verschiedenen Lesereplikaten. Beispielsweise könnten Sie bestimmte Lesereplikate für Analysen oder Sharding verwenden.
- Für eine DB-Instance, die Read Replicas in verschiedenen hat AWS-Regionen, um verschiedene Datenbanken oder Tabellen in verschiedenen zu replizieren. AWS-Regionen

> **Note**
>
> Sie können auch Replikationsfilter verwenden, um anzugeben, welche Datenbanken und Tabellen mit einer primären My SQL DB-Instance repliziert werden, die als Replikat in einer Topologie für eingehende Replikation konfiguriert ist. Weitere Informationen zu dieser Konfiguration finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

Themen

- Einstellung der Replikationsfilterparameter für My RDS SQL
- Einschränkungen bei der Replikationsfilterung RDS für My SQL
- Beispiele für Replikationsfilter RDS für My SQL
- Anzeigen der Replikationsfilter für ein Lesereplikat

Einstellung der Replikationsfilterparameter für My RDS SQL

Um Replikationsfilter zu konfigurieren, legen Sie die folgenden Filterparameter für die Replikation fest:

- `replicate-do-db` – Repliziert Änderungen der angegebenen Datenbanken. Wenn Sie diesen Parameter für ein Lesereplikat festlegen, werden nur die im Parameter angegebenen Datenbanken repliziert.
- `replicate-ignore-db` – Repliziert keine Änderungen der angegebenen Datenbanken. Wenn der Parameter `replicate-do-db` für ein Lesereplikat festgelegt ist, wird dieser Parameter nicht ausgewertet.
- `replicate-do-table` – Repliziert Änderungen der angegebenen Tabellen. Wenn Sie diesen Parameter für ein Lesereplikat festlegen, werden nur die im Parameter angegebenen Tabellen repliziert. Wenn der Parameter `replicate-do-db` oder `replicate-ignore-db` festgelegt ist, müssen Sie die Datenbank, welche die angegebenen Tabellen enthält, in die Replikation mit dem Lesereplikat einbeziehen.
- `replicate-ignore-table` – Repliziert keine Änderungen der angegebenen Tabellen. Wenn der Parameter `replicate-do-table` für ein Lesereplikat festgelegt ist, wird dieser Parameter nicht ausgewertet.
- `replicate-wild-do-table` – Repliziert Tabellen basierend auf den angegebenen Namensmustern für Datenbanken und Tabellen. Die Platzhalterzeichen `%` und `_` werden unterstützt. Wenn der Parameter `replicate-do-db` oder `replicate-ignore-db` festgelegt ist, müssen Sie die Datenbank, welche die angegebenen Tabellen enthält, in die Replikation mit dem Lesereplikat einbeziehen.
- `replicate-wild-ignore-table` – Repliziert keine Tabellen basierend auf den angegebenen Namensmustern für Datenbanken und Tabellen. Die Platzhalterzeichen `%` und `_` werden unterstützt. Wenn die Parameter `replicate-do-table` oder `replicate-wild-do-table` für ein Lesereplikat festgelegt sind, wird dieser Parameter nicht ausgewertet.

Die Parameter werden in der angegebenen Reihenfolge ausgewertet. Weitere Informationen zur Funktionsweise dieser Parameter finden Sie in der SQL Dokumentation Meine:

- Allgemeine Informationen finden Sie unter Optionen und Variablen für Replikatserver.
- Informationen darüber, wie Filterparameter für die Datenbankreplikation ausgewertet werden, finden Sie unter Optionen zur Auswertung der Replikation auf Datenbankebene und für die binäre Protokollierung.

- Informationen darüber, wie Filterparameter für die Tabellenreplikation ausgewertet werden, finden Sie unter Optionen zur Auswertung der Replikation auf Tabellenebene.

Standardmäßig hat jeder dieser Parameter einen leeren Wert. Sie können diese Parameter auf jedem Lesereplikat verwenden, um Replikationsfilter festzulegen, zu ändern und zu löschen. Wenn Sie einen dieser Parameter festlegen, trennen Sie die einzelnen Filter durch ein Komma voneinander.

Sie können die Platzhalterzeichen `%` und `_` in den Parametern `replicate-wild-do-table` und `replicate-wild-ignore-table` verwenden. Der Platzhalter `%` entspricht einer beliebigen Anzahl von Zeichen, und der Platzhalter `_` entspricht nur einem Zeichen.

Das binäre Protokollierungsformat der Quell-DB-Instance ist wichtig für die Replikation, da es den Datensatz der Datenänderungen bestimmt. Die Einstellung des Parameters `binlog_format` bestimmt, ob die Replikation zeilenbasiert oder anweisungsbasiert ist. Weitere Informationen finden Sie unter Konfiguration von RDS für MySQL-Binärprotokollierung für Single-AZ-Datenbanken.

> **Note**
>
> Alle Anweisungen der Data Definition Language (DDL) werden als Anweisungen repliziert, unabhängig von der `binlog_format` Einstellung in der Quell-DB-Instance.

Einschränkungen bei der Replikationsfilterung RDS für My SQL

Die folgenden Einschränkungen gelten für die Replikationsfilterung RDS für MySQL:

- Jeder Filterparameter für die Replikation hat ein Limit von 2.000 Zeichen.
- Kommas werden in Replikationsfiltern für Parameterwerte nicht unterstützt. In einer Liste von Parametern können Kommas nur als Werttrennzeichen verwendet werden. Wird beispielsweise `ParameterValue='`a,b`'` nicht unterstützt, ist es aber`ParameterValue='a,b'`.
- Die `--binlog-ignore-db` Optionen My SQL `--binlog-do-db` und für die Filterung von Binärprotokollen werden nicht unterstützt.
- Die Replikationsfilterung unterstützt keine XA-Transaktionen.

  Weitere Informationen finden Sie unter Einschränkungen für XA-Transaktionen in der SQL Dokumentation „Meine“.

Beispiele für Replikationsfilter RDS für My SQL

Um die Replikationsfilter für ein Lesereplikat zu konfigurieren, ändern Sie die Parameter der Replikationsfilter in der Parametergruppe, die dem Lesereplikat zugeordnet ist.

> **Note**
>
> Eine Standard-Parametergruppe kann nicht modifiziert werden. Erstellen Sie eine neue Parametergruppe und ordnen Sie diese der Lesereplika zu, wenn die Lesereplika eine Standardparametergruppe verwendet. Weitere Informationen zu DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Sie können Parameter in einer Parametergruppe mithilfe von AWS Management Console AWS CLI, oder festlegen RDSAPI. Weitere Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS. Wenn Sie Parameter in einer Parametergruppe festlegen, verwenden alle DB-Instances, die der Parametergruppe zugeordnet sind, diese Parametereinstellungen. Wenn Sie Parameter der Replikationsfilter in einer Parametergruppe festlegen, stellen Sie sicher, dass die Parametergruppe nur Lesereplikaten zugeordnet ist. Lassen Sie die Parameter der Replikationsfilter für Quell-DB-Instances leer.

In den folgenden Beispielen werden die Parameter mithilfe von festgeleg AWS CLI. Diese Beispiele sind `ApplyMethod` auf `immediate` so eingestellt, dass die Parameteränderungen unmittelbar nach Abschluss des CLI Befehls vorgenommen werden. Wenn Sie möchten, dass eine ausstehende Änderung nach dem Neustart des Lesereplikats angewendet wird, legen Sie `ApplyMethod` auf `pending-reboot` fest.

In den folgenden Beispielen werden Replikationsfilter festgelegt:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Einschließen von Datenbanken in die Replikation

Das folgende Beispiel schließt die Datenbanken `mydb1` und `mydb2` in die Replikation ein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Example Einschließen von Tabellen in die Replikation

Das folgende Beispiel schließt die Tabellen `table1` und `table2` in der Datenbank `mydb1` in die Replikation ein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Example Einschließen von Tabellen in die Replikation mit Platzhalterzeichen

Das folgende Beispiel schließt Tabellen mit Namen, die mit `order` und `return` beginnen, in Datenbank `mydb` in die Replikation ein.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Example Ausschließen von Datenbanken von der Replikation

Das folgende Beispiel schließt die Datenbanken `mydb5` und `mydb6` von der Replikation aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Example Ausschließen von Tabellen von der Replikation

Das folgende Beispiel schließt die Tabellen `table1` in Datenbank `mydb5` und `table2` in Datenbank `mydb6` von der Replikation aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Example Ausschließen von Tabellen von der Replikation mit Platzhalterzeichen

Das folgende Beispiel schließt Tabellen mit Namen, die mit `order` und `return` beginnen, in Datenbank `mydb7` von der Replikation aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Anzeigen der Replikationsfilter für ein Lesereplikat

Sie können die Replikationsfilter für ein Lesereplikat wie folgt anzeigen:

- Überprüfen Sie die Einstellungen der Parameter der Replikationsfilter in der dem Lesereplikat zugeordneten Parametergruppe.

  Detaillierte Anweisungen finden Sie unter Parameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen.

- Stellen Sie in einem My SQL Client eine Verbindung zur Read Replica her und führen Sie die `SHOW REPLICA STATUS` Anweisung aus.

  In der Ausgabe werden in den folgenden Feldern die Replikationsfilter für das Lesereplikat angezeigt:

  - `Replicate_Do_DB`
  - `Replicate_Ignore_DB`
  - `Replicate_Do_Table`
  - `Replicate_Ignore_Table`
  - `Replicate_Wild_Do_Table`
  - `Replicate_Wild_Ignore_Table`

  Weitere Informationen zu diesen Feldern finden Sie unter Überprüfen des Replikationsstatus in der SQL Dokumentation My.

Konfiguration der verzögerten Replikation mit My SQL

Sie können die verzögerte Replikation als Strategie für die Notfallwiederherstellung einsetzen. Für die verzögerte Replikation geben Sie die Mindestanzahl von Sekunden an, um welche die Replikation von der Quelle in das Lesereplikat verzögert werden soll. Wenn es zu einem Notfall kommt, weil beispielsweise eine Tabelle versehentlich gelöscht wurde, können Sie das Problem mit den folgenden Schritten schnell beheben:

- Beenden Sie die Replikation im Lesereplikat, bevor die Änderung, die den Notfall verursacht hat, an das Lesereplikat gesendet wird.

  Verwenden Sie die gespeicherte Prozedur mysql.rds_stop_replication, um die Replikation zu stoppen.
- Starten Sie die Replikation und geben Sie an, dass die Replikation automatisch an einer bestimmten Position in der Protokolldatei stoppen soll.

  Mit der gespeicherten Prozedur geben Sie eine Position unmittelbar vor Eintreten des Notfalls an.
- Stufen Sie das Lesereplikat zur neuen Quell-DB-Instance hoch, indem Sie der Anleitung unter folge Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Note

- RDSBei My SQL 8.4 wird verzögerte Replikation für My SQL 8.4.3 und höher unterstützt. On RDS für My SQL 8.0 wird verzögerte Replikation für My SQL 8.0.28 und höher unterstützt. On RDS für My SQL 5.7 wird verzögerte Replikation für My SQL 5.7.44 und höher unterstützt.
- Verwenden Sie gespeicherte Prozeduren, um die verzögerte Replikation zu konfigurieren. Sie können die verzögerte Replikation nicht mit dem AWS Management Console AWS CLI, dem oder Amazon konfigurieren RDSAPI.
- Sie können die Replikation auf der Grundlage globaler Transaktions-Identifikatoren (GTIDs) in einer Konfiguration mit verzögerter Replikation für die folgenden Versionen verwenden:
  - RDSfür meine SQL Version 5.7.44 und höhere Versionen 5.7
  - RDSfür meine SQL Version 8.0.28 und höhere 8.0-Versionen
  - RDSfür meine SQL Version 8.4.3 und höhere 8.4-Versionen

  Wenn Sie die GTID basierte Replikation verwenden, verwenden Sie die gespeicherte Prozedur anstelle der gespeicherten Prozedur. Weitere Hinweise zur GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Themen

- Konfigurieren der verzögerten Replikation während der Erstellung von Read Replicas
- Ändern der verzögerten Replikation einer vorhandenen Read Replica
- Festlegen einer Position zum Stoppen der Replikation zu einer Read Replica
- Hochstufen eines Lesereplikats

Konfigurieren der verzögerten Replikation während der Erstellung von Read Replicas

Um die verzögerte Replikation für zukünftig aus einer DB-Instance erstellte Lesereplikate zu konfigurieren, führen Sie die gespeicherte Prozedur mysql.rds_set_configuration mit dem Parameter `target delay` aus.

Konfigurieren Sie die verzögerte Replikation während der Lesereplikat-Erstellung wie folgt:

1. Stellen Sie über einen My SQL Client eine Verbindung zur My SQL DB-Instance her, um als Master-Benutzer als Quelle für Read Replicas zu dienen.
2. Führen Sie die gespeicherte Prozedur mysql.rds_set_configuration mit dem Parameter `target delay` aus.

   Führen Sie beispielsweise die folgende gespeicherte Prozedur aus, um anzugeben, dass die Replikation um mindestens eine Stunde (3.600 Sekunden) für jedes Lesereplikat verzögert werden soll, das aus der aktuellen DB-Instance erstellt wird.

   ```
   call mysql.rds_set_configuration('target delay', 3600);
   ```

   > **Note**
   >
   > Nach dem Ausführen dieser gespeicherten Prozedur wird jede Read Replica, die Sie mit AWS CLI oder Amazon erstellen, so konfiguriert, dass die Replikation um die angegebene Anzahl von Sekunden verzögert RDS API wird.

Ändern der verzögerten Replikation einer vorhandenen Read Replica

Führen Sie die gespeicherte Prozedur mysql.rds_set_source_delay aus, um die verzögerte Replikation eines vorhandenen Lesereplikats zu ändern.

Ändern Sie die verzögerte Replikation eines existierenden Lesereplikats wie folgt:

1. Stellen Sie über My SQL Client eine Verbindung mit der Read Replica als Master-Benutzer her.
2. Verwenden Sie die gespeicherte Prozedur mysql.rds_stop_replication, um die Replikation zu stoppen.
3. Führen Sie die gespeicherte Prozedur mysql.rds_set_source_delay aus.

   Führen Sie beispielsweise die folgende gespeicherte Prozedur aus, um anzugeben, dass die Replikation für das Lesereplikat um mindestens eine Stunde (3600 Sekunden) verzögert werden soll.

   ```
   call mysql.rds_set_source_delay(3600);
   ```

4. Verwenden Sie die gespeicherte Prozedur mysql.rds_start_replication, um die Replikation zu starten.

Festlegen einer Position zum Stoppen der Replikation zu einer Read Replica

Nachdem Sie die Replikation für ein Lesereplikat gestoppt haben, können Sie die Replikation mit der gespeicherten Prozedur starten und dann an einer angegebenen Position in der Binärprotokolldatei stoppen lassen.

Starten Sie die Replikation für ein Lesereplikat und stoppen Sie die Replikation an einer bestimmten Position wie folgt:

1. Stellen Sie über einen My SQL Client als Masterbenutzer eine Verbindung zur My SQL DB-Quellinstanz her.
2. Führen Sie die gespeicherte Prozedur aus.

   Das folgende Beispiel initiiert die Replikation und repliziert die Änderungen, bis die Position `120` in der Binärprotokolldatei `mysql-bin-changelog.000777` erreicht wird. Beispiel: In einem Szenario der Notfallwiederherstellung liegt die Position `120` unmittelbar vor der Katastrophe.

   ```
   call mysql.rds_start_replication_until(
     'mysql-bin-changelog.000777',
     120);
   ```

Die Replikation stoppt automatisch, sobald der Stopppunkt erreicht ist. Das folgende RDS Ereignis wird generiert:`Replication has been stopped since the replica reached the stop point specified by the rds_start_replication_until stored procedure`.

Hochstufen eines Lesereplikats

Nachdem die Replikation gestoppt wurde, können Sie in einem Szenario der Notfallwiederherstellung ein Lesereplikat zur neuen Quell-DB-Instance hochstufen. Weitere Informationen zum Hochstufen eines Lesereplikats finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Aktualisierung von Read Replicas mit My SQL

Lesereplikate wurden zur Unterstützung von Leseabfragen entwickelt. Jedoch könnten gelegentliche Updates notwendig sein. Sie könnten beispielsweise einen Index hinzufügen, um die spezifischen Abfragetypen zu optimieren, die auf das Replica zugreifen.

Zwar können Sie Updates aktivieren, indem Sie den Parameter `read_only` in der DB-Parametergruppe für das Lesereplikat auf `0` setzen. Wir raten jedoch davon ab, da Probleme auftreten können, wenn das Lesereplikat nicht mehr mit der Quell-DB-Instance kompatibel ist. Für Wartungsvorgänge empfehlen wir, Blau/Grün-Bereitstellungen zu verwenden. Weitere Informationen finden Sie unter Verwendung von Blau/Grün-Bereitstellungen für Datenbankaktualisierungen.

Wenn Sie den Schreibschutz für ein Lesereplikat deaktivieren, ändern Sie den Wert des Parameters `read_only` sobald wie möglich wieder in `1`.

Arbeiten mit Multi-AZ-Read Replica-Bereitstellungen mit My SQL

Sie können ein Lesereplikat aus Single-AZ- oder Multi-AZ-DB-Instance-Bereitstellungen erstellen. Sie können Multi-AZ-Bereitstellungen verwenden, um die Haltbarkeit und Verfügbarkeit kritischer Daten zu verbessern, aber Sie können Multi-AZ nicht zweitranging noch für schreibgeschützte Abfragen verwenden. Stattdessen können Sie Lesereplikate aus Multi-AZ-DB-Instances mit hohem Datenverkehr erstellen, um schreibgeschützte Abfragen auslagern zu können. Wenn die Quell-Instance einer Multi-AZ-Bereitstellung ein Failover zur sekundären Instance durchführt, werden alle zugehörigen Lesereplikate automatisch auf die Verwendung der sekundären (jetzt primären) Instance als Replikationsquelle umgeschaltet. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Sie können ein Lesereplikat als Multi-AZ-DB-Instance erstellen. Amazon RDS erstellt eine Standby-Version Ihres Replikats in einer anderen Availability Zone zur Failover-Unterstützung für das Replikat. Das Erstellen Ihres Lesereplikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist.

Verwenden von kaskadierenden Read Replicas mit for My RDS SQL

RDSfor My SQL unterstützt kaskadierende Read Replicas. Mit kaskadierenden Read Replicas können Sie Lesevorgänge skalieren, ohne Ihre Quelle für My DB-Instance unnötig zu belasten. RDS SQL

Bei kaskadierenden Read Replicas sendet Ihre RDS For My SQL DB-Instance Daten an die erste Read Replica in der Kette. Dieses Lesereplikat sendet dann Daten an das zweite Replikat in der Kette usw. Das Endergebnis ist, dass alle Read Replicas in der Kette die Änderungen aus der RDS For My SQL DB-Instance enthalten, jedoch ohne den Overhead ausschließlich für die Quell-DB-Instance.

Sie können eine Reihe von bis zu drei Read Replicas in einer Kette aus einer Quelle RDS für My SQL DB-Instance erstellen. Nehmen wir zum Beispiel an, Sie haben eine RDS For My SQL DB-Instance,`mysql-main`. Sie haben die folgenden Möglichkeiten:

- Beginnend mit `mysql-main` erstellen Sie das erste Lesereplikat in der Kette `read-replica-1`.
- Als Nächstes erstellen Sie ab `read-replica-1` das nächste Lesereplikat in der Kette `read-replica-2`.
- Schließlich erstellen Sie ab `read-replica-2` das nächste Lesereplikat in der Kette `read-replica-3`.

Sie können kein weiteres Lesereplikat über dieses dritte kaskadierende Lesereplikat hinaus in der Reihe für `mysql-main` erstellen. Eine vollständige Reihe von Instances, von einer „RDSFor My SQL Source“ -DB-Instance bis zum Ende einer Reihe von kaskadierenden Read Replicas, kann aus höchstens vier DB-Instances bestehen.

Damit kaskadierende Read Replicas funktionieren, müssen für jede Quelle RDS für My SQL DB-Instance automatische Backups aktiviert sein. Erstellen Sie zuerst das Lesereplikat und ändern Sie es dann, um automatische Backups für das Lesereplikat für zu aktivieren. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats.

Wie bei jedem Lesereplikat können Sie ein Lesereplikat, das Teil einer Kaskade ist, hochstufen. Wenn Sie ein Lesereplikat aus einer Kette von Lesereplikaten hochstufen, wird dieses Replikat aus der Kette entfernt. Angenommen, Sie möchten einen Teil der Workload von Ihrer `mysql-main`-DB-Instance zu einer neuen Instance verschieben, die nur von der Buchhaltung verwendet wird. Ausgehend von der Kette von drei Lesereplikaten aus dem Beispiel entscheiden Sie sich, `read-replica-2` hochzustufen. Die Kette ist wie folgt betroffen:

- Durch Hochstufen von `read-replica-2` wird es aus der Replikationskette entfernt.
  - Es ist jetzt eine vollständige DB-Instance mit Lese-/Schreibzugriff.
  - Die Replizierung auf `read-replica-3` wird fortgesetzt wie vor der Hochstufung.
- Ihre `mysql-main` setzt die Replizierung auf `read-replica-1` fort.

Weitere Informationen über das Hochstufen von Lesereplikaten finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Überwachung der Replikationsverzögerung für My SQL Read Replicas

Für My SQL Read Replicas können Sie die Replikationsverzögerung in Amazon überwachen, CloudWatch indem Sie sich die RDS `ReplicaLag` Amazon-Metrik ansehen. Die Kennzahl `ReplicaLag` meldet den Wert des Feldes `Seconds_Behind_Master` des Befehls `SHOW REPLICA STATUS`.

Die häufigsten Ursachen für Replikationsverzögerungen bei My SQL sind die folgenden:

- Ein Netzwerkausfall.
- Schreiben in Tabellen, die verschiedene Indizes in einem Lesereplikat haben. Wenn der Parameter `read_only` in einem Lesereplikat auf `0` gesetzt ist, kann die Replikation fehlschlagen, wenn das Lesereplikat nicht mehr mit der Quell-DB-Instance kompatibel ist. Nachdem Sie Wartungsarbeiten für ein Lesereplikat durchgeführt haben, sollten Sie den Parameter `read_only` wieder zurück auf 1 setzen.
- Verwenden Sie eine nicht transaktionale Speicher-Engine wie My. ISAM Die Replikation wird nur für die InnoDB-Speicher-Engine auf My SQL unterstützt.

Wenn die Metrik `ReplicaLag` 0 erreicht, hat das Replica den Stand der Quell-DB-Instance erreicht. Wenn die Metrik `ReplicaLag` -1 zurückgibt, ist die Replikation aktuell nicht aktiv. `ReplicaLag` = -1 ist gleich `Seconds_Behind_Master` = `NULL`.

Starten und Beenden der Replikation mit My SQL Read Replicas

Sie können den Replikationsprozess auf einer RDS Amazon-DB-Instance beenden und neu starten, indem Sie die gespeicherten Systemprozeduren mysql.rds_stop_replication und aufrufenmysql.rds_start_replication. Sie können dies tun, wenn Sie zwischen zwei RDS Amazon-Instances für lang andauernde Operationen wie die Erstellung großer Indizes replizieren. Sie müssen Replikation auch anhalten und starten, wenn Sie Datenbanken importieren oder exportieren. Weitere Informationen erhalten Sie unter Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit und Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation.

Wenn die Replikation für mehr als 30 aufeinanderfolgende Tage unterbrochen wird, entweder manuell oder aufgrund eines Replikationsfehlers, RDS beendet Amazon die Replikation zwischen der Quell-DB-Instance und allen Read Replicas. um erhöhten Speicheranforderungen in der Quell-DB-Instance vorzubeugen und lange Failover-Zeiten zu vermeiden. Die Lesereplikat-DB-Instance ist

weiterhin verfügbar. Die Replikation kann jedoch nicht fortgesetzt werden, da die vom Lesereplikat benötigten Binärprotokolle aus der Quell-DB-Instance nach Beendigung der Replikation gelöscht wurden. Sie können ein neues Lesereplikat für die Quell-DB-Instance erstellen, um die Replikation erneut aufzunehmen.

Behebung eines My SQL Read Replica-Problems

Bei My SQL DB-Instances führen Read Replicas in einigen Fällen zu Replikationsfehlern oder Dateninkonsistenzen (oder beidem) zwischen der Read Replica und ihrer Quell-DB-Instance. Dieses Problem tritt auf, wenn einige Binärprotokoll (Binlog)-Ereignisse oder InnoDB-Wiederholungsprotokolle während eines Fehlers des Lesereplikats oder der Quell-DB-Instance nicht bereinigt werden. In diesen Fällen müssen Sie die Lesereplikate manuell löschen und neu erstellen. Sie können das Risiko minimieren, indem Sie die folgenden Parameterwerte einstellen: `sync_binlog=1` und `innodb_flush_log_at_trx_commit=1`. Diese Einstellungen könnten die Leistung reduzieren, daher ist es ratsam, sie vor der Implementierung in einer Produktionsumgebung ausgiebig zu testen.

> **Warning**
>
> In der Parametergruppe, die mit der Quell-DB-Instance verknüpft ist, sollten diese Parameterwerte beibehalten werden: `sync_binlog=1` und `innodb_flush_log_at_trx_commit=1`. Diese Parameter sind dynamisch. Wenn Sie diese Einstellungen nicht verwenden möchten, empfehlen wir Ihnen, diese Werte vorübergehend festzulegen, bevor Sie einen Vorgang für die Quell-DB-Instance ausführen, der einen Neustart verursachen könnte. Zu diesen Vorgängen gehören unter anderem Neustart, Neustart mit Failover, das Aktualisieren der Datenbankversion und das Ändern der DB-Instance-Klasse oder ihres Speichers. Die gleiche Empfehlung gilt für das Erstellen neuer Lesereplikate für die Quell-DB-Instance.
>
> Die Nichteinhaltung dieser Anleitung erhöht das Risiko, dass Lesereplikate Replikationsfehler oder Dateninkonsistenzen (oder beides) zwischen dem Lesereplikat und seiner DB-Quell-Instance aufweisen.

Die Replikationstechnologien für My sind asynchron. SQL Da sie asynchron sind, steigt gelegentlich `BinLogDiskUsage` in der Quell-DB-Instance an und `ReplicaLag` ist im Lesereplikat zu erwarten. Beispielsweise kann auf der Quell-DB-Instance eine große Anzahl von Schreibvorgängen gleichzeitig ausgeführt werden. Dagegen werden Schreibvorgänge zum Lesereplikat über einen einzigen I/O-Thread seriell abgearbeitet. Dies kann zu einer Verzögerung zwischen der Quell-DB-Instance und

dem Lesereplikat führen. Weitere Informationen zu schreibgeschützten Replikaten finden Sie in der SQL Dokumentation zu My unter Details zur Replikationsimplementierung.

Sie können folgende Dinge tun, um die Verzögerungszeit zwischen Aktualisierungen einer Quell-DB-Instance und der nachfolgenden Aktualisierung des Lesereplikats zu reduzieren:

- Passen Sie die Speichergröße und die DB-Instance-Klasse eines Lesereplikats an die der Quell-DB-Instance an.
- Stellen Sie sicher, dass Parametereinstellungen in den DB-Parametergruppen, die von der Quell-DB-Instance und den Lesereplikaten verwendet werden, kompatibel sind. Weitere Informationen und ein Beispiel finden Sie in der Beschreibung des `max_allowed_packet`-Parameters weiter unten in diesem Abschnitt.

Amazon RDS überwacht den Replikationsstatus Ihrer Read Replicas und aktualisiert das `Replication State` Feld der Read Replica-Instance auf den `Error` Fall, dass die Replikation aus irgendeinem Grund beendet wird. Ein Beispiel könnte sein, wenn DML Abfragen, die auf Ihrer Read Replica ausgeführt werden, in Konflikt mit den Aktualisierungen stehen, die auf der Quell-DB-Instance vorgenommen wurden.

Sie können die Details des zugehörigen Fehlers, der von der My SQL Engine ausgelöst wurde, überprüfen, indem Sie sich das `Replication Error` Feld ansehen. Ereignisse, die den Status des Lesereplikats angeben, werden ebenfalls generiert, einschließlich RDS-EVENT-0045, RDS-EVENT-0046 und RDS-EVENT-0047. Weitere Informationen über Ereignisse und Abonnements zu Ereignissen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten. Wenn die Meldung Meine SQL Fehlermeldung zurückgegeben wird, überprüfen Sie die Fehlernummer in der Dokumentation Meine SQL Fehlermeldung.

Ein häufiges Problem, das Replikationsfehler verursachen kann, besteht, wenn der Wert für den `max_allowed_packet`-Parameter eines Lesereplikats niedriger ist als der Wert für den `max_allowed_packet`-Parameter der Quell-DB-Instance. Der `max_allowed_packet`-Parameter ist ein benutzerdefinierter Parameter, den Sie in einer DB-Parametergruppe festlegen können. Sie geben `max_allowed_packet` damit die maximale DML Codegröße an, die in der Datenbank ausgeführt werden kann. In einigen Fällen ist der `max_allowed_packet`-Wert in der mit einer Quell-DB-Instance verknüpften DB-Parametergruppe kleiner als der `max_allowed_packet`-Wert in der mit dem Lesereplikat der Quelle verknüpften DB-Parametergruppe. In diesen Fällen kann der Replikationsprozess den Fehler `Packet bigger than 'max_allowed_packet' bytes` auslösen und die Replikation beenden. Um den Fehler zu beheben, lassen Sie die DB-

Quell-Instance und das Lesereplikat DB-Parametergruppen mit denselben `max_allowed_packet`-Parameterwerten verwenden.

Weitere Situationen, bei denen Replikationsfehler auftreten können, sind die Folgenden:

- Schreibvorgänge auf Tabellen in einem Lesereplikat. In einigen Fällen können Sie Indizes für ein Lesereplikat erstellen, die sich von den Indizes in der Quell-DB-Instance unterscheiden. Wenn Sie dies tun, setzen Sie den Parameter `read_only` auf `0`, um die Indizes zu erstellen. Wenn Sie in einem Lesereplikat in Tabellen schreiben, kann die Replikation unterbrochen werden, wenn das Lesereplikat nicht mehr mit der Quell-DB-Instance kompatibel ist. Nachdem Sie Wartungsarbeiten für ein Lesereplikat durchgeführt haben, sollten Sie den Parameter `read_only` wieder zurück auf 1 setzen.
- Verwenden Sie eine nicht transaktionale Speicher-Engine wie My. ISAM Read Replicas erfordern eine transaktionale Speicher-Engine. Die Replikation wird nur für die InnoDB-Speicher-Engine auf My SQL unterstützt.
- Verwenden von nicht-deterministischen Abfragen wie `SYSDATE()`. Weitere Informationen finden Sie unter Erkennen sicherer und nicht sicherer Anweisungen in der binären Protokollierung.

Wenn Sie entscheiden, dass Sie einen Fehler problemlos überspringen können, folgen Sie den Schritten im Abschnitt Überspringen des aktuellen Replikationsfehlers für RDS für My SQL. Andernfalls können Sie zuerst das Lesereplikat löschen. Anschließend erstellen Sie eine Instance mit derselben DB-Instance-Kennung, sodass der Endpunkt mit dem des alten Lesereplikats übereinstimmt. Wird ein Replikationsfehler behoben, ändert sich das Feld `Replication State` zu Replicating.

Verwenden der GTID basierten Replikation

Im folgenden Inhalt wird erklärt, wie globale Transaktions-Identifikatoren (GTIDs) mit der Binärprotokollreplikation (Binlog) zwischen Amazon RDS for My SQL DB-Instances verwendet werden.

Wenn Sie die Binlog-Replikation verwenden und mit der GTID basierten Replikation mit My nicht vertraut sindSQL, finden Sie in der Dokumentation zu My weitere Informationen unter Replikation mit globalen Transaktions-Identifikatoren. SQL

GTIDDie basierte Replikation wird für die folgenden Versionen unterstützt:

- Alles RDS für meine SQL 8.4-Versionen

- Alles RDS für Meine SQL 8.0-Versionen
- Alles RDS für Meine SQL 5.7-Versionen

Alle My SQL DB-Instances in einer Replikationskonfiguration müssen diese Versionsanforderung erfüllen.

Themen

- Überblick über globale Transaktions-Identifikatoren () GTIDs
- Parameter für die GTID basierte Replikation
- Aktivierung der GTID basierten Replikation für neue Read Replicas RDS für My SQL
- Aktivierung der GTID basierten Replikation für bestehende Read Replicas RDS für My SQL
- Deaktivierung der GTID basierten Replikation für eine My SQL DB-Instance mit Read Replicas und DB-Cluster

Überblick über globale Transaktions-Identifikatoren () GTIDs

Globale Transaktions-Identifikatoren (GTIDs) sind eindeutige Identifikatoren, die für meine Transaktionen generiert wurden. SQL Sie können sie verwendenGTIDs, um die Binlog-Replikation zu vereinfachen und Fehler zu beheben.

My SQL verwendet zwei verschiedene Arten von Transaktionen für die Binlog-Replikation:

- GTIDTransaktionen — Transaktionen, die durch a GTID identifiziert werden.
- Anonyme Transaktionen — Transaktionen, denen kein Konto GTID zugewiesen wurde.

In einer Replikationskonfiguration GTIDs sind sie für alle DB-Instances eindeutig. GTIDsvereinfachen Sie die Replikationskonfiguration, da Sie bei ihrer Verwendung nicht auf die Positionen in den Protokolldateien verweisen müssen. GTIDserleichtern außerdem die Nachverfolgung replizierter Transaktionen und die Feststellung, ob die Quell-Instance und die Replikate konsistent sind.

Sie können die GTID basierte Replikation verwenden, um Daten mit RDS For My SQL Read Replicas zu replizieren. Sie können die GTID basierte Replikation konfigurieren, wenn Sie neue Read Replicas erstellen, oder Sie können vorhandene Read Replicas in eine nutzerbasierte Replikation konvertieren. GTID

Sie können die GTID basierte Replikation auch in einer Konfiguration mit verzögerter Replikation mit RDS for My verwendenSQL. Weitere Informationen finden Sie unter Konfiguration der verzögerten Replikation mit My SQL.

Parameter für die GTID basierte Replikation

Verwenden Sie die folgenden Parameter, um die GTID basierte Replikation zu konfigurieren.

Parameter	Zulässige Werte	Beschreibung
`gtid_mode`	`OFF`, `OFF_PERMISSIVE` , `ON_PERMISSIVE` , `ON`	`OFF`gibt an, dass es sich bei neuen Transakti onen um anonyme Transaktionen handelt (d. h., es gibt keine`GTIDs`) und dass eine Transaktion anonym sein muss, damit sie repliziert werden kann. `OFF_PERMISSIVE` gibt an, dass neue Transaktionen anonyme Transaktionen sind und alle Transaktionen repliziert werden können. `ON_PERMISSIVE` gibt an, dass es sich bei neuen Transaktionen um GTID Transaktionen handelt, dass aber alle Transaktionen repliziert werden können. `ON`gibt an, dass es sich bei neuen GTID Transaktionen um Transaktionen handelt und dass eine Transaktion eine Transaktion sein muss, um repliziert zu werden. GTID
`enforce_gtid_consistency`	`OFF`, `ON`, `WARN`	`OFF`erlaubt Transaktionen, die GTID Konsistenz zu verletzen. `ON`verhindert, dass Transaktionen gegen die GTID Konsistenz verstoßen. `WARN`ermöglicht Transaktionen, gegen die GTID Konsistenz zu verstoßen, generiert jedoch eine Warnung, wenn ein Verstoß auftritt.

> **Note**
>
> In der AWS Management Console wird der `gtid_mode` Parameter als angezeigt`gtid-mode`.

Verwenden Sie für die GTID basierte Replikation diese Einstellungen für die Parametergruppe für Ihre DB-Instance oder Read Replica:

- `0N`und `ON_PERMISSIVE` gelten nur für die ausgehende Replikation von einer RDS DB-Instance. Beide Werte führen dazu, dass Ihre RDS DB-Instance GTIDs für replizierte Transaktionen verwendet wird. `0N`erfordert, dass die Zieldatenbank auch die GTID basierte Replikation verwendet. `ON_PERMISSIVE`macht die GTID basierte Replikation in der Zieldatenbank optional.
- `OFF_PERMISSIVE`, falls gesetzt, bedeutet dies, dass Ihre RDS DB-Instances eingehende Replikation aus einer Quelldatenbank akzeptieren können. Sie können dies unabhängig davon tun, ob die Quelldatenbank die GTID basierte Replikation verwendet.
- `OFF`, falls gesetzt, bedeutet dies, dass Ihre RDS DB-Instance nur eingehende Replikation von Quelldatenbanken akzeptiert, die keine GTID basierte Replikation verwenden.

Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Aktivierung der GTID basierten Replikation für neue Read Replicas RDS für My SQL

Wenn die GTID basierte Replikation für eine RDS For My SQL DB-Instance aktiviert ist, wird die GTID basierte Replikation automatisch für Read Replicas der DB-Instance konfiguriert.

Um die GTID basierte Replikation für neue Read Replicas zu aktivieren

1. Die Parametergruppe, die der DB-Instance zugeordnet ist, muss über die folgenden Parametereinstellungen verfügen:

   - `gtid_mode` – `ON` oder `ON_PERMISSIVE`
   - `enforce_gtid_consistency` – `ON`

   Weitere Informationen zum Einstellen von Konfigurationsparametern unter Verwendung von Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.
2. Wenn Sie die Parametergruppe der DB-Instance geändert haben, müssen Sie die DB-Instance neu starten. Weitere Information dazu finden Sie unter Neustarten einer DB-Instance.

3. Erstellen Sie mindestens ein Lesereplikat der DB-Instance. Weitere Information dazu finden Sie unter Erstellen eines Lesereplikats.

Amazon RDS versucht, mithilfe der eine GTID basierte Replikation zwischen der My SQL DB-Instance und den Read Replicas einzurichten. `MASTER_AUTO_POSITION` Schlägt der Versuch fehl, RDS verwendet Amazon Protokolldateipositionen für die Replikation mit den Read Replicas. Weitere Informationen zu finden Sie `MASTER_AUTO_POSITION` unter GTIDAutomatische Positionierung in der SQL Dokumentation Meine.

Aktivierung der GTID basierten Replikation für bestehende Read Replicas RDS für My SQL

Für eine bestehende My SQL DB-Instance mit Read Replicas, die keine basierte Replikation verwendet, können Sie die GTID basierte Replikation zwischen der DB-Instance und den Read Replicas konfigurierenGTID.

Um die GTID basierte Replikation für bestehende Read Replicas zu aktivieren

1. Wenn die DB-Instance oder eine Read Replica eine 8.0-Version von RDS for My unter 8.0.26 verwendet, aktualisieren Sie die DB-Instance oder das Read Replica auf SQL Version 8.0.26 oder eine höhere My 8.0-Version. SQL Alle RDS Versionen von My SQL 8.4 und 5.7 unterstützen die basierte Replikation. GTID

   Weitere Informationen finden Sie unter Upgrades der DB-Engine RDS für MySQL.
2. (Optional) Setzen Sie die GTID Parameter zurück, testen Sie das Verhalten der DB-Instance und lesen Sie Repliken:

   a. Die Parametergruppe, die der DB-Instance zugeordnet ist, und jedes Lesereplikat muss den Wert für den Parameter `enforce_gtid_consistency` auf `WARN` gesetzt haben.

      Weitere Informationen zum Einstellen von Konfigurationsparametern unter Verwendung von Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

   b. Wenn Sie die Parametergruppe der DB-Instance geändert haben, müssen Sie die DB-Instance neu starten. Wenn Sie die Parametergruppe des Lesereplikats geändert haben, müssen Sie das Lesereplikat neu starten.

      Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

c. Führen Sie Ihre DB-Instance und Lesereplikate mit Ihrem normalen Workload aus und überwachen Sie die Protokolldateien.

   Wenn Sie Warnungen zu GTID -inkompatiblen Transaktionen sehen, passen Sie Ihre Anwendung so an, dass sie nur GTID -kompatible Funktionen verwendet. Stellen Sie sicher, dass die DB-Instance keine Warnungen über GTID -inkompatible Transaktionen generiert, bevor Sie mit dem nächsten Schritt fortfahren.

3. Setzen Sie die GTID Parameter für die GTID basierte Replikation zurück, die anonyme Transaktionen zulässt, bis die Read Replicas alle verarbeitet haben.

   a. Die Parametergruppe, die der DB-Instance zugeordnet ist, und jedes Lesereplikat muss die folgenden Parametereinstellungen haben:

      - `gtid_mode` – `ON_PERMISSIVE`
      - `enforce_gtid_consistency` – `ON`

   b. Wenn Sie die Parametergruppe der DB-Instance geändert haben, müssen Sie die DB-Instance neu starten. Wenn Sie die Parametergruppe des Lesereplikats geändert haben, müssen Sie das Lesereplikat neu starten.

4. Warten Sie, bis alle anonymen Transaktionen abgeschlossen sind. Um zu überprüfen, ob diese repliziert sind, gehen Sie wie folgt vor:

   a. Führen Sie die folgende Anweisung auf Ihrer Quell-DB-Instance aus.

      Mein 8.4 SQL

      ```
      SHOW BINARY LOG STATUS;
      ```

      Meine SQL 5.7 und 8.0

      ```
      SHOW MASTER STATUS;
      ```

      Notieren Sie die Werte in den Spalten `File` und `Position`.

   b. Verwenden Sie bei jedem Lesereplikat die Datei- und Positionsinformationen der Quellinstance im vorherigen Schritt, um die folgende Abfrage auszuführen.

      ```
      SELECT MASTER_POS_WAIT('file', position);
      ```

Führen Sie die folgende Anweisung aus, wenn der Dateiname `mysql-bin-changelog.000031` lautet und die Position `107` ist.

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

Wenn das Lesereplikat über die angegebene Position hinausgeht, wird die Abfrage sofort zurückgegeben. Andernfalls wartet die Funktion. Gehen Sie zum nächsten Schritt über, wenn die Abfrage für alle Lesereplikate zurückgegeben wird.

5. Setzen Sie die GTID Parameter nur für die GTID basierte Replikation zurück.

   a. Die Parametergruppe, die der DB-Instance zugeordnet ist, und jedes Lesereplikat muss die folgenden Parametereinstellungen haben:

      - `gtid_mode` – `ON`
      - `enforce_gtid_consistency` – `ON`

   b. Starten Sie die DB-Instance und jedes Lesereplikat neu.

6. Führen Sie auf jedem Lesereplikat die folgende Prozedur aus.

   Meine Hauptversionen SQL 8.4 und höher

   ```
   CALL mysql.rds_set_source_auto_position(1);
   ```

   Meine Hauptversionen SQL 8.0 und niedriger

   ```
   CALL mysql.rds_set_master_auto_position(1);
   ```

Deaktivierung der GTID basierten Replikation für eine My SQL DB-Instance mit Read Replicas und DB-Cluster

Sie können die GTID basierte Replikation für eine Meine SQL DB-Instance mit Read Replicas.

So deaktivieren Sie die GTID basierte Replikation für , eine My SQL DB-Instance mit Read Replicas

1. Führen Sie für jede Read Replica das folgende Verfahren aus:

   Meine SQL Hauptversionen 8.4 und höher

```
CALL mysql.rds_set_source_auto_position(0);
```

Meine Hauptversionen SQL 8.0 und niedriger

```
CALL mysql.rds_set_master_auto_position(0);
```

2. Setzen Sie den Wert für `gtid_mode` auf `ON_PERMISSIVE` zurück.

   a. Stellen Sie sicher, dass die Parametergruppe, die der My SQL DB-Instance und jeder Read Replica zugeordnet ist, auf `gtid_mode` `ON_PERMISSIVE` gesetzt ist.

      Weitere Informationen zum Einstellen von Konfigurationsparametern unter Verwendung von Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

   b. Starten Sie die My SQL DB-Instance und jede Read Replica neu. Weitere Informationen zum Neustarten finden Sie unter Neustarten einer DB-Instance.

3. Setzen Sie den Wert für `gtid_mode` auf `OFF_PERMISSIVE` zurück.

   a. Stellen Sie sicher, dass die Parametergruppe, die der My SQL DB-Instance und jeder Read Replica zugeordnet ist, auf `gtid_mode` `OFF_PERMISSIVE` eingestellt ist.

   b. Starten Sie die My SQL DB-Instance und jede Read Replica neu.

4. Warten Sie, bis alle GTID Transaktionen auf alle Read Replicas angewendet wurden. Gehen Sie wie folgt vor, um zu überprüfen, ob diese angewendet wurden:

   a. Führen Sie auf der My SQL DB-Instance den folgenden Befehl aus:

      Mein SQL 8.4

      ```
      SHOW BINARY LOG STATUS
      ```

      Meine SQL 5.7 und 8.0

      ```
      SHOW MASTER STATUS
      ```

      Ihre Ausgabe sollte der folgenden Ausgabe ähneln.

      ```
      File                        Position
      ```

```
-----------------------------------
mysql-bin-changelog.000031      107
-----------------------------------
```

Notieren Sie die Datei und Position in Ihrer Ausgabe.

b. Verwenden Sie für jedes Read Replica die Datei- und Positionsinformationen aus der Quellinstanz im vorherigen Schritt, um die folgende Abfrage auszuführen:

Meine SQL 8.4 und My SQL 8.0.26 und höher Meine SQL 8.0-Versionen

```
SELECT SOURCE_POS_WAIT('file', position);
```

Mein SQL 5.7

```
SELECT MASTER_POS_WAIT('file', position);
```

Wenn der Dateiname beispielsweise lautet `mysql-bin-changelog.000031` und die Position lautet`107`, führen Sie die folgende Anweisung aus:

Meine SQL 8.4 und My SQL 8.0.26 und höher Meine SQL 8.0-Versionen

```
SELECT SOURCE_POS_WAIT('mysql-bin-changelog.000031', 107);
```

Mein SQL 5.7

```
SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
```

5. Setzen Sie die GTID Parameter zurück, um die GTID basierte Replikation zu deaktivieren.

a. Stellen Sie sicher, dass die Parametergruppe, die der My SQL DB-Instance und jeder Read Replica zugeordnet ist, die folgenden Parametereinstellungen hat:

- `gtid_mode` – `OFF`
- `enforce_gtid_consistency` – `OFF`

b. Starten Sie die My SQL DB-Instance und jede Read Replica neu.

Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Sie können die Replikation zwischen einer RDS for My SQL - oder MariaDB-DB-Instance und einer My- SQL oder MariaDB-Instance, die sich außerhalb von Amazon befindet, RDS mithilfe der binären Protokolldateireplikation einrichten.

Themen

- Bevor Sie beginnen
- Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Bevor Sie beginnen

Sie konfigurieren die Replikation mithilfe der Position der Binärprotokolldatei von replizierten Transaktionen.

Die zum Starten der Replikation auf einer Amazon RDS DB-Instance erforderlichen Berechtigungen sind eingeschränkt und stehen Ihrem RDS Amazon-Master-Benutzer nicht zur Verfügung. Stellen Sie daher sicher, dass Sie die mysql.rds_start_replication Befehle Amazon RDS oder, und verwenden, um die Replikation zwischen Ihrer Live-Datenbank und Ihrer RDS Amazon-Datenbank einzurichten.

Um das binäre Protokollierungsformat für eine My SQL - oder MariaDB-Datenbank festzulegen, aktualisieren Sie den `binlog_format` Parameter. Wenn Ihre DB-Instance die standardmäßige DB-Instance-Parametergruppe verwendet, erstellen Sie eine neue DB-Parametergruppe, um den `binlog_format` Parameter zu ändern. In MariaDB und My SQL 8.0 und niedrigeren Versionen ist `binlog_format` standardmäßig. `MIXED` Sie können `binlog_format` jedoch auch auf `ROW` oder `STATEMENT` einstellen, wenn Sie ein spezifisches Format des Binärprotokolls (binlog) benötigen. Starten Sie Ihre DB-Instance neu, damit die Änderungen übernommen werden. In My SQL 8.4 und höheren Versionen ist standardmäßig. `binlog_format` `ROW`

Informationen zum Einstellen des Parameters `binlog_format` erhalten Sie unter Konfiguration von RDS für MySQL-Binärprotokollierung für Single-AZ-Datenbanken. Informationen zu den Auswirkungen der verschiedenen Typen von Meine SQL Replikation finden Sie unter Vor- und Nachteile der anweisungsbasierten und zeilenbasierten Replikation in der Dokumentation zu My. SQL

Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance

Folgen Sie diesen Richtlinien, wenn Sie eine externe Quell-Instance und ein Replikat bei Amazon RDS einrichten:

- Überwachen Sie Failover-Ereignisse für die Amazon RDS DB-Instance, die Ihr Replikat ist. Tritt ein Failover auf, kann die DB-Instance, die Ihr Replikat ist, auf einem neuen Host mit einer anderen Netzwerkadresse wiederhergestellt werden. Weitere Informationen zum Überwachen von Failover-Ereignissen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.
- Bewahren Sie die Binärprotokolle auf Ihrer Quell-Instance auf, bis Sie sichergestellt haben, dass sie auf das Replikat angewendet wurden. Durch das Aufbewahren können Sie sicherstellen, dass Sie Ihre Quell-Instance im Fall eines Ausfalls wiederherstellen können.
- Aktivieren Sie automatische Backups auf Ihrer Amazon RDS DB-Instance. Das Einschalten automatischer Sicherungen stellt sicher, dass Sie Ihr Replikat zu einem bestimmten Zeitpunkt wiederherstellen können, wenn Sie die Quell-Instance und das Replikat neu synchronisieren müssen. Informationen zu point-in-time Backups und Wiederherstellungen finden Sie unterDaten sichern, wiederherstellen und exportieren.

Konfigurieren Sie die Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance wie folgt:

1. Machen Sie die My SQL - oder MariaDB-Quellinstanz schreibgeschützt.

   ```
   mysql> FLUSH TABLES WITH READ LOCK;
   mysql> SET GLOBAL read_only = ON;
   ```

2. Führen Sie den `SHOW MASTER STATUS` Befehl auf der My SQL - oder MariaDB-Quellinstanz aus, um den Binlog-Speicherort zu ermitteln.

   Sie erhalten eine Ausgabe, ähnlich der im folgenden Beispiel.

   ```
   File                      Position
   ------------------------------------
    mysql-bin-changelog.000031      107
   ------------------------------------
   ```

3. Kopieren Sie die Datenbank von der externen Instance in die Amazon RDS DB-Instance mit`mysqldump`. Für große Datenbanken empfiehlt es sich, die Prozedur in zu verwende Import von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

   > **Note**
   >
   > Zwischen der Option `-p` und dem eingegebenen Passwort darf kein Leerzeichen vorhanden sein.

   Um den Hostnamen, den Benutzernamen, den Port und das Passwort für die Verbindung mit Ihrer RDS Amazon-DB-Instance anzugeben `--host--user (-u)`, verwenden Sie die `-p` Optionen, `--port` und im `mysql` Befehl. Der Hostname ist beispielsweise der Name des Domain Name Service (DNS) vom RDS Amazon-DB-Instance-

Endpunkt`myinstance.123456789012.us-east-1.rds.amazonaws.com`. Sie finden den Endpunktwert in den Instance-Details in der AWS Management Console.

4. Machen Sie die My SQL - oder MariaDB-Quellinstanz wieder beschreibbar.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Weitere Informationen zum Erstellen von Backups zur Verwendung bei der Replikation finden Sie in der Dokumentation Meine. SQL

5. Fügen Sie im die IP-Adresse des Servers AWS Management Console, der die externe Datenbank hostet, zur Sicherheitsgruppe Virtual Private Cloud (VPC) für die RDS Amazon-DB-Instance hinzu. Weitere Informationen zum Ändern einer VPC Sicherheitsgruppe finden Sie unter Sicherheitsgruppen für Sie VPC im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

   Die IP-Adresse kann sich ändern, wenn die folgenden Bedingungen erfüllt sind:

   - Sie verwenden eine öffentliche IP-Adresse für die Kommunikation zwischen der externen Quell-Instance und der DB-Instance.
   - Die externe Quell-Instance wurde gestoppt und neu gestartet.

   Wenn diese Bedingungen erfüllt sind, überprüfen Sie die IP-Adresse, bevor Sie sie hinzufügen.

   Möglicherweise müssen Sie auch Ihr lokales Netzwerk so konfigurieren, dass Verbindungen von der IP-Adresse Ihrer RDS Amazon-DB-Instance aus zugelassen werden. Sie tun dies, damit Ihr lokales Netzwerk mit Ihrer externen My SQL - oder MariaDB-Instance kommunizieren kann. Verwenden Sie den `host` Befehl, um die IP-Adresse der RDS Amazon-DB-Instance zu ermitteln.

```
host db_instance_endpoint
```

   Der Hostname ist der DNS Name des RDS Amazon-DB-Instance-Endpunkts.

6. Stellen Sie mit einem Client Ihrer Wahl eine Verbindung zur Quell-Instance her und erstellen Sie den für die Replikation zu verwendenden Benutzer. Verwenden Sie dieses Konto ausschließlich für die Replikation und beschränken Sie es auf Ihre Domäne, um die Sicherheit zu erhöhen. Im Folgenden wird ein Beispiel gezeigt.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

7. Erteilen Sie für die externe Instance die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` für Ihren Replikationsbenutzer. Erteilen Sie beispielsweise die Sonderrechte `REPLICATION CLIENT` und `REPLICATION SLAVE` in allen Datenbank für den '`repl_user`'-Benutzer für Ihre Domäne, mit dem folgenden Befehl.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Machen Sie die Amazon RDS DB-Instance zum Replikat. Stellen Sie dazu zunächst als Master-Benutzer eine Verbindung zur Amazon RDS DB-Instance her. Identifizieren Sie dann die externe My SQL - oder MariaDB-Datenbank als Quellinstanz, indem Sie den Befehl or verwenden. Verwenden Sie den Namen und die Position der Protokolldatei, die Sie in Schritt 2 festgelegt haben. Die folgenden Befehle sind Beispiele.

   Mein SQL 8.4

```
CALL mysql.rds_set_external_source ('mysourceserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

   MariaDB und My SQL 8.0 und 5.7

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1);
```

   > **Note**
   >
   > RDSBei My können Sie wählenSQL, ob Sie die verzögerte Replikation verwenden möchten, indem Sie stattdessen die mysql.rds_set_external_source_with_delay (für meine Hauptversionen 8.4 und höher) RDS SQL oder die mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL gespeicherte Prozedur ausführen. Ein Grund RDS für SQL die Verwendung der verzögerten Replikation ist die Aktivierung der Notfallwiederherstellung mit der gespeicherten Prozedur. Derzeit RDS

unterstützt MariaDB verzögerte Replikation, unterstützt das Verfahren jedoch nicht. `mysql.rds_start_replication_until`

9. Geben Sie auf der Amazon RDS DB-Instance den mysql.rds_start_replication Befehl ein, um die Replikation zu starten.

```
CALL mysql.rds_start_replication;
```

Konfiguration multi-source-replication für Amazon RDS for MySQL

Bei der Replikation mit mehreren Quellen können Sie eine Amazon RDS for MySQL MySQL-DB-Instance als Replikat einrichten, das binäre Protokollereignisse von mehr als einer RDS for MySQL-Quell-DB-Instance empfängt. Multisource-Replikation wird für RDS for MySQL-DB-Instances unterstützt, auf denen die folgenden Engine-Versionen ausgeführt werden:

- Alle MySQL 8.4-Versionen
- Nebenversionen 8.0.35 und höher
- Nebenversionen 5.7.44 und höher

Informationen zur MySQL-Replikation mit mehreren Quellen finden Sie unter MySQL Multisource-Replikation in der MySQL-Dokumentation. Die MySQL-Dokumentation enthält detaillierte Informationen zu dieser Funktion. In diesem Thema wird beschrieben, wie Sie die Multisource-Replikationskanäle auf Ihren RDS für MySQL-DB-Instances konfigurieren und verwalten.

Anwendungsfälle für die Replikation mit mehreren Quellen

Die folgenden Fälle eignen sich gut für die Verwendung der Multisource-Replikation auf RDS für MySQL:

- Anwendungen, die mehrere Shards auf separaten DB-Instances zu einem einzigen Shard zusammenführen oder kombinieren müssen.
- Anwendungen, die Berichte aus Daten generieren müssen, die aus mehreren Quellen konsolidiert wurden.
- Anforderungen zur Erstellung konsolidierter langfristiger Backups von Daten, die auf mehrere RDS für MySQL-DB-Instances verteilt sind.

Voraussetzungen für die Replikation mit mehreren Quellen

Bevor Sie die Replikation mit mehreren Quellen konfigurieren, müssen Sie die folgenden Voraussetzungen erfüllen.

- Stellen Sie sicher, dass für jede Quell-RDS für MySQL-DB-Instance automatische Backups aktiviert sind. Durch die Aktivierung automatischer Backups wird die binäre Protokollierung aktiviert. Informationen zum Aktivieren automatischer Backups finden Sie unterthe section called “Aktivieren von automatisierten Backups”.
- Um Replikationsfehler zu vermeiden, wurde empfohlen, Schreibvorgänge in die Quell-DB-Instances zu blockieren. Sie können dies tun, indem Sie den `read-only` Parameter `ON` in einer benutzerdefinierten Parametergruppe, die an die RDS for MySQL-Quell-DB-Instance angehängt ist, auf setzen. Sie können das AWS Management Console oder das verwenden AWS CLI , um eine neue benutzerdefinierte Parametergruppe zu erstellen oder eine bestehende zu ändern. Weitere Informationen erhalten Sie unter the section called “Erstellen einer DB-Parametergruppe” und the section called “Ändern von Parametern in einer DB-Parametergruppe”.
- Fügen Sie für jede Quell-DB-Instance die IP-Adresse der Amazon Virtual Private Cloud (VPC) -Sicherheitsgruppe für die Multisource-DB-Instance hinzu. Um die IP-Adresse einer Quell-DB-Instance zu identifizieren, können Sie den Befehl ausführen. `dig` *`RDS Endpoint`* Führen Sie den Befehl von einer EC2 Amazon-Instance in derselben VPC aus wie die Ziel-DB-Instance mit mehreren Quellen.
- Verwenden Sie für jede Quell-DB-Instance einen Client, um eine Verbindung zur DB-Instance herzustellen, und erstellen Sie einen Datenbankbenutzer mit den erforderlichen Rechten für die Replikation, wie im folgenden Beispiel gezeigt.

```
CREATE USER 'repl_user' IDENTIFIED BY 'password';
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user';
```

> **Note**
>
> Ab MySQL 8.4 ist das `REPLICATION SLAVE` Privileg veraltet und wurde durch ersetzt. `REPLICATION REPLICA` Verwenden Sie für MySQL 8.4 und spätere Versionen stattdessen die folgende Syntax:
>
> ```
> CREATE USER 'repl_user' IDENTIFIED BY 'password';
> GRANT REPLICATION CLIENT, REPLICATION REPLICA ON *.* TO 'repl_user';
> ```

Konfiguration von Multisource-Replikationskanälen auf RDS für MySQL-DB-Instances

Die Konfiguration von Replikationskanälen mit mehreren Quellen ähnelt der Konfiguration der Replikation aus einzelnen Quellen. Bei der Replikation mit mehreren Quellen aktivieren Sie zunächst die binäre Protokollierung auf der Quellinstanz. Anschließend importieren Sie Daten aus den Quellen in das Multiquellen-Replikat. Anschließend starten Sie die Replikation von jeder Quelle aus, indem Sie die binären Log-Koordinaten oder die automatische GTID-Positionierung verwenden.

Gehen Sie wie folgt vor, um eine RDS for MySQL-DB-Instance als Multisource-Replikat von zwei oder mehr RDS for MySQL-DB-Instances zu konfigurieren.

Themen

- Schritt 1: Importieren Sie Daten aus den Quell-DB-Instances in das Multiquellen-Replikat
- Schritt 2: Starten Sie die Replikation von den Quell-DB-Instances auf das Multiquellen-Replikat

Schritt 1: Importieren Sie Daten aus den Quell-DB-Instances in das Multiquellen-Replikat

Führen Sie die folgenden Schritte für jede Quell-DB-Instance aus.

Bevor Sie die Daten aus einer Quelle in das Multiquellen-Replikat importieren, ermitteln Sie die aktuelle binäre Protokolldatei und Position, indem Sie den `SHOW MASTER STATUS` Befehl ausführen. Notieren Sie sich diese Details, damit Sie sie im nächsten Schritt verwenden können. In dieser Beispielausgabe ist die Datei `mysql-bin-changelog.000031` und die Position ist`107`.

> **Note**
>
> Ab MySQL 8.4 ist der `SHOW MASTER STATUS` Befehl veraltet und wurde durch ersetzt. `SHOW BINARY LOG STATUS` Verwenden Sie für MySQL 8.4 und spätere Versionen `SHOW BINARY LOG STATUS` stattdessen.

```
File                        Position
-----------------------------------
mysql-bin-changelog.000031      107
-----------------------------------
```

Kopieren Sie nun die Datenbank von der Quell-DB-Instance in das Multisource-Replikat, indem Sie`mysqldump`, wie im folgenden Beispiel, Folgendes verwenden.

```
mysqldump --databases database_name \
 --single-transaction \
 --compress \
 --order-by-primary \
 -u RDS_user_name \
 -p RDS_password \
 --host=RDS Endpoint | mysql \
 --host=RDS Endpoint \
 --port=3306 \
 -u RDS_user_name \
-p RDS_password
```

Nach dem Kopieren der Datenbank können Sie den schreibgeschützten Parameter auf der Quell-DB-Instance 0FF auf setzen.

Schritt 2: Starten Sie die Replikation von den Quell-DB-Instances auf das Multiquellen-Replikat

Verwenden Sie für jede Quell-DB-Instance die administrativen Benutzeranmeldedaten, um eine Verbindung mit der Instance herzustellen, und führen Sie die folgenden beiden gespeicherten Prozeduren aus. Diese gespeicherten Prozeduren konfigurieren die Replikation auf einem Kanal und starten die Replikation. In diesem Beispiel werden der Name und die Position der Binlog-Datei aus der Beispielausgabe im vorherigen Schritt verwendet.

```
CALL mysql.rds_set_external_source_for_channel('mysourcehost.example.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 1, 'channel_1');
CALL mysql.rds_start_replication_for_channel('channel_1');
```

Weitere Hinweise zur Verwendung dieser und anderer gespeicherter Prozeduren zur Einrichtung und Verwaltung Ihrer Replikationskanäle finden Sie unterthe section called “Verwaltung der Replikation mit mehreren Quellen”.

Verwenden von Filtern bei der Replikation mit mehreren Quellen

Sie können Replikationsfilter verwenden, um anzugeben, mit welchen Datenbanken und Tabellen in einem Multiquellenreplikat repliziert werden. Replikationsfilter können Datenbanken und Tabellen in die Replikation einbeziehen oder sie von der Replikation ausschließen. Weitere Informationen zu Replikationsfiltern finden Sie unter. the section called “Konfigurieren von Replikationsfiltern”

Bei der Replikation mit mehreren Quellen können Sie Replikationsfilter global oder auf Kanalebene konfigurieren. Die Filterung auf Kanalebene ist nur für unterstützte DB-Instances verfügbar, auf denen

Version 8.0 oder Version 8.4 ausgeführt wird. Die folgenden Beispiele zeigen, wie Filter global oder auf Kanalebene konfiguriert werden.

Beachten Sie die folgenden Anforderungen und das folgende Verhalten bei der Filterung bei der Replikation mit mehreren Quellen:

- Die Kanalnamen müssen in umgekehrte Anführungszeichen (``) gesetzt werden.
- Wenn Sie die Replikationsfilter in der Parametergruppe ändern, werden die Multiquellen-Replikate `sql_thread` für alle Kanäle mit Aktualisierungen neu gestartet, um die Änderungen dynamisch zu übernehmen. Wenn ein Update einen globalen Filter beinhaltet, werden alle Replikationskanäle im laufenden Zustand neu gestartet.
- Alle globalen Filter werden vor allen kanalspezifischen Filtern angewendet.
- Wenn ein Filter global und auf Kanalebene angewendet wird, wird nur der Filter auf Kanalebene angewendet. Wenn die Filter beispielsweise auf `replicate_ignore_db` gesetzt sind`replicate_ignore_db="db1,`channel_22`:db2"`, `db1` wird „auf“ auf alle Kanäle mit Ausnahme `channel_22` von angewendet und `channel_22` ignoriert nur Änderungen von. `db2`

Beispiel 1: Einen globalen Filter einrichten

Im folgenden Beispiel ist die `temp_data` Datenbank in jedem Kanal von der Replikation ausgeschlossen.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-ignore-
db,ParameterValue='temp_data',ApplyMethod=immediate"
```

Beispiel 2: Einstellen eines Filters auf Kanalebene

Im folgenden Beispiel sind Änderungen aus der `sample22` Datenbank nur im Kanal enthalten. `channel_22` Ebenso sind Änderungen aus der `sample99` Datenbank nur im Kanal enthalten`channel_99`.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-parameter-group \
```

```
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-do-db,ParameterValue='\`channel_22\`:sample22,
\`channel_99\`:sample99',ApplyMethod=immediate"
```

Überwachung von Replikationskanälen mit mehreren Quellen

Sie können einzelne Kanäle in einem Replikat mit mehreren Quellen mithilfe der folgenden Methoden überwachen:

- Um den Status aller Kanäle oder eines bestimmten Kanals zu überwachen, stellen Sie eine Verbindung zum Multiquellen-Replikat her und führen Sie den Befehl or aus. `SHOW REPLICA STATUS` `SHOW REPLICA STATUS FOR CHANNEL '`*`channel_name`*`'` Weitere Informationen finden Sie unter Überprüfen des Replikationsstatus in der MySQL-Dokumentation.
- Verwenden Sie die RDS-Ereignisbenachrichtigung, um eine Benachrichtigung zu erhalten, wenn ein Replikationskanal gestartet, gestoppt oder entfernt wird. Weitere Informationen finden Sie unter the section called “Mit RDS Amazon-Event-Benachrichtigungen arbeiten”.
- Um die Verzögerung für einen bestimmten Kanal zu überwachen, überprüfen Sie die `ReplicationChannelLag` entsprechende Metrik. Datenpunkte für diese Metrik haben einen Zeitraum von 60 Sekunden (1 Minute) und sind für 15 Tage verfügbar. Verwenden Sie die Instanz-ID und den Namen des Replikationskanals, um die Verzögerung des Replikationskanals für einen Kanal zu ermitteln. Um eine Benachrichtigung zu erhalten, wenn diese Verzögerung einen bestimmten Schwellenwert überschreitet, können Sie einen CloudWatch Alarm einrichten. Weitere Informationen finden Sie unter the section called “Überwachung von RDS mit CloudWatch”.

Überlegungen und bewährte Methoden für die Replikation mehrerer Quellen

Bevor Sie die Multiquellenreplikation auf RDS for MySQL verwenden, sollten Sie sich die folgenden Überlegungen und bewährten Methoden ansehen:

- Stellen Sie sicher, dass eine als Multiquellen-Replikat konfigurierte DB-Instance über ausreichende Ressourcen wie Durchsatz, Arbeitsspeicher, CPU und IOPS verfügt, um die Arbeitslast mehrerer Quell-Instances zu bewältigen.
- Überwachen Sie regelmäßig die Ressourcennutzung auf Ihrem Multisource-Replikat und passen Sie die Speicher- oder Instance-Konfiguration an, um die Arbeitslast zu bewältigen, ohne Ressourcen zu belasten.
- Sie können die Multithread-Replikation auf einem Replikat mit mehreren Quellen konfigurieren, indem Sie die Systemvariable `replica_parallel_workers` auf einen Wert größer als setzen.

`0` In diesem Fall entspricht die Anzahl der Threads, die jedem Kanal zugewiesen sind, dem Wert dieser Variablen zuzüglich eines Koordinator-Threads zur Verwaltung der Anwender-Threads.

- Konfigurieren Sie die Replikationsfilter entsprechend, um Konflikte zu vermeiden. Um eine gesamte Datenbank auf einem Replikat in eine andere Datenbank zu replizieren, können Sie die `--replicate-rewrite-db` Option verwenden. Sie können beispielsweise alle Tabellen in Datenbank A auf einer Replikatinstanz in Datenbank B replizieren. Dieser Ansatz kann hilfreich sein, wenn alle Quellinstanzen dieselbe Schema-Benennungskonvention verwenden. Informationen zu dieser `--replicate-rewrite-db` Option finden Sie unter Replica Server Options and Variables in der MySQL-Dokumentation.
- Um Replikationsfehler zu vermeiden, sollten Sie vermeiden, in das Replikat zu schreiben. Es wurde empfohlen, den `read_only` Parameter auf Replikaten mit mehreren Quellen zu aktivieren, um Schreibvorgänge zu blockieren. Auf diese Weise können Replikationsprobleme vermieden werden, die durch widersprüchliche Schreibvorgänge verursacht werden.
- Um die Leistung von Lesevorgängen wie Sortiervorgängen und High-Load-Joins, die auf dem Replikat mit mehreren Quellen ausgeführt werden, zu erhöhen, sollten Sie die Verwendung von RDS-optimierten Lesevorgängen in Betracht ziehen. Diese Funktion kann bei Abfragen hilfreich sein, die von großen temporären Tabellen oder Sortierdateien abhängen. Weitere Informationen finden Sie unter the section called “Verbesserung der Abfrageleistung mit RDS optimierten Lesevorgängen”.
- Um die Verzögerung bei der Replikation zu minimieren und die Leistung eines Replikats mit mehreren Quellen zu verbessern, sollten Sie die Aktivierung optimierter Schreibvorgänge in Betracht ziehen. Weitere Informationen finden Sie unter the section called “Verbesserung der Schreibleistung mit RDS Optimized Writes for My SQL”.
- Führen Sie Verwaltungsvorgänge (z. B. das Ändern der Konfiguration) jeweils auf einem Kanal durch und vermeiden Sie, dass Änderungen an mehreren Kanälen von mehreren Verbindungen aus vorgenommen werden. Diese Praktiken können zu Konflikten bei Replikationsvorgängen führen. Beispielsweise kann die gleichzeitige Ausführung von `rds_skip_repl_error_for_channel` `rds_start_replication_for_channel` Prozeduren über mehrere Verbindungen dazu führen, dass Ereignisse auf einem anderen Kanal als beabsichtigt übersprungen werden.
- Sie können Backups auf einer Replikationsinstanz mit mehreren Quellen aktivieren und Daten aus dieser Instance in einen Amazon S3 S3-Bucket exportieren, um sie für langfristige Zwecke zu speichern. Es ist jedoch wichtig, auch Backups mit entsprechender Aufbewahrung für die einzelnen Quell-Instances zu konfigurieren. Informationen zum Exportieren von Snapshot-Daten

nach Amazon S3 finden Sie unterthe section called “Exportieren von DB-Snapshot-Daten nach Amazon S3”.

- Um den Lese-Workload auf ein Replikat mit mehreren Quellen zu verteilen, können Sie Read Replicas aus einem Replikat mit mehreren Quellen erstellen. Sie können diese Read Replicas AWS-Regionen je nach den Anforderungen Ihrer Anwendung an unterschiedlichen Orten platzieren. Weitere Informationen über Lesereplikate finden Sie unter the section called “Meine SQL Read Replicas”.

Einschränkungen für die Replikation mehrerer Quellen auf RDS for MySQL

Die folgenden Einschränkungen gelten für die Replikation mehrerer Quellen auf RDS for MySQL:

- Derzeit unterstützt RDS for MySQL die Konfiguration von maximal 15 Kanälen für ein Multisource-Replikat.
- Eine Read Replica-Instanz kann nicht als Multisource-Replikat konfiguriert werden.
- Um die Multiquellreplikation auf RDS für MySQL mit Engine-Version 5.7 zu konfigurieren, muss das Leistungsschema auf der Replikatinstanz aktiviert sein. Die Aktivierung des Leistungsschemas ist auf RDS für MySQL, auf dem die Engine-Version 8.0 oder 8.4 ausgeführt wird, optional.
- Für RDS for MySQL, auf dem Engine Version 5.7 ausgeführt wird, gelten Replikationsfilter für alle Replikationskanäle. Für RDS for MySQL, auf dem die Engine-Version 8.0 oder 8.4 ausgeführt wird, können Sie Filter konfigurieren, die für alle Replikationskanäle oder für einzelne Kanäle gelten.
- Durch das Wiederherstellen eines RDS-Snapshots oder das Durchführen eines Point-in-time-Restore (PITR) werden keine Replikatkanalkonfigurationen mit mehreren Quellen wiederhergestellt.
- Wenn Sie eine Read Replica eines Multisource-Replikats erstellen, werden nur Daten aus der Multiquell-Instance repliziert. Es stellt keine Kanalkonfiguration wieder her.
- MySQL unterstützt nicht die Einrichtung einer unterschiedlichen Anzahl parallel Worker für jeden Kanal. Jeder Kanal erhält die gleiche Anzahl parallel Worker, basierend auf dem `replica_parallel_workers` Wert.

Die folgenden zusätzlichen Einschränkungen gelten, wenn Ihr Replikationsziel mit mehreren Quellen ein Multi-AZ-DB-Cluster ist:

- Ein Kanal muss für eine Quell-Instance von RDS for MySQL konfiguriert werden, bevor Schreibvorgänge auf diese Instanz ausgeführt werden.

- Für jede Quell-RDS for MySQL-Instance muss die GTID-basierte Replikation aktiviert sein.
- Ein Failover-Ereignis auf dem DB-Cluster entfernt die Konfiguration für die Replikation mit mehreren Quellen. Um diese Konfiguration wiederherzustellen, müssen die Konfigurationsschritte wiederholt werden.

Konfiguration von Active-Active-Clustern für My RDS SQL

Ein Active-Active-Cluster in Amazon RDS ist eine Datenbankkonfiguration, bei der mehrere Knoten aktiv Lese- und Schreibvorgänge abwickeln und die Arbeitslast auf die Instances verteilen, um die Verfügbarkeit und Skalierbarkeit zu verbessern. Jeder Knoten im Cluster wird synchronisiert, um die Datenkonsistenz aufrechtzuerhalten, was eine hohe Verfügbarkeit und einen schnelleren Failover bei einem Knotenausfall ermöglicht

Mithilfe des My SQL Group Replication-Plug-ins können Sie einen aktiv-aktiven Cluster RDS für for My SQL einrichten. Das Group Replication-Plug-In wird RDS für My SQL DB-Instances unterstützt, auf denen die folgenden Versionen ausgeführt werden:

- Alle Meine SQL 8.4-Versionen
- Meine SQL Nebenversionen 8.0.35 und höher

Informationen zu My SQL Group Replication finden Sie unter Group Replication in der Dokumentation MySQL. Die SQL Dokumentation „Meine“ enthält ausführliche Informationen zu dieser Funktion.
In diesem Thema wird beschrieben, wie Sie das Plug-in auf Ihren RDS for My SQL DB-Instances konfigurieren und verwalten.

> **Note**
>
> Der Kürze halber beziehen sich alle Erwähnungen von „aktiv-aktivem“ Cluster in diesem Thema auf aktiv-aktive Cluster, die das My Group Replication-Plugin verwenden. SQL

Anwendungsfälle für aktiv-aktive Cluster

Die folgenden Fälle eignen sich gut für die Verwendung von aktiv-aktiven Clustern:

- Anwendungen, die alle DB-Instances im Cluster benötigen, um Schreibvorgänge zu unterstützen. Das Gruppenreplikations-Plug-In sorgt dafür, dass die Daten auf jeder DB-Instance im Active-Active-Cluster konsistent bleiben. Weitere Informationen dazu, wie das funktioniert, finden Sie unter Gruppenreplikation in der Dokumentation „MeineSQL“.
- Anwendungen, die eine kontinuierliche Verfügbarkeit der Datenbank erfordern. Bei einem aktiv-aktiven Cluster werden die Daten auf allen DB-Instances im Cluster gespeichert. Wenn eine DB-Instance ausfällt, kann die Anwendung den Datenverkehr auf eine andere DB-Instance im Cluster umleiten.

- Anwendungen, die zum Zwecke des Lastenausgleichs möglicherweise Lese- und Schreibvorgänge auf verschiedene DB-Instances im Cluster aufteilen müssen. Mit einem aktiven und aktiven Cluster können Ihre Anwendungen Lesetraffic an bestimmte DB-Instances und Schreibdatenverkehr an andere senden. Sie können auch jederzeit wechseln, an welche DB-Instances Lese- oder Schreibvorgänge gesendet werden sollen.

Themen

- Einschränkungen und Überlegungen für Active-Active-Cluster
- Vorbereitung auf einen VPC aktivenübergreifenden aktiven Cluster
- Erforderliche Parametereinstellungen für Active-Active-Cluster
- Konvertierung einer vorhandenen DB-Instance in einen Active-Active-Cluster
- Einrichtung eines Active-Active-Clusters mit neuen DB-Instances
- Hinzufügen einer DB-Instance zu einem Active-Active-Cluster
- Überwachung von aktiv-aktiven Clustern
- Beenden der Gruppenreplikation auf einer DB-Instance in einem Active-Active-Cluster
- Umbenennen einer DB-Instance in einem Active-Active-Cluster
- Entfernen einer DB-Instance aus einem Active-Active-Cluster

Einschränkungen und Überlegungen für Active-Active-Cluster

Active-Active-Cluster in Amazon RDS bieten verbesserte Verfügbarkeit und Skalierbarkeit, indem sie Workloads auf mehrere Instances verteilen. Bei der Verwendung dieser Architektur sind jedoch wichtige Einschränkungen und Überlegungen zu beachten.

In den folgenden Abschnitten werden wichtige Faktoren wie Verzögerungen bei der Replikation, Konfliktlösung, Ressourcenzuweisung und Failover-Verhalten beschrieben. Das Verständnis dieser Überlegungen kann dazu beitragen, optimale Leistung und Zuverlässigkeit in Active-Active-Clusterbereitstellungen sicherzustellen.

Themen

- Einschränkungen für „Meine RDS Active-Active-Cluster“ SQL
- Überlegungen und bewährte Methoden RDS für Meine Active-Active-Cluster SQL

Einschränkungen für „Meine RDS Active-Active-Cluster“ SQL

Die folgenden Einschränkungen gelten für Active-Active-Cluster für My: RDS SQL

- Der Master-Benutzername darf nicht `rdsgrprepladmin` für DB-Instances in einem aktiv-aktiven Cluster verwendet werden. Dieser Benutzername ist für Gruppenreplikationsverbindungen reserviert.
- Bei DB-Instances mit Read Replicas in aktiv-aktiven Clustern `Replicating` kann ein anderer verlängerter Replikationsstatus dazu führen, dass Protokolldateien die Speicherlimits überschreiten. Hinweise zum Status von Read Replicas finden Sie unter. Überwachen der Lesereplikation
- Blaue/grüne Bereitstellungen werden für DB-Instances in einem Active-Active-Cluster nicht unterstützt. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Blue/Green Deployments für Datenbank-Updates.
- Die Kerberos-Authentifizierung wird für DB-Instances in einem Active-Active-Cluster nicht unterstützt. Weitere Informationen finden Sie unter KerberosAuthentifizierung für Amazon RDS for MySQL verwenden.
- Die DB-Instances in einem Multi-AZ-DB-Cluster können einem Active-Active-Cluster nicht hinzugefügt werden. Die DB-Instances in einer Multi-AZ-DB-Instance-Bereitstellung können jedoch zu einem aktiv-aktiven Cluster hinzugefügt werden. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.
- Tabellen, die keinen Primärschlüssel haben, werden in einem aktiv-aktiven Cluster nicht repliziert, da Schreibvorgänge vom Gruppenreplikations-Plug-In abgelehnt werden.
- Nicht-InnoDB-Tabellen werden in einem Active-Active-Cluster nicht repliziert.
- Active-Active-Cluster unterstützen keine gleichzeitigen DML DDL Anweisungen auf verschiedenen DB-Instances im Cluster.
- Sie können einen Active-Active-Cluster nicht so konfigurieren, dass er den Single-Primary-Modus für den Replikationsmodus der Gruppe verwendet. Für diese Konfiguration empfehlen wir, stattdessen einen Multi-AZ-DB-Cluster zu verwenden. Weitere Informationen finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen für Amazon RDS.
- Die Replikation mehrerer Quellen wird für DB-Instances in einem Active-Active-Cluster nicht unterstützt.
- Ein regionsübergreifender Active-Active-Cluster kann die Überprüfung durch eine Zertifizierungsstelle (CA) für Gruppenreplikationsverbindungen nicht erzwingen.

Überlegungen und bewährte Methoden RDS für Meine Active-Active-Cluster SQL

Lesen Sie sich vor der Verwendung von RDS For My SQL active-active clusters die folgenden Überlegungen und bewährten Methoden durch:

- Aktiv-aktive Cluster können nicht mehr als neun DB-Instances haben.
- Mit dem Gruppenreplikations-Plug-In können Sie die garantierten Transaktionskonsistenzen des aktiv-aktiven Clusters steuern. Weitere Informationen finden Sie unter Transaktionskonsistenzgarantien in meiner SQL Dokumentation.
- Konflikte sind möglich, wenn verschiedene DB-Instances dieselbe Zeile in einem Active-Active-Cluster aktualisieren. Informationen zu Konflikten und Konfliktlösung finden Sie unter Gruppenreplikation in der Dokumentation MeineSQL.
- Aus Gründen der Fehlertoleranz sollten Sie mindestens drei DB-Instances in Ihren Active-Active-Cluster aufnehmen. Es ist möglich, einen Active-Active-Cluster mit nur einer oder zwei DB-Instances zu konfigurieren, der Cluster ist jedoch nicht fehlertolerant. Informationen zur Fehlertoleranz finden Sie unter Fehlertoleranz in der Dokumentation Meine Dokumentation. SQL
- Wenn eine DB-Instance einem bestehenden Active-Active-Cluster beitritt und dieselbe Engine-Version wie die niedrigste Engine-Version im Cluster ausführt, tritt die DB-Instance im Lese-Schreibmodus bei.
- Wenn eine DB-Instance einem vorhandenen Active-Active-Cluster beitritt und eine höhere Engine-Version als die niedrigste Engine-Version im Cluster ausgeführt wird, muss die DB-Instance im schreibgeschützten Modus bleiben.
- Wenn Sie die Gruppenreplikation für eine DB-Instance aktivieren, indem Sie ihren `rds.group_replication_enabled` Parameter `1` in der DB-Parametergruppe auf setzen, die Replikation aber nicht gestartet wurde oder nicht gestartet werden konnte, wird die DB-Instance in den super-read-only Modus versetzt, um Dateninkonsistenzen zu vermeiden. Informationen zum super-read-only Modus finden Sie unter Meine SQL Dokumentation.
- Sie können eine DB-Instance in einem Active-Active-Cluster aktualisieren, aber die DB-Instance ist schreibgeschützt, bis alle anderen DB-Instances im Active-Active-Cluster auf dieselbe Engine-Version oder eine höhere Engine-Version aktualisiert wurden. Wenn Sie eine DB-Instance aktualisieren, tritt die DB-Instance nach Abschluss des Upgrades automatisch demselben Active-Active-Cluster bei. Um zu verhindern, dass eine DB-Instance unbeabsichtigt in den schreibgeschützten Modus wechselt, deaktivieren Sie für diese automatische Aktualisierung kleinerer Versionen. Informationen zum Upgrade einer My SQL DB-Instance finden Sie unter. Upgrades der DB-Engine RDS für MySQL

- Sie können eine DB-Instance in einer Multi-AZ-DB-Instance-Bereitstellung zu einem vorhandenen Active-Active-Cluster hinzufügen. Sie können auch eine Single-AZ-DB-Instance in einem Active-Active-Cluster in eine Multi-AZ-DB-Instance-Bereitstellung konvertieren. Wenn eine primäre DB-Instance in einer Multi-AZ-Bereitstellung ausfällt, wird für diese primäre Instance ein Failover auf die Standby-Instance ausgeführt. Die neue primäre DB-Instance tritt nach Abschluss des Failovers automatisch demselben Cluster bei. Weitere Informationen zu Multi-AZ-DB-Instance-Bereitstellungen finden Sie unter. Multi-AZ-DB-Instance-Bereitstellungen für Amazon RDS
- Wir empfehlen, dass die Wartungsfenster der DB-Instances in einem aktiv-aktiven Cluster unterschiedlich lang sind. Durch diese Vorgehensweise wird vermieden, dass mehrere DB-Instances im Cluster gleichzeitig aus Wartungsgründen offline gehen. Weitere Informationen finden Sie unter Amazon RDS-Wartungsfenster.
- Aktiv-aktive Cluster können SSL für Verbindungen zwischen DB-Instances verwendet werden. Um SSL Verbindungen zu konfigurieren, legen Sie die Parameter group_replication_recovery_use_ssl und group_replication_ssl_mode fest. Die Werte für diese Parameter müssen für alle DB-Instances im Active-Active-Cluster übereinstimmen.

  Derzeit unterstützen aktiv-aktive Cluster keine Überprüfung durch eine Zertifizierungsstelle (CA) für Verbindungen zwischen. AWS-Regionen Daher muss der Parameter group_replication_ssl_mode auf (Standard) oder für regionsübergreifende Cluster gesetzt sein. `DISABLED` `REQUIRED`
- Ein RDS for My active-active Cluster wird im Multi-Primary-Modus ausgeführt. SQL Der Standardwert von group_replication_enforce_update_everywhere_checks ist und der Parameter ist statisch. `ON` Wenn dieser Parameter auf gesetzt ist, können Anwendungen keine Daten in eine Tabelle einfügen`ON`, für die kaskadierende Fremdschlüsseleinschränkungen gelten.
- An RDS for My SQL active-active Cluster verwendet stattdessen den My SQL Communication Stack für die Verbindungssicherheit. XCOM Weitere Informationen finden Sie unter Kommunikationsstapel für die Verwaltung der Verbindungssicherheit in der Dokumentation „MeineSQL“.
- Wenn eine DB-Parametergruppe einer DB-Instance in einem aktiv-aktiven Cluster zugeordnet ist, empfehlen wir, diese DB-Parametergruppe nur anderen DB-Instances zuzuordnen, die sich im Cluster befinden.
- Aktiv-aktive Cluster unterstützen RDS nur My DB-Instances. SQL Auf diesen DB-Instances müssen unterstützte Versionen der DB-Engine ausgeführt werden.
- Wenn eine DB-Instance in einem Active-Active-Cluster einen unerwarteten Ausfall hat, RDS wird die Wiederherstellung der DB-Instance automatisch gestartet. Wenn die DB-Instance nicht wiederhergestellt werden kann, empfehlen wir, sie durch eine neue DB-Instance zu ersetzen,

indem Sie eine point-in-time Wiederherstellung mit einer fehlerfreien DB-Instance im Cluster durchführen. Detaillierte Anweisungen finden Sie unter Hinzufügen einer DB-Instance zu einem aktiv-aktiven Cluster mithilfe der Wiederherstellung point-in-time.

- Sie können eine DB-Instance in einem aktiv-aktiven Cluster löschen, ohne dass dies Auswirkungen auf die anderen DB-Instances im Cluster hat. Weitere Informationen zum Löschen einer DB-Instance finden Sie unter Löschen einer DB-Instance.
- Wenn eine DB-Instance unbeabsichtigt einen aktiv-aktiven Cluster verlässt, ändert sich der Parameter standardmäßig auf. `group_replication_exit_state_action` `OFFLINE_MODE` In diesem Zustand ist der Zugriff auf die DB-Instance nicht möglich und Sie müssen die DB-Instance neu starten, um sie wieder online zu schalten und dem Cluster wieder beizutreten. Sie können dieses Verhalten ändern, indem Sie den `group_replication_exit_state_action` Parameter in einer benutzerdefinierten Parametergruppe ändern. Wenn Sie den Parameter auf einstellen`READ_ONLY`, wechselt die DB-Instance, wenn sie versehentlich einen Cluster verlässt, in einen Super-Read-Only-Status, anstatt offline zu gehen.

Vorbereitung auf einen VPC aktivenübergreifenden aktiven Cluster

Sie können einen Active-Active-Cluster mit Amazon RDS for My SQL DB-Instances in mehr als einer konfigurieren. VPC VPCsSie können sich im selben AWS-Region oder in einem anderen befinden. AWS-Regionen

> **Note**
>
> Das Senden von Datenverkehr zwischen mehreren AWS-Regionen kann zusätzliche Kosten verursachen. Weitere Informationen finden Sie unter Überblick über die Datenübertragungskosten für gängige Architekturen.

Wenn Sie einen Active-Active-Cluster in einem einzigen Cluster konfigurierenVPC, können Sie diese Schritte überspringen und mit fortfahren. Einrichtung eines Active-Active-Clusters mit neuen DB-Instances

So bereiten Sie sich auf einen aktiv-aktiven Cluster mit DB-Instances in mehr als einer vor VPC

1. Stellen Sie sicher, dass die IPv4 Adressbereiche in den CIDR Blöcken die folgenden Anforderungen erfüllen:

- Die IPv4 Adressbereiche in den CIDR Blöcken von VPCs dürfen sich nicht überschneiden.
- Alle IPv4 Adressbereiche in den CIDR Blöcken müssen entweder niedriger `128.0.0.0/`*`subnet_mask`* oder höher als 128.0.0.0/ sein*`subnet_mask`*.

Die folgenden Bereiche veranschaulichen diese Anforderungen:

- `10.1.0.0/16`in der einen VPC und `10.2.0.0/16` in der anderen VPC wird es unterstützt.
- `172.1.0.0/16`in der einen VPC und `172.2.0.0/16` in der anderen VPC wird unterstützt.
- `10.1.0.0/16`in der einen VPC und `10.1.0.0/16` in der anderen VPC wird nicht unterstützt, da sich die Bereiche überschneiden.
- `10.1.0.0/16`in der einen VPC und `172.1.0.0/16` in der anderen VPC wird nicht unterstützt, weil einer unten `128.0.0.0/`*`subnet_mask`* und der andere oben ist`128.0.0.0/`*`subnet_mask`*.

Informationen zu CIDR Blöcken finden Sie unter VPCCIDRBlöcke im VPCAmazon-Benutzerhandbuch.

2. Stellen Sie in jedem Fall sicherVPC, dass sowohl DNS Auflösung als auch DNS Hostnamen aktiviert sind.

   Anweisungen finden Sie unter DNSAttribute für Sie anzeigen und aktualisieren VPC im VPCAmazon-Benutzerhandbuch.

3. Konfigurieren Sie die VPCs so, dass Sie den Verkehr zwischen ihnen auf eine der folgenden Arten weiterleiten können:

   - Erstellen Sie eine VPC Peering-Verbindung zwischen denVPCs.

     Anweisungen finden Sie im Amazon VPC Peering Guide unter Eine VPC Peering-Verbindung erstellen. Stellen Sie in jedem Fall sicherVPC, dass es Regeln für eingehenden Datenverkehr für Ihre Sicherheitsgruppen gibt, die auf Sicherheitsgruppen im Peering verweisen. VPC Auf diese Weise kann Datenverkehr zu und von Instances fließen, die der Sicherheitsgruppe zugeordnet sind, auf die im Peering verwiesen wird. VPC Anweisungen finden Sie unter Aktualisieren Sie Ihre Sicherheitsgruppen, um Peer-Sicherheitsgruppen zu referenzieren im Amazon VPC Peering Guide.

   - Erstellen Sie ein Transit-Gateway zwischen denVPCs.

Anweisungen finden Sie unter Erste Schritte mit Transit-Gateways in Amazon VPC Transit Gateways. Stellen Sie in jeder Sicherheitsgruppe sicherVPC, dass es Regeln für eingehenden Datenverkehr für Ihre Sicherheitsgruppen gibt, die Datenverkehr von der jeweils anderen Gruppe zulassenVPC, z. B. Regeln für eingehenden Datenverkehr, die die jeweils andere Gruppe spezifizieren. CIDR VPC Dadurch kann der Datenverkehr zu und von Instances fließen, die der referenzierten Sicherheitsgruppe im Active-Active-Cluster zugeordnet sind. Weitere Informationen finden Sie unter Steuern des Datenverkehrs zu Ihren AWS Ressourcen mithilfe von Sicherheitsgruppen im VPCAmazon-Benutzerhandbuch.

Erforderliche Parametereinstellungen für Active-Active-Cluster

Die Konfiguration von Parametern für Active-Active-Cluster in Amazon RDS for My SQL ist für die Aufrechterhaltung einer gleichbleibenden Leistung und Betriebsstabilität unerlässlich. In dieser Tabelle sind die wichtigsten Parameter aufgeführt, die die Replikation, Konfliktlösung und Arbeitslastverteilung steuern. Eine korrekte Konfiguration gewährleistet eine effiziente Synchronisation zwischen den Knoten, minimiert die Replikationsverzögerung und optimiert die Ressourcennutzung in verteilten Umgebungen oder Umgebungen mit hohem Datenaufkommen.

Parameter	Beschreibung	Erforderliche Einstellung
`binlog_format`	Legt das binäre Protokoll ierungsformat fest. Der Standardwert RDS für My SQL 8.0-Versionen und niedriger ist`MIXED`. Der Standardwert RDS für Meine SQL 8.4-Versi onen ist`ROW`. Weitere Informati onen finden Sie unter Meine SQL Dokumentation.	`ROW`
`enforce_gtid_consi stency`	Erzwingt GTID Konsisten z bei der Ausführung von Anweisungen. Der Standardw ert für RDS für My SQL ist`OFF`. Weitere Informationen finden	`ON`

Parameter	Beschreibung	Erforderliche Einstellung
	Sie in der SQL Dokumentation zu My.	
`group_replication_group_name`	Setzt den Namen der Gruppenreplikation auf UUID a. Das UUID Format ist`11111111-2222-3333-4444-555555555555` . Sie können ein My generiere n, SQL UUID indem Sie eine Verbindung zu einer My SQL DB-Instance herstellen und es ausführen`SELECT UUID()`. Der Wert muss für alle DB-Instances im Active-Active-Cluster identisch sein. Weitere Informationen finden Sie unter Meine SQL Dokumentation.	Ein Mein SQL UUID
`gtid-mode`	Auf Kontrollen GTID basierend e Protokollierung. Der Standardwert RDS für My SQL ist`OFF_PERMISSIVE` . Weitere Informationen finden Sie in der SQL Dokumentation zu My.	`ON`

Parameter	Beschreibung	Erforderliche Einstellung
`rds.custom_dns_resolution`	Gibt an, ob die DNS Auflösung vom DNS Amazon-Server in Ihrem zulässig istVPC. DNSDie Auflösung muss aktiviert sein, wenn die Gruppenreplikation mit dem `rds.group_replication_enabled` Parameter aktiviert ist. DNSDie Auflösung kann nicht aktiviert werden, wenn die Gruppenreplikation mit dem `rds.group_replication_enabled` Parameter deaktiviert ist. Weitere Informationen finden Sie unter DNSAmazon-Server im VPCAmazon-Benutzerhandbuch.	1
`rds.group_replication_enabled`	Gibt an, ob die Gruppenreplikation für eine DB-Instance aktiviert ist. Die Gruppenreplikation muss auf einer DB-Instance in einem aktiv-aktiven Cluster aktiviert sein.	1
`replica_preserve_commit_order` (RDSfür My SQL 8.4 und höhere Versionen) oder `slave_preserve_commit_order` (RDSfür My SQL 8.0)	Steuert die Reihenfolge, in der Transaktionen auf einem Replikat festgeschrieben werden. Der Standardwert für RDS für My SQL ist`ON`. Weitere Informationen finden Sie in der SQL Dokumentation zu My.	`ON`

Konvertierung einer vorhandenen DB-Instance in einen Active-Active-Cluster

Die DB-Engine-Version der DB-Instance, die Sie zu einem Active-Active-Cluster migrieren möchten, muss eine der folgenden Versionen sein:

- Alle meine 8.4-Versionen SQL
- Meine SQL Nebenversionen 8.0.35 und höher

Wenn Sie die Engine-Version aktualisieren müssen, finden Sie weitere Informationen unter. Upgrades der DB-Engine RDS für MySQL

Wenn Sie einen Active-Active-Cluster mit DB-Instances in mehr als einer einrichten, stellen Sie sicherVPC, dass Sie die Voraussetzungen unter erfüllen. Vorbereitung auf einen VPC aktivenübergreifenden aktiven Cluster

Gehen Sie wie folgt vor, um eine bestehende DB-Instance auf einen Active-Active-Cluster für for My zu migrieren. RDS SQL

Themen

- Schritt 1: Legen Sie die Active-Active-Cluster-Parameter in einer oder mehreren benutzerdefinierten Parametergruppen fest
- Schritt 2: Ordnen Sie die DB-Instance einer DB-Parametergruppe zu, für die die erforderlichen Gruppenreplikationsparameter festgelegt sind
- Schritt 3: Erstellen Sie den Active-Active-Cluster
- Schritt 4: Erstellen Sie zusätzliche Instances RDS für My SQL DB für den Active-Active-Cluster
- Schritt 5: Initialisieren Sie die Gruppe auf der DB-Instance, die Sie konvertieren
- Schritt 6: Starten Sie die Replikation auf den anderen DB-Instances im Active-Active-Cluster
- Schritt 7: (Empfohlen) Überprüfen Sie den Status des Active-Active-Clusters

Schritt 1: Legen Sie die Active-Active-Cluster-Parameter in einer oder mehreren benutzerdefinierten Parametergruppen fest

Die RDS for My SQL DB-Instances in einem Active-Active-Cluster müssen einer benutzerdefinierten Parametergruppe zugeordnet werden, die die richtigen Einstellungen für die erforderlichen Parameter

hat. Informationen zu den Parametern und den jeweils erforderlichen Einstellungen finden Sie unter. Erforderliche Parametereinstellungen für Active-Active-Cluster

Sie können diese Parameter in neuen Parametergruppen oder in vorhandenen Parametergruppen festlegen. Um jedoch zu verhindern, dass versehentlich DB-Instances beeinträchtigt werden, die nicht Teil des Active-Active-Clusters sind, empfehlen wir dringend, eine neue benutzerdefinierte Parametergruppe zu erstellen. Die DB-Instances in einem aktiv-aktiven Cluster können derselben DB-Parametergruppe oder unterschiedlichen DB-Parametergruppen zugeordnet werden.

Sie können das AWS Management Console oder verwenden AWS CLI , um eine neue benutzerdefinierte Parametergruppe zu erstellen. Weitere Informationen finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS. Im folgenden Beispiel wird der create-db-parameter-group AWS CLI Befehl ausgeführt, um eine benutzerdefinierte DB-Parametergruppe mit dem Namen *myactivepg* My SQL 8.0 RDS zu erstellen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

Sie können auch das AWS Management Console oder das verwenden AWS CLI , um die Parameter in der benutzerdefinierten Parametergruppe festzulegen. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Im folgenden Beispiel wird der modify-db-parameter-group AWS CLI Befehl ausgeführt, um die Parameter RDS für My SQL 8.0 festzulegen. Um dieses Beispiel mit RDS für My SQL 8.4 zu verwenden, ändern Sie `slave_preserve_commit_order` `zureplica_preserve_commit_order`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^
```

```
 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Schritt 2: Ordnen Sie die DB-Instance einer DB-Parametergruppe zu, für die die erforderlichen Gruppenreplikationsparameter festgelegt sind

Ordnen Sie die DB-Instance einer Parametergruppe zu, die Sie im vorherigen Schritt erstellt oder geändert haben. Detaillierte Anweisungen finden Sie unter Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon.

Starten Sie die DB-Instance neu, damit die neuen Parametereinstellungen wirksam werden. Detaillierte Anweisungen finden Sie unter Neustarten einer DB-Instance.

Schritt 3: Erstellen Sie den Active-Active-Cluster

Stellen Sie in der DB-Parametergruppe, die der DB-Instance zugeordnet ist, den `group_replication_group_seeds` Parameter auf den Endpunkt der DB-Instance ein, die Sie konvertieren.

Sie können das AWS Management Console oder das verwenden AWS CLI , um den Parameter festzulegen. Sie müssen die DB-Instance nicht neu starten, nachdem Sie diesen Parameter gesetzt haben. Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Im folgenden Beispiel wird der modify-db-parameter-group AWS CLI Befehl ausgeführt, um die Parameter festzulegen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
```

```
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Schritt 4: Erstellen Sie zusätzliche Instances RDS für My SQL DB für den Active-Active-Cluster

Um zusätzliche DB-Instances für den Active-Active-Cluster zu erstellen, führen Sie eine point-in-time Wiederherstellung auf der DB-Instance durch, die Sie konvertieren. Detaillierte Anweisungen finden Sie unter Hinzufügen einer DB-Instance zu einem aktiv-aktiven Cluster mithilfe der Wiederherstellung point-in-time.

Ein aktiv-aktiver Cluster kann bis zu neun DB-Instances haben. Führen Sie eine point-in-time Wiederherstellung auf der DB-Instance durch, bis Sie die Anzahl der DB-Instances haben, die Sie für den Cluster benötigen. Stellen Sie bei der Ausführung sicher point-in-recovery, dass Sie die DB-Instance, die Sie hinzufügen, einer DB-Parametergruppe zuordnen, für die der Wert auf `rds.group_replication_enabled` festgelegt ist1. Andernfalls wird die Gruppenreplikation auf der neu hinzugefügten DB-Instance nicht gestartet.

Schritt 5: Initialisieren Sie die Gruppe auf der DB-Instance, die Sie konvertieren

Initialisieren Sie die Gruppe und starten Sie die Replikation:

1. Stellen Sie in einem SQL Client eine Connect zu der DB-Instance her, die Sie konvertieren. Weitere Informationen zum Herstellen einer Verbindung zu einer RDS For My SQL DB-Instance finden Sie unterVerbindung zu Ihrer MySQL-DB-Instance herstellen.
2. Führen Sie auf dem SQL Client die folgenden gespeicherten Prozeduren aus und *`group_replication_user_password`* ersetzen Sie sie durch das Kennwort für den `rdsgrprepladmin` Benutzer. Der `rdsgrprepladmin` Benutzer ist für Gruppenreplikationsverbindungen in einem Active-Active-Cluster reserviert. Das Passwort für diesen Benutzer muss auf allen DB-Instances in einem Active-Active-Cluster identisch sein.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   call mysql.rds_group_replication_start(1);
   ```

In diesem Beispiel wird der `binlog retention hours` Wert auf festgelegt168, was bedeutet, dass binäre Protokolldateien sieben Tage lang auf der DB-Instance aufbewahrt werden. Sie können diesen Wert an Ihre Anforderungen anpassen.

In diesem Beispiel wird `1` in der `mysql.rds_group_replication_start` gespeicherten Prozedur angegeben, dass eine neue Gruppe mit der aktuellen DB-Instance initialisiert werden soll.

Weitere Hinweise zu den im Beispiel aufgerufenen gespeicherten Prozeduren finden Sie unterVerwaltung von aktiv-aktiven Clustern.

Schritt 6: Starten Sie die Replikation auf den anderen DB-Instances im Active-Active-Cluster

Verwenden Sie für jede DB-Instance im Active-Active-Cluster einen SQL Client, um eine Verbindung mit der Instance herzustellen, und führen Sie die folgenden gespeicherten Prozeduren aus. Ersetzen Sie es *`group_replication_user_password`* durch das Passwort für den `rdsgrprepladmin` Benutzer.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

In diesem Beispiel wird der `binlog retention hours` Wert auf festgelegt168, was bedeutet, dass binäre Protokolldateien auf jeder DB-Instance sieben Tage lang aufbewahrt werden. Sie können diesen Wert an Ihre Anforderungen anpassen.

In diesem Beispiel wird `0` in der `mysql.rds_group_replication_start` gespeicherten Prozedur angegeben, dass die aktuelle DB-Instance einer vorhandenen Gruppe hinzugefügt werden soll.

> **Tip**
>
> Stellen Sie sicher, dass Sie diese gespeicherten Prozeduren auf allen anderen DB-Instances im Active-Active-Cluster ausführen.

Schritt 7: (Empfohlen) Überprüfen Sie den Status des Active-Active-Clusters

Um sicherzustellen, dass jedes Mitglied des Clusters korrekt konfiguriert ist, überprüfen Sie den Status des Clusters, indem Sie eine Verbindung zu einer DB-Instance im Active-Active-Cluster herstellen und den folgenden Befehl ausführen: SQL

```
SELECT * FROM performance_schema.replication_group_members;
```

Ihre Ausgabe sollte `ONLINE` für jede DB-Instance wie in `MEMBER_STATE` der folgenden Beispielausgabe angezeigt werden:

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Informationen zu den möglichen `MEMBER_STATE` Werten finden Sie unter Gruppenreplikations-Serverstatus in der SQL Dokumentation Meine.

Einrichtung eines Active-Active-Clusters mit neuen DB-Instances

Gehen Sie wie folgt vor, um einen Active-Active-Cluster mit neuen Amazon RDS for My SQL DB-Instances einzurichten.

Wenn Sie einen Active-Active-Cluster mit DB-Instances in mehr als einer einrichten, stellen Sie sicherVPC, dass Sie die Voraussetzungen unter erfüllen. Vorbereitung auf einen VPC aktivenübergreifenden aktiven Cluster

Themen

- Schritt 1: Legen Sie die Active-Active-Cluster-Parameter in einer oder mehreren benutzerdefinierten Parametergruppen fest
- Schritt 2: Erstellen Sie neue Instances RDS für My SQL DB für den Active-Active-Cluster
- Schritt 3: Geben Sie die DB-Instances im Active-Active-Cluster an
- Schritt 4: Initialisieren Sie die Gruppe auf einer DB-Instance und starten Sie die Replikation
- Schritt 5: Starten Sie die Replikation auf den anderen DB-Instances im Active-Active-Cluster
- Schritt 6: (Empfohlen) Überprüfen Sie den Status des Active-Active-Clusters
- Schritt 7: (Optional) Importieren Sie Daten in eine DB-Instance im Active-Active-Cluster

Schritt 1: Legen Sie die Active-Active-Cluster-Parameter in einer oder mehreren benutzerdefinierten Parametergruppen fest

Die RDS for My SQL DB-Instances in einem Active-Active-Cluster müssen einer benutzerdefinierten Parametergruppe zugeordnet werden, die die richtigen Einstellungen für die erforderlichen Parameter hat. Informationen zu den Parametern und den jeweils erforderlichen Einstellungen finden Sie unter. Erforderliche Parametereinstellungen für Active-Active-Cluster

Sie können diese Parameter in neuen Parametergruppen oder in vorhandenen Parametergruppen festlegen. Um jedoch zu verhindern, dass versehentlich DB-Instances beeinträchtigt werden, die nicht Teil des Active-Active-Clusters sind, empfehlen wir dringend, eine neue benutzerdefinierte Parametergruppe zu erstellen. Die DB-Instances in einem aktiv-aktiven Cluster können derselben DB-Parametergruppe oder unterschiedlichen DB-Parametergruppen zugeordnet werden.

Sie können das AWS Management Console oder das verwenden AWS CLI , um eine neue benutzerdefinierte Parametergruppe zu erstellen. Weitere Informationen finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS. Im folgenden Beispiel wird der create-db-parameter-group AWS CLI Befehl ausgeführt, um eine benutzerdefinierte DB-Parametergruppe mit dem Namen *myactivepg* My SQL 8.0 RDS zu erstellen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-parameter-group \
```

```
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

Sie können auch das AWS Management Console oder das verwenden AWS CLI , um die Parameter in der benutzerdefinierten Parametergruppe festzulegen. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Im folgenden Beispiel wird der modify-db-parameter-group AWS CLI Befehl ausgeführt, um die Parameter RDS für My SQL 8.0 festzulegen. Um dieses Beispiel mit RDS für My SQL 8.4 zu verwenden, ändern Sie `slave_preserve_commit_order` zu`replica_preserve_commit_order`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \
```

```
 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Schritt 2: Erstellen Sie neue Instances RDS für My SQL DB für den Active-Active-Cluster

Active-Active-Cluster werden für die folgenden Versionen von RDS for My DB-Instances unterstützt: SQL

- Alle Versionen von My SQL Version 8.4
- Meine SQL Version 8.0.35 und höhere Nebenversionen

Sie können bis zu neun neue DB-Instances für den Cluster erstellen.

Sie können das AWS Management Console oder das verwenden AWS CLI , um neue DB-Instances zu erstellen. Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen

einer Amazon RDS-DB-Instance. Wenn Sie die DB-Instance erstellen, ordnen Sie sie einer DB-Parametergruppe zu, die Sie im vorherigen Schritt erstellt oder geändert haben.

Schritt 3: Geben Sie die DB-Instances im Active-Active-Cluster an

Legen Sie in der DB-Parametergruppe, die jeder DB-Instance zugeordnet ist, den `group_replication_group_seeds` Parameter auf die Endpunkte der DB-Instances fest, die Sie in den Cluster aufnehmen möchten.

Sie können das AWS Management Console oder das verwenden AWS CLI , um den Parameter festzulegen. Sie müssen die DB-Instance nicht neu starten, nachdem Sie diesen Parameter gesetzt haben. Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Im folgenden Beispiel wird der modify-db-parameter-group AWS CLI Befehl ausgeführt, um die Parameter festzulegen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

> **Tip**
>
> Stellen Sie sicher, dass Sie den `group_replication_group_seeds` Parameter in jeder DB-Parametergruppe festlegen, die einer DB-Instance im Active-Active-Cluster zugeordnet ist.

Schritt 4: Initialisieren Sie die Gruppe auf einer DB-Instance und starten Sie die Replikation

Sie können eine beliebige neue Datenbank auswählen, um die Gruppe zu initialisieren und die Replikation zu starten. Führen Sie dazu die folgenden Schritte aus:

1. Wählen Sie eine DB-Instance im Active-Active-Cluster aus und stellen Sie eine Verbindung zu dieser DB-Instance auf einem Client her. SQL Weitere Informationen zum Herstellen einer Verbindung zu einer RDS For My SQL DB-Instance finden Sie unter. Verbindung zu Ihrer MySQL-DB-Instance herstellen
2. Führen Sie auf dem SQL Client die folgenden gespeicherten Prozeduren aus und *`group_replication_user_password`* ersetzen Sie sie durch das Kennwort für den `rdsgrprepladmin` Benutzer. Der `rdsgrprepladmin` Benutzer ist für Gruppenreplikationsverbindungen in einem Active-Active-Cluster reserviert. Das Passwort für diesen Benutzer muss auf allen DB-Instances in einem Active-Active-Cluster identisch sein.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   call mysql.rds_group_replication_start(1);
   ```

   In diesem Beispiel wird der `binlog retention hours` Wert auf festgelegt168, was bedeutet, dass binäre Protokolldateien sieben Tage lang auf der DB-Instance aufbewahrt werden. Sie können diesen Wert an Ihre Anforderungen anpassen.

   In diesem Beispiel wird 1 in der `mysql.rds_group_replication_start` gespeicherten Prozedur angegeben, dass eine neue Gruppe mit der aktuellen DB-Instance initialisiert werden soll.

Weitere Hinweise zu den im Beispiel aufgerufenen gespeicherten Prozeduren finden Sie unterVerwaltung von aktiv-aktiven Clustern.

Schritt 5: Starten Sie die Replikation auf den anderen DB-Instances im Active-Active-Cluster

Verwenden Sie für jede DB-Instance im Active-Active-Cluster einen SQL Client, um eine Verbindung mit der Instance herzustellen, und führen Sie die folgenden gespeicherten Prozeduren aus. Ersetzen Sie es *`group_replication_user_password`* durch das Passwort für den `rdsgrprepladmin` Benutzer.

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

In diesem Beispiel wird der `binlog retention hours` Wert auf festgelegt168, was bedeutet, dass binäre Protokolldateien auf jeder DB-Instance sieben Tage lang aufbewahrt werden. Sie können diesen Wert an Ihre Anforderungen anpassen.

In diesem Beispiel wird `0` in der `mysql.rds_group_replication_start` gespeicherten Prozedur angegeben, dass die aktuelle DB-Instance einer vorhandenen Gruppe hinzugefügt werden soll.

> **Tip**
>
> Stellen Sie sicher, dass Sie diese gespeicherten Prozeduren auf allen anderen DB-Instances im Active-Active-Cluster ausführen.

Schritt 6: (Empfohlen) Überprüfen Sie den Status des Active-Active-Clusters

Um sicherzustellen, dass jedes Mitglied des Clusters korrekt konfiguriert ist, überprüfen Sie den Status des Clusters, indem Sie eine Verbindung zu einer DB-Instance im Active-Active-Cluster herstellen und den folgenden Befehl ausführen: SQL

```
SELECT * FROM performance_schema.replication_group_members;
```

Ihre Ausgabe sollte `ONLINE` für jede DB-Instance wie in `MEMBER_STATE` der folgenden Beispielausgabe angezeigt werden:

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Informationen zu den möglichen `MEMBER_STATE` Werten finden Sie unter Gruppenreplikations-Serverstatus in der SQL Dokumentation Meine.

Schritt 7: (Optional) Importieren Sie Daten in eine DB-Instance im Active-Active-Cluster

Sie können Daten aus einer My SQL Database in eine DB-Instance im Active-Active-Cluster importieren. Nachdem die Daten importiert wurden, repliziert die Gruppenreplikation sie auf die anderen DB-Instances im Cluster.

Hinweise zum Importieren von Daten finden Sie unterImport von Daten in eine Amazon RDS for MySQL MySQL-Datenbank mit reduzierter Ausfallzeit.

Hinzufügen einer DB-Instance zu einem Active-Active-Cluster

Sie können einem SQL Active-Active-Cluster von Amazon RDS for My eine DB-Instance hinzufügen, indem Sie einen DB-Snapshot oder eine DB-Instance zu einem bestimmten Zeitpunkt wiederherstellen. Ein aktiv-aktiver Cluster kann bis zu neun DB-Instances umfassen.

Wenn Sie eine DB-Instance auf einen bestimmten Zeitpunkt wiederherstellen, umfasst sie in der Regel neuere Transaktionen als eine DB-Instance, die aus einem DB-Snapshot wiederhergestellt wurde. Wenn die DB-Instance über neuere Transaktionen verfügt, müssen zu Beginn der Replikation weniger Transaktionen angewendet werden. Die point-in-time Wiederherstellung zum Hinzufügen einer DB-Instance zu einem Cluster ist also in der Regel schneller als die Wiederherstellung aus einem DB-Snapshot.

Themen

- Hinzufügen einer DB-Instance zu einem aktiv-aktiven Cluster mithilfe der Wiederherstellung point-in-time
- Hinzufügen einer DB-Instance zu einem Active-Active-Cluster mithilfe eines DB-Snapshots

Hinzufügen einer DB-Instance zu einem aktiv-aktiven Cluster mithilfe der Wiederherstellung point-in-time

Sie können eine DB-Instance zu einem aktiv-aktiven Cluster hinzufügen, indem Sie eine point-in-time Wiederherstellung auf einer DB-Instance im Cluster durchführen.

Hinweise zur Wiederherstellung einer DB-Instance zu einem anderen Zeitpunkt finden Sie AWS-Region unter. Automatisierte Backups auf ein anderes replizieren AWS-Region

So fügen Sie mithilfe der Wiederherstellung eine DB-Instance zu einem Active-Active-Cluster hinzu point-in-time

1. Erstellen Sie eine neue DB-Instance, indem Sie eine point-in-time Wiederherstellung auf einer DB-Instance im Active-Active-Cluster durchführen.

   Sie können eine point-in-time Wiederherstellung auf jeder DB-Instance im Cluster durchführen, um die neue DB-Instance zu erstellen. Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

   > **Important**
   >
   > Ordnen Sie währenddessen die neue DB-Instance einer DB-Parametergruppe zu, für die die Active-Active-Cluster-Parameter festgelegt sind. point-in-time-recovery Andernfalls wird die Gruppenreplikation auf der neuen DB-Instance nicht gestartet.

Informationen zu den Parametern und den jeweils erforderlichen Einstellungen finden Sie unterErforderliche Parametereinstellungen für Active-Active-Cluster.

> **Tip**
>
> Wenn Sie vor Beginn der point-in-time Wiederherstellung einen Snapshot der DB-Instance erstellen, können Sie möglicherweise den Zeitaufwand für die Anwendung von Transaktionen auf der neuen DB-Instance reduzieren.

2. Fügen Sie die DB-Instance dem `group_replication_group_seeds` Parameter in jeder DB-Parametergruppe hinzu, die einer DB-Instance im Active-Active-Cluster zugeordnet ist, einschließlich der DB-Parametergruppe, die Sie der neuen DB-Instance zugeordnet haben.

   Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

3. Stellen Sie in einem SQL Client eine Verbindung mit der neuen DB-Instance her und rufen Sie die mysql.rds_group_replication_set_recovery_channel gespeicherte Prozedur auf. Ersetzen *`group_replication_user_password`* mit dem Passwort für den `rdsgrprepladmin` Benutzer.

```
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
```

4. Rufen Sie mithilfe des SQL Clients die mysql.rds_group_replication_start gespeicherte Prozedur auf, um die Replikation zu starten:

```
call mysql.rds_group_replication_start(0);
```

Hinzufügen einer DB-Instance zu einem Active-Active-Cluster mithilfe eines DB-Snapshots

Sie können einem aktiv-aktiven Cluster eine DB-Instance hinzufügen, indem Sie einen DB-Snapshot einer DB-Instance im Cluster erstellen und dann den DB-Snapshot wiederherstellen.

Hinweise zum Kopieren eines Snapshots in einen anderen finden Sie AWS-Region unter. the section called “Regionsübergreifendes Kopieren”

So fügen Sie mithilfe eines DB-Snapshots eine DB-Instance zu einem Active-Active-Cluster hinzu

1. Erstellen Sie einen DB-Snapshot einer DB-Instance im Active-Active-Cluster.

   Sie können einen DB-Snapshot jeder DB-Instance im Cluster erstellen. Detaillierte Anweisungen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
2. Stellen Sie eine DB-Instance aus dem DB-Snapshot wieder her.

   Ordnen Sie während des Snapshot-Wiederherstellungsvorgangs die neue DB-Instance einer DB-Parametergruppe zu, für die die Active-Active-Cluster-Parameter festgelegt sind. Informationen zu den Parametern und den jeweils erforderlichen Einstellungen finden Sie unter. Erforderliche Parametereinstellungen für Active-Active-Cluster

   Hinweise zum Wiederherstellen einer DB-Instance aus einem DB-Snapshot finden Sie unterWiederherstellung auf einer DB-Instance.
3. Fügen Sie die DB-Instance dem `group_replication_group_seeds` Parameter in jeder DB-Parametergruppe hinzu, die einer DB-Instance im Active-Active-Cluster zugeordnet ist, einschließlich der DB-Parametergruppe, die Sie der neuen DB-Instance zugeordnet haben.

   Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.
4. Stellen Sie in einem SQL Client eine Verbindung mit der neuen DB-Instance her und rufen Sie die mysql.rds_group_replication_set_recovery_channel gespeicherte Prozedur auf. Ersetzen *`group_replication_user_password`* mit dem Passwort für den `rdsgrprepladmin` Benutzer.

   ```
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   ```
5. Rufen Sie mithilfe des SQL Clients die mysql.rds_group_replication_start gespeicherte Prozedur auf, um die Replikation zu starten:

   ```
   call mysql.rds_group_replication_start(0);
   ```

Überwachung von aktiv-aktiven Clustern

Die Überwachung von aktiv-aktiven Clustern in Amazon RDS for My SQL ist entscheidend für die Überwachung der Leistung, Replikationsintegrität und Knotensynchronisierung. Sie können Ihren

Active-Active-Cluster überwachen, indem Sie eine Verbindung zu einer DB-Instance im Cluster herstellen und den folgenden Befehl ausführen: SQL

```
SELECT * FROM performance_schema.replication_group_members;
```

Ihre Ausgabe sollte `ONLINE` für jede DB-Instance wie in `MEMBER_STATE` der folgenden Beispielausgabe angezeigt werden:

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+----------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+----------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+----------------
+----------------------------+
3 rows in set (0.00 sec)
```

Informationen zu den möglichen `MEMBER_STATE` Werten finden Sie unter Gruppenreplikations-Serverstatus in der SQL Dokumentation Meine.

Beenden der Gruppenreplikation auf einer DB-Instance in einem Active-Active-Cluster

Sie können die Gruppenreplikation auf einer DB-Instance in einem aktiv-aktiven Cluster beenden. Wenn Sie die Gruppenreplikation beenden, wird die DB-Instance in den super-read-only Modus versetzt, bis die Replikation neu gestartet oder diese DB-Instance aus dem aktiv-aktiven Cluster entfernt wird. Informationen zum super-read-only Modus finden Sie unter Meine Dokumentation. SQL

Um die Gruppenreplikation für einen Active-Active-Cluster vorübergehend zu beenden

1. Stellen Sie mithilfe eines Clients eine Connect zu einer DB-Instance im Active-Active-Cluster her. SQL

   Weitere Informationen zum Herstellen einer Verbindung zu einer RDS For My SQL DB-Instance finden Sie unter. Verbindung zu Ihrer MySQL-DB-Instance herstellen
2. Rufen Sie im SQL Client die mysql.rds_group_replication_stop gespeicherte Prozedur auf:

   ```
   call mysql.rds_group_replication_stop();
   ```

Umbenennen einer DB-Instance in einem Active-Active-Cluster

Sie können den Namen einer DB-Instance in einem Active-Active-Cluster ändern. Um mehr als eine DB-Instance in einem aktiv-aktiven Cluster umzubenennen, tun Sie dies jeweils für eine DB-Instance. Benennen Sie also eine DB-Instance um und fügen Sie sie erneut dem Cluster hinzu, bevor Sie die nächste DB-Instance umbenennen.

Um eine DB-Instance in einem aktiv-aktiven Cluster umzubenennen

1. Stellen Sie eine Connect zur DB-Instance in einem SQL Client her und rufen Sie die mysql.rds_group_replication_stop gespeicherte Prozedur auf:

   ```
   call mysql.rds_group_replication_stop();
   ```
2. Benennen Sie die DB-Instance um, indem Sie den Anweisungen unter folgenUmbenennen einer DB-Instance.
3. Ändern Sie den `group_replication_group_seeds` Parameter in jeder DB-Parametergruppe, die einer DB-Instance im Active-Active-Cluster zugeordnet ist.

   Ersetzen Sie in der Parametereinstellung den alten DB-Instance-Endpunkt durch den neuen DB-Instance-Endpunkt. Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.
4. Stellen Sie eine Connect zur DB-Instance in einem SQL Client her und rufen Sie die mysql.rds_group_replication_start gespeicherte Prozedur auf:

   ```
   call mysql.rds_group_replication_start(0);
   ```

Entfernen einer DB-Instance aus einem Active-Active-Cluster

Wenn Sie eine DB-Instance aus einem aktiv-aktiven Cluster entfernen, wird sie wieder zu einer eigenständigen DB-Instance.

Um eine DB-Instance aus einem aktiv-aktiven Cluster zu entfernen

1. Stellen Sie eine Connect zur DB-Instance in einem SQL Client her und rufen Sie die mysql.rds_group_replication_stop gespeicherte Prozedur auf:

   ```
   call mysql.rds_group_replication_stop();
   ```

2. Ändern Sie den `group_replication_group_seeds` Parameter für die DB-Instances, die im Active-Active-Cluster verbleiben.

   Löschen Sie im `group_replication_group_seeds` Parameter die DB-Instance, die Sie aus dem Active-Active-Cluster entfernen möchten. Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

3. Ändern Sie die Parameter der DB-Instance, die Sie aus dem Active-Active-Cluster entfernen, sodass sie nicht mehr Teil des Clusters ist.

   Sie können die DB-Instance entweder einer anderen Parametergruppe zuordnen oder die Parameter in der DB-Parametergruppe ändern, die der DB-Instance zugeordnet ist. Zu den zu ändernden Parametern gehören `group_replication_group_namerds.group_replication_enabled`, und`group_replication_group_seeds`. Weitere Hinweise zu aktiven/aktiven Clusterparametern finden Sie unter. Erforderliche Parametereinstellungen für Active-Active-Cluster

   Wenn Sie die Parameter in einer DB-Parametergruppe ändern, stellen Sie sicher, dass die DB-Parametergruppe nicht mit anderen DB-Instances im aktiv-aktiven Cluster verknüpft ist.

4. Starten Sie die DB-Instance neu, die Sie aus dem Active-Active-Cluster entfernt haben, damit die neuen Parametereinstellungen wirksam werden.

   Detaillierte Anweisungen finden Sie unter Neustarten einer DB-Instance.

Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation

Um Daten von einer RDS for My SQL DB-Instance in eine My SQL Instance zu exportieren, die außerhalb von Amazon ausgeführt wirdRDS, können Sie die Replikation verwenden. In diesem Szenario ist die My SQL DB-Instance die My SQL DB-Quell-Instance, und die My SQL Instance, die außerhalb von Amazon ausgeführt wird, RDS ist die externe My SQL Database.

Die externe SQL Datenbank Meine Datenbank kann entweder lokal in Ihrem Rechenzentrum oder auf einer EC2 Amazon-Instance ausgeführt werden. Die externe My SQL Database-Instanz muss dieselbe Version wie die My SQL DB-Quell-Instance oder eine neuere Version ausführen.

Die Replikation auf eine externe My SQL Database wird nur während der Zeit unterstützt, die für den Export einer Datenbank aus der My SQL DB-Quellinstanz benötigt wird. Die Replikation sollte beendet werden, wenn die Daten exportiert wurden und Anwendungen auf die externe My SQL Instance zugreifen können.

Im Folgenden sind die durchzuführenden Schritte aufgeführt. In den folgenden Abschnitten wird jeder Schritt im Detail erklärt.

1. Bereiten Sie eine externe My SQL DB-Instance vor.
2. Bereiten Sie die My SQL DB-Quellinstanz für die Replikation vor.
3. Verwenden Sie das Hilfsprogramm mysqldump, um die Datenbank von der My SQL DB-Quell-Instance auf die externe My Database zu übertragen. SQL
4. Starten Sie die Replikation zur externen Datenbank Meine Datenbank. SQL
5. Nachdem der Exportvorgang abgeschlossen ist, stoppt die Replikation.

Bereiten Sie eine externe Datenbank vom Typ Meine SQL Datenbank vor

Führen Sie die folgenden Schritte aus, um die externe SQL Datenbank Meine Datenbank vorzubereiten.

So bereiten Sie die externe SQL Datenbank Meine Datenbank vor

1. Installieren Sie die externe SQL Datenbank Meine Datenbank.

2. Stellen Sie als Masterbenutzer eine Connect mit der externen SQL Datenbank Meine Datenbank her. Erstellen Sie dann die Benutzer, die erforderlich sind, um die Administratoren, Anwendungen und Services zu unterstützen, die auf die Datenbank zugreifen.
3. Folgen Sie den Anweisungen in der SQL Dokumentation Meine, um die externe SQL Datenbank Meine Datenbank als Replikat vorzubereiten. Weitere Informationen finden Sie unter Einstellung der Replikatkonfiguration in der Dokumentation Meine SQL Daten.
4. Konfigurieren Sie eine Ausgangsregel für die externe Datenbank „Meine SQL Datenbank“, sodass sie während des Exports als Lesereplikat fungiert. Die Ausgangsregel ermöglicht es der externen My SQL Database, während der Replikation eine Verbindung mit der My SQL DB-Quell-Instance herzustellen. Geben Sie eine Ausgangsregel an, die Transmission Control Protocol (TCP) -Verbindungen zum Port und zur IP-Adresse der My SQL DB-Quellinstanz zulässt.

   Geben Sie die entsprechenden Ausgangsregeln für Ihre Umgebung an:

   - Wenn die externe SQL Datenbank Meine Datenbank in einer EC2 Amazon-Instance in einer virtuellen privaten Cloud (VPC) läuft, die auf dem VPC Amazon-Service basiert, geben Sie die Ausgangsregeln in einer VPC Sicherheitsgruppe an. Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
   - Wenn die externe SQL Datenbank „Meine Datenbank“ lokal installiert ist, geben Sie die Ausgangsregeln in einer Firewall an.
5. Wenn die externe SQL Datenbank Meine Datenbank in einem ausgeführt wirdVPC, konfigurieren Sie zusätzlich zur VPC Ausgangsregel der Sicherheitsgruppe Regeln für die Zugriffskontrollliste (ACL):

   - Konfigurieren Sie eine ACL Eingangsregel, die den TCP Datenverkehr zu den Ports 1024—65535 von der IP-Adresse der My DB-Quell-Instance aus ermöglicht. SQL
   - Konfigurieren Sie eine ACL Ausgangsregel, die ausgehenden TCP Datenverkehr zum Port und zur IP-Adresse der My DB-Quellinstanz zulässt. SQL

   Weitere Informationen zum VPC Amazon-Netzwerk ACLs finden Sie unter Netzwerk ACLs im VPCAmazon-Benutzerhandbuch.
6. (Optional) Legen Sie den Parameter `max_allowed_packet` auf die maximale Größe fest, um Replikationsfehler zu vermeiden. Wir empfehlen diese Einstellung.

Bereiten Sie die My SQL DB-Quell-Instance vor

Führen Sie die folgenden Schritte aus, um die My SQL DB-Quellinstanz als Replikationsquelle vorzubereiten.

So bereiten Sie die My SQL DB-Quell-Instance vor

1. Stellen Sie sicher, dass Ihr Client-Computer über genügend freien Speicherplatz verfügt, um die Binärprotokolle während des Einrichtens der Replikation zu speichern.
2. Stellen Sie eine Connect der My SQL DB-Quell-Instance her und erstellen Sie ein Replikationskonto, indem Sie den Anweisungen unter Creating a User for Replication in der SQL Dokumentation My folgen.
3. Konfigurieren Sie Eingangsregeln auf dem System, auf dem die My SQL DB-Quell-Instance ausgeführt wird, damit die externe My SQL Database während der Replikation eine Verbindung herstellen kann. Geben Sie eine Eingangsregel an, die TCP Verbindungen zu dem von der My SQL DB-Quell-Instance verwendeten Port von der IP-Adresse der externen My SQL database aus ermöglicht.
4. Geben Sie die Ausgangregeln an:

   - Wenn die My SQL DB-Quell-Instance in einer ausgeführt wirdVPC, geben Sie die Eingangsregeln in einer VPC Sicherheitsgruppe an. Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.
5. Wenn die Quell-Instance „Meine SQL DB“ in einer ausgeführt wirdVPC, konfigurieren Sie zusätzlich zur Regel für den Zugriff auf Sicherheitsgruppen weitere VPC ACL Regeln:

   - Konfigurieren Sie eine ACL Eingangsregel, um TCP Verbindungen zu dem von der RDS Amazon-Instance verwendeten Port von der IP-Adresse der externen My SQL database aus zuzulassen.
   - Konfigurieren Sie eine ACL Ausgangsregel, um TCP Verbindungen von den Ports 1024—65535 zur IP-Adresse der externen Datenbank Meine Datenbank zuzulassen. SQL

   Weitere Informationen zum VPC Amazon-Netzwerk ACLs finden Sie unter Netzwerk ACLs im VPCAmazon-Benutzerhandbuch.
6. Stellen Sie sicher, dass der Aufbewahrungszeitraum für Backup auf eine ausreichend lange Dauer eingestellt ist, damit keine Binärprotokolle während des Exportvorgangs bereinigt werden. Wenn Protokolle bereinigt werden, bevor der Exportvorgang abgeschlossen ist, müssen

Sie die Replikation von Anfang an neu starten. Weitere Informationen zum Einstellen des Aufbewahrungszeitraums für Backups finden Sie unter Einführung in Backups.

7. Verwenden Sie die gespeicherte Prozedur `mysql.rds_set_configuration`, um das Binärprotokoll für den Aufbewahrungszeitraum auf eine ausreichend lange Dauer einzustellen, damit die Binärprotokolle während des Exportvorgangs nicht bereinigt werden. Weitere Informationen finden Sie unter Zugriff auf MySQL-Binärprotokolle.
8. Erstellen Sie eine RDS Amazon-Read-Replica aus der My SQL DB-Quell-Instance, um weiterhin sicherzustellen, dass die Binärprotokolle der My SQL DB-Quell-Instance nicht gelöscht werden. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats.
9. Nachdem die Amazon RDS Read Replica erstellt wurde, rufen Sie die `mysql.rds_stop_replication` gespeicherte Prozedur auf, um den Replikationsvorgang zu beenden. Die My SQL DB-Quell-Instance löscht ihre binären Protokolldateien nicht mehr, sodass sie für den Replikationsprozess verfügbar sind.
10. (Optional) Legen Sie sowohl den Parameter `max_allowed_packet` als auch den Parameter `slave_max_allowed_packet` auf die maximale Größe fest, um Replikationsfehler zu vermeiden. Die maximale Größe für beide Parameter beträgt 1 GB. Wir empfehlen diese Einstellung für beide Parameter. Weitere Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Kopieren der Datenbank

Führen Sie die folgenden Schritte aus, um die Datenbank zu kopieren.

So kopieren Sie die Datenbank:

1. Connect zur RDS Read Replica der My SQL DB-Quell-Instance her und führen Sie die SQL `SHOW REPLICA STATUS\G` My-Anweisung aus. Beachten Sie die Werte für Folgendes:
   - `Master_Host`
   - `Master_Port`
   - `Master_Log_File`
   - `Exec_Master_Log_Pos`

> **Note**
>
> Frühere Versionen von My SQL wurden `SHOW SLAVE STATUS` anstelle von `SHOW REPLICA STATUS` verwendet. Wenn Sie eine SQL Version von My vor 8.0.23 verwenden, verwenden Sie. `SHOW SLAVE STATUS`

2. Verwenden Sie das Hilfsprogramm mysqldump, um einen Snapshot zu erstellen, der die Daten von Amazon RDS auf Ihren lokalen Client-Computer kopiert. Stellen Sie sicher, dass Ihr Client-Computer ausreichend Speicherplatz zur Verfügung hat, um die zu replizierenden `mysqldump`-Dateien aus den Datenbanken zu speichern. Dieser Vorgang kann für große Datenbanken einige Stunden in Anspruch nehmen. Folgen Sie den Anweisungen unter Erstellen eines Daten-Snapshots mithilfe von mysqldump in der Dokumentation „Meine“. SQL

   Im folgenden Beispiel wird `mysqldump` auf einem Client ausgeführt und schreibt das Dump in eine Datei.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   mysqldump -h source_MySQL_DB_instance_endpoint \
       -u user \
       -ppassword \
       --port=3306 \
       --single-transaction \
       --routines \
       --triggers \
       --databases  database database2 > path/rds-dump.sql
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   mysqldump -h source_MySQL_DB_instance_endpoint ^
       -u user ^
       -ppassword ^
       --port=3306 ^
       --single-transaction ^
       --routines ^
       --triggers ^
       --databases  database database2 > path\rds-dump.sql
   ```

Sie können die Sicherungsdatei in die externe SQL Datenbank Meine Datenbank laden. Weitere Informationen finden Sie unter Backups im SQL -Format Reloading in der Dokumentation MySQL. Sie können ein anderes Hilfsprogramm ausführen, um die Daten in die externe Datenbank Meine SQL Datenbank zu laden.

Abschließen des Exportvorgangs

Führen Sie die folgenden Schritte aus, um den Export abzuschließen.

So schließen Sie den Export ab:

1. Verwenden Sie die SQL `CHANGE MASTER` Anweisung My, um die externe SQL Datenbank My zu konfigurieren. Geben Sie die ID und das Passwort des Benutzers an, dem `REPLICATION SLAVE`-Berechtigungen erteilt wurden. Geben Sie die `Exec_Master_Log_Pos` Werte `Master_HostMaster_Port`,`Relay_Master_Log_File`, und an, die Sie aus der SQL `SHOW REPLICA STATUS\G` My-Anweisung erhalten haben, die Sie auf der RDS Read Replica ausgeführt haben. Weitere Informationen finden Sie in der SQL Dokumentation CHANGEMASTERzu My unter TO Statement.

   > **Note**
   >
   > Frühere Versionen von My SQL wurden `SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie eine SQL Version von My vor 8.0.23 verwenden, verwenden Sie. `SHOW SLAVE STATUS`

2. Verwenden Sie den SQL `START REPLICA` Befehl My, um die Replikation von der My SQL DB-Quell-Instance zur externen My SQL Database zu initiieren.

   Dadurch wird die Replikation von der My SQL DB-Quell-Instance gestartet und alle Quelländerungen exportiert, die nach dem Beenden der Replikation von der Amazon RDS Read Replica vorgenommen wurden.

   > **Note**
   >
   > Frühere Versionen von My SQL wurden `START SLAVE` anstelle von `START REPLICA` verwendet. Wenn Sie eine SQL Version von My vor 8.0.23 verwenden, verwenden Sie. `START SLAVE`

3. Führen Sie den SQL `SHOW REPLICA STATUS\G` Befehl My für die externe SQL Datenbank Meine Datenbank aus, um zu überprüfen, ob sie als Read Replica funktioniert. Weitere Informationen zur Interpretation der Ergebnisse finden Sie unter SHOWSLAVE| REPLICA STATUS Statement in der SQL Dokumentation My.
4. Nachdem die Replikation auf der externen My SQL Database-Instance die My SQL DB-Quell-Instance eingeholt hat, verwenden Sie den SQL `STOP REPLICA` Befehl My, um die Replikation von der My SQL DB-Quell-Instance aus zu beenden.

   > **Note**
   >
   > Frühere Versionen von My SQL wurden `STOP SLAVE` anstelle von verwendet`STOP REPLICA`. Wenn Sie eine SQL Version von My vor 8.0.23 verwenden, verwenden Sie. `STOP SLAVE`

5. Rufen Sie auf der Amazon RDS Read Replica die `mysql.rds_start_replication` gespeicherte Prozedur auf. Auf diese Weise kann Amazon RDS damit beginnen, die binären Protokolldateien aus der My SQL DB-Quell-Instance zu löschen.

Optionen für Meine SQL DB-Instances

Im Folgenden finden Sie eine Beschreibung der Optionen oder zusätzlichen Funktionen, die für RDS Amazon-Instances verfügbar sind, auf denen die My SQL DB-Engine ausgeführt wird. Damit diese Optionen aktiviert werden, können Sie diese einer benutzerdefinierten Optionsgruppe hinzufügen und anschließend der Optionsgruppe für Ihre DB-Instance zuordnen. Weitere Informationen über das Arbeiten mit Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.

Amazon RDS unterstützt die folgenden Optionen für MySQL:

Option	Options-ID	Engine-Versionen
MariaDB Audit Plugin-Unterstützung für My SQL	`MARIADB_AUDIT_PLUGIN`	Alle meine SQL 8.4-Versionen Meine SQL 8.0.28 und höhere 8.0-Versionen Alle meine 5.7-Versionen SQL
Unterstützung für MySQL-memcached	`MEMCACHED`	Alle meine SQL 5.7- und 8.0-Versionen

MariaDB Audit Plugin-Unterstützung für My SQL

Amazon RDS bietet ein Audit-Plugin für My SQL Database-Instances an, das auf dem Open-Source-MariaDB-Audit-Plugin basiert. Weitere Informationen finden Sie im Audit-Plug-In für das My SQL GitHub Server-Repository.

> **Note**
>
> Das Audit-Plugin für My SQL basiert auf dem MariaDB Audit Plugin. In diesem Artikel bezeichnen wir es als MariaDB-Audit-Plugin.

Das MariaDB-Audit-Plugin zeichnet Datenbankaktivitäten auf, z. B. Benutzer, die sich bei der Datenbank anmelden, in der Datenbank ausgeführte Abfragen und vieles mehr. Der Datensatz der Datenbankaktivität wird in einer Protokolldatei gespeichert.

Audit-Plugin-Optionseinstellungen

Amazon RDS unterstützt die folgenden Einstellungen für die MariaDB Audit Plugin-Option.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	Der Speicherort der Protokolldatei. Die Protokolldatei beinhaltet den Datensatz der Aktivitäten die in festgelegt wurde `SERVER_AUDIT_EVENTS` . Weitere Informationen erhalten Sie unter Anzeigen und Auflisten von Datenbank-Protokolldateien und MySQL-Datenbank-Protokolldateien.
`SERVER_AUDIT_FILE_ROTATE_SIZE`	1 – 1000000000	1000000	Die Größe in Bytes, die bei Erreichen der Datei dazu führt, dass die Datei rotiert. Weitere Informationen finden Sie unter Überblick über RDS-for-MySQL-Datenbankprotokolle.
`SERVER_AUDIT_FILE_`	0 – 100	9	Die Anzahl der zu speichernden Protokoll rotationen, wenn `server_audit_outpu`

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
ROTATIONS			`t_type=file` . Wenn der Wert auf 0 festgelegt ist, rotiert die Protokolldatei niemals. Weitere Informationen erhalten Sie unter Überblick über RDS-for-MySQL-Datenbankprotokolle und Herunterladen einer Datenbank-Protokolldatei.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_EVENTS	CONNECT, QUERY, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	Die Arten von Aktivitäten, die im Protokoll aufgezeichnet werden sollen. Die Installation des MariaDB Audit Plugins ist selbst protokolliert. • CONNECT: Protokollieren Sie erfolgreiche und nicht erfolgreiche Verbindungen zur Datenbank und trennen Sie die Verbindung zur Datenbank. • QUERY: Protokollieren Sie den Text aller Abfragen, die für die Datenbank ausgeführt werden. • QUERY_DDL : Ähnlich wie das QUERY Ereignis, gibt jedoch nur Abfragen in der Datendefinitionssprache (DDL) (CREATEALTER, usw.) zurück. • QUERY_DML : Ähnlich wie das QUERY Ereignis, gibt jedoch nur Abfragen (INSERTUPDATE, usw. usw.DML) zur Datenmanipulationssprache (SELECT) zurück. • QUERY_DML_NO_SELECT : : Ähnlich dem QUERY_DML -Ereignis, protokolliert jedoch keine SELECT-Abfragen. • QUERY_DCL : Ähnlich dem QUERY Ereignis, gibt jedoch nur Abfragen (GRANTREVOKE, usw.DCL) in Data Control Language () zurück. For MySQL, TABLE wird nicht unterstützt.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_INCL_USERS	Mehrere kommaseparierte Werte	Keine	Füge nur Aktivität von den angegebenen Benutzern ein. Standardmäßig werden Aktivitäten für alle Benutzer aufgezeichnet. SERVER_AUDIT_INCL_USERS und schließen SERVER_AUDIT_EXCL_USERS sich gegenseitig aus. Wenn Sie Werte hinzufügen SERVER_AUDIT_INCL_USERS , stellen Sie sicher, dass keine Werte hinzugefügt werden SERVER_AUDIT_EXCL_USERS .

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
SERVER_AUDIT_EXCL_USERS	Mehrere kommaseparierte Werte	Keine	Aktivität von den angegebenen Benutzern ausschließen. Standardmäßig werden Aktivitäten für alle Benutzer aufgezeichnet. SERVER_AUDIT_INCL_USERS und schließen SERVER_AUDIT_EXCL_USERS sich gegenseitig aus. Wenn Sie Werte hinzufügen SERVER_AUDIT_EXCL_USERS , stellen Sie sicher, dass keine Werte hinzugefügt werden SERVER_AUDIT_INCL_USERS . Der rdsadmin-Benutzer fragt die Datenbank jede Sekunde ab, um den Zustand der Datenbank zu überprüfen. Abhängig von Ihren anderen Einstellungen kann diese Aktivität dazu führen, dass die Größe Ihrer Protokolldatei sehr schnell sehr groß wird. Wenn Sie diese Aktivität nicht aufzeichnen müssen, fügen Sie den Benutzer rdsadmin zur Liste SERVER_AUDIT_EXCL_USERS hinzu. **Note** CONNECTDie Aktivität wird stets für alle Benutzer erfasst, auch wenn ein Benutzer für diese Optionseinstellung angegeben ist.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
`SERVER_AUDIT_LOGGING`	`ON`	`ON`	Die Protokollierung ist aktiv. Der einzige gültige Wert ist `ON`. Amazon RDS unterstützt die Deaktivierung der Protokollierung nicht. Wenn Sie die Protokollierung deaktivieren möchten, entfernen Sie das MariaDB-Audit-Plugin. Weitere Informationen finden Sie unter Entfernen des MariaDB-Audit-Plugins.
`SERVER_AUDIT_QUERY_LOG_LIMIT`	0 – 2147483647	1024	Die Längenbegrenzung der Abfragezeichenfolge in einem Datensatz.

Hinzufüge des MariaDB-Audit-Plugins

Der allgemeine Vorgang zum Hinzufügen des MariaDB-Audit-Plugins zu einer DB-Instance ist wie folgt:

- Erstellen einer neuen Optionsgruppe oder Kopieren oder Ändern einer bestehenden Optionsgruppe
- Hinzufügen der Option zur Optionsgruppe
- Zuordnen der Optionsgruppe zu einer DB-Instance

Nachdem Sie das MariaDB-Audit-Plugin hinzugefügt haben, müssen Sie Ihre DB-Instance nicht neu starten. Sobald die Optionsgruppe aktiv ist, beginnt sofort die Überwachung.

> **Important**
>
> Das Hinzufügen des MariaDB-Audit-Plugin zu einer DB-Instance kann einen Ausfall verursachen. Sie sollten das MariaDB-Audit-Plugin während eines Wartungsfensters hinzufügen oder wenn die Arbeitslast der Datenbank gering ist.

So fügen Sie das MariaDB-Audit-Plugin hinzu

1. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe. Wählen Sie mysql für Engine und 5.7, 8.0 oder 8.4 für die Major-Engine-Version. Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die PLUGIN Option MARIADBAUDIT_ _ zur Optionsgruppe hinzu und konfigurieren Sie die Optionseinstellungen. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Audit-Plugin-Optionseinstellungen.
3. Wenden Sie die Optionsgruppe auf eine neue oder vorhandene DB-Instance an.
   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Format des Prüfprotokolls

Protokolldateien werden als kommagetrennte Variablendateien (CSV) im Format UTF -8 dargestellt.

> **Tip**
>
> Protokolldateieinträge folgen keiner sequenziellen Reihenfolge. Um die Einträge zu sortieren, verwenden Sie den Wert des Zeitstempels. Sie müssen eventuell alle Protokolldateien überprüfen, um die aktuellen Ereignisse zu sehen. Um mehr Flexibilität beim Sortieren und Durchsuchen der Protokolldaten zu erhalten, aktivieren Sie die Einstellung zum Hochladen der Auditprotokolle CloudWatch und zum Anzeigen dieser Protokolle über die Benutzeroberfläche. CloudWatch
>
> Um Auditdaten mit mehr Feldtypen und mit JSON formatierter Ausgabe anzuzeigen, können Sie auch die Funktion Database Activity Streams verwenden. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mithilfe von Datenbankaktivitätsstreams.

Die Prüfprotokolldateien beinhalten die folgenden durch Kommata getrennten Informationen in Zeilen in der festgelegten Reihenfolge:

Feld	Beschreibung
Zeitstempel	`YYYYMMDD`, gefolgt von `HH:MI:SS` (24-Stunden-Uhr) für das protokollierte Ereignis.
serverhost	Der Name der Instance, für die das Ereignis protokolliert wird.
username	Der verbundene Benutzername des Benutzers.
host	Der Host, von dem sich der Benutzer verbunden hat.
connectionid	Die Nummer der Verbindungs-ID für den protokollierte Vorgang.
queryid	Die Nummer der Abfragen-ID, die verwendet werden kann, um Ereignisse für relationale Tabellenereignisse und zugehörige Abfragen zu finden. Für `TABLE`-Ereignisse werden mehrere Zeilen hinzugefügt.
operation	Der dokumentierte Aktionstyp. Mögliche Werte sind: `CONNECT`, `QUERY`, `READ`, `WRITE`, `CREATE`, `ALTER`, `RENAME` und `DROP`.
Datenbank	Die aktive Datenbank, wie vom `USE`-Befehl eingestellt.
Objekt	Bei `QUERY`-Ereignissen gibt dieser Wert die Abfrage an, die die Datenbank ausgeführt hat. Bei `TABLE`-Ereignissen gibt er den Tabellennamen an.
retcode	Der zurückgegebene Code des protokollierten Vorgangs.
connection_type	Der Sicherheitsstatus der Verbindung zum Server. Die möglichen Werte sind: • `0` – nicht definiert • 1— TCP /IP • `2` – Socket • `3` – Named-Pipe • `4` – SSL/TLS • `5` – gemeinsamer Arbeitsspeicher

Anzeigen und Herunterladen des MariaDB-Audit-Plugin-Protokolls

Nachdem Sie das MariaDB Audit-Plugin aktiviert haben, greifen Sie auf die Ergebnisse in den Protokolldateien genauso zu, wie auf andere textbasierte Protokolldateien. Die Auditprotokolldateien werden unter gespeicher `/rdsdbdata/log/audit/`. Weitere Informationen zum Anzeigen einer Protokolldatei in der Konsole finden Sie unter Anzeigen und Auflisten von Datenbank-Protokolldateien. Weitere Informationen zum Herunterladen der Protokolldatei finden Sie unter Herunterladen einer Datenbank-Protokolldatei.

Ändern der Einstellungen für das MariaDB-Audit-Plugin

Nachdem Sie das MariaDB-Audit-Plugin aktiviert haben, können Sie die Einstellungen ändern. Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Audit-Plugin-Optionseinstellungen.

Entfernen des MariaDB-Audit-Plugins

Amazon unterstützt das Deaktivieren der Protokollierung im MariaDB Audit Plugin RDS nicht. Sie können das Plugin jedoch aus einer DB-Instance entfernen. Wenn Sie das MariaDB-Audit-Plugin entfernen, wird die DB-Instance automatisch erneut gestartet, um die Überwachung zu beenden.

Führen Sie einen der folgenden Schritte aus, um das MariaDB-Audit-Plugin aus einer DB-Instance zu entfernen:

- Entfernen Sie das MariaDB-Audit-Plugin aus ihrer zugehörigen Optionsgruppe wie folgt: Diese Änderung wirkt sich auf alle DB-Instances aus, die die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe
- Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, die im Plugin nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Unterstützung für MySQL-memcached

Amazon RDS unterstützt die Verwendung der Schnittstelle memcached für InnoDB-Tabellen, die in MySQL 5.6 eingeführt wurde. Die API memcached erlaubt Anwendungen, InnoDB-Tabellen zu verwenden, ähnlich wie es in NoSQL-Schlüsselwert-Datenspeichern möglich ist.

> **Note**
>
> Die Memcached-Schnittstelle ist in MySQL 8.4 nicht mehr verfügbar. Wenn Sie Ihre DB-Instances auf MySQL 8.4 aktualisieren, müssen Sie sie memcached in vorhandenen Optionsgruppen deaktivieren.

Die memcached-Benutzeroberfläche ist ein einfacher, schlüsselbasierter Cache. Anwendungen verwenden memcached, um im Cache Schlüsselwert-Datenpaare einzufügen, zu manipulieren oder abzurufen. Mit MySQL 5.6 wurde ein Plugin eingeführt, das einen Daemon-Service implementiert, der Daten von InnoDB-Tabellen über das memcached-Protokoll freigibt. Weitere Informationen zum MySQL-memcached-Plug-in finden Sie unter InnoDB-Integration mit memcached.

So aktivieren Sie Memcached-Unterstützung für eine DB-Instance von RDS for MySQL

1. Bestimmen der Sicherheitsgruppe, die für den kontrollierten Zugriff auf die memcached-Schnittstelle verwendet werden soll. Wenn die vom Anwendungs-Set bereits verwendete SQL-Schnittstelle dieselbe ist, die auf die memcached-Schnittstelle zugreifen wird, können Sie die bestehende VPC-Sicherheitsgruppe verwenden, die von der SQL-Schnittstelle verwendet wird. Wenn ein anderes Anwendungs-Set auf die memcached-Schnittstelle zugreifen wird, definieren Sie eine neue VPC- oder DB-Sicherheitsgruppe. Weitere Informationen über das Verwalten von Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen
2. Erstellen einer benutzerdefinierten DB-Optionsgruppe, Auswählen von MySQL als Engine-Typ und Version. Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
3. Fügen Sie die Option MEMCACHED zur Optionsgruppe hinzu. Angeben des Ports, den die memcached-Schnittstelle verwenden soll, und der Sicherheitsgruppe, die für die Kontrolle des Zugriffs auf die Schnittstelle verwendet werden soll. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

4. Ändern der Optionseinstellungen, um die memcached-Parameter zu konfigurieren, falls notwendig. Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung.
5. Wenden Sie die Optionsgruppe auf eine Instance an. Amazon RDS aktiviert die memcached-Unterstützung für diese Instance, wenn die Optionsgruppe angewendet wurde:
   - Sie können die memcached-Unterstützung für eine neue Instance aktivieren, indem Sie beim Starten der Instance die benutzerdefinierte Optionsgruppe angeben. Weitere Informationen über das Starten einer MySQL-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Sie können die memcached-Unterstützung für eine bestehende Instance aktivieren, indem Sie beim Ändern der Instance eine benutzerdefinierte Optionsgruppe angeben. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.
6. Angeben, auf welche Spalten in Ihren MySQL-Tabellen über die memcached-Schnittstelle zugegriffen werden darf. Das Plugin memcached erstellt eine Katalog-Tabelle mit dem Namen containers in einer dedizierten Datenbank mit dem Namen innodb_memcache. Sie können eine Zeile in der containers-Tabelle einfügen, um eine InnoDB-Tabelle für den Zugriff durch memcached zu berechtigen. Sie können eine Spalte in der InnoDB-Tabelle festlegen, die verwendet werden soll, um memcached-Schlüsselwerte zu speichern, und eine oder mehrere Spalten, die dazu verwendet werden sollen, um die Datenwerte zu speichern, die mit dem Schlüssel zusammenhängen. Sie können einen Namen angeben, den eine memcached-Anwendung verwenden soll, um sich auf diese Spalten zu beziehen. Weitere Details zum Einfügen von Zeilen in der containers-Tabelle finden Sie unter InnoDB memcached Plugin Internals. Ein Beispiel für die Zuordnung einer InnoDB-Tabelle und den Zugriff über memcached finden Sie unter Schreiben von Anwendungen für das InnoDB-memcached Plugin.
7. Wenn sich die Anwendungen, die auf die memcached Schnittstelle zugreifen, auf anderen Computern oder EC2 Instanzen befinden als die Anwendungen, die die SQL-Schnittstelle verwenden, fügen Sie die Verbindungsinformationen für diese Computer der VPC-Sicherheitsgruppe hinzu, die der MySQL-Instance zugeordnet ist. Weitere Informationen über das Verwalten von Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Sie können die memcached-Unterstützung für eine Instance deaktivieren, indem Sie die Instance ändern und die Standard-Sicherheitsgruppe für Ihre MySQL-Version angeben. Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sicherheitsüberlegungen zu MySQL memcached

Das memcached-Protokoll unterstützt keine Benutzer-Authentifizierung. Weitere Informationen zu memcached-Sicherheitsüberlegungen von MySQL finden Sie unter Sicherheitsüberlegungen für das InnoDB Memcached Plugin in der MySQL-Dokumentation.

Sie können folgende Aktionen durchführen, um die Sicherheit der memcached-Schnittstelle zu erhöhen:

- Geben Sie einen anderen Port als den Standard-Port 11211 an, wenn Sie die Option MEMCACHED Ihrer Sicherheitsgruppe hinzufügen.
- Stellen Sie sicher, dass Sie die memcached Schnittstelle einer VPC-Sicherheitsgruppe zuordnen, die den Zugriff auf bekannte, vertrauenswürdige Clientadressen und EC2 Instances beschränkt. Weitere Informationen über das Verwalten von Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Verbindungsinformationen zu MySQL memcached

Um auf eine memcached-Schnittstelle zuzugreifen, müssen in einer Anwendung der DNS-Name der Amazon RDS-Instance und die memcached-Portnummer angegeben sein. Wenn eine Instance beispielsweise den DNS-Namen my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com trägt und die Memcached-Schnittstelle den Port 11212 verwendet, würde die Verbindungsinformation in PHP wie folgt aussehen:

```
<?php

$cache = new Memcache;
$cache->connect('my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com',11212);
?>
```

So finden Sie den DNS-Namen und den Memcached-Port einer MySQL-DB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke von die Region aus AWS Management Console, die die DB-Instance enthält.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.

4. Wählen Sie den Namen der MySQL DB-Instance, um deren Details anzuzeigen.
5. Beachten Sie im Bereich Verbinden den Wert im Feld Endpunkt. Der DNS-Name stimmt mit dem Endpunkt überein. Beachten Sie ebenfalls, dass der Port im Bereich Verbinden nicht für den Zugriff auf die memcached-Schnittstelle verwendet wird.
6. Achten Sie im Bereich Details auf den aufgeführten Namen im Feld Optionsgruppe.
7. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
8. Wählen Sie den Namen der Optionsgruppe aus, die von der MySQL-DB-Instance verwendet wird, um die Details für die Optionsgruppe anzuzeigen. Achten Sie im Bereich Optionen auf den Wert der Einstellung Port für die Option MEMCACHED.

Optionseinstellungen in MySQL memcached

Amazon RDS gibt die MySQL-memcached-Parameter als Optionseinstellungen in der Amazon RDS-MEMCACHED-Option frei.

MySQL memcached-Parameter

- DAEMON_MEMCACHED_R_BATCH_SIZE – Eine Zahl, die angibt, wie viele memcached-Leseoperationen durchgeführt (erhalten) werden müssen, bevor ein COMMIT ausgeführt wird, um eine neue Transaktion zu starten. Der erlaubte Wertebereich liegt zwischen 1 und 4294967295; der Standardwert ist 1. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- DAEMON_MEMCACHED_W_BATCH_SIZE – Eine Zahl, die angibt, wie viele memcached-Schreiboperationen, wie Addition, Set oder Erhöhung, durchgeführt werden müssen, bevor ein COMMIT ausgeführt wird, um eine neue Transaktion zu starten. Der erlaubte Wertebereich liegt zwischen 1 und 4294967295; der Standardwert ist 1. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- INNODB_API_BK_COMMIT_INTERVAL – Eine Zahl, die angibt, wie oft für Verbindungen im Leerlauf, die die InnoDB-memcached-Schnittstelle verwenden, ein Auto-Commit durchgeführt werden muss. Der erlaubte Wertebereich liegt zwischen 1 und 1073741824; der Standardwert ist 5. Die Option wird angewendet, ohne dass ein Neustart der Instance notwendig ist.
- INNODB_API_DISABLE_ROWLOCK – Ein boolescher Wert, der die Verwendung von Zeilensperren aktiviert (1 (wahr)) oder deaktiviert (0 (falsch)), wenn die InnoDB-memcached-Schnittstelle verwendet wird. Der Standardwert lautet 0 (falsch). Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- INNODB_API_ENABLE_MDL – Ein boolescher Wert, der, wenn auf 0 (falsch) festgelegt, die vom InnoDB-memcached-Plug-in verwendete Tabelle sperrt, damit diese durch DDL über die SQL-

Schnittstelle nicht verworfen oder geändert werden kann. Der Standardwert lautet 0 (falsch). Die Option wird nicht angewendet, bis die Instance neu gestartet wird.

- INNODB_API_TRX_LEVEL – Eine Zahl, die das Level der Transaktionsisolation für Abfragen festlegt, die von der memcached-Schnittstelle verarbeitet werden. Zulässige Werte liegen zwischen 0 und 3. Der Standardwert ist 0. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.

Amazon RDS konfiguriert diese MySQL-memcached-Parameter und sie können nicht geändert werden: DAEMON_MEMCACHED_LIB_NAME, DAEMON_MEMCACHED_LIB_PATH und INNODB_API_ENABLE_BINLOG. Die Parameter, die MySQL-Administratoren mithilfe von daemon_memcached_options festlegen, sind als einzelne MEMCACHED-Optionseinstellung in Amazon RDS verfügbar.

MySQL-daemon_memcached_options-Parameter

- BINDING_PROTOCOL – Ein String, der das zu verwendende verbindliche Protokoll angibt. Die zulässigen Werte sind auto, ascii oder binary. Der Standardwert lautet auto. Das bedeutet, dass der Server mit dem Client das Protokoll automatisch vereinbart. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- BACKLOG_QUEUE_LIMIT – Eine Zahl, die festlegt, wie viele Netzwerkverbindungen in der Warteschlange stehen dürfen, um von verarbeitet zu werden memcached. Das Erhöhen dieser Beschränkung könnte Fehler reduzieren, die ein Client erhält, wenn er nicht mit der memcached-Instance verbunden ist, aber es verbessert nicht die Leistung des Servers. Der erlaubte Wertebereich liegt zwischen 1 und 2048; der Standardwert ist 1024. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- CAS_DISABLED – Ein boolescher Wert, der die Verwendung von compare und swap (CAS) aktiviert (1 (wahr)) oder deaktiviert (0 (falsch)), welche die Pro-Element-Größe um 8 Bytes reduzieren. Der Standardwert lautet 0 (falsch). Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- CHUNK_SIZE – Eine Zahl, die das Minimum der Chunk-Größe festlegt, um die kleinsten Schlüssel, Werte und Kennzeichnungen für ein Element bereitzustellen. Die zulässigen Werte liegen zwischen 1 und 48. Der Standardwert lautet 48. Mit einem niedrigeren Wert können Sie die Effizienz des Arbeitsspeichers signifikant verbessern. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- CHUNK_SIZE_GROWTH_FACTOR – Eine Float-Zahl, die die Größe neuer Chunks steuert. Die Größe eines neuen Chunks ist die Größe des Vielfachen des vorherigen Chunks

CHUNK_SIZE_GROWTH_FACTOR. Der erlaubte Wertebereich liegt zwischen 1 und 2; der Standardwert ist 1,25. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.

- ERROR_ON_MEMORY_EXHAUSTED – Ein boolescher Wert, der, wenn auf 1 (wahr) festgelegt, angibt, dass memcached eine Fehlermeldung zurückgibt, anstatt Elemente zu entfernen, wenn kein freier Arbeitsspeicher mehr für die Elemente verfügbar ist. Wenn auf 0 (falsch) festgelegt, wird memcached Elemente exmittieren, falls kein freier Arbeitsspeicher mehr verfügbar ist. Der Standardwert lautet 0 (falsch). Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- MAX_SIMULTANEOUS_CONNECTIONS – Eine Zahl, die die maximale Anzahl an gleichzeitigen Verbindungen angibt. Wenn dieser Wert unter dem von 10 festgelegt wird, kann MySQL nicht gestartet werden. Der erlaubte Wertebereich liegt zwischen 10 und 1024; der Standardwert ist 1024. Die Option wird nicht angewendet, bis die Instance neu gestartet wird.
- VERBOSITY – Eine Zeichenfolge, die das Informationslevel des vom memcached-Service erstellten MySQL-Fehlerprotokolls angibt. Der Standardwert ist v. Die Option wird erst wirksam, wenn die Instance neu gestartet wird. Die zulässigen Werte lauten:
  - v – Protokolliert Fehler und Warnungen während der Ausführung der Hauptereignisschleife.
  - vv – Protokolliert zusätzlich zu den von v protokollierten Informationen auch jeden Client-Befehl und die Antwort.
  - vvv – Protokolliert zusätzlich zu den von vv protokollierten Informationen auch interne Zustandsübergänge.

Amazon RDS konfiguriert diese MySQL-DAEMON_MEMCACHED_OPTIONS-Parameter und sie können nicht geändert werden: DAEMON_PROCESS, LARGE_MEMORY_PAGES, MAXIMUM_CORE_FILE_LIMIT, MAX_ITEM_SIZE, LOCK_DOWN_PAGE_MEMORY, MASK, IDFILE, REQUESTS_PER_EVENT, SOCKET und USER.

Parameter für My SQL

Standardmäßig verwendet eine My SQL DB-Instance eine DB-Parametergruppe, die für eine My SQL Database spezifisch ist. Diese Parametergruppe enthält Parameter für die My SQL Database-Engine. Weitere Informationen zum Arbeiten mit Parametergruppen und zum Festlegen von Parametern finden Sie unter Parametergruppen für Amazon RDS.

RDSfür Meine SQL Parameter sind auf die Standardwerte der Speicher-Engine gesetzt, die Sie ausgewählt haben. Weitere Informationen zu Meine SQL Parameter finden Sie in der SQLDokumentation Meine. Weitere Informationen zu My SQL Storage Engines finden Sie unterUnterstützte Speicher-Engines für RDS for MySQL.

Sie können die Parameter, die für eine bestimmte RDS SQL Version von My verfügbar sind, in der RDS Konsole oder im anzeigen AWS CLI. Informationen zum Anzeigen der Parameter in der Gruppe Meine SQL Parameter in der RDS Konsole finden Sie unterParameterwerte für eine DB-Parametergruppe in Amazon RDS Amazon anzeigen.

Mithilfe von können Sie die AWS CLI Parameter für eine SQL Version von RDS for My anzeigen, indem Sie den `describe-engine-default-parameters`Befehl ausführen. Geben Sie einen der folgenden Werte für die Option `--db-parameter-group-family` an:

- `mysql8.4`
- `mysql8.0`
- `mysql5.7`

Um beispielsweise die Parameter RDS für Meine SQL Version 8.0 anzuzeigen, führen Sie den folgenden Befehl aus.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0
```

Die Ausgabe sieht folgendermaßen oder ähnlich aus.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "activate_all_roles_on_login",
                "ParameterValue": "0",
```

```
                "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": true
            },
            {
                "ParameterName": "allow-suspicious-udfs",
                "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
            {
                "ParameterName": "auto_generate_certs",
                "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
        ...
```

Führen Sie den folgenden Befehl aus, um nur die änderbaren Parameter RDS für Meine SQL Version 8.0 aufzulisten.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Allgemeine DBA Aufgaben für My SQL DB-Instances

Im folgenden Inhalt finden Sie Beschreibungen der RDS Amazon-spezifischen Implementierungen einiger allgemeiner DBA Aufgaben für DB-Instances, auf denen die My SQL Database-Engine ausgeführt wird. Um ein Managed-Serviceerlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind.

Informationen zur Arbeit mit „Meine SQL Protokolldateien“ bei Amazon RDS finden Sie unterMySQL-Datenbank-Protokolldateien.

Grundlegendes zu vordefinierten Benutzern

Amazon erstellt RDS automatisch mehrere vordefinierte Benutzer mit neuen Instances RDS für My SQL DB. Vordefinierte Benutzer und ihre Rechte können nicht geändert werden. Sie können die Rechte für diese vordefinierten Benutzer nicht löschen, umbenennen oder ändern. Jeder entsprechende Versuch führt zu einem Fehler.

- rdsadmin — Ein Benutzer, der erstellt wurde, um viele der Verwaltungsaufgaben zu erledigen, die ein Administrator mit entsprechenden `superuser` Rechten in einer eigenständigen Datenbank von My SQL database ausführen würde. Dieser Benutzer wird intern von RDS for My SQL für viele Verwaltungsaufgaben verwendet.
- rdsrepladmin — Ein Benutzer, der intern von Amazon verwendet wird, RDS um Replikationsaktivitäten RDS für My SQL DB-Instances und -Cluster zu unterstützen.

Informationen zu anderen allgemeinen DBA Aufgaben finden Sie in den folgenden Themen.

Themen

- Rollenbasiertes Privilegienmodell für for My RDS SQL
- Dynamische Privilegien RDS für for My SQL
- Beenden einer Sitzung oder Abfrage RDS für My SQL
- Überspringen des aktuellen Replikationsfehlers für RDS für My SQL
- Arbeiten mit InnoDB-Tablespaces zur Verbesserung der Wiederherstellungszeiten nach Abstürzen für My RDS SQL
- Verwaltung des globalen Statusverlaufs für RDS for My SQL
- Konfiguration der Pufferpoolgröße und der Redo-Log-Kapazität in My 8.4 SQL

Rollenbasiertes Privilegienmodell für for My RDS SQL

Ab SQL Version 8.0.36 von My können Sie die Tabellen in der Datenbank nicht direkt ändern. RDS `mysql` Insbesondere können Sie keine Datenbankbenutzer erstellen, indem Sie Operationen mit der Datenmanipulationssprache (DML) an den `grant` Tabellen ausführen. Stattdessen verwenden Sie Meine SQL Kontoverwaltungsanweisungen wie`CREATE USER`, und`GRANT`, um Benutzern rollenbasierte Berechtigungen `REVOKE` zu gewähren. Sie können auch keine anderen Objekte wie gespeicherte Prozeduren in der `mysql`-Datenbank erstellen. Sie können immer noch die `mysql`-Tabellen abfragen. Wenn Sie die binäre Protokollreplikation verwenden, werden Änderungen, die direkt an den `mysql` Tabellen auf der Quell-DB-Instance vorgenommen wurden, nicht auf den Zielcluster repliziert.

In einigen Fällen verwendet Ihre Anwendung möglicherweise Verknüpfungen, um Benutzer oder andere Objekte zu erstellen, indem Sie sie in die`mysql`-Tabellen Wenn ja, ändern Sie Ihren Anwendungscode, um die entsprechenden Anweisungen wie `CREATE USER` zu verwenden.

Verwenden Sie eine der folgenden Methoden, um Metadaten für Datenbankbenutzer während der Migration aus einer externen SQL Datenbank von My Database zu exportieren:

- Verwenden Sie das Instance-Dump-Hilfsprogramm von My SQL Shell mit einem Filter, um Benutzer, Rollen und Zuschüsse auszuschließen. Das folgende Beispiel zeigt Ihnen die zu verwendende Befehlssyntax. Stellen Sie sicher, dass `outputUrl` das leer ist.

  ```
  mysqlsh user@host -- util.dumpInstance(outputUrl,{excludeSchemas:['mysql'],users:
   true})
  ```

  Weitere Informationen finden Sie unter Instance Dump Utility, Schema Dump Utility und Table Dump Utility im My SQL Reference Manual.

- Verwenden Sie das Client-Hilfsprogramm`mysqlpump`. Dieses Beispiel umfasst alle Tabellen mit Ausnahme der Tabellen in der `mysql` Systemdatenbank. Es enthält `CREATE USER` auch `GRANT` Anweisungen zur Reproduktion aller SQL Benutzer vom Typ My in der migrierten Datenbank.

  ```
  mysqlpump --exclude-databases=mysql --users
  ```

  Das `mysqlpump` Client-Hilfsprogramm ist mit My SQL 8.4 nicht mehr verfügbar. Nutzen Sie stattdessen `mysqldump`.

Um die Verwaltung von Berechtigungen für viele Benutzer oder Anwendungen zu vereinfachen, können SieCREATE ROLE-Anweisung zum Erstellen einer Rolle mit einer Reihe von Berechtigungen. Dann können Sie dieGRANTundSET ROLE-Anweisungen und diecurrent_roleFunktion, um Benutzern oder Anwendungen Rollen zuzuweisen, die aktuelle Rolle zu wechseln und zu überprüfen, welche Rollen in Kraft sind. Weitere Informationen zum rollenbasierten Berechtigungssystem in My SQL 8.0 finden Sie unter Using Roles im My SQL Reference Manual.

> **⚠ Important**
>
> Wir empfehlen Ihnen, den Hauptbenutzer nicht direkt in Ihren Anwendungen zu verwenden. Bleiben Sie stattdessen bei der bewährten Methode, einen Datenbankbenutzer zu verwenden, der mit den Mindestberechtigungen erstellt wurde, die für Ihre Anwendung erforderlich sind.

Ab Version 8.0.36 SQL enthält RDS for My eine spezielle Rolle, die über alle der folgenden Rechte verfügt. Der Name der Rolle lautet `rds_superuser_role`. Dem primären Administratorbenutzer für jede DB-Instance wurde diese Rolle bereits zugewiesen. Die`rds_superuser_role`enthält die folgenden Berechtigungen für alle Datenbankobjekte:

- `ALTER`
- `APPLICATION_PASSWORD_ADMIN`
- `ALTER ROUTINE`
- `CREATE`
- `CREATE ROLE`
- `CREATE ROUTINE`
- `CREATE TEMPORARY TABLES`
- `CREATE USER`
- `CREATE VIEW`
- `DELETE`
- `DROP`
- `DROP ROLE`
- `EVENT`
- `EXECUTE`

- INDEX
- INSERT
- LOCK TABLES
- PROCESS
- REFERENCES
- RELOAD
- REPLICATION CLIENT
- REPLICATION SLAVE
- ROLE_ADMIN
- SET_USER_ID
- SELECT
- SHOW DATABASES
- SHOW VIEW
- TRIGGER
- UPDATE
- XA_RECOVER_ADMIN

Die Rollendefinition umfasst auchWITH GRANT OPTIONdamit ein Administratorbenutzer diese Rolle anderen Benutzern gewähren kann. Insbesondere muss der Administrator alle Rechte gewähren, die für die Durchführung der binären Protokollreplikation mit My SQL Cluster als Ziel erforderlich sind.

> **Tip**
>
> Verwenden Sie die folgende Anweisung, um die vollständigen Details der Berechtigungen einzusehen.
>
> ```
> SHOW GRANTS FOR rds_superuser_role@'%';
> ```

Wenn Sie Zugriff mithilfe von Rollen in RDS My SQL Version 8.0.36 und höher gewähren, aktivieren Sie die Rolle auch mithilfe der Anweisung SET ROLE *role_name* orSET ROLE ALL. Im folgenden Beispiel wird gezeigt, wie dies geschieht. Ersetzen Sie den entsprechenden Rollennamen für CUSTOM_ROLE.

```
# Grant role to user
mysql> GRANT CUSTOM_ROLE TO 'user'@'domain-or-ip-address'

# Check the current roles for your user. In this case, the CUSTOM_ROLE role has not
 been activated.
# Only the rds_superuser_role is currently in effect.
mysql> SELECT CURRENT_ROLE();
+-------------------------+
| CURRENT_ROLE()          |
+-------------------------+
| `rds_superuser_role`@`%` |
+-------------------------+
1 row in set (0.00 sec)

# Activate all roles associated with this user using SET ROLE.
# You can activate specific roles or all roles.
# In this case, the user only has 2 roles, so we specify ALL.
mysql> SET ROLE ALL;
Query OK, 0 rows affected (0.00 sec)

# Verify role is now active
mysql> SELECT CURRENT_ROLE();
+-------------------------------------------------+
| CURRENT_ROLE()                                  |
+-------------------------------------------------+
| `CUSTOM_ROLE`@`%`,`rds_superuser_role`@`%` |
+-------------------------------------------------+
```

Dynamische Privilegien RDS für for My SQL

Dynamische Rechte sind Meine SQL Rechte, die Sie mithilfe der GRANT Anweisung explizit gewähren können. Je nach Ihrer Version von RDS for My SQL RDS können Sie damit nur bestimmte dynamische Rechte gewähren. RDSverbietet einige dieser Rechte, da sie bestimmte Datenbankvorgänge wie Replikation und Sicherung beeinträchtigen können.

Die folgende Tabelle zeigt, welche dieser Rechte Sie für verschiedene SQL Versionen von My gewähren können. Wenn Sie ein Upgrade von einer SQL Version unter Meine Version als 8.0.36 auf Version 8.0.36 oder höher durchführen, müssen Sie möglicherweise Ihren Anwendungscode aktualisieren, wenn die Gewährung einer bestimmten Berechtigung nicht mehr zulässig ist.

Recht	Meine Version 8.0.35 und niedriger SQL	Meine Nebenversionen SQL 8.0.36 und höher	Mein SQL 8.4.3 und höher
ALLOW_NONEXISTENT_DEFINER	Nicht verfügbar	Nicht verfügbar	Nicht erlaubt
APPLICATION_PASSWORD_ADMIN	Zulässig	Zulässig	Zulässig
AUDIT_ABORT_EXEMPT	Zulässig	Nicht erlaubt	Nicht erlaubt
AUDIT_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
AUTHENTICATION_POLICY_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
BACKUP_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
BINLOG_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
BINLOG_ENCRYPTION_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
CLONE_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
CONNECTION_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
ENCRYPTION_KEY_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
FIREWALL_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
FIREWALL_EXEMPT	Zulässig	Nicht erlaubt	Nicht erlaubt
FIREWALL_USER	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt

Recht	Meine Version 8.0.35 und niedriger SQL	Meine Nebenversionen SQL 8.0.36 und höher	Mein SQL 8.4.3 und höher
FLUSH_OPTIMIZER_COSTS	Zulässig	Zulässig	Zulässig
FLUSH_PRIVILEGES	Nicht verfügbar	Nicht verfügbar	Zulässig
FLUSH_STATUS	Zulässig	Zulässig	Zulässig
FLUSH_TABLES	Zulässig	Zulässig	Zulässig
FLUSH_USER_RESOURCES	Zulässig	Zulässig	Zulässig
GROUP_REPLICATION_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
GROUP_REPLICATION_STREAM	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
INNODB_REDO_LOG_ARCHIVE	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
INNODB_REDO_LOG_ENABLE	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
MASKING_DICTIONARIES_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
NDB_STORED_USER	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
OPTIMIZE_LOCAL_TABLE	Nicht verfügbar	Nicht verfügbar	Nicht erlaubt
PASSWORDLESS_USER_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt

Recht	Meine Version 8.0.35 und niedriger SQL	Meine Nebenversionen SQL 8.0.36 und höher	Mein SQL 8.4.3 und höher
PERSIST_RO_ _VARIABLES ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
REPLICATION_APPLIER	Zulässig	Nicht erlaubt	Nicht erlaubt
REPLICATION_SLAVE_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
RESOURCE_GROUP_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
RESOURCE_GROUP_USER	Zulässig	Nicht erlaubt	Nicht erlaubt
ROLE_ADMIN	Zulässig	Zulässig	Zulässig
SENSITIVE_VARIABLES_OBSERVER	Zulässig	Zulässig	Zulässig
SERVICE_CONNECTION_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
SESSION_VARIABLES_ADMIN	Zulässig	Zulässig	Zulässig
SET_ANY_DEFINER	Nicht verfügbar	Nicht verfügbar	Zulässig
SET_ _ID USER	Zulässig	Zulässig	Nicht verfügbar
SHOW_ROUTINE	Zulässig	Zulässig	Zulässig
SKIP_QUERY_REWRITE	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt

Recht	Meine Version 8.0.35 und niedriger SQL	Meine Nebenversionen SQL 8.0.36 und höher	Mein SQL 8.4.3 und höher
SYSTEM_USER	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
SYSTEM_VARIABLES_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
TABLE_ENCRYPTION_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
TELEMETRY_LOG_ADMIN	Zulässig	Nicht erlaubt	Nicht erlaubt
TP_ _ CONNECTION ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
TRANSACTION_GTID_TAG	Nicht verfügbar	Nicht verfügbar	Nicht erlaubt
VERSION_TOKEN_ADMIN	Nicht erlaubt	Nicht erlaubt	Nicht erlaubt
XA_ _ RECOVER ADMIN	Zulässig	Zulässig	Zulässig

Beenden einer Sitzung oder Abfrage RDS für My SQL

Sie können Benutzersitzungen oder Abfragen in DB-Instances mit den Befehlen `rds_kill` und `rds_kill_query` beenden. Stellen Sie zuerst eine Verbindung zu Ihrer My SQL DB-Instance her und geben Sie dann den entsprechenden Befehl ein, wie im Folgenden gezeigt. Weitere Informationen finden Sie unter Verbindung zu Ihrer MySQL-DB-Instance herstellen.

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

Um beispielsweise die Sitzung zu beenden, die auf Thread 99 ausgeführt wird, würden Sie Folgendes eingeben:

```
CALL mysql.rds_kill(99);
```

Um die Abfrage zu beenden, die auf Thread 99 ausgeführt wird, würden Sie Folgendes eingeben:

```
CALL mysql.rds_kill_query(99);
```

Überspringen des aktuellen Replikationsfehlers für RDS für My SQL

Amazon RDS stellt einen Mechanismus bereit, mit dem Sie einen Fehler für Ihre Lesereplikate überspringen können, wenn der Fehler dazu führt, dass Ihr Lesereplikat aufhört zu reagieren und der Fehler keine Auswirkungen auf die Integrität Ihrer Daten hat.

> **Note**
>
> Sie sollten zunächst überprüfen, ob der Fehler sicher übersprungen werden kann. Stellen Sie in einem My SQL Utility eine Verbindung zur Read Replica her und führen Sie den folgenden My SQL -Befehl aus.
>
> ```
> SHOW REPLICA STATUS\G
> ```
>
> Informationen zu den zurückgegebenen Werten finden Sie in der SQL Dokumentation Meine. Frühere Versionen von und My SQL wurden `SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie eine SQL Version von Meine vor 8.0.23 verwenden, verwenden Sie. `SHOW SLAVE STATUS`

Sie können einen Fehler in Ihrem Lesereplikat wie folgt überspringen.

Themen

- Aufrufen der Prozedur mysql.rds_skip_repl_error
- Festlegen des Parameters slave_skip_error

Aufrufen der Prozedur mysql.rds_skip_repl_error

Amazon RDS bietet eine gespeicherte Prozedur, die Sie aufrufen können, um einen Fehler bei Ihren Read Replicas zu überspringen. Stellen Sie zunächst eine Verbindung mit Ihrer MySQL-DB-Instance her und geben Sie anschließend den entsprechenden Befehl aus, wie im Folgenden gezeigt. Weitere Informationen finden Sie unter Verbindung zu Ihrer MySQL-DB-Instance herstellen.

Um den Fehler zu überspringen, können Sie den folgenden Befehl ausgeben:

```
CALL mysql.rds_skip_repl_error;
```

Dieser Befehl hat keine Auswirkungen, wenn Sie ihn auf der Quell-DB-Instance oder einem Lesereplikat ausführen, für den kein Replikationsfehler aufgetreten ist.

Weitere Informationen, z. B. zu den Versionen von My, die SQL diese unterstützen`mysql.rds_skip_repl_error`, finden Sie untermysql.rds_skip_repl_error.

> **Important**
>
> Wenn Sie versuchen, aufzurufen `mysql.rds_skip_repl_error` und der folgende Fehler auftritt:`ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist`, aktualisieren Sie Ihre My SQL DB-Instance auf die neueste Nebenversion oder eine der unter aufgeführten Mindestnebenversionenmysql.rds_skip_repl_error.

Festlegen des Parameters slave_skip_error

Um einen oder mehrere Fehler zu überspringen, können Sie die `slave_skip_errors` statischer Parameter für das Lesereplikat setzen. Sie können diesen Parameter so einstellen, dass ein oder mehrere spezifische Replikationsfehlercodes übersprungen werden. Derzeit können Sie diesen Parameter nur RDS für My SQL 5.7 DB-Instances festlegen. Nachdem Sie die Einstellung für diesen Parameter geändert haben, starten Sie Ihre DB-Instance neu, damit die neue Einstellung wirksam wird. Informationen zum Einstellen dieses Parameters finden Sie in der SQLDokumentation Meine.

Wir empfehlen, diesen Parameter in einer separaten DB-Parametergruppe einzustellen. Sie können diese DB-Parametergruppe nur den Lese-Replikaten zuordnen, die Fehler überspringen müssen. Nach dieser Best Practice werden die potenziellen Auswirkungen auf andere DB-Instances und Lese-Replikate reduziert.

> **Important**
>
> Das Festlegen eines nicht standardmäßigen Werts für diesen Parameter kann zu Inkonsistenz der Replikation führen. Stellen Sie diesen Parameter nur auf einen nicht standardmäßigen Wert ein, wenn Sie andere Optionen zur Behebung des Problems ausgeschöpft haben und Sie sich der möglichen Auswirkungen auf die Daten Ihres Read Replica sicher sind.

Arbeiten mit InnoDB-Tablespaces zur Verbesserung der Wiederherstellungszeiten nach Abstürzen für My RDS SQL

Jede Tabelle in My SQL besteht aus einer Tabellendefinition, Daten und Indizes. Die My SQL Storage Engine InnoDB speichert Tabellendaten und Indizes in einem Tablespace. InnoDB erstellt einen globalen, freigegebenen Tabellenraum, der ein Datenverzeichnis und andere relevante Metadaten enthält, und kann Tabellendaten und Indizes enthalten. InnoDB kann darüber hinaus für jede Tabelle und Partition eigene Tabellenräume erstellen. Diese getrennten Tabellenräume werden in Dateien mit der Erweiterung .ibd gespeichert. Die Kopfzeile der einzelnen Tabellenräume enthalten eine Zahl, welche diese eindeutig identifiziert.

Amazon RDS stellt einen Parameter in der Gruppe Meine SQL Parameter mit dem Namen bereit`innodb_file_per_table`. Dieser Parameter steuert, ob InnoDB neue Tabellendaten und Indizes in den gemeinsamen Tabellenraum (durch Setzen des Parameterwertes auf 0) oder in einzelne Tabellenräume (durch Setzen des Parameterwertes auf 1) einfügt. Amazon RDS setzt den Standardwert für den `innodb_file_per_table` Parameter auf 1, sodass Sie einzelne InnoDB-Tabellen löschen und den von diesen Tabellen verwendeten Speicher für die DB-Instance zurückgewinnen können. In der Mehrzahl der Anwendungsfälle stellt die Festlegung des Parameters `innodb_file_per_table` auf 1 die empfohlene Einstellung dar.

Sie sollten den `innodb_file_per_table` Parameter auf 0 setzen, wenn Sie über eine große Anzahl von Tabellen verfügen, z. B. über 1000 Tabellen, wenn Sie Standardspeicher (magnetisch) oder SSD Allzweckspeicher verwenden, oder auf über 10.000 Tabellen, wenn Sie bereitgestellten IOPS Speicher verwenden. Wenn Sie diesen Parameter auf 0 festlegen, werden keine einzelnen Tabellenräume erstellt. Dies kann die Zeit für die Datenbankwiederherstellung nach einem Absturz verkürzen.

Meine SQL verarbeitet jede Metadatendatei, einschließlich Tablespaces, während des Wiederherstellungszyklus nach einem Absturz. Die Zeit, die My benötigtSQL, um die

Metadateninformationen im gemeinsam genutzten Tablespace zu verarbeiten, ist vernachlässigbar gering im Vergleich zu der Zeit, die für die Verarbeitung von Tausenden von Tablespace-Dateien benötigt wird, wenn es mehrere Tablespaces gibt. Da die Tabellenraumnummer in den Kopfzeilen der einzelnen Dateien gespeichert wird, kann die Gesamtzeit für das Lesen aller Tabellenraumdateien mehrere Stunden betragen. Beispielsweise kann die Verarbeitung von einer Million InnoDB-Tabellenräumen in einem Standardspeicher während eines Wiederherstellungszyklus nach einem Absturz zwischen fünf und acht Stunden betragen. In einigen Fällen kann InnoDB feststellen, dass im Anschluss an einen Wiederherstellungszyklus nach einem Absturz eine zusätzliche Bereinigung erforderlich ist. Dann wird ein weiterer Absturzwiederherstellungszyklus gestartet, was die Wiederherstellungszeit verlängert. Denken Sie daran, dass ein Absturzwiederherstellungszyklus zusätzlich zur Verarbeitung von Tabellenrauminformationen auch Rollbacks von Transaktionen, die Reparatur beschädigter Seiten und andere Operationen umfasst.

Da sich der Parameter `innodb_file_per_table` in einer Parametergruppe befindet, können Sie den Parameterwert ändern, indem Sie die von Ihrer DB-Instance verwendete Parametergruppe bearbeiten, anstatt die DB-Instance neu starten zu müssen. Nach dem Ändern der Einstellung, beispielsweise von 1 (Erstellen einzelner Tabellen) in 0 (Verwenden freigegebener Tabellenräume) werden dem freigegebenen Tabellenraum neue InnoDB-Tabellen hinzugefügt, während die vorhandenen Tabellen weiterhin über einzelne Tabellenräume verfügen. Um eine InnoDB-Tabelle zum freigegebenen Tabellenraum zu verschieben, müssen Sie den Befehl `ALTER TABLE` verwenden.

Migrieren mehrerer Tabellenräume zum freigegebenen Tabellenraum

Sie können die Metadaten einer InnoDB-Tabelle vom eigenen Tabellenraum zum freigegebenen Tabellenraum verschieben. Hierdurch werden die Tabellenmetadaten entsprechend der Parametereinstellung `innodb_file_per_table` neu erstellt. Stellen Sie zuerst eine Verbindung zu Ihrer My SQL DB-Instance her und geben Sie dann die entsprechenden Befehle ein, wie im Folgenden dargestellt. Weitere Informationen finden Sie unter Verbindung zu Ihrer MySQL-DB-Instance herstellen.

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

Zum Beispiel gibt die folgende Abfrage für jede nicht im freigegebenen Tabellenraum enthaltene InnoDB-Tabelle eine `ALTER TABLE`-Anweisung zurück.

Für My SQL 5.7 DB-Instances:

```
SELECT CONCAT('ALTER TABLE `',
```

```
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '``', '```'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '``', '```'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Für meine SQL 8.4- und 8.0-DB-Instances:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '``', '```'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '``', '```'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Beim Neuaufbau einer SQL My-Tabelle, um die Metadaten der Tabelle in den gemeinsam genutzten Tablespace zu verschieben, ist vorübergehend zusätzlicher Speicherplatz erforderlich, um die Tabelle neu zu erstellen. Daher muss für die DB-Instance Speicherplatz verfügbar sein. Während der Neuerstellung ist die Tabelle gesperrt und für Abfragen nicht verfügbar. Im Fall kleiner Tabellen oder von Tabellen, auf die nicht häufig zugegriffen wird, stellt dies möglicherweise kein Problem dar. Im Fall großer Tabellen oder von Tabellen, auf die in einer stark gleichzeitigen Umgebung häufig zugegriffen wird, können Sie Tabellen in einem Lesereplikat neu erstellen.

Sie können ein Lesereplikat erstellen und Tabellenmetadaten zum freigegebenen Tabellenraum in dem Lesereplikat verschieben. Die ALTER TABLE Anweisung blockiert zwar den Zugriff auf die Read Replica, die Quell-DB-Instance ist davon jedoch nicht betroffen. Die Quell-DB-Instance generiert weiterhin Binärprotokolle, während das Lesereplikat während der Neuerstellung der Tabelle folgt. Da für die Neuerstellung zusätzlicher Speicherplatz benötigt wird und die Wiedergabeprotokolldatei sehr groß werden kann, sollten Sie ein Lesereplikat erstellen, dessen Speicher größer als der der Quell-DB-Instance ist.

Führen Sie die folgenden Schritte aus, um ein Lesereplikat zu erstellen und InnoDB-Tabellen, die den freigegebenen Tabellenraum verwenden sollen, neu zu erstellen:

1. Stellen Sie sicher, dass die Sicherungsbeibehaltung in der Quell-DB-Instance aktiviert ist, damit die Binärprotokollierung aktiviert ist.
2. Verwenden Sie AWS Management Console oder AWS CLI , um eine Read Replica für die Quell-DB-Instance zu erstellen. Da die Erstellung eines Lesereplikats viele derselben Verfahren umfasst wie eine Wiederherstellung nach einem Absturz, kann der Erstellungsvorgang eine Weile dauern,

wenn es eine große Zahl von InnoDB-Tabellenräumen gibt. Teilen Sie dem Lesereplikat mehr Speicherplatz zu, als zurzeit in der Quell-DB-Instance verwendet wird.

3. Wenn das Lesereplikat erstellt wurde, erstellen Sie eine Parametergruppe mit den Parametereinstellungen `read_only = 0` und `innodb_file_per_table = 0`. Ordnen Sie dann die Parametergruppe dem Lesereplikat zu.
4. Geben Sie die folgende SQL Anweisung für alle Tabellen aus, die Sie auf das Replikat migrieren möchten:

```
ALTER TABLE name ENGINE = InnoDB
```

5. Wenn alle `ALTER TABLE`-Anweisungen für das Lesereplikat ausgeführt wurden, überprüfen Sie, ob das Lesereplikat mit der Quell-DB-Instance verknüpft ist und die beiden Instances synchronisiert sind.
6. Verwenden Sie die Konsole oder stufen CLI Sie die Read Replica zur Instanz herauf. Achten Sie darauf, dass für die Parametergruppe, die für die neue eigenständige DB-Instance verwendet wird, der Parameter `innodb_file_per_table` auf 0 festgelegt ist. Ändern Sie den Namen der neuen eigenständigen DB-Instance, und verweisen Sie alle Anwendungen auf die neue eigenständige DB-Instance.

Verwaltung des globalen Statusverlaufs für RDS for My SQL

> **Tip**
>
> Um die Datenbankleistung zu analysieren, können Sie auch Performance Insights auf Amazon verwendenRDS. Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.

My SQL verwaltet viele Statusvariablen, die Informationen über den Betrieb liefern. Ihre Werte können Ihnen helfen, Probleme mit Sperren oder Arbeitsspeichern in einer DB-Instance zu entdecken. Die Werte dieser Statusvariablen sind seit dem letzten Zeitpunkt, an dem die DB-Instance gestartet wurde, kumulativ. Sie können die meisten Statusvariablen auf 0 zurücksetzen, indem Sie den Befehl `FLUSH STATUS` verwenden.

Um die Überwachung dieser Werte im Zeitverlauf zu ermöglichen, RDS bietet Amazon eine Reihe von Verfahren, mit denen die Werte dieser Statusvariablen im Zeitverlauf erfasst und zusammen mit allen Änderungen seit dem letzten Snapshot in eine Tabelle geschrieben werden. Diese Infrastruktur,

Global Status History (GoSH) genannt, ist auf allen My SQL DB-Instances ab Version 5.5.23 installiert. GoSH ist standardmäßig deaktiviert.

Um GoSH zu aktivieren, müssen Sie zunächst den Ereignis-Scheduler aus einer DB-Parametergruppe aktivieren, indem Sie den Parameter `event_scheduler` auf `ON` festlegen. Stellen Sie für My SQL DB-Instances, auf denen My SQL 5.7 ausgeführt wird, den Parameter ebenfalls auf ein. `show_compatibility_56` 1 Weitere Informationen zum Erstellen und Ändern einer DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS. Informationen zu den Nebenwirkungen der Aktivierung dieses Parameters finden Sie unter show_compatibility_56 im My 5.7-Referenzhandbuch. SQL

Sie können anschließend die Verfahren in der folgenden Tabelle verwenden, um GoSH zu aktivieren und zu konfigurieren. Stellen Sie zuerst eine Verbindung zu Ihrer My SQL DB-Instance her und geben Sie dann die entsprechenden Befehle ein, wie im Folgenden dargestellt. Weitere Informationen finden Sie unter Verbindung zu Ihrer MySQL-DB-Instance herstellen. Führen Sie für jede Prozedur den folgenden Befehl aus und ersetzen Sie ihn *`procedure-name`*:

```
CALL procedure-name;
```

In der folgenden Tabelle sind alle Prozeduren aufgeführt, die Sie *`procedure-name`*im vorherigen Befehl verwenden können.

Verfahren	Beschreibung
`mysql.rds_enable_gsh_collector`	Aktiviert GoSH, um Standard-Snapshots in zeitlichen Abständen zu erstellen, die durch angegeben werde `rds_set_gsh_collector` .
`mysql.rds_set_gsh_collector`	Gibt den zeitlichen Abstand in Minuten für die periodische Generierung von Snapshots an. Der Standardwert ist 5.
`mysql.rds_disable_gsh_collector`	Deaktiviert Snapshots.
`mysql.rds_collect_global_status_history`	Generiert einen Snapshot auf Anforderung.
`mysql.rds_enable_gsh_rotation`	Aktiviert die Rotation der Inhalte der Tabelle `mysql.rds_global_status_history` zu `mysql.rds`

Verfahren	Beschreibung
	`_global_status_history_old` in zeitlichen Abständen, die durch `rds_set_gsh_rotation` angegeben werden.
`mysql.rds_set_gsh_rotation`	Gibt den zeitlichen Abstand in Tagen für die periodische Tabellenrotation an. Der Standardwert ist 7.
`mysql.rds_disable_gsh_rotation`	Deaktiviert die Tabellenrotation.
`mysql.rds_rotate_global_status_history`	Rotiert die Inhalte der Tabelle `mysql.rds_global_status_history` bei Anforderung zu `mysql.rds_global_status_history_old` .

Wenn GoSH ausgeführt wird, können Sie Abfragen für die Tabellen ausführen, in die GoSH schreibt. Um beispielsweise eine Abfrage für das Trefferverhältnis des InnoDB-Pufferpools auszuführen, würden Sie die folgende Abfrage ausgeben:

```
select a.collection_end, a.collection_start, (( a.variable_Delta-b.variable_delta)/
a.variable_delta)*100 as "HitRatio"
    from mysql.rds_global_status_history as a join mysql.rds_global_status_history as b
 on a.collection_end = b.collection_end
    where a. variable_name = 'Innodb_buffer_pool_read_requests' and b.variable_name =
 'Innodb_buffer_pool_reads'
```

Konfiguration der Pufferpoolgröße und der Redo-Log-Kapazität in My 8.4 SQL

In My SQL 8.4 RDS aktiviert Amazon den `innodb_dedicated_server` Parameter standardmäßig. Mit dem `innodb_dedicated_server` Parameter berechnet die Datenbank-Engine die `innodb_redo_log_capacity` Parameter `innodb_buffer_pool_size` und. Informationen zur Berechnung dieser Parameter finden Sie unter Configuring InnoDB Buffer Pool Size and Redo Log in der Dokumentation MySQL.

`innodb_dedicated_server`Wenn diese Option aktiviert ist, wird der `innodb_buffer_pool_size` Parameter auf der Grundlage des Speichers der DB-Instance-

Klasse berechnet. Die folgende Tabelle zeigt den erkannten Serverspeicher und die entsprechende Pufferpoolgröße.

Serverspeicher wurde erkannt	Größe des Pufferpools
< 1 GB	Der Standardwert ist 128 MB
1 GB bis 4 GB	*Detected server memory** 0,5
> 4 GB	*Detected server memory** 0,75

Der `innodb_redo_log_capacity` Parameter skaliert automatisch mit der Instanzklasse auf (Anzahl von vCPUs /2) GB bis zu einem Maximum von 16 GB. Größere Instance-Klassen verfügen über eine größere Redo-Log-Kapazität, wodurch die Leistung und Stabilität bei schreibintensiven Workloads verbessert werden kann.

Stellen Sie vor dem Upgrade von My SQL 8.0 auf My SQL 8.4 sicher, dass Sie Ihren Speicherplatz vergrößern, um einer möglichen Erhöhung der Größe der Redo-Logs Rechnung zu tragen, die nach Abschluss des Upgrades auftreten kann. Weitere Informationen finden Sie unter Steigern der DB-Instance-Speicherkapazität.

Wenn Sie nicht möchten, dass der `innodb_dedicated_server` Parameter die Werte für die `innodb_redo_log_capacity` Parameter `innodb_buffer_pool_size` und berechnet, können Sie diese Werte überschreiben, indem Sie spezifische Werte für sie in einer benutzerdefinierten Parametergruppe festlegen. Alternativ können Sie den `innodb_dedicated_server` Parameter deaktivieren und Werte für die `innodb_redo_log_capacity` Parameter `innodb_buffer_pool_size` und in einer benutzerdefinierten Parametergruppe festlegen. Weitere Informationen finden Sie unter Standard- und benutzerdefinierte Parametergruppen.

Wenn Sie den `innodb_dedicated_server` Parameter deaktivieren, indem Sie ihn auf setzen, `0` und keine Werte für die `innodb_redo_log_capacity` Parameter `innodb_buffer_pool_size` und festlegen, RDS setzt Amazon die beiden letztgenannten Parameter auf 128 MB bzw. 100 MB. Diese Standardwerte führen zu einer schlechten Leistung bei größeren Instance-Klassen.

Lokale Zeitzone für My SQL DB-Instances

Standardmäßig ist die Zeitzone für eine My SQL DB-Instance Universal Time Coordinated (UTC). Sie können die Zeitzone für Ihre DB-Instance auf die lokale Zeitzone für Ihre Anwendung einstellen.

Setzen Sie den Parameter `time_zone` in der Parametergruppe für Ihre DB-Instance auf einen der unterstützten Werte, die weiter unten in diesem Abschnitt gelistet sind. Wenn Sie den Parameter `time_zone` für eine Parametergruppe setzen, wird bei allen DB-Instances und Lesereplikaten, die diese Parametergruppe verwenden, die neue lokale Zeitzone eingestellt. Weitere Informationen zum Einstellen von Parametern in einer Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

Nachdem Sie die lokale Zeitzone eingestellt haben, werden alle neuen Verbindungen zur Datenbank die Änderung reflektieren. Wenn Sie keine offenen Verbindungen zu Ihrer Datenbank haben, wenn Sie die lokale Zeitzone ändern, sehen Sie die lokale Zeitzonenaktualisierung nicht, nachdem Sie die Verbindung schließen und eine neue Verbindung öffnen.

Sie können eine andere lokale Zeitzone für eine DB-Instance sowie ein oder mehrere ihrer Lesereplikate einstellen. Verwenden Sie eine andere Parametergruppe für die DB-Instance und das/die Replica/s und stellen Sie den Parameter `time_zone` in jeder Parametergruppe auf eine andere lokale Zeit ein.

Wenn Sie eine Across-Replikation durchführen AWS-Regionen, verwenden die Quell-DB-Instance und die Read Replica unterschiedliche Parametergruppen (Parametergruppen gelten nur für eine AWS-Region). Sie müssen den Parameter `time_zone` in der Parametergruppe der Instance und des Lesereplikats einstellen, um dieselbe lokale Zeitzone für jede Instance zu verwenden.

Wenn Sie eine DB-Instance aus einem DB-Snapshot wiederherstellen, ist die lokale Zeitzone auf eingestellt. UTC Sie können die Zeitzone auf Ihre lokale Zeitzone einstellen, nachdem die Wiederherstellung abgeschlossen ist. Wenn Sie die DB-Instance auf einen Zeitpunkt wiederherstellen, ist die lokale Zeitzone für die wiederhergestellte DB-Instance die Zeitzoneneinstellung von der Parametergruppe für die wiederhergestellte DB-Instance.

Die Internet Assigned Numbers Authority (IANA) veröffentlicht mehrmals im Jahr neue Zeitzonen unter https://www.iana.org/time-zones. Jedes Mal, wenn eine neue, kleinere Wartungsversion von My RDS veröffentlicht wirdSQL, wird sie mit den aktuellen Zeitzonendaten zum Zeitpunkt der Veröffentlichung ausgeliefert. Wenn Sie die neueste Version RDS für Meine SQL Versionen verwenden, verfügen Sie über aktuelle Zeitzonendaten vonRDS. Wenn Sie sichergehen möchten, dass Ihre DB-Instance über aktuelle Zeitzonendaten verfügt, empfehlen wir ein Upgrade auf eine

höhere DB-Engine-Version. Alternativ können Sie die Zeitzonentabellen in MariaDB-DB-Instances manuell ändern. Dazu können Sie SQL Befehle verwenden oder das Tool mysql_tzinfo_to_sql in einem Client ausführen. SQL Nachdem Sie die Zeitzonendaten manuell aktualisiert haben, starten Sie Ihre DB-Instance neu, damit die Änderungen wirksam werden. RDSändert oder setzt die Zeitzonendaten laufender DB-Instances nicht zurück. Neue Zeitzonendaten werden nur installiert, wenn Sie ein Upgrade der Datenbank-Engine-Version durchführen.

Sie können Ihre lokale Zeitzone auf die folgenden Werte einstellen.

Bereich	Zeitzone
Afrika	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
Amerika	America/Araguaina, America/Asuncion, America/Bogota, America/Buenos_Aires, America/Caracas, America/Chihuahua, America/Cuiaba, America/Denver, America/Fortaleza, America/Guatemala, America/Halifax, America/Manaus, America/Matamoros, America/Monterrey, America/Montevideo, America/Phoenix, America/Santiago, America/Tijuana
Asien	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damascus, Asia/Dhaka, Asia/Irkutsk, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Krasnoyarsk, Asia/Magadan, Asia/Muscat, Asia/Novosibirsk, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantik/Azoren
Australien	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Hobart, Australia/Perth, Australia/Sydney
Brasilien	Brazil/DeNoronha, Brazil/East
Kanada	Canada/Newfoundland, Canada/Saskatchewan, Canda/Yukon
Europa	Europe/Amsterdam, Europe/Athens, Europe/Dublin, Europe/Helsinki, Europe/Istanbul, Europe/Kaliningrad Europe/Moscow, Europe/Paris, Europe/Prague, Europe/Sarajevo

Bereich	Zeitzone
Pazifik	Pacific/Auckland, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Samoa
US	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific
UTC	UTC

Bekannte Probleme und Einschränkungen für Amazon RDS for MySQL

Bekannte Probleme und Einschränkungen bei der Arbeit Amazon RDS for MySQL sind wie folgt.

Themen

- Reserviertes Wort InnoDB
- Vollständiges Storage-Verhalten für Amazon RDS for MySQL
- Inkonsistente Größe des InnoDB-Buffer-Pools
- Index-Merge-Optimierung zeigt falsche Ergebnisse an
- MySQL-Parameterausnahmen für Amazon RDS-DB-Instances
- MySQL-Dateigrößenlimits in Amazon RDS
- MySQL Keyring-Plugin wird nicht unterstützt
- Benutzerdefinierte Ports
- Einschränkungen bei gespeicherten MySQL-Prozeduren
- GTID-basierte Replikation mit einer externen Quell-Instance
- MySQL-Standardauthentifizierungs-Plugin
- Überschreiben von innodb_buffer_pool_size
- Upgrade von MySQL 5.7 auf MySQL 8.4
- InnoDB-Seitenkomprimierung

Reserviertes Wort InnoDB

`InnoDB` ist ein reserviertes Wort für RDS for MySQL. Sie können diesen Namen für eine MySQL-Datenbank nicht verwenden.

Vollständiges Storage-Verhalten für Amazon RDS for MySQL

Wenn der Speicher für eine MySQL-DB-Instance voll ist, kann es zu Inkonsistenzen bei Metadaten, Diskatorkonsistenzen und verwaisten Tabellen kommen. Um diese Probleme zu vermeiden, stoppt Amazon RDS automatisch eine DB-Instance, die den `storage-full` Status erreicht.

Eine MySQL-DB-Instance erreicht den `storage-full` Status in den folgenden Fällen:

- Die DB-Instance verfügt über weniger als 20.000 MiB Speicher, und der verfügbare Speicher erreicht 200 MiB oder weniger.
- Die DB-Instance verfügt über mehr als 102.400 MiB Speicher, und der verfügbare Speicher erreicht 1024 MiB oder weniger.
- Die DB-Instance verfügt über zwischen 20.000 MiB und 102.400 MiB Speicher und verfügt über weniger als 1% des verfügbaren Speichers.

Nachdem eine DB-Instance automatisch Amazon RDS gestoppt wurde, weil sie den `storage-full` Status erreicht hat, können Sie sie immer noch ändern. Um die DB-Instance neu zu starten, führen Sie mindestens einen der folgenden Schritte aus:

- Ändern Sie die DB-Instance, um die automatische Speicherung zu aktivieren.

  Weitere Informationen zum Autoscaling von Storage finden Sie unter Automatische Kapazitätsverwaltung mit automatischer Amazon RDS-Speicherskalierung.
- Ändern Sie die DB-Instance, um ihre Speicherkapazität zu erhöhen.

  Weitere Informationen zur Erhöhung der Speicherkapazität finden Sie unter Steigern der DB-Instance-Speicherkapazität.

Nachdem Sie eine dieser Änderungen vorgenommen haben, wird die DB-Instance automatisch neu gestartet. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Inkonsistente Größe des InnoDB-Buffer-Pools

Für MySQL 5.7 gibt es aktuell einen Bug beim Verwalten der Größe des InnoDB-Buffer-Pools. MySQL 5.7 könnte den Wert des Parameters `innodb_buffer_pool_size` an einen großen Wert anpassen, was dazu führen kann, dass der InnoDB-Buffer-Pool zu groß wird und dadurch zu viel Arbeitsspeicher verbraucht. Dieser Effekt kann dazu führen, dass die Ausführung der MySQL-Datenbank-Engine beendet wird oder die Engine nicht gestartet werden kann. Dieses Problem ist häufiger bei DB-Instance-Klassen vorhanden, die weniger Arbeitsspeicher zur Verfügung haben.

Setzen Sie den Wert des Parameters `innodb_buffer_pool_size` auf ein Vielfaches des Produkts der Parameterwerte `innodb_buffer_pool_instances` und `innodb_buffer_pool_chunk_size`, um das Problem zu beheben. Sie könnten beispielsweise den Parameterwert `innodb_buffer_pool_size` auf das achtfache des Produkts der

Parameterwerte `innodb_buffer_pool_instances` und `innodb_buffer_pool_chunk_size` setzen, wie im folgenden Beispiel gezeigt.

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

Einzelheiten zu diesem MySQL 5.7-Bug finden Sie unter https://bugs.mysql.com/bug.php? id=79379 in der MySQL-Dokumentation.

Index-Merge-Optimierung zeigt falsche Ergebnisse an

Abfragen, die die Optimierung der Indexzusammenführung verwenden, können aufgrund eines Fehlers im MySQL-Abfrageoptimierer, der in MySQL 5.5.37 eingeführt wurde, falsche Ergebnisse zurückgeben. Wenn Sie eine Abfrage für eine Tabelle mit mehreren Indizes ausführen, scannt der Optimierer die Zeilenbereiche anhand der Indizes, führt die Ergebnisse jedoch nicht korrekt zusammen. Weitere Informationen zum Bug im Abfragenoptimierer finden Sie unter http://bugs.mysql.com/bug.php?id=72745 und http://bugs.mysql.com/bug.php?id=68194 in der MySQL-Bug-Datenbank.

Denken Sie beispielsweise an eine Abfrage für eine Tabelle mit zwei Indizes, wobei die Suchmuster auf die indizierten Spalten verweisen.

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

In diesem Fall durchsucht die Suchmaschine beide Indizes. Aufgrund des Fehlers sind die zusammengeführten Ergebnisse jedoch falsch.

Um dieses Problem zu beheben, können Sie eine der folgenden Aktionen ausführen:

- Stellen Sie den Parameter `optimizer_switch` in Ihrer DB-Parametergruppe für Ihre MySQL-DB-Instance auf `index_merge=off` ein. Weitere Informationen über das Einstellen von Parametern in DB-Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.
- Führen Sie für Ihre MySQL-DB-Instance ein Upgrade auf MySQL Version 5.7 oder 8.0 durch. Weitere Informationen finden Sie unter Upgrades der DB-Engine RDS für MySQL.
- Wenn Sie Ihre Instance nicht upgraden oder den Parameter `optimizer_switch` nicht ändern können, können Sie alternativ einen Index für die Abfrage explizit bestimmen, beispielsweise so:

```
SELECT * FROM table1
USE INDEX covering_index
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

Weitere Informationen finden Sie unter Index-Merge-Optimierung in der MySQL-Dokumentation.

MySQL-Parameterausnahmen für Amazon RDS-DB-Instances

Einige MySQL-Parameter erfordern besondere Beachtung bei der Verwendung in einer Amazon RDS-DB-Instance.

lower_case_table_names

Da Amazon RDS ein Dateisystem mit Berücksichtigung von Groß- und Kleinschreibung verwendet, wird die Festlegung des Werts 2 für den Serverparameter `lower_case_table_names` (Namen werden wie angegeben gespeichert, aber in Kleinbuchstaben verglichen) nicht unterstützt. Nachfolgend sind die unterstützten Werte für Amazon RDS for MySQL DB-Instances aufgeführt:

- 0 (Namen werden wie angegeben gespeichert und bei Vergleichen wird die Groß-/Kleinschreibung berücksichtigt) wird für alle RDS-für-MySQL-Versionen unterstützt.
- 1 (Namen werden in Kleinbuchstaben gespeichert und Vergleiche unterscheiden nicht zwischen Groß- und Kleinschreibung) wird für RDS for MySQL Version 5.7, Version 8.0.28 und höhere Versionen 8.0 und Version 8.4 unterstützt.

Legen Sie den Parameter `lower_case_table_names` in einer benutzerdefinierten DB-Parametergruppe fest, bevor Sie eine DB-Instance erstellen. Stellen Sie dann die benutzerdefinierte DB-Parametergruppe ein, wenn Sie die DB-Instance erstellen.

Wenn eine Parametergruppe mit einer MySQL-DB-Instance mit einer niedrigeren Version als 8.0 verknüpft ist, empfehlen wir, die Parameter `lower_case_table_names` in der Parametergruppe nicht zu ändern. Eine Änderung könnte zu Inkonsistenzen bei point-in-time Wiederherstellungs-Backups und Read Replica-DB-Instances führen.

Wenn eine Parametergruppe mit einer MySQL-DB-Instance der Version 8.0 oder 8.4 verknüpft ist, können Sie den `lower_case_table_names` Parameter in der Parametergruppe nicht ändern.

Lesereplikate sollten immer den selben `lower_case_table_names`-Parameterwert wie die Quell-DB-Instance verwenden.

long_query_time

Sie können den Parameter `long_query_time` auf einen Fließkommawert einstellen, damit Sie langsame Abfragen im MySQL-Slow-Query-Protokoll in Mikrosekunden-Auflösung protokollieren können. Sie können einen Wert von z. B. 0,1 Sekunden einstellen (100 Millisekunden), um das Debuggen bei langsamen Transaktionen, die weniger als eine Sekunde dauern, zu erleichtern.

MySQL-Dateigrößenlimits in Amazon RDS

Für MySQL-Versionen 8.0 und höhere DB-Instances beträgt die maximale Dateigröße 16 TiB. Bei der Verwendung von file-per-table Tablespaces begrenzt die maximale Dateigröße die Größe einer InnoDB-Tabelle auf 16 TiB. file-per-tableInnoDB-Tablespaces (mit Tabellen in jeweils einem eigenen Tablespace) sind standardmäßig für MySQL-DB-Instances festgelegt. Weitere Informationen finden Sie unter InnoDB-Grenzwerte in der MySQL-Dokumentation.

> **Note**
>
> Einige existierende DB-Instances haben eine Untergrenze. Beispielsweise haben MySQL-DB-Instances, die vor April 2014 erstellt wurden, ein Datei- und Tabellenlimit von 2 TB. Dieses Dateilimit von 2 TB gilt auch für DB-Instances oder Lesereplikate, die aus DB-Snapshots erstellt wurden, die vor April 2014 gemacht wurden, unabhängig davon wann die DB-Instance erstellt wurde.

Die Verwendung von file-per-table InnoDB-Tablespaces hat je nach Ihrer Anwendung Vor- und Nachteile. Informationen zum besten Ansatz für Ihre Anwendung finden Sie unter File-per-table Tablespaces in der MySQL-Dokumentation.

Es wird nicht empfohlen, die Tabellen bis zur maximal möglichen Größe anwachsen zu lassen. Generell hat es sich bewährt, Daten in kleinere Tabellen zu partitionieren, wodurch sich die Leistung und die Wiederherstellungszeiten verbessern.

Eine Möglichkeit, mit der Sie eine große Tabelle in kleinere Tabellen aufteilen können, ist die Partitionierung. Die Partitionierung verteilt Teile Ihrer großen Tabelle in separate Dateien auf der Basis von Regeln, die Sie angeben. Wenn Sie beispielsweise Transaktionen nach Datum speichern, können Sie Partitionierungsregeln erstellen, mit denen ältere Transaktionen in separate Dateien partitioniert werden. Anschließend können Sie regelmäßig die historischen Transaktionsdaten archivieren, die für Ihre Anwendung nicht ständig verfügbar sein müssen. Weitere Informationen finden Sie unter Partitionierung in der MySQL-Dokumentation.

Da es keine einzelne Systemtabelle oder Ansicht gibt, in der die Größe aller Tabellen und des Tabellenraums des InnoDB-Systems angegeben ist, müssen Sie mehrere Tabellen abfragen, um die Größe der Tabellenräume zu ermitteln.

So ermitteln Sie die Größe des Tabellenraums des InnoDB-Systems und des Tabellenraums des Datenwörterbuchs

- Verwenden Sie den folgenden SQL-Befehl, um zu bestimmen, ob einer Ihrer Tabellenräume zu groß ist und eventuell partitioniert werden sollte.

  > **Note**
  >
  > Der Datenwörterbuch-Tablespace ist spezifisch für MySQL 8.0 und höhere Versionen.

```
select FILE_NAME,TABLESPACE_NAME, ROUND(((TOTAL_EXTENTS*EXTENT_SIZE)
/1024/1024/1024), 2)  as "File Size (GB)" from information_schema.FILES
where tablespace_name in ('mysql','innodb_system');
```

So ermitteln Sie die Größe von InnoDB-Benutzertabellen außerhalb des Tabellenraums des InnoDB-Systems (für MySQL 5.7-Versionen)

- Verwenden Sie den folgenden SQL-Befehl, um zu bestimmen, ob eine Ihrer Tabellen zu groß ist und evtl. partitioniert werden sollte.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Um die Größe von InnoDB-Benutzertabellen außerhalb des InnoDB-System-Tablespace zu bestimmen (für MySQL 8.0 und höhere Versionen)

- Verwenden Sie den folgenden SQL-Befehl, um zu bestimmen, ob eine Ihrer Tabellen zu groß ist und evtl. partitioniert werden sollte.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
```

```
FROM information_schema.INNODB_TABLESPACES ORDER BY 3 DESC;
```

So ermitteln Sie die Größe von Nicht-InnoDB-Benutzertabellen

- Verwenden Sie den folgenden SQL-Befehl, um zu bestimmen, ob eine Ihrer Nicht-InnoDB-Benutzertabellen zu groß ist.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Um file-per-table InnoDB-Tablespaces zu aktivieren

- Setzen Sie den Parameter innodb_file_per_table in der Parametergruppe für die DB-Instance auf `1`.

Um file-per-table InnoDB-Tablespaces zu deaktivieren

- Setzen Sie den Parameter innodb_file_per_table in der Parametergruppe für die DB-Instance auf `0`.

Weitere Informationen über das Updaten von Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Wenn Sie file-per-table InnoDB-Tablespaces aktiviert oder deaktiviert haben, können Sie einen `ALTER TABLE` Befehl ausführen, um eine Tabelle vom globalen Tablespace in ihren eigenen Tablespace oder von ihrem eigenen Tablespace in den globalen Tablespace zu verschieben, wie im folgenden Beispiel gezeigt:

```
ALTER TABLE table_name TABLESPACE=innodb_file_per_table;
```

MySQL Keyring-Plugin wird nicht unterstützt

Derzeit unterstützt Amazon RDS für MySQL das MySQL-Amazon-Web-Services-Keyring-Plugin `keyring_aws` nicht.

Benutzerdefinierte Ports

Amazon RDS blockiert Verbindungen zum benutzerdefinierten Port 33060 für die MySQL-Engine. Wählen Sie einen anderen Port für Ihre MySQL-Engine.

Einschränkungen bei gespeicherten MySQL-Prozeduren

Die gespeicherten Prozeduren mysql.rds_kill und mysql.rds_kill_query können Sitzungen oder Abfragen von MySQL-Benutzern mit Benutzernamen, die länger als 16 Zeichen sind, in den folgenden Versionen von RDS für MySQL nicht beenden:

- 8.0.32 und niedrigere 8-Versionen
- 5.7.41 und niedrigere 5.7-Versionen

GTID-basierte Replikation mit einer externen Quell-Instance

Amazon RDS unterstützt die auf globalen Transaktions-Identifikatoren (GTIDs) basierende Replikation von einer externen MySQL-Instance in eine Amazon RDS for MySQL MySQL-DB-Instance, für die während der Konfiguration GTID_PURGED gesetzt werden muss. Diese Funktionalität wird jedoch nur von RDS for MySQL 8.0.37 und höheren Versionen unterstützt.

MySQL-Standardauthentifizierungs-Plugin

RDS für MySQL Version 8.0.34 und höhere Versionen 8.0 verwenden das `mysql_native_password` Plugin. Sie können die `default_authentication_plugin`-Einstellung nicht ändern.

RDS für MySQL Version 8.4 und höhere Versionen verwenden das `caching_sha2_password` Plugin als Standard-Authentifizierungs-Plugin. Sie können das Standard-Authentifizierungs-Plugin für MySQL 8.4 ändern. Das `mysql_native_password` Plugin funktioniert immer noch mit MySQL 8.4, aber die Unterstützung dieses Plugins endet mit MySQL 8.4. Um das Standard-Authentifizierungs-Plugin zu ändern, erstellen Sie eine benutzerdefinierte Parametergruppe und ändern Sie den Wert des `authentication_policy` Parameters. Weitere Informationen finden Sie unter the section called “Standard- und benutzerdefinierte Parametergruppen”.

Überschreiben von innodb_buffer_pool_size

Bei Mikro- oder kleinen DB-Instance-Klassen kann der Standardwert für den `innodb_buffer_pool_size` Parameter von dem Wert abweichen, der beim Ausführen des folgenden Befehls zurückgegeben wird:

```
mysql> SELECT @@innodb_buffer_pool_size;
```

Dieser Unterschied kann auftreten, wenn Amazon RDS im Rahmen der Verwaltung der DB-Instance-Klassen den Standardwert überschreiben muss. Bei Bedarf können Sie den Standardwert überschreiben und ihn auf einen Wert setzen, den Ihre DB-Instance-Klasse unterstützt. Um einen gültigen Wert zu ermitteln, addieren Sie die Speichernutzung und den gesamten auf Ihrer DB-Instance verfügbaren Speicher. Weitere Informationen finden Sie unter Amazon RDS-Instance-Typen.

Wenn Ihre DB-Instance nur 4 GB Arbeitsspeicher hat, können Sie sie nicht `innodb_buffer_pool_size` auf 8 GB festlegen, aber Sie können sie möglicherweise auf 3 GB setzen, je nachdem, wie viel Speicher Sie für andere Parameter zugewiesen haben.

Wenn der von Ihnen eingegebene Wert zu groß ist, senkt Amazon RDS den Wert auf die folgenden Grenzwerte:

- Micro-DB-Instance-Klassen: 256 MB
- db.t4g.micro DB-Instance-Klassen: 128 MB

Upgrade von MySQL 5.7 auf MySQL 8.4

Sie können nicht direkt von MySQL 5.7 auf MySQL 8.4 aktualisieren. Sie müssen zuerst ein Upgrade von MySQL 5.7 auf MySQL 8.0 und dann ein Upgrade von MySQL 8.0 auf MySQL 8.4 durchführen. Weitere Informationen finden Sie unter Hauptversions-Upgrades für RDS for MySQL.

InnoDB-Seitenkomprimierung

Die InnoDB-Seitenkomprimierung funktioniert nicht mit Amazon RDS-DB-Instances, die eine Dateisystem-Blockgröße von 16 KB haben, da die Dateisystem-Blockgröße kleiner als die InnoDB-Seitengröße sein muss. Ab Februar 2024 haben alle neu erstellten DB-Instances eine Dateisystem-Blockgröße von 16 KB, was den Durchsatz erhöht und den IOPS-Verbrauch bei Seitenlöschungen senkt.

RDSals Referenz für meine SQL gespeicherte Prozedur

In diesen Themen werden gespeicherte Systemprozeduren beschrieben, die für RDS Amazon-Instances verfügbar sind, auf denen die My SQL DB-Engine ausgeführt wird. Diese Prozeduren müssen vom Hauptbenutzer ausgeführt werden.

Themen

- Erfassung und Pflege der globalen Statushistorie
- Konfiguration, Starten und Stoppen der Binärprotokollreplikation (Binlog)
- Beenden einer Sitzung oder Abfrage
- Verwaltung von aktiv-aktiven Clustern
- Verwaltung der Replikation mit mehreren Quellen
- Transaktionen replizieren mit GTIDs
- Rotieren der Abfrage-Logs
- Konfiguration des Binärprotokolls einrichten und anzeigen
- Wärmen des InnoDB-Caches

Erfassung und Pflege der globalen Statushistorie

Amazon RDS bietet eine Reihe von Verfahren, mit denen Schnappschüsse der Werte von Statusvariablen im Laufe der Zeit erstellt und diese zusammen mit allen Änderungen seit dem letzten Snapshot in eine Tabelle geschrieben werden. Diese Infrastruktur wird als Global Status History bezeichnet. Weitere Informationen finden Sie unter Verwaltung des globalen Statusverlaufs für RDS for My SQL.

Die folgenden gespeicherten Verfahren verwalten, wie die Global Status History erfasst und verwaltet wird.

Themen

- mysql.rds_collect_global_status_history
- mysql.rds_disable_gsh_collector
- mysql.rds_disable_gsh_rotation
- mysql.rds_enable_gsh_collector
- mysql.rds_enable_gsh_rotation
- mysql.rds_rotate_global_status_history
- mysql.rds_set_gsh_collector
- mysql.rds_set_gsh_rotation

mysql.rds_collect_global_status_history

Generiert einen Snapshot auf Anforderung für den globalen Statusverlauf (Global Status History, GoSH).

Syntax

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_disable_gsh_collector

Deaktiviert die periodische Generierung von Snapshots des globalen Statusverlaufs (Global Status History, GoSH).

Syntax

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_disable_gsh_rotation

Schaltet die Rotation der `mysql.global_status_history`-Tabelle aus.

Syntax

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_enable_gsh_collector

Aktiviert den globalen Statusverlauf (Global Status History, GoSH), um Standard-Snapshots in zeitlichen Abständen, die mithilfe von `rds_set_gsh_collector` festgelegt wurden, zu generieren.

Syntax

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_enable_gsh_rotation

Aktiviert die Rotation der Inhalte der Tabelle `mysql.global_status_history` zu `mysql.global_status_history_old` in zeitlichen Abständen, die durch `rds_set_gsh_rotation` angegeben werden.

Syntax

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

Rotiert die Inhalte der Tabelle `mysql.global_status_history` bei Anforderung zu `mysql.global_status_history_old`.

Syntax

```
CALL mysql.rds_rotate_global_status_history;
```

mysql.rds_set_gsh_collector

Gibt den zeitlichen Abstand für die Generierung von aufeinander folgenden Snapshots durch den globalen Statusverlauf (Global Status History, GoSH) an.

Syntax

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

Parameter

intervalPeriod

Der zeitliche Abstand in Minuten für die periodische Generierung von Snapshots. Der Standardwert ist 5.

mysql.rds_set_gsh_rotation

Gibt den zeitlichen Abstand in Tagen für die periodische Rotation der Tabelle `mysql.global_status_history` an.

Syntax

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

Parameter

intervalPeriod

Der zeitliche Abstand in Tagen für die periodische Tabellenrotation. Der Standardwert ist 7.

Konfiguration, Starten und Stoppen der Binärprotokollreplikation (Binlog)

Die folgenden gespeicherten Prozeduren steuern, wie Transaktionen von einer externen Datenbank in RDS für My oder von RDS for My SQL SQL in eine externe Datenbank repliziert werden.

Wenn Sie diese gespeicherten Prozeduren verwenden, um die Replikation mit einem mit konfigurierten Replikationsbenutzer zu verwaltencaching_sha2_password, müssen Sie TLS bei der Konfiguration Folgendes angebenSOURCE_SSL=1: caching_sha2_passwordist das Standard-Authentifizierungs-Plug-In RDS für My SQL 8.4. Weitere Informationen finden Sie unterVerschlüsseln mit/SSLTLS.

Informationen zur Konfiguration, Verwendung und Verwaltung von Read Replicas finden Sie unterthe section called “Meine SQL Read Replicas”.

Themen

-
-
-
-
-
-
- mysql.rds_set_external_master_with_auto_position (für meine Hauptversionen 8.0 und niedriger) RDS SQL
- mysql.rds_set_external_source_with_auto_position (für meine Hauptversionen 8.4 und höher) RDS SQL
- mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL
- mysql.rds_set_external_source_with_delay (für meine Hauptversionen 8.4 und höher) RDS SQL
- mysql.rds_set_external_source_gtid_purged
-
-
- mysql.rds_set_source_delay
- mysql.rds_skip_repl_error
- mysql.rds_start_replication

-
- mysql.rds_stop_replication

Ändert die Protokollposition der Quelldatenbankinstance in den Anfang des nächsten Binärprotokolls auf der Quelldatenbankinstance. Verwenden Sie diese Prozedur nur dann, wenn Sie für ein Lesereplikat bei der Replikation einen I/O-Fehler mit der Fehlernummer 1236 erhalten.

Syntax

```
CALL mysql.rds_next_master_log(
curr_master_log
);
```

Parameter

curr_master_log

Der Index der aktuellen Master-Protokolldatei. Der Index ist im Dateinamen codiert. Eine aktuelle Datei mit dem Namen `mysql-bin-changelog.012345` hat beispielsweise den Index 12345. Um den Namen der aktuellen Master-Protokolldatei zu ermitteln, führen Sie den Befehl `SHOW REPLICA STATUS` aus. Sie finden den Namen anschließend im Feld `Master_Log_File`.

Nutzungshinweise

Die Prozedur `mysql.rds_next_master_log` muss vom Hauptbenutzer ausgeführt werden.

Warning

`mysql.rds_next_master_log` sollte nur dann aufgerufen werden, wenn die Replikation nach einem Failover einer als Replikationsquelle fungierenden Multi-AZ-DB-Instance fehlschlägt und das Feld `Last_IO_Errno` im von `SHOW REPLICA STATUS` zurückgegebenen Ergebnis einen I/O-Fehler mit der Nummer 1236 meldet.
Ein Aufruf von `mysql.rds_next_master_log` kann zu Datenverlust im Lesereplikat führen, falls Transaktionen in der Quell-Instance nicht in das binäre Protokoll auf der Festplatte geschrieben wurden, bevor das Failover-Ereignis auftrat. Sie können die Wahrscheinlichkeit verringern, dass dies passiert, indem Sie die Quellinstanzparameter und auf festlegen, auch wenn dies die Leistung beeinträchtigen könnte. `sync_binlog`

`innodb_support_xa` 1 Weitere Informationen finden Sie unter Behebung eines My SQL Read Replica-Problems.

Beispiele

Gehen Sie davon aus, dass die Replikation auf einer RDSfür My SQL Read Replica fehlschlägt. Die Ausführung von `SHOW REPLICA STATUS\G` für das Lesereplikat gibt das folgende Ergebnis zurück:

```
*************************** 1. row ***************************
             Replica_IO_State:
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: MasterUser
                  Source_Port: 3306
                Connect_Retry: 10
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: relaylog.012340
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Source_Log_Pos: 30223232
              Relay_Log_Space: 5248928866
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Source_SSL_Allowed: No
           Source_SSL_CA_File:
           Source_SSL_CA_Path:
              Source_SSL_Cert:
            Source_SSL_Cipher:
               Source_SSL_Key:
        Seconds_Behind_Master: NULL
```

```
Source_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Source_Server_Id: 67285976
```

Den Angeben im Feld `Last_IO_Errno` ist zu entnehmen, dass die Instance eine I/O-Fehlermeldung mit der Nummer 1236 erhalten hat. Dem Feld `Master_Log_File` ist zudem zu entnehmen, dass die betroffene Protokolldatei den Namen `mysql-bin-changelog.012345` aufweist und ihr Index folglich `12345` lautet. Zur Behebung des Fehlers können Sie dann `mysql.rds_next_master_log` mit dem folgenden Parameter aufrufen:

```
CALL mysql.rds_next_master_log(12345);
```

Ändert die Protokollposition der Quelldatenbankinstance in den Anfang des nächsten Binärprotokolls auf der Quelldatenbankinstance. Verwenden Sie diese Prozedur nur dann, wenn Sie für ein Lesereplikat bei der Replikation einen I/O-Fehler mit der Fehlernummer 1236 erhalten.

Syntax

```
CALL mysql.rds_next_source_log(
curr_source_log
);
```

Parameter

curr_source_log

Der Index der aktuellen Quell-Protokolldatei. Der Index ist im Dateinamen codiert. Eine aktuelle Datei mit dem Namen `mysql-bin-changelog.012345` hat beispielsweise den Index 12345. Um den Namen der aktuellen Quell-Protokolldatei zu ermitteln, führen Sie den Befehl `SHOW REPLICA STATUS` aus. Sie finden den Namen anschließend im Feld `Source_Log_File`.

Nutzungshinweise

Der Administrator muss das `mysql.rds_next_source_log`-Verfahren ausführen.

> **Warning**
>
> `mysql.rds_next_source_log` sollte nur dann aufgerufen werden, wenn die Replikation nach einem Failover einer als Replikationsquelle fungierenden Multi-AZ-DB-Instance fehlschlägt und das Feld `Last_IO_Errno` im von `SHOW REPLICA STATUS` zurückgegebenen Ergebnis einen I/O-Fehler mit der Nummer 1236 meldet.
> Ein Aufruf von `mysql.rds_next_source_log` kann zu Datenverlust im Lesereplikat führen, falls Transaktionen in der Quell-Instance nicht in das binäre Protokoll auf der Festplatte geschrieben wurden, bevor das Failover-Ereignis auftrat. Sie können die Wahrscheinlichkeit verringern, dass dies passiert, indem Sie die Quellinstanzparameter `sync_binlog` und auf festlegen, auch wenn dies die Leistung beeinträchtigen könnte. `innodb_support_xa` 1 Weitere Informationen finden Sie unter Behebung eines My SQL Read Replica-Problems.

Beispiele

Gehen Sie davon aus, dass die Replikation auf einer RDSfür My SQL Read Replica fehlschlägt. Die Ausführung von `SHOW REPLICA STATUS\G` für das Lesereplikat gibt das folgende Ergebnis zurück:

```
*************************** 1. row ***************************
             Replica_IO_State:
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: MasterUser
                  Source_Port: 3306
                Connect_Retry: 10
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: relaylog.012340
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
```

```
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Source_Log_Pos: 30223232
              Relay_Log_Space: 5248928866
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Source_SSL_Allowed: No
           Source_SSL_CA_File:
           Source_SSL_CA_Path:
              Source_SSL_Cert:
            Source_SSL_Cipher:
               Source_SSL_Key:
        Seconds_Behind_Source: NULL
Source_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from source when reading data from
 binary log: 'Client requested source to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Source_Server_Id: 67285976
```

Den Angeben im Feld `Last_IO_Errno` ist zu entnehmen, dass die Instance eine I/O-Fehlermeldung mit der Nummer 1236 erhalten hat. Dem Feld `Source_Log_File` ist zudem zu entnehmen, dass die betroffene Protokolldatei den Namen `mysql-bin-changelog.012345` aufweist und ihr Index folglich `12345` lautet. Zur Behebung des Fehlers können Sie dann `mysql.rds_next_source_log` mit dem folgenden Parameter aufrufen:

```
CALL mysql.rds_next_source_log(12345);
```

Konfiguriert eine My SQL DB-Instance RDSfür My SQL neu, sodass sie keine Read Replica einer My-Instance mehr ist, die außerhalb von SQL Amazon ausgeführt wird. RDS

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_reset_external_master;
```

Nutzungshinweise

Die Prozedur `mysql.rds_reset_external_master` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer SQL My-Instance entfernt werden soll, die außerhalb von Amazon RDS ausgeführt wird.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Weitere Informationen zur Verwendung der Replikation zum Importieren von Daten aus einer Instance von My, die außerhalb von Amazon SQL ausgeführt wirdRDS, finden Sie unterKonfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

Konfiguriert eine My SQL DB-Instance RDSfür My SQL neu, sodass sie keine Read Replica einer My-Instance mehr ist, die außerhalb von SQL Amazon ausgeführt wird. RDS

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_reset_external_source;
```

Nutzungshinweise

Der Administrator muss das `mysql.rds_reset_external_source`-Verfahren ausführen. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer SQL My-Instance entfernt werden soll, die außerhalb von Amazon RDS ausgeführt wird.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten. Weitere Informationen zur Verwendung der Replikation zum Importieren von Daten aus einer Instance von My, die außerhalb von Amazon SQL ausgeführt wirdRDS, finden Sie unterKonfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

Konfiguriert eine RDSMy SQL DB-Instance als Read Replica einer My-Instance, die außerhalb von Amazon SQL ausgeführt wird. RDS

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

> **Note**
>
> Sie können die gespeicherte Prozedur mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL zum Konfigurieren einer externer Quelldatenbank-Instance und einer verzögerten Replikation verwenden.

Syntax

```
CALL mysql.rds_set_external_master (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der SQL My-Instance, die außerhalb von Amazon ausgeführt wird und die Quelldatenbank-Instance werden RDS soll.

host_port

Der Port, der von der SQL My-Instance verwendet wird, die außerhalb von Amazon ausgeführt wirdRDS, um als Quelldatenbank-Instance konfiguriert zu werden. Wenn Ihre Netzwerkkonfiguration eine Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen für die My SQL Instance, die außerhalb von Amazon ausgeführt wirdRDS. Es wird empfohlen, ein Benutzerkonto bereitzustellen, das ausschließlich für die Replikation mit der externen Instance genutzt wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

mysql_binary_log_file_name

Der Name des Binärprotokolls auf der Quelldatenbank-Instance, die die Replikationsinformationen enthält.

mysql_binary_log_file_location

Die Position in der binären Protokolldatei `mysql_binary_log_file_name`, ab der bei der Replikation die Replikationsinformationen gelesen werden.

Sie können den Namen und den Speicherort der Binlog-Datei ermitteln, indem Sie `SHOW MASTER STATUS`auf der Quelldatenbankinstanz starten.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `MASTER_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_master` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer My SQL Instance konfiguriert werden soll, die außerhalb von Amazon RDS ausgeführt wird.

Vor der Ausführung `mysql.rds_set_external_master` müssen Sie die Instance von My SQL running external RDS to Amazon als Quelldatenbank-Instance konfigurieren. Um eine Verbindung mit der SQL My-Instance herzustellen, die außerhalb von Amazon ausgeführt wirdRDS, müssen Sie `replication_user_password` Werte angeben`replication_user_name`, die auf einen Replikationsbenutzer hinweisen, der über `REPLICATION SLAVE` Berechtigungen für die externe Instance von My verfügt `REPLICATION CLIENT`SQL.

Um eine externe Instance von My SQL als Quelldatenbank-Instance zu konfigurieren

1. Stellen Sie mit dem SQL My-Client Ihrer Wahl eine Verbindung zur externen Instanz von My her SQL und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   Mein SQL 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   Meine SQL 8,0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   > **Note**
   >
   > Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der externen Instanz von My SQL `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer 'repl_user' für Ihre Domäne die Berechtigungen `REPLICATION CLIENT` und `REPLICATION SLAVE` für alle Datenbanken erteilt.

   Mein SQL 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   Meine SQL 8,0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

Um die verschlüsselte Replikation zu verwenden, konfigurieren Sie die Quelldatenbank-Instance für die Verwendung von SSL Verbindungen.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Nach dem Aufruf `mysql.rds_set_external_master` zur Konfiguration einer Amazon RDS DB-Instance als Read Replica können Sie die Read Replica aufrufenmysql.rds_start_replication, um den Replikationsprozess zu starten. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Wenn aufgerufen `mysql.rds_set_external_master` wird, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set master` in den `mysql.rds_replication_status` Tabellen `mysql.rds_history` und auf.

Beispiele

Wenn sie auf einer My SQL DB-Instance ausgeführt wird, konfiguriert das folgende Beispiel die DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS

```
call mysql.rds_set_external_master(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
```

```
  120,
  1);
```

Konfiguriert eine RDSMy SQL DB-Instance als Read Replica einer My-Instance, die außerhalb von Amazon SQL ausgeführt wird. RDS

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_source (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der SQL My-Instance, die außerhalb von Amazon ausgeführt wird und die Quelldatenbank-Instance werden RDS soll.

host_port

Der Port, der von der SQL My-Instance verwendet wird, die außerhalb von Amazon ausgeführt wirdRDS, um als Quelldatenbank-Instance konfiguriert zu werden. Wenn Ihre Netzwerkkonfiguration eine Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen für die My SQL Instance, die außerhalb von Amazon ausgeführt wirdRDS. Es wird empfohlen, ein Benutzerkonto bereitzustellen, das ausschließlich für die Replikation mit der externen Instance genutzt wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

mysql_binary_log_file_name

Der Name des Binärprotokolls auf der Quelldatenbank-Instance, die die Replikationsinformationen enthält.

mysql_binary_log_file_location

Die Position in der binären Protokolldatei `mysql_binary_log_file_name`, ab der bei der Replikation die Replikationsinformationen gelesen werden.

Sie können den Namen und den Speicherort der Binlog-Datei ermitteln, indem Sie `SHOW MASTER STATUS`auf der Quelldatenbankinstanz starten.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `SOURCE_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

Nutzungshinweise

Der Administrator muss das `mysql.rds_set_external_source`-Verfahren ausführen. Dieses Verfahren muss auf der DB-Instance ausgeführt werden, um sie als Read Replica einer SQL My-Instance zu konfigurieren, die außerhalb von Amazon RDS ausgeführt wird.

Vor der Ausführung `mysql.rds_set_external_source` müssen Sie die Instance von My SQL running external RDS to Amazon als Quelldatenbank-Instance konfigurieren. Um eine Verbindung mit der SQL My-Instance herzustellen, die außerhalb von Amazon ausgeführt wirdRDS, müssen Sie `replication_user_password` Werte angeben`replication_user_name`, die auf einen Replikationsbenutzer hinweisen, der über `REPLICATION SLAVE` Berechtigungen für die externe Instance von My verfügt `REPLICATION CLIENT`SQL.

Um eine externe Instance von My SQL als Quelldatenbank-Instance zu konfigurieren

1. Stellen Sie mit dem SQL My-Client Ihrer Wahl eine Verbindung zur externen Instanz von My her SQL und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   > **Note**
   >
   > Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der externen Instanz von My SQL `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer 'repl_user' für Ihre Domäne die Berechtigungen `REPLICATION CLIENT` und `REPLICATION SLAVE` für alle Datenbanken erteilt.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

Um die verschlüsselte Replikation zu verwenden, konfigurieren Sie die Quelldatenbankinstanz für die Verwendung von SSL Verbindungen. Importieren Sie außerdem mit der Prozedur mysql.rds_import_binlog_ssl_material das Zertifizierungsstellenzertifikat, das Clientzertifikat und den Clientschlüssel in die DB-Instance bzw. den DB-Cluster.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances.

Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Nach einem Aufruf `mysql.rds_set_external_source` zur Konfiguration einer for My SQL DB-Instance als Read Replica können Sie die Read Replica aufrufenmysql.rds_start_replication, um den Replikationsprozess zu starten. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Wenn aufgerufen `mysql.rds_set_external_source` wird, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set master` in den `mysql.rds_replication_status` Tabellen `mysql.rds_history` und auf.

Beispiele

Wenn sie auf einer DB-Instance ausgeführt wird, konfiguriert das folgende Beispiel die DB-Instance als Read Replica einer My-Instance, die außerhalb von SQL Amazon läuft. RDS

```
call mysql.rds_set_external_source(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  1);
```

mysql.rds_set_external_master_with_auto_position (für meine Hauptversionen 8.0 und niedriger) RDS SQL

Konfiguriert eine RDS for My SQL DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS Bei diesem Verfahren werden auch verzögerte Replikation und Replikation auf der Grundlage globaler Transaktions-Identifikatoren () konfiguriert. GTIDs

Important

Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von

Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_master_with_auto_position (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der SQL My-Instance, die außerhalb von Amazon ausgeführt wird und die Quelldatenbank-Instance werden RDS soll.

host_port

Der Port, der von der SQL My-Instance verwendet wird, die außerhalb von Amazon ausgeführt wirdRDS, um als Quelldatenbank-Instance konfiguriert zu werden. Wenn Ihre Netzwerkkonfiguration eine Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen für die My SQL Instance, die außerhalb von Amazon ausgeführt wirdRDS. Es wird empfohlen, ein Benutzerkonto bereitzustellen, das ausschließlich für die Replikation mit der externen Instance genutzt wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `MASTER_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quelldatenbank-Instance zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_master_with_auto_position` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer My SQL Instance konfiguriert werden soll, die außerhalb von Amazon RDS ausgeführt wird.

Dieses Verfahren wird für alle Versionen von My SQL 5.7 und RDS RDS für My SQL 8.0.26 und höhere Versionen 8.0 unterstützt.

Vor der Ausführung `mysql.rds_set_external_master_with_auto_position` müssen Sie die Instance von My SQL running external RDS to Amazon als Quelldatenbank-Instance konfigurieren. Um eine Verbindung mit der SQL My-Instance herzustellen, die außerhalb von Amazon ausgeführt wirdRDS, müssen Sie Werte für `replication_user_name` und angeben`replication_user_password`. Diese Werte müssen einen Replikationsbenutzer angeben, der über `REPLICATION SLAVE` Berechtigungen für die externe Instance von My verfügt `REPLICATION CLIENT`SQL.

Um eine externe Instanz von My SQL als Quelldatenbankinstanz zu konfigurieren

1. Stellen Sie mit dem SQL My-Client Ihrer Wahl eine Verbindung zur externen Instanz von My her SQL und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der externen Instanz von My SQL `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer `REPLICATION CLIENT` für Ihre Domäne die Berechtigungen `REPLICATION SLAVE` und `'repl_user'` für alle Datenbanken erteilt.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Weitere Informationen finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Stellen Sie vor dem Aufrufen sicher`mysql.rds_set_external_master_with_auto_position`, dass Sie die `gtid_purged` Systemvariable aufrufen, the section called “mysql.rds_set_external_source_gtid_purged” um sie auf einen bestimmten GTID Bereich aus einer externen Quelle festzulegen.

Nach dem Aufruf `mysql.rds_set_external_master_with_auto_position` zur Konfiguration einer Amazon RDS DB-Instance als Read Replica können Sie die Read Replica

aufrufenmysql.rds_start_replication, um den Replikationsprozess zu starten. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Wenn Sie anrufen`mysql.rds_set_external_master_with_auto_position`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set master` in den `mysql.rds_replication_status` Tabellen `mysql.rds_history` und auf.

Für die Notfallwiederherstellung können Sie diese Prozedur mit der gespeicherten Prozedur oder verwenden. Um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen, können Sie die Prozedur `mysql.rds_set_external_master_with_auto_position` ausführen. Nachdem die Prozedur `mysql.rds_start_replication_until_gtid` die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance).

Um das `mysql.rds_rds_start_replication_until_gtid` Verfahren verwenden zu können, muss die GTID basierte Replikation aktiviert sein. Um eine bestimmte GTID basierte Transaktion zu überspringen, von der bekannt ist, dass sie zu einem Notfall führt, können Sie die gespeicherte Prozedur verwenden. Weitere Informationen zum Arbeiten mit der GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Beispiele

Wenn sie auf einer My SQL DB-Instance ausgeführt wird, konfiguriert das folgende Beispiel die DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS Es legt die minimale Replikationsverzögerung auf der My SQL DB-Instance auf eine Stunde (3.600 Sekunden) fest. Eine Änderung gegenüber der My SQL Source Database-Instance, die extern zu Amazon ausgeführt wird, wird mindestens eine Stunde lang RDS nicht auf die My SQL DB-Instance-Lesereplik angewendet.

```
call mysql.rds_set_external_master_with_auto_position(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  1,
  3600);
```

mysql.rds_set_external_source_with_auto_position (für meine Hauptversionen 8.4 und höher) RDS SQL

Konfiguriert eine RDS for My SQL DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS Bei diesem Verfahren werden auch verzögerte Replikation und Replikation auf der Grundlage globaler Transaktions-Identifikatoren () konfiguriert. GTIDs

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_source_with_auto_position (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der SQL My-Instance, die außerhalb von Amazon ausgeführt wird und die Quelldatenbank-Instance werden RDS soll.

host_port

Der Port, der von der SQL My-Instance verwendet wird, die außerhalb von Amazon ausgeführt wirdRDS, um als Quelldatenbank-Instance konfiguriert zu werden. Wenn Ihre Netzwerkkonfiguration eine Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen für die My SQL Instance, die außerhalb von Amazon ausgeführt wirdRDS. Es wird empfohlen, ein Benutzerkonto bereitzustellen, das ausschließlich für die Replikation mit der externen Instance genutzt wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `SOURCE_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quelldatenbank-Instance zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

Nutzungshinweise

Der Administrator muss das `mysql.rds_set_external_source_with_auto_position`-Verfahren ausführen. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer My SQL Instance konfiguriert werden soll, die außerhalb von Amazon RDS ausgeführt wird.

Vor der Ausführung `mysql.rds_set_external_source_with_auto_position` müssen Sie die Instance von My SQL running external RDS to Amazon als Quelldatenbank-Instance konfigurieren. Um eine Verbindung mit der SQL My-Instance herzustellen, die außerhalb

von Amazon ausgeführt wirdRDS, müssen Sie Werte für `replication_user_name` und angeben`replication_user_password`. Diese Werte müssen einen Replikationsbenutzer angeben, der über `REPLICATION SLAVE` Berechtigungen für die externe Instance von My verfügt `REPLICATION CLIENT`SQL.

Um eine externe Instanz von My SQL als Quelldatenbankinstanz zu konfigurieren

1. Stellen Sie mit dem SQL My-Client Ihrer Wahl eine Verbindung zur externen Instanz von My her SQL und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der externen Instanz von My SQL `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer `REPLICATION CLIENT` für Ihre Domäne die Berechtigungen `REPLICATION SLAVE` und `'repl_user'` für alle Datenbanken erteilt.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Weitere Informationen finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Stellen Sie vor dem Aufrufen sicher`mysql.rds_set_external_source_with_auto_position`, dass Sie die `gtid_purged` Systemvariable aufrufen, the section called

“mysql.rds_set_external_source_gtid_purged” um sie auf einen bestimmten GTID Bereich aus einer externen Quelle festzulegen.

Nach dem Aufruf `mysql.rds_set_external_source_with_auto_position` zur Konfiguration einer Amazon RDS DB-Instance als Read Replica können Sie die Read Replica aufrufenmysql.rds_start_replication, um den Replikationsprozess zu starten. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Wenn Sie anrufen`mysql.rds_set_external_source_with_auto_position`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set master` in den `mysql.rds_replication_status` Tabellen `mysql.rds_history` und auf.

Für die Notfallwiederherstellung können Sie diese Prozedur mit der gespeicherten Prozedur oder verwenden. Um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen, können Sie die Prozedur `mysql.rds_set_external_source_with_auto_position` ausführen. Nachdem die Prozedur `mysql.rds_start_replication_until_gtid` die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance).

Um das `mysql.rds_rds_start_replication_until_gtid` Verfahren verwenden zu können, muss die GTID basierte Replikation aktiviert sein. Um eine bestimmte GTID basierte Transaktion zu überspringen, von der bekannt ist, dass sie zu einem Notfall führt, können Sie die gespeicherte Prozedur verwenden. Weitere Informationen zum Arbeiten mit der GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Beispiele

Wenn sie auf einer My SQL DB-Instance ausgeführt wird, konfiguriert das folgende Beispiel die DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS Es legt die minimale Replikationsverzögerung auf der My SQL DB-Instance auf eine Stunde (3.600 Sekunden) fest. Eine Änderung gegenüber der My SQL Source Database-Instance, die extern zu Amazon ausgeführt wird, wird mindestens eine Stunde lang RDS nicht auf die My SQL DB-Instance-Lesereplik angewendet.

```
call mysql.rds_set_external_source_with_auto_position(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
```

```
  1,
  3600);
```

mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL

Konfiguriert eine RDS for My SQL DB-Instance als Read Replica einer My-Instance, die außerhalb von Amazon SQL läuft, RDS und konfiguriert die verzögerte Replikation.

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_master_with_delay(
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der SQL My-Instance, die außerhalb von Amazon ausgeführt wird und RDS die Quelldatenbank-Instance wird.

host_port

Der Port, der von der SQL My-Instance verwendet wird, die außerhalb von Amazon ausgeführt wirdRDS, um als Quelldatenbank-Instance konfiguriert zu werden. Wenn Ihre

Netzwerkkonfiguration eine SSH Portreplikation beinhaltet, bei der die Portnummer konvertiert wird, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen für die My SQL Instance, die außerhalb von Amazon ausgeführt wirdRDS. Es wird empfohlen, ein Benutzerkonto bereitzustellen, das ausschließlich für die Replikation mit der externen Instance genutzt wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

mysql_binary_log_file_name

Der Name des Binärprotokolls auf der Quelldatenbank-Instance, die die Replikationsinformationen enthält.

mysql_binary_log_file_location

Die Position innerhalb der für `mysql_binary_log_file_name` angegebenen binären Protokolldatei, ab der bei der Replikation die Replikationsinformationen gelesen werden.

Sie können den Namen und den Speicherort der Binlog-Datei ermitteln, indem Sie `SHOW MASTER STATUS` auf der Quelldatenbankinstanz starten.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `MASTER_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quelldatenbank-Instance zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_master_with_delay` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer My SQL Instance konfiguriert werden soll, die außerhalb von Amazon RDS ausgeführt wird.

Vor der Ausführung `mysql.rds_set_external_master_with_delay` müssen Sie die Instance von My SQL running external RDS to Amazon als Quelldatenbank-Instance konfigurieren. Um eine Verbindung mit der SQL My-Instance herzustellen, die außerhalb von Amazon ausgeführt wirdRDS, müssen Sie Werte für `replication_user_name` und angeben`replication_user_password`. Diese Werte müssen einen Replikationsbenutzer angeben, der über `REPLICATION SLAVE` Berechtigungen für die externe Instance von My verfügt `REPLICATION CLIENT`SQL.

Um eine externe Instanz von My SQL als Quelldatenbankinstanz zu konfigurieren

1. Stellen Sie mit dem SQL My-Client Ihrer Wahl eine Verbindung zur externen Instanz von My her SQL und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der externen Instanz von My SQL `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer `REPLICATION CLIENT` für Ihre Domäne die Berechtigungen `REPLICATION SLAVE` und `'repl_user'` für alle Datenbanken erteilt.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Weitere Informationen finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Nach dem Aufruf `mysql.rds_set_external_master_with_delay` zur Konfiguration einer Amazon RDS DB-Instance als Read Replica können Sie die Read Replica aufrufenmysql.rds_start_replication, um den Replikationsprozess zu starten. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Wenn Sie anrufen`mysql.rds_set_external_master_with_delay`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set master` in den `mysql.rds_replication_status` Tabellen `mysql.rds_history` und auf.

Für die Notfallwiederherstellung können Sie diese Prozedur mit der gespeicherten Prozedur oder verwenden. Um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen, können Sie die Prozedur `mysql.rds_set_external_master_with_delay` ausführen. Nachdem die Prozedur `mysql.rds_start_replication_until` die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance).

Um das `mysql.rds_rds_start_replication_until_gtid` Verfahren verwenden zu können, muss die GTID basierte Replikation aktiviert sein. Um eine bestimmte GTID basierte Transaktion zu überspringen, von der bekannt ist, dass sie zu einem Notfall führt, können Sie die gespeicherte Prozedur verwenden. Weitere Informationen zum Arbeiten mit der GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Das `mysql.rds_set_external_master_with_delay` Verfahren ist in diesen Versionen von RDS for My verfügbarSQL:

- Meine SQL 8.0.26 und höhere 8.0-Versionen

- Alle 5.7-Versionen

Beispiele

Wenn sie auf einer My SQL DB-Instance ausgeführt wird, konfiguriert das folgende Beispiel die DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS Es legt die minimale Replikationsverzögerung auf der My SQL DB-Instance auf eine Stunde (3.600 Sekunden) fest. Eine Änderung gegenüber der My SQL Source Database-Instance, die extern zu Amazon ausgeführt wird, wird mindestens eine Stunde lang RDS nicht auf die My SQL DB-Instance-Lesereplik angewendet.

```
call mysql.rds_set_external_master_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  1,
  3600);
```

mysql.rds_set_external_source_with_delay (für meine Hauptversionen 8.4 und höher) RDS SQL

Konfiguriert eine RDS for My SQL DB-Instance als Read Replica einer My-Instance, die außerhalb von Amazon SQL läuft, RDS und konfiguriert die verzögerte Replikation.

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf `1`. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_source_with_delay (
  host_name
  , host_port
```

```
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der SQL My-Instance, die außerhalb von Amazon ausgeführt wird und RDS die Quelldatenbank-Instance wird.

host_port

Der Port, der von der SQL My-Instance verwendet wird, die extern zu Amazon läuftRDS, um als Quelldatenbank-Instance konfiguriert zu werden. Wenn Ihre Netzwerkkonfiguration eine SSH Portreplikation beinhaltet, bei der die Portnummer konvertiert wird, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen für die My SQL Instance, die außerhalb von Amazon ausgeführt wirdRDS. Es wird empfohlen, ein Benutzerkonto bereitzustellen, das ausschließlich für die Replikation mit der externen Instance genutzt wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

mysql_binary_log_file_name

Der Name des Binärprotokolls auf der Quelldatenbank-Instance, die die Replikationsinformationen enthält.

mysql_binary_log_file_location

Die Position innerhalb der für `mysql_binary_log_file_name` angegebenen binären Protokolldatei, ab der bei der Replikation die Replikationsinformationen gelesen werden.

Sie können den Namen und den Speicherort der Binlog-Datei ermitteln, indem Sie `SHOW MASTER STATUS` auf der Quelldatenbankinstanz starten.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `SOURCE_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quelldatenbank-Instance zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

Nutzungshinweise

Der Administrator muss das `mysql.rds_set_external_source_with_delay`-Verfahren ausführen. Dieses Verfahren muss auf der My SQL DB-Instance ausgeführt werden, die als Read Replica einer My SQL Instance konfiguriert werden soll, die außerhalb von Amazon RDS ausgeführt wird.

Vor der Ausführung `mysql.rds_set_external_source_with_delay` müssen Sie die Instance von My SQL running external RDS to Amazon als Quelldatenbank-Instance konfigurieren. Um eine Verbindung mit der SQL My-Instance herzustellen, die außerhalb von Amazon ausgeführt wirdRDS, müssen Sie Werte für `replication_user_name` und angeben`replication_user_password`. Diese Werte müssen einen Replikationsbenutzer angeben, der über `REPLICATION SLAVE` Berechtigungen für die externe Instance von My verfügt `REPLICATION CLIENT`SQL.

Um eine externe Instanz von My SQL als Quelldatenbankinstanz zu konfigurieren

1. Stellen Sie mit dem SQL My-Client Ihrer Wahl eine Verbindung zur externen Instanz von My her SQL und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der externen Instanz von My SQL `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer `REPLICATION CLIENT` für Ihre Domäne die Berechtigungen `REPLICATION SLAVE` und `'repl_user'` für alle Datenbanken erteilt.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
IDENTIFIED BY 'SomePassW0rd'
```

Weitere Informationen finden Sie unter Konfigurieren der Replikation der Binärprotokolldateiposition mit einer externen Quell-Instance.

> **Note**
>
> Wir empfehlen, wenn möglich, Read Replicas zu verwenden, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu verwalten. In diesem Fall sollten Sie nur diese und andere replikationsbezogene gespeicherte Prozeduren verwenden. Diese Methoden ermöglichen komplexere Replikationstopologien zwischen RDS Amazon-DB-Instances. Wir bieten diese gespeicherten Prozeduren hauptsächlich an, um die Replikation mit SQL My-Instances zu ermöglichen, die außerhalb von Amazon ausgeführt RDS werden. Informationen zur Verwaltung der Replikation zwischen RDS Amazon-DB-Instances finden Sie unterArbeiten mit DB-Instance-Lesereplikaten.

Nach dem Aufruf `mysql.rds_set_external_source_with_delay` zur Konfiguration einer Amazon RDS DB-Instance als Read Replica können Sie die Read Replica aufrufenmysql.rds_start_replication, um den Replikationsprozess zu starten. Zudem haben Sie die Möglichkeit, mit einem Aufruf von die Lesereplikat-Konfiguration zu entfernen.

Wenn Sie anrufen`mysql.rds_set_external_source_with_delay`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set master` in den `mysql.rds_replication_status` Tabellen `mysql.rds_history` und auf.

Für die Notfallwiederherstellung können Sie diese Prozedur mit der gespeicherten Prozedur oder verwenden. Um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen, können Sie die Prozedur

`mysql.rds_set_external_source_with_delay` ausführen. Nachdem die Prozedur `mysql.rds_start_replication_until` die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance).

Um das `mysql.rds_rds_start_replication_until_gtid` Verfahren verwenden zu können, muss die GTID basierte Replikation aktiviert sein. Um eine bestimmte GTID basierte Transaktion zu überspringen, von der bekannt ist, dass sie zu einem Notfall führt, können Sie die gespeicherte Prozedur verwenden. Weitere Informationen zum Arbeiten mit der GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Beispiele

Wenn sie auf einer My SQL DB-Instance ausgeführt wird, konfiguriert das folgende Beispiel die DB-Instance als Read Replica einer Instance von My, die außerhalb von SQL Amazon läuft. RDS Es legt die minimale Replikationsverzögerung auf der My SQL DB-Instance auf eine Stunde (3.600 Sekunden) fest. Eine Änderung gegenüber der My SQL Source Database-Instance, die extern zu Amazon ausgeführt wird, wird mindestens eine Stunde lang RDS nicht auf die My SQL DB-Instance-Lesereplik angewendet.

```
call mysql.rds_set_external_source_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  1,
  3600);
```

mysql.rds_set_external_source_gtid_purged

Setzt die Systemvariable gtid_purged auf einen angegebenen Bereich aus einer externen Quelle. GTID Der `gtid_purged` Wert ist für die Konfiguration der GTID basierten Replikation erforderlich, um die Replikation mithilfe der auto Positionierung wieder aufzunehmen.

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von

Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_source_gtid_purged(
  server_uuid
  , start_pos
  , end_pos
);
```

Parameter

server_uuid

Die universell eindeutige Kennung (UUID) des externen Servers, von dem der GTID Bereich importiert wird.

start_pos

Die Startposition des GTID festzulegenden Bereichs.

end_pos

Die Endposition des festzulegenden GTID Bereichs.

Nutzungshinweise

Das `mysql.rds_set_external_source_gtid_purged` Verfahren ist nur in den Versionen My SQL 8.0.37 und höher 8.0 verfügbar.

Rufen Sie an, `mysql.rds_set_external_source_gtid_purged` bevor Sie anrufenmysql.rds_set_external_master_with_auto_position (für meine Hauptversionen 8.0 und niedriger) RDS SQL,mysql.rds_set_external_source_with_auto_position (für meine Hauptversionen 8.4 und höher) RDS SQL, oder. mysql.rds_set_external_source_with_auto_position_for_channel

Stellen Sie vor dem Anruf sicher`mysql.rds_set_external_source_gtid_purged`, dass alle aktiven Replikationskanäle für die Datenbank gestoppt sind. Um den Status eines Kanals zu überprüfen, verwenden Sie die SQL Anweisung `SHOW REPLICA STATUS` My. Um die Replikation auf einem Kanal zu beenden, rufen Sie anthe section called “mysql.rds_stop_replication_for_channel”.

Der GTID Bereich, den Sie angeben, muss eine Obermenge des vorhandenen `GTID_PURGED` Werts sein. Diese gespeicherte Prozedur überprüft die folgenden Werte, bevor sie den `GTID_PURGED` Wert festlegt:

- Das `server_uuid` ist gültig.
- Der Wert von `start_pos` ist größer als `0` und kleiner als der Wert von`end_pos`.
- Der Wert von `end_pos` ist größer oder gleich dem Wert von`start_pos`.

Wenn der GTID Satz auf Ihrem externen Server mehrere Wertebereiche enthält, sollten Sie erwägen, die Prozedur mehrmals mit unterschiedlichen GTID festgelegten Werten aufzurufen.

Wenn Sie anrufen`mysql.rds_set_external_source_gtid_purged`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set gtid_purged` in der `mysql.rds_history` Tabelle auf.

Wenn Sie den `gtid_purged` Wert für das Backup, das Sie für die Replikation verwenden, nicht angemessen festlegen, kann dies dazu führen, dass während des Replikationsvorgangs Transaktionen fehlen oder doppelt vorhanden sind. Führen Sie die folgenden Schritte aus, um den richtigen `gtid_purged` Wert festzulegen.

Um den gtid_purged-Wert für das Replikat festzulegen

1. Ermitteln Sie den Zeitpunkt oder die spezifische Sicherungsdatei, die als Ausgangspunkt für die Replikation verwendet werden soll. Dies kann ein logisches Backup (eine Mysqldump-Datei) oder ein physisches Backup (ein RDS Amazon-Snapshot) sein.
2. Ermitteln Sie den Wert. `gtid_executed` Dieser Wert stellt die Menge aller Daten darGTIDs, die auf dem Server festgeschrieben wurden. Gehen Sie auf der Quellinstanz wie folgt vor, um diesen Wert abzurufen:
   - Führen Sie die SQL Anweisung `SELECT @@GLOBAL.GTID_EXECUTED;` zu dem Zeitpunkt aus, zu dem das Backup erstellt wurde.
   - Wenn das jeweilige Backup-Hilfsprogramm verwandte Optionen enthält, extrahieren Sie den Wert aus der Backup-Datei. Weitere Informationen finden Sie unter Meine SQL Dokumentation zu dieser set-gtid-purgedOption.
3. Ermitteln Sie den `gtid_purged` Wert, der für den Aufruf von verwendet werden sollmysql.rds_set_external_source_gtid_purged. Der `gtid_purged` Wert sollte alle Elemente enthaltenGTIDs, die auf der Quellinstanz ausgeführt wurden und nicht mehr

für die Replikation benötigt werden. Daher sollte der `gtid_purged` Wert eine Teilmenge des `gtid_executed` Werts sein, den Sie im vorherigen Schritt abgerufen haben.

Um den `gtid_purged` Wert zu ermitteln, identifizieren Sie diejenigen, GTIDs die nicht in der Sicherung enthalten sind und nicht mehr für die Replikation benötigt werden. Sie können dies tun, indem Sie die Binärprotokolle analysieren oder ein Tool wie mysqlbinlog verwenden, um diejenigen zu findenGTIDs, die aus den Binärprotokollen gelöscht wurden.

Wenn Sie über ein konsistentes Backup verfügen, das alle Binärprotokolle bis zum Backup-Punkt umfasst, können Sie den Wert alternativ so einstellen, dass er dem `gtid_purged` `gtid_executed` Wert am Backup-Punkt entspricht.

4. Nachdem Sie den entsprechenden `gtid_purged` Wert ermittelt haben, der mit Ihrem Backup konsistent ist, rufen Sie die `mysql.rds_set_external_source_gtid_purged` gespeicherte Prozedur auf Ihrer RDS for My SQL DB-Instance auf, um den Wert festzulegen.

Beispiele

Bei der Ausführung auf einer My SQL DB-Instance legt das folgende Beispiel den GTID Bereich von einem externen SQL My-Server mit der UUID `12345678-abcd-1234-efgh-123456789abc` Startposition und der 1 Endposition von fest`100`. Der resultierende GTID Wert ist auf gesetzt`+12345678-abcd-1234-efgh-123456789abc:1-100`.

```
CALL mysql.rds_set_external_source_gtid_purged('12345678-abcd-1234-efgh-123456789abc',
 1, 100);
```

Legt fest, dass der Replikationsmodus entweder auf binären Protokolldateipositionen oder auf globalen Transaktions-Identifikatoren (GTIDs) basiert.

Syntax

```
CALL mysql.rds_set_master_auto_position (
auto_position_mode
);
```

Parameter

auto_position_mode

Ein Wert, der angibt, ob die Positionsreplikation der Protokolldatei oder die GTID basierte Replikation verwendet werden soll:

- `0` – Verwendung der auf der Binärprotokolldateiposition basierenden Replikationsmethode. Der Standardwert ist `0`.
- 1— Verwenden Sie die GTID basierte Replikationsmethode.

Nutzungshinweise

Die Prozedur `mysql.rds_set_master_auto_position` muss vom Hauptbenutzer ausgeführt werden.

Dieses Verfahren wird RDS für alle Versionen von My SQL 5.7 und RDS für My SQL 8.0.26 und höher 8.0 unterstützt.

Legt fest, dass der Replikationsmodus entweder auf binären Protokolldateipositionen oder auf globalen Transaktions-Identifikatoren () GTIDs basiert.

Syntax

```
CALL mysql.rds_set_source_auto_position (auto_position_mode);
```

Parameter

auto_position_mode

Ein Wert, der angibt, ob die Positionsreplikation der Protokolldatei oder die GTID basierte Replikation verwendet werden soll:

- `0` – Verwendung der auf der Binärprotokolldateiposition basierenden Replikationsmethode. Der Standardwert ist `0`.
- 1— Verwenden Sie die GTID basierte Replikationsmethode.

Nutzungshinweise

Der Administrator muss das `mysql.rds_set_source_auto_position`-Verfahren ausführen.

mysql.rds_set_source_delay

Legt die Mindestanzahl von Sekunden fest, in der die Replikation von der Quelldatenbankinstance auf das aktuelle Lesereplikat verzögert werden soll. Verwenden Sie diese Prozedur, wenn Sie mit einem Lesereplikat verbunden sind, um die Replikation von der zugehörigen Quelldatenbankinstance zu verzögern.

Syntax

```
CALL mysql.rds_set_source_delay(
delay
);
```

Parameter

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quelldatenbank-Instance zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

Nutzungshinweise

Die Prozedur `mysql.rds_set_source_delay` muss vom Hauptbenutzer ausgeführt werden.

Für die Notfallwiederherstellung können Sie diese Prozedur mit der gespeicherten Prozedur oder verwenden. Um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen, können Sie die Prozedur `mysql.rds_set_source_delay` ausführen. Nachdem die Prozedur `mysql.rds_start_replication_until` oder `mysql.rds_start_replication_until_gtid` die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance).

Um das `mysql.rds_rds_start_replication_until_gtid` Verfahren verwenden zu können, muss die GTID basierte Replikation aktiviert sein. Um eine bestimmte GTID basierte Transaktion zu überspringen, von der bekannt ist, dass sie zu einem Notfall führt, können Sie die gespeicherte Prozedur verwenden. Weitere Informationen zur GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Das `mysql.rds_set_source_delay` Verfahren ist in diesen Versionen von RDS for My verfügbarSQL:

- Alles RDS für meine SQL 8.4-Versionen
- Meine SQL 8.0.26 und höhere 8.0-Versionen
- Alle 5.7-Versionen

Beispiele

Um die Replikation von der Quelldatenbankinstance zum aktuellen Lesereplikat um mindestens eine Stunde (3.600 Sekunden) zu verzögern, können Sie `mysql.rds_set_source_delay` mit dem folgenden Parameter aufrufen:

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_skip_repl_error

Überspringt und löscht einen Replikationsfehler in einer My DB-Read Replica. SQL

Syntax

```
CALL mysql.rds_skip_repl_error;
```

Nutzungshinweise

Der Hauptbenutzer muss die Prozedur `mysql.rds_skip_repl_error` auf einem Lesereplikat ausführen. Weitere Informationen zu dieser Prozedur finden Sie unter Aufrufen der Prozedur mysql.rds_skip_repl_error.

Führen Sie den Befehl My aus, um festzustellen, ob Fehler vorliegen. SQL `SHOW REPLICA STATUS\G` Wenn ein Replikationsfehler nicht als kritisch eingestuft, ist, können Sie `mysql.rds_skip_repl_error` ausführen, um den Fehler zu überspringen. Wenn mehrere Fehler aufgetreten sind, löscht `mysql.rds_skip_repl_error` den ersten Fehler und weist darauf hin, dass weitere Fehlermeldungen anhängig sind. In diesem Fall können Sie mithilfe von `SHOW REPLICA STATUS\G` die angemessene Vorgehensweise bei der Handhabung des nächsten Fehlers ermitteln. Informationen zu den zurückgegebenen Werten finden Sie in der SHOWREPLICASTATUSErklärung unter Meine SQL Dokumentation.

Weitere Informationen zur Behebung von Replikationsfehlern mit Amazon RDS finden Sie unterBehebung eines My SQL Read Replica-Problems.

Fehler „Replication stopped (Replikation gestoppt)“

Wenn Sie die Prozedur `mysql.rds_skip_repl_error` aufrufen, wird möglicherweise eine Fehlermeldung angezeigt, die besagt, dass das Replikat ausgefallen oder deaktiviert ist.

Diese Fehlermeldung wird angezeigt, wenn Sie die Prozedur auf der primären Instance statt auf dem Lesereplikat ausführen. Sie müssen diese Prozedur auf dem Lesereplikat ausführen, damit sie funktioniert.

Diese Fehlermeldung wird möglicherweise auch angezeigt, wenn Sie die Prozedur zwar auf dem Lesereplikat ausführen, die Replikation jedoch nicht neu gestartet werden kann.

Wenn Sie eine größere Anzahl von Fehlern überspringen müssen, kann die Dauer der Replikationsverzögerung den standardmäßigen Aufbewahrungszeitraum für binäre Protokolldateien (binlog) überschreiten. In diesem Fall kann ein schwerwiegender Fehler auftreten, da Binlogdateien gelöscht werden, bevor sie auf der Read Replica wiedergegeben wurden. Diese Bereinigung führt zur Beendigung der Replikation, und Sie können den Befehl `mysql.rds_skip_repl_error` nicht mehr aufrufen, um Replikationsfehler zu überspringen und zu ignorieren.

Sie können dieses Problem verringern, indem Sie die Anzahl der Stunden erhöhen, die die Binärprotokolldateien auf Ihrer Quelldatenbankinstance aufbewahrt werden. Nachdem Sie die Aufbewahrungsdauer für binäre Protokolldateien verlängert haben, können Sie die Replikation neu starten und nach Bedarf den Befehl `mysql.rds_skip_repl_error` aufrufen.

Verwenden Sie zur Festlegung der Aufbewahrungszeit für Binärprotokolldateien die Prozedur mysql.rds_set_configuration und legen Sie einen Konfigurationsparameter für `'binlog retention hours'` zusammen mit der Stundenanzahl für die Aufbewahrung der Binärprotokolldateien im DB-Cluster fest. Beim folgenden Beispiel wird die Aufbewahrungszeit für binäre Protokolle auf 48 Stunden festgelegt.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_start_replication

Initiiert die Replikation von einem aus.

> **Note**
>
> Sie können die gespeicherte Prozedur oder verwenden, um die Replikation von einer RDSMy SQL DB-Instance aus zu initiieren und die Replikation am angegebenen Speicherort der binären Protokolldatei zu beenden.

Syntax

```
CALL mysql.rds_start_replication;
```

Nutzungshinweise

Die Prozedur `mysql.rds_start_replication` muss vom Hauptbenutzer ausgeführt werden.

Um Daten von einer Instance von My SQL external nach Amazon zu importierenRDS, rufen Sie die Read Replica `mysql.rds_start_replication` auf, um den Replikationsprozess nach dem Aufruf zu starten oder die Replikationskonfiguration zu erstellen. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance.

Um Daten in eine Instance von My SQL external to Amazon zu exportierenRDS, rufen Sie `mysql.rds_start_replication` und `mysql.rds_stop_replication` auf der Read Replica auf, um einige Replikationsaktionen zu steuern, z. B. das Löschen von Binärprotokollen. Weitere Informationen finden Sie unter Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation.

Darüber hinaus können Sie `mysql.rds_start_replication` für das Lesereplikat aufrufen, um einen zuvor durch einen Aufruf von `mysql.rds_stop_replication` gestoppten Replikationsprozess wieder zu starten. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Initiiert die Replikation von einem und stoppt die Replikation am angegebenen Speicherort der binären Protokolldatei.

Syntax

```
CALL mysql.rds_start_replication_until (
```

```
replication_log_file
  , replication_stop_point
);
```

Parameter

replication_log_file

Der Name des Binärprotokolls auf der Quelldatenbank-Instance, die die Replikationsinformationen enthält.

replication_stop_point

Die Position im `replication_log_file`-Binärprotokoll, an der die Replikation stoppt.

Nutzungshinweise

Die Prozedur `mysql.rds_start_replication_until` muss vom Hauptbenutzer ausgeführt werden.

Das `mysql.rds_start_replication_until` Verfahren ist in diesen Versionen von RDS for My SQL verfügbar:

- Alles RDS für meine SQL 8.4-Versionen
- Meine SQL 8.0.26 und höhere 8.0-Versionen
- Alle 5.7-Versionen

Sie können diese Prozedur mit verzögerter Replikation für die Notfallwiederherstellung verwenden. Wenn Sie die verzögerte Replikation konfiguriert haben, können Sie diese Prozedur verwenden, um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen. Nachdem diese Prozedur die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Sie können die verzögerte Replikation mit den folgenden gespeicherten Prozeduren konfigurieren:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL

- mysql.rds_set_external_source_with_delay (für meine Hauptversionen 8.4 und höher) RDS SQL
- mysql.rds_set_source_delay

Der für den Parameter `replication_log_file` angegebene Dateiname muss mit dem Binlogdateinamen der Quelldatenbankinstance übereinstimmen.

Wenn der Parameter `replication_stop_pointe`ine Stoppposition angibt, die in der Vergangenheit liegt, wird die Replikation sofort gestoppt.

Beispiele

Das folgende Beispiel initiiert die Replikation und repliziert die Änderungen, bis die Position `120` in der Binärprotokolldatei `mysql-bin-changelog.000777` erreicht wird.

```
call mysql.rds_start_replication_until(
  'mysql-bin-changelog.000777',
  120);
```

mysql.rds_stop_replication

Stoppt die Replikation von einer My SQL DB-Instance aus.

Syntax

```
CALL mysql.rds_stop_replication;
```

Nutzungshinweise

Die Prozedur `mysql.rds_stop_replication` muss vom Hauptbenutzer ausgeführt werden.

Wenn Sie die Replikation so konfigurieren, dass Daten aus einer Instance von My importiert werden, die extern zu Amazon SQL läuftRDS, rufen Sie die Read Replica `mysql.rds_stop_replication` auf, um den Replikationsvorgang nach Abschluss des Imports zu beenden. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance.

Wenn Sie die Replikation so konfigurieren, dass Daten in eine Instance von My SQL external to Amazon exportiert werdenRDS, rufen Sie `mysql.rds_start_replication` und `mysql.rds_stop_replication` auf der Read Replica auf, um einige Replikationsaktionen zu

steuern, z. B. das Löschen von Binärprotokollen. Weitere Informationen finden Sie unter Exportieren von Daten aus einer My SQL DB-Instance mithilfe von Replikation.

Sie können es auch verwenden`mysql.rds_stop_replication`, um die Replikation zwischen zwei RDS Amazon-DB-Instances zu beenden. In der Regel wird eine Replikation gestoppt, um eine länger dauernde Operation im Lesereplikat – zum Beispiel das Erstellen eines umfangreichen Indexes – durchzuführen. Ein gestoppter Replikationsvorgang kann durch Aufruf von mysql.rds_start_replication für das Lesereplikat wieder gestartet werden. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Beenden einer Sitzung oder Abfrage

Die folgenden gespeicherten Prozeduren beenden eine Sitzung oder Abfrage.

Themen

- mysql.rds_kill
- mysql.rds_kill_query

mysql.rds_kill

Beendet eine Verbindung zum MySQL-Server.

Syntax

```
CALL mysql.rds_kill(processID);
```

Parameter

processID

Die ID des Verbindungs-Threads, der beendet werden soll.

Nutzungshinweise

Jede Verbindung zum MySQL-Server wird in einem eigenen Thread ausgeführt. Um eine Verbindung zu beenden, verwenden Sie die Prozedur `mysql.rds_kill` und übergeben ihr als Parameter die Thread-ID der Verbindung. Die Thread-ID erhalten Sie mithilfe des MySQL-Befehls SHOW PROCESSLIST.

Weitere Informationen zu Einschränkungen finden Sie unter Einschränkungen bei gespeicherten MySQL-Prozeduren.

Beispiele

Im folgenden Beispiel wird eine Verbindung mit der Thread-ID 4243 beendet:

```
CALL mysql.rds_kill(4243);
```

mysql.rds_kill_query

Beendet eine an den MySQL-Server übermittelte Abfrage.

Syntax

```
CALL mysql.rds_kill_query(processID);
```

Parameter

processID

Die Identität des Prozesses oder Threads, der die zu beendende Abfrage ausführt.

Nutzungshinweise

Um eine an den MySQL-Server übermittelte Abfrage zu beenden, verwenden Sie die Prozedur `mysql_rds_kill_query` und übergeben die ID des Threads, der die Abfrage ausführt. Die Prozedur beendet dann die Verbindung.

Die Abfrage-ID erhalten Sie mithilfe der MySQL-Tabelle INFORMATION_SCHEMA PROCESSLIST oder des MySQL-Befehls SHOW PROCESSLIST. Der Wert in der ID-Spalte von `SHOW PROCESSLIST` oder `SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST` ist die *processID*.

Weitere Informationen zu Einschränkungen finden Sie unter Einschränkungen bei gespeicherten MySQL-Prozeduren.

Beispiele

Im folgenden Beispiel wird eine Abfrage mit der Thread-ID 230040 beendet:

```
CALL mysql.rds_kill_query(230040);
```

Verwaltung von aktiv-aktiven Clustern

Mit den folgenden gespeicherten Prozeduren werden Meine SQL aktiv-aktiven Cluster eingerichtet und verwaltetRDS. Weitere Informationen finden Sie unter the section called “Konfiguration von aktiv-aktiven Clustern”.

Diese gespeicherten Prozeduren sind nur RDS für My SQL DB-Instances verfügbar, auf denen die folgenden Versionen ausgeführt werden:

- Alle Meine SQL 8.4-Versionen
- Meine SQL Nebenversionen 8.0.35 und höher

Themen

- mysql.rds_group_replication_advance_gtid
- mysql.rds_group_replication_create_user
- mysql.rds_group_replication_set_recovery_channel
- mysql.rds_group_replication_start
- mysql.rds_group_replication_stop

mysql.rds_group_replication_advance_gtid

Erzeugt einen Platzhalter GTIDs auf der aktuellen DB-Instance.

Syntax

```
CALL mysql.rds_group_replication_advance_gtid(
begin_id
, end_id
, server_uuid
);
```

Parameter

begin_id

Die zu erstellende Start-Transaktions-ID.

end_id

Die zu erstellende End-Transaktions-ID.

begin_id

Die `group_replication_group_name` für die zu erstellende Transaktion. Der `group_replication_group_name` ist UUID in der DB-Parametergruppe, die der DB-Instance zugeordnet ist, als a angegeben.

Nutzungshinweise

In einem aktiv-aktiven Cluster müssen alle GTID Transaktionen, die auf der neuen DB-Instance ausgeführt werden, auf den anderen Mitgliedern des Clusters existieren, damit eine DB-Instance einer Gruppe beitreten kann. In ungewöhnlichen Fällen kann eine neue DB-Instance mehr Transaktionen haben, wenn Transaktionen ausgeführt werden, bevor die Instance der Gruppe hinzugefügt wird. In diesem Fall können Sie keine vorhandenen Transaktionen entfernen, aber Sie können dieses Verfahren verwenden, um den entsprechenden Platzhalter GTIDs auf den anderen DB-Instances in der Gruppe zu erstellen. Stellen Sie zuvor sicher, dass sich die Transaktionen nicht auf die replizierten Daten auswirken.

Wenn Sie diese Prozedur aufrufen, `server_uuid:begin_id-end_id` werden GTID Transaktionen von mit leerem Inhalt erstellt. Um Replikationsprobleme zu vermeiden, sollten Sie dieses Verfahren unter keinen anderen Bedingungen anwenden.

> **Important**
>
> Vermeiden Sie es, diese Prozedur aufzurufen, wenn der Active-Active-Cluster normal funktioniert. Rufen Sie dieses Verfahren nur auf, wenn Sie die möglichen Konsequenzen der von Ihnen erstellten Transaktionen kennen. Das Aufrufen dieser Prozedur kann zu inkonsistenten Daten führen.

Beispiel

Im folgenden Beispiel wird ein Platzhalter in GTIDs der aktuellen DB-Instance erstellt. :

```
CALL mysql.rds_group_replication_advance_gtid(5, 6,
 '11111111-2222-3333-4444-555555555555');
```

mysql.rds_group_replication_create_user

Erstellt den Replikationsbenutzer `rdsgrprepladmin` für die Gruppenreplikation auf der DB-Instance.

Syntax

```
CALL mysql.rds_group_replication_create_user(
replication_user_password
);
```

Parameter

`replication_user_password`

Das Passwort des Replikationsbenutzers`rdsgrprepladmin`.

Nutzungshinweise

- Das Passwort des `rdsgrprepladmin` Replikationsbenutzers muss auf allen DB-Instances in einem aktiv-aktiven Cluster identisch sein.
- Der `rdsgrprepladmin` Benutzername ist für Gruppenreplikationsverbindungen reserviert. Kein anderer Benutzer, einschließlich des Masterbenutzers, kann diesen Benutzernamen haben.

Beispiel

Im folgenden Beispiel wird der Replikationsbenutzer `rdsgrprepladmin` für die Gruppenreplikation auf der DB-Instance erstellt:

```
CALL mysql.rds_group_replication_create_user('password');
```

mysql.rds_group_replication_set_recovery_channel

Legt den `group_replication_recovery` Kanal für einen aktiv-aktiven Cluster fest. Das Verfahren verwendet den reservierten `rdsgrprepladmin` Benutzer, um den Kanal zu konfigurieren.

Syntax

```
CALL mysql.rds_group_replication_set_recovery_channel(
```

```
replication_user_password);
```

Parameter

replication_user_password

Das Passwort des Replikationsbenutzers`rdsgrprepladmin`.

Nutzungshinweise

Das Passwort des `rdsgrprepladmin` Replikationsbenutzers muss auf allen DB-Instances in einem aktiv-aktiven Cluster identisch sein. Ein Aufruf von `mysql.rds_group_replication_create_user` gibt das Passwort an.

Beispiel

Im folgenden Beispiel wird der `group_replication_recovery` Kanal für einen Active-Active-Cluster festgelegt:

```
CALL mysql.rds_group_replication_set_recovery_channel('password');
```

mysql.rds_group_replication_start

Startet die Gruppenreplikation auf der aktuellen DB-Instance.

Syntax

```
CALL mysql.rds_group_replication_start(
bootstrap
);
```

Parameter

bootstrap

Ein Wert, der angibt, ob eine neue Gruppe initialisiert oder einer vorhandenen Gruppe beigetreten werden soll. `1`initialisiert eine neue Gruppe mit der aktuellen DB-Instance. `0`verbindet die aktuelle DB-Instance mit einer vorhandenen Gruppe, indem eine Verbindung zu den Endpunkten hergestellt wird, die im `group_replication_group_seeds` Parameter in der DB-Parametergruppe definiert sind, die der DB-Instance zugeordnet ist.

Beispiel

Das folgende Beispiel initialisiert eine neue Gruppe mit der aktuellen DB-Instance:

```
CALL mysql.rds_group_replication_start(1);
```

mysql.rds_group_replication_stop

Stoppt die Gruppenreplikation auf der aktuellen DB-Instance.

Syntax

```
CALL mysql.rds_group_replication_stop();
```

Nutzungshinweise

Wenn Sie die Replikation auf einer DB-Instance beenden, hat dies keine Auswirkungen auf andere DB-Instances im Active-Active-Cluster.

Verwaltung der Replikation mit mehreren Quellen

Mit den folgenden gespeicherten Prozeduren werden Replikationskanäle auf einem SQL Multiquellen-Replikat RDS für My eingerichtet und verwaltet. Weitere Informationen finden Sie unter the section called “Konfiguration der Replikation mit mehreren Quellen”.

Diese gespeicherten Prozeduren sind nur RDS für My SQL DB-Instances verfügbar, auf denen die folgenden Engine-Versionen ausgeführt werden:

- Alle 8.4-Versionen
- Nebenversionen 8.0.35 und höher
- Nebenversionen 5.7.44 und höher

Wenn Sie gespeicherte Prozeduren verwenden, um die Replikation mit einem Replikationsbenutzer zu verwalten`caching_sha2_passwword`, müssen Sie bei der Konfiguration Folgendes angeben`TLS: SOURCE_SSL=1` `caching_sha2_password`ist das Standard-Authentifizierungs-Plugin RDS für My SQL 8.4.

> **Note**
>
> Obwohl sich diese Dokumentation auf Quell-DB-Instances wie RDS auf My SQL DB-Instances bezieht, funktionieren diese Verfahren auch für My SQL Instances, die außerhalb von Amazon ausgeführt RDS werden.

Themen

- mysql.rds_next_source_log_for_channel
- mysql.rds_reset_external_source_for_channel
- mysql.rds_set_external_source_for_channel
- mysql.rds_set_external_source_with_auto_position_for_channel
- mysql.rds_set_external_source_with_delay_for_channel
- mysql.rds_set_source_auto_position_for_channel
- mysql.rds_set_source_delay_for_channel
- mysql.rds_skip_repl_error_for_channel
- mysql.rds_start_replication_for_channel

- mysql.rds_start_replication_until_for_channel
- mysql.rds_start_replication_until_gtid_for_channel
- mysql.rds_stop_replication_for_channel

mysql.rds_next_source_log_for_channel

Ändert die Protokollposition der Quell-DB-Instance auf den Anfang des nächsten Binärprotokolls auf der Quell-DB-Instance für den Kanal. Verwenden Sie dieses Verfahren nur, wenn Sie den Replikations-I/O-Fehler 1236 auf einem Replikat mit mehreren Quellen erhalten.

Syntax

```
CALL mysql.rds_next_source_log_for_channel(
curr_master_log,
channel_name
);
```

Parameter

curr_master_log

Der Index der aktuellen Quell-Protokolldatei. Der Index ist im Dateinamen codiert. Eine aktuelle Datei mit dem Namen `mysql-bin-changelog.012345` hat beispielsweise den Index 12345. Um den Namen der aktuellen Quell-Protokolldatei zu ermitteln, führen Sie den Befehl `SHOW REPLICA STATUS FOR CHANNEL '`*channel_name*`'` aus. Sie finden den Namen anschließend im Feld `Source_Log_File`.

channel_name

Der Name des Replikationskanals auf dem Multiquellenreplikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_next_source_log_for_channel` muss vom Hauptbenutzer ausgeführt werden. Wenn beispielsweise ein IO_Thread-Fehler auftritt, können Sie dieses Verfahren verwenden, um alle Ereignisse in der aktuellen Binärprotokolldatei zu überspringen und die Replikation mit der nächsten binären Protokolldatei für den in angegebenen Kanal fortzusetzen. `channel_name`

Beispiel

Gehen Sie davon aus, dass die Replikation auf einem Kanal auf einem Replikat mit mehreren Quellen fehlschlägt. Die Ausführung `SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G` auf dem Multiquellen-Replikat gibt das folgende Ergebnis zurück:

```
mysql> SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G
*************************** 1. row ***************************
             Replica_IO_State: Waiting for source to send event
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: ReplicationUser
                  Source_Port: 3306
                Connect_Retry: 60
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: replica-relay-bin.000003
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:.
              .
              .
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
               .
               .
                 Channel_name: channel_1
              .
              .
 -- Some fields are omitted in this example output
```

Den Angeben im Feld `Last_IO_Errno` ist zu entnehmen, dass die Instance eine I/O-Fehlermeldung mit der Nummer 1236 erhalten hat. Dem Feld `Source_Log_File` ist zudem zu entnehmen, dass die betroffene Protokolldatei den Namen `mysql-bin-changelog.012345`

aufweist und ihr Index folglich 12345 lautet. Um den Fehler zu beheben, können Sie `mysql.rds_next_source_log_for_channel` mit den folgenden Parametern aufrufen:

```
CALL mysql.rds_next_source_log_for_channel(12345,'channel_1');
```

mysql.rds_reset_external_source_for_channel

Stoppt den Replikationsvorgang auf dem angegebenen Kanal und entfernt den Kanal und die zugehörigen Konfigurationen aus dem Multiquellenreplikat.

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_reset_external_source_for_channel (channel_name);
```

Parameter

channel_name

Der Name des Replikationskanals auf dem Multiquellenreplikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_reset_external_source_for_channel` muss vom Hauptbenutzer ausgeführt werden. Bei diesem Verfahren werden alle Relay-Logs gelöscht, die zu dem zu entfernenden Kanal gehören.

mysql.rds_set_external_source_for_channel

Konfiguriert einen Replikationskanal auf einer RDS for My SQL DB-Instance, um die Daten von einer anderen RDS für My DB-Instance zu replizieren. SQL

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

> **Note**
>
> Sie können stattdessen die the section called "mysql.rds_set_external_source_with_delay_for_channel" gespeicherte Prozedur verwenden, um diesen Kanal mit verzögerter Replikation zu konfigurieren.

Syntax

```
CALL mysql.rds_set_external_source_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , channel_name
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der RDS für Meine SQL Quell-DB-Instance.

host_port

Der Port, der von der RDS for My SQL Source DB-Instance verwendet wird. Wenn Ihre Netzwerkkonfiguration die Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen RDS für die „Meine SQL Quell-DB-Instance“. Wir empfehlen, dass Sie ein Konto angeben, das ausschließlich für die Replikation mit der Quell-DB-Instance verwendet wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

mysql_binary_log_file_name

Der Name des Binärprotokolls auf der Quell-DB-Instance, das die Replikationsinformationen enthält.

mysql_binary_log_file_location

Die Position in der binären Protokolldatei `mysql_binary_log_file_name`, ab der bei der Replikation die Replikationsinformationen gelesen werden.

Sie können den Namen und den Speicherort der Binlog-Datei ermitteln, indem Sie sie `SHOW BINARY LOG STATUS` auf der Quell-DB-Instance ausführen.

> **Note**
>
> Frühere Versionen von My SQL wurden `SHOW MASTER STATUS` anstelle von `SHOW BINARY LOG STATUS` verwendet. Wenn Sie eine SQL Version von My vor 8.4 verwenden, verwenden Sie`SHOW MASTER STATUS`.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `SOURCE_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

`channel_name`

Der Name des Replikationskanals. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_source_for_channel` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf dem Ziel RDS für My SQL DB-Instance ausgeführt werden, auf dem Sie den Replikationskanal erstellen.

Konfigurieren Sie vor der Ausführung `mysql.rds_set_external_source_for_channel` einen Replikationsbenutzer auf der Quell-DB-Instance mit den für das Multisource-Replikat erforderlichen Rechten. Um das Multisource-Replikat mit der Quell-DB-Instance zu verbinden, müssen Sie die `replication_user_password` Werte eines Replikationsbenutzers angeben`replication_user_name`, der über `REPLICATION SLAVE` Berechtigungen für die Quell-DB-Instance verfügt`REPLICATION CLIENT`.

Um einen Replikationsbenutzer auf der Quell-DB-Instance zu konfigurieren

1. Stellen Sie mit dem My SQL Client Ihrer Wahl eine Verbindung zur Quell-DB-Instance her und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   > **Important**
   >
   > Aus Sicherheitsgründen sollten Sie ein anderes Passwort als den in den folgenden Beispielen gezeigten Platzhalterwert angeben.

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der Quell-DB-Instance `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer 'repl_user' für Ihre Domäne die Berechtigungen `REPLICATION CLIENT` und `REPLICATION SLAVE` für alle Datenbanken erteilt.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Um die verschlüsselte Replikation zu verwenden, konfigurieren Sie die Quell-DB-Instance für die Verwendung von SSL Verbindungen.

Nach einem Aufruf `mysql.rds_set_external_source_for_channel` zur Konfiguration dieses Replikationskanals können Sie das Replikat aufrufenmysql.rds_start_replication_for_channel, um den Replikationsprozess auf dem Kanal zu starten. Sie können aufrufenthe section called “mysql.rds_reset_external_source_for_channel”, um die Replikation auf dem Kanal zu beenden und die Kanalkonfiguration aus dem Replikat zu entfernen.

Wenn Sie anrufen`mysql.rds_set_external_source_for_channel`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set channel source` in der `mysql.rds_history` Tabelle ohne kanalspezifische Details und in der `mysql.rds_replication_status` Tabelle mit dem Kanalnamen auf. Diese Informationen werden nur für interne Zwecke und zu Überwachungszwecken aufgezeichnet. Um den gesamten Verfahrensaufruf zu Prüfungszwecken aufzuzeichnen, sollten Sie je nach den spezifischen Anforderungen Ihrer Anwendung die Aktivierung von Auditprotokollen oder allgemeinen Protokollen in Betracht ziehen.

Beispiele

Bei der Ausführung auf einer RDS for My SQL DB-Instance konfiguriert das folgende Beispiel einen Replikationskanal, der nach dieser DB-Instance benannt `channel_1` ist, so, dass Daten aus der durch Host `sourcedb.example.com` und Port angegebenen Quelle repliziert werden. `3306`

```
call mysql.rds_set_external_source_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
```

```
  'mysql-bin-changelog.0777',
  120,
  0,
  'channel_1');
```

mysql.rds_set_external_source_with_auto_position_for_channel

Konfiguriert einen Replikationskanal auf einer RDS for My SQL DB-Instance mit einer optionalen Replikationsverzögerung. Die Replikation basiert auf globalen Transaktions-Identifikatoren ()GTIDs.

> **Important**
>
> Um diese Prozedur auszuführen, muss `autocommit` aktiviert sein. Um dies zu aktivieren, setzen Sie den `autocommit`-Parameter auf 1. Weitere Informationen zum Ändern von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_source_with_auto_position_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
  , channel_name
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der RDS für Meine SQL Quell-DB-Instance.

host_port

Der Port, der von der RDS for My SQL Source DB-Instance verwendet wird. Wenn Ihre Netzwerkkonfiguration die Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen RDS für die „Meine SQL Quell-DB-Instance“. Wir empfehlen, dass Sie ein Konto angeben, das ausschließlich für die Replikation mit der Quell-DB-Instance verwendet wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `SOURCE_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quell-DB-Instance aus zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

channel_name

Der Name des Replikationskanals. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_source_with_auto_position_for_channel` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf dem Ziel RDS für My SQL DB-Instance ausgeführt werden, auf dem Sie den Replikationskanal erstellen.

Konfigurieren Sie vor der Ausführung `rds_set_external_source_with_auto_position_for_channel` einen Replikationsbenutzer auf der Quell-DB-Instance mit den für das Multisource-Replikat erforderlichen Rechten. Um das Multisource-Replikat mit der Quell-DB-Instance zu verbinden, müssen Sie die `replication_user_password` Werte eines Replikationsbenutzers angeben`replication_user_name`, der über `REPLICATION SLAVE` Berechtigungen für die Quell-DB-Instance verfügt`REPLICATION CLIENT`.

Um einen Replikationsbenutzer auf der Quell-DB-Instance zu konfigurieren

1. Stellen Sie mit dem My SQL Client Ihrer Wahl eine Verbindung zur Quell-DB-Instance her und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   > **Important**
   >
   > Aus Sicherheitsgründen sollten Sie ein anderes Passwort als den in den folgenden Beispielen gezeigten Platzhalterwert angeben.

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der Quell-DB-Instance `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer 'repl_user' für Ihre Domäne die Berechtigungen `REPLICATION CLIENT` und `REPLICATION SLAVE` für alle Datenbanken erteilt.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
   ```

Um die verschlüsselte Replikation zu verwenden, konfigurieren Sie die Quell-DB-Instance für die Verwendung von SSL Verbindungen.

Stellen Sie vor dem Aufruf sicher`mysql.rds_set_external_source_with_auto_position_for_channel`, dass Sie die `gtid_purged` Systemvariable aufrufen, the section called “mysql.rds_set_external_source_gtid_purged” um sie auf einen bestimmten GTID Bereich aus einer externen Quelle zu setzen.

Nachdem Sie aufgerufen habenmysql.rds_set_external_source_with_auto_position_for_channel, um eine RDS Amazon-DB-Instance als Read Replica auf einem bestimmten Channel zu konfigurieren, können Sie die Read Replica aufrufenthe section called “mysql.rds_start_replication_for_channel”, um den Replikationsprozess auf diesem Kanal zu starten.

Nach einem Aufruf mysql.rds_set_external_source_with_auto_position_for_channel zur Konfiguration dieses Replikationskanals können Sie das Replikat aufrufenmysql.rds_start_replication_for_channel, um den Replikationsprozess auf dem Kanal zu starten. Sie können aufrufenthe section called “mysql.rds_reset_external_source_for_channel”, um die Replikation auf dem Kanal zu beenden und die Kanalkonfiguration aus dem Replikat zu entfernen.

Beispiele

Bei der Ausführung auf einer RDS for My SQL DB-Instance konfiguriert das folgende Beispiel einen Replikationskanal, der nach dieser DB-Instance benannt channel_1 ist, so, dass er Daten aus der durch Host sourcedb.example.com und Port angegebenen Quelle repliziert. Die minimale Replikationsverzögerung 3306 wird auf eine Stunde (3.600 Sekunden) festgelegt. Das bedeutet, dass eine Änderung gegenüber der Quelle RDS für Meine SQL DB-Instance mindestens eine Stunde lang nicht auf das Multiquellen-Replikat angewendet wird.

```
call mysql.rds_set_external_source_with_auto_position_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  1,
  3600,
  'channel_1');
```

mysql.rds_set_external_source_with_delay_for_channel

Konfiguriert einen Replikationskanal auf einer RDS for My SQL DB-Instance mit einer bestimmten Replikationsverzögerung.

> **Important**
>
> Um diese Prozedur auszuführen, muss autocommit aktiviert sein. Um dies zu aktivieren, setzen Sie den autocommit-Parameter auf 1. Weitere Informationen zum Ändern von

Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Syntax

```
CALL mysql.rds_set_external_source_with_delay_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
  , channel_name
);
```

Parameter

host_name

Der Hostname oder die IP-Adresse der RDS für Meine SQL Quell-DB-Instance.

host_port

Der Port, der von der RDS for My SQL Source DB-Instance verwendet wird. Wenn Ihre Netzwerkkonfiguration die Secure Shell (SSH) -Portreplikation beinhaltet, die die Portnummer konvertiert, geben Sie die Portnummer an, die von verfügbar gemacht wirdSSH.

replication_user_name

Die ID eines Benutzers mit `REPLICATION CLIENT` und `REPLICATION SLAVE` Berechtigungen RDS für die „Meine SQL Quell-DB-Instance“. Wir empfehlen, dass Sie ein Konto angeben, das ausschließlich für die Replikation mit der Quell-DB-Instance verwendet wird.

replication_user_password

Das zu dem in `replication_user_name` angegebenen User-ID gehörige Passwort.

mysql_binary_log_file_name

Der Name des Binärprotokolls auf der Quell-DB-Instance enthält die Replikationsinformationen.

mysql_binary_log_file_location

Die Position innerhalb der für `mysql_binary_log_file_name` angegebenen binären Protokolldatei, ab der bei der Replikation die Replikationsinformationen gelesen werden.

Sie können den Namen und den Speicherort der Binlog-Datei ermitteln, indem Sie `SHOW BINARY LOG STATUS` auf der Quelldatenbankinstanz starten.

> **Note**
>
> Frühere Versionen von My SQL wurden `SHOW MASTER STATUS` anstelle von verwendet`SHOW BINARY LOG STATUS`. Wenn Sie eine SQL Version von My vor 8.4 verwenden, verwenden Sie`SHOW MASTER STATUS`.

ssl_encryption

Ein Wert, der angibt, ob die Secure Socket Layer (SSL) -Verschlüsselung für die Replikationsverbindung verwendet wird. 1 gibt an, dass SSL Verschlüsselung verwendet wird, 0 gibt an, dass keine Verschlüsselung verwendet wird. Der Standardwert ist 0.

> **Note**
>
> Die Option `SOURCE_SSL_VERIFY_SERVER_CERT` wird nicht unterstützt. Diese Option ist auf 0 gesetzt, was bedeutet, dass die Verbindung verschlüsselt ist, aber die Zertifikate nicht überprüft werden.

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quell-DB-Instance aus zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

channel_name

Der Name des Replikationskanals. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_set_external_source_with_delay_for_channel` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren muss auf dem Ziel RDS für My SQL DB-Instance ausgeführt werden, auf dem Sie den Replikationskanal erstellen.

Konfigurieren Sie vor der Ausführung `mysql.rds_set_external_source_with_delay_for_channel` einen Replikationsbenutzer auf der Quell-DB-Instance mit den für das Multisource-Replikat erforderlichen Rechten. Um das Multisource-Replikat mit der Quell-DB-Instance zu verbinden, müssen Sie die `replication_user_password` Werte eines Replikationsbenutzers angeben`replication_user_name`, der über `REPLICATION SLAVE` Berechtigungen für die Quell-DB-Instance verfügt`REPLICATION CLIENT`.

Um einen Replikationsbenutzer auf der Quell-DB-Instance zu konfigurieren

1. Stellen Sie mit dem My SQL Client Ihrer Wahl eine Verbindung zur Quell-DB-Instance her und erstellen Sie ein Benutzerkonto, das für die Replikation verwendet werden soll. Im Folgenden wird ein Beispiel gezeigt.

   > **Important**
   >
   > Aus Sicherheitsgründen sollten Sie ein anderes Passwort als den in den folgenden Beispielen gezeigten Platzhalterwert angeben.

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. `REPLICATION CLIENT`Erteilen Sie Ihrem Replikationsbenutzer auf der Quell-DB-Instance `REPLICATION SLAVE` Rechte. Im folgenden Beispiel werden dem Benutzer 'repl_user' für Ihre Domäne die Berechtigungen `REPLICATION CLIENT` und `REPLICATION SLAVE` für alle Datenbanken erteilt.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
   ```

Um die verschlüsselte Replikation zu verwenden, konfigurieren Sie die Quell-DB-Instance für die Verwendung von SSL Verbindungen.

Nach einem Aufruf `mysql.rds_set_external_source_with_delay_for_channel` zur Konfiguration dieses Replikationskanals können Sie das Replikat aufrufen[mysql.rds_start_replication_for_channel](), um den Replikationsprozess auf dem Kanal zu starten. Sie können aufrufen[the section called “mysql.rds_reset_external_source_for_channel”](), um die Replikation auf dem Kanal zu beenden und die Kanalkonfiguration aus dem Replikat zu entfernen.

Wenn Sie anrufen`mysql.rds_set_external_source_with_delay_for_channel`, RDS zeichnet Amazon die Uhrzeit, den Benutzer und eine Aktion von `set channel source` in der `mysql.rds_history` Tabelle ohne kanalspezifische Details und in der `mysql.rds_replication_status` Tabelle mit dem Kanalnamen auf. Diese Informationen werden nur für interne Zwecke und zu Überwachungszwecken aufgezeichnet. Um den gesamten Verfahrensaufruf zu Prüfungszwecken aufzuzeichnen, sollten Sie je nach den spezifischen Anforderungen Ihrer Anwendung die Aktivierung von Auditprotokollen oder allgemeinen Protokollen in Betracht ziehen.

Beispiele

Bei der Ausführung auf einer RDS For My SQL DB-Instance konfiguriert das folgende Beispiel einen Replikationskanal, der `channel_1` auf dieser DB-Instance benannt ist, so, dass Daten aus der durch Host `sourcedb.example.com` und Port angegebenen Quelle repliziert werden. Die minimale Replikationsverzögerung `3306` wird auf eine Stunde (3.600 Sekunden) festgelegt. Das bedeutet, dass eine Änderung gegenüber der Quelle RDS für Meine SQL DB-Instance mindestens eine Stunde lang nicht auf das Multiquellen-Replikat angewendet wird.

```
call mysql.rds_set_external_source_with_delay_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.000777',
  120,
  1,
  3600,
  'channel_1');
```

mysql.rds_set_source_auto_position_for_channel

Legt fest, dass der Replikationsmodus für den angegebenen Kanal entweder auf den Positionen der binären Protokolldateien oder auf globalen Transaktions-Identifikatoren (GTIDs) basiert.

Syntax

```
CALL mysql.rds_set_source_auto_position_for_channel (
auto_position_mode
 , channel_name
);
```

Parameter

auto_position_mode

Ein Wert, der angibt, ob die Positionsreplikation der Protokolldatei oder die GTID basierte Replikation verwendet werden soll:

- `0` – Verwendung der auf der Binärprotokolldateiposition basierenden Replikationsmethode. Der Standardwert ist `0`.
- 1— Verwenden Sie die GTID basierte Replikationsmethode.

channel_name

Der Name des Replikationskanals auf dem Multiquellenreplikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_set_source_auto_position_for_channel` muss vom Hauptbenutzer ausgeführt werden. Dieses Verfahren startet die Replikation auf dem angegebenen Kanal neu, um den angegebenen Modus für die auto Positionierung anzuwenden.

Beispiele

Im folgenden Beispiel wird der auto Positionierungsmodus für channel_1 so eingestellt, dass er die GTID basierte Replikationsmethode verwendet.

```
call mysql.rds_set_source_auto_position_for_channel(1,'channel_1');
```

mysql.rds_set_source_delay_for_channel

Legt die Mindestanzahl von Sekunden fest, um die Replikation von der Quelldatenbank-Instance zum Multiquellen-Replikat für den angegebenen Kanal zu verzögern.

Syntax

```
CALL mysql.rds_set_source_delay_for_channel(delay, channel_name);
```

Parameter

delay

Die Mindestanzahl von Sekunden, um die Replikation von der Quell-DB-Instance aus zu verzögern.

Die Obergrenze für diesen Parameter beträgt einen Tag (86 400 Sekunden).

channel_name

Der Name des Replikationskanals auf dem Multiquellen-Replikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_set_source_delay_for_channel` muss vom Hauptbenutzer ausgeführt werden. Um dieses Verfahren zu verwenden, rufen Sie zuerst auf, `mysql.rds_stop_replication_for_channel` um die Replikation zu beenden. Rufen Sie dann diese Prozedur auf, um den Wert für die Replikationsverzögerung festzulegen. Wenn die Verzögerung festgelegt ist, rufen Sie auf, `mysql.rds_start_replication_for_channel` um die Replikation neu zu starten.

Beispiele

Im folgenden Beispiel wird die Verzögerung für die Replikation von der Quelldatenbank-Instance auf dem `channel_1` Multiquellen-Replikat auf mindestens eine Stunde (3.600 Sekunden) festgelegt.

```
CALL mysql.rds_set_source_delay_for_channel(3600,'channel_1');
```

mysql.rds_skip_repl_error_for_channel

Überspringt ein binäres Protokollereignis und löscht einen Replikationsfehler auf einem My SQL DB-Replikat mit mehreren Quellen für den angegebenen Kanal.

Syntax

```
CALL mysql.rds_skip_repl_error_for_channel(channel_name);
```

Parameter

channel_name

Der Name des Replikationskanals auf dem Multiquellenreplikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Der Hauptbenutzer muss die Prozedur `mysql.rds_skip_repl_error_for_channel` auf einem Lesereplikat ausführen. Sie können dieses Verfahren auf ähnliche Weise `mysql.rds_skip_repl_error` verwenden, um einen Fehler in einer Read Replica zu überspringen. Weitere Informationen finden Sie unter Aufrufen der Prozedur mysql.rds_skip_repl_error.

> **Note**
>
> Um Fehler bei der GTID basierten Replikation zu überspringen, empfehlen wir, the section called “mysql.rds_skip_transaction_with_gtid” stattdessen das Verfahren zu verwenden.

Führen Sie den SQL `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` Befehl My aus, um festzustellen, ob Fehler vorliegen. Wenn ein Replikationsfehler nicht als kritisch eingestuft, ist, können Sie `mysql.rds_skip_repl_error_for_channel` ausführen, um den Fehler zu überspringen. Wenn mehrere Fehler vorliegen, `mysql.rds_skip_repl_error_for_channel` löscht der erste Fehler auf dem angegebenen Replikationskanal und warnt dann, dass weitere Fehler vorhanden sind. In diesem Fall können Sie mithilfe von `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` die angemessene Vorgehensweise bei der Handhabung des nächsten Fehlers ermitteln. Informationen zu den zurückgegebenen Werten finden Sie in der SHOWREPLICASTATUSErklärung unter Meine SQL Dokumentation.

mysql.rds_start_replication_for_channel

Initiiert die Replikation von einer RDS For My SQL DB-Instance zu einem Multisource-Replikat auf dem angegebenen Kanal.

> **Note**
>
> Sie können die mysql.rds_start_replication_until_gtid_for_channel gespeicherte Prozedur mysql.rds_start_replication_until_for_channel oder verwenden, um die Replikation von einer RDS for My SQL DB-Instance aus zu initiieren und die Replikation am angegebenen Speicherort der Binärprotokolldatei zu beenden.

Syntax

```
CALL mysql.rds_start_replication_for_channel(channel_name);
```

Parameter

channel_name

Der Name des Replikationskanals auf dem Multiquellenreplikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_start_replication_for_channel` muss vom Hauptbenutzer ausgeführt werden. Nachdem Sie die Daten aus der Quelle RDS für Meine SQL DB-Instance importiert haben, führen Sie diesen Befehl für das Multiquellen-Replikat aus, um die Replikation auf dem angegebenen Kanal zu starten.

Beispiele

Im folgenden Beispiel wird die Replikation auf dem `channel_1` Multiquellen-Replikat gestartet.

```
CALL mysql.rds_start_replication_for_channel('channel_1');
```

mysql.rds_start_replication_until_for_channel

Initiiert die Replikation von einer RDS for My SQL DB-Instance auf dem angegebenen Kanal und stoppt die Replikation am angegebenen Speicherort der binären Protokolldatei.

Syntax

```
CALL mysql.rds_start_replication_until_for_channel (
replication_log_file
  , replication_stop_point
  , channel_name
);
```

Parameter

replication_log_file

Der Name des Binärprotokolls auf der Quell-DB-Instance enthält die Replikationsinformationen.

replication_stop_point

Die Position im `replication_log_file`-Binärprotokoll, an der die Replikation stoppt.

channel_name

Der Name des Replikationskanals auf dem Multiquellen-Replikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_start_replication_until_for_channel` muss vom Hauptbenutzer ausgeführt werden. Bei diesem Verfahren wird die Replikation gestartet und dann beendet, wenn die angegebene Position der Binlogdatei erreicht ist. Bei diesem Verfahren werden sowohl das `SQL_THREAD` als auch beendet. `IO_THREAD`

Der für den `replication_log_file` Parameter angegebene Dateiname muss mit dem Binlog-Dateinamen der Quell-DB-Instance übereinstimmen.

Wenn der `replication_stop_point` Parameter einen Stopp angibt, der in der Vergangenheit liegt, wird die Replikation sofort gestoppt.

Beispiele

Das folgende Beispiel initiiert die Replikation am und repliziert die Änderungen`channel_1`, bis sie den Speicherort `120` in der `mysql-bin-changelog.000777` Binärprotokolldatei erreichen.

```
call mysql.rds_start_replication_until_for_channel(
  'mysql-bin-changelog.000777',
  120,
  'channel_1'
  );
```

mysql.rds_start_replication_until_gtid_for_channel

Initiiert die Replikation auf dem angegebenen Kanal von einer RDS for My SQL DB-Instance aus und stoppt die Replikation an der angegebenen globalen Transaktions-ID ()GTID.

Syntax

```
CALL mysql.rds_start_replication_until_gtid_for_channel(gtid,channel_name);
```

Parameter

gtid

GTIDDanach soll die Replikation beendet werden.

channel_name

Der Name des Replikationskanals auf dem Multiquellenreplikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_start_replication_until_gtid_for_channel` muss vom Hauptbenutzer ausgeführt werden. Das Verfahren startet die Replikation auf dem angegebenen Kanal und wendet alle Änderungen bis zum angegebenen GTID Wert an. Anschließend wird die Replikation auf dem Kanal beendet.

Wenn der Parameter `gtid` eine Transaktion angibt, die bereits von dem Replikat ausgeführt wurde, wird die Replikation sofort gestoppt.

Bevor Sie dieses Verfahren ausführen, müssen Sie die Multithread-Replikation deaktivieren, indem Sie den Wert `replica_parallel_workers` oder `slave_parallel_workers` auf `0` setzen.

Beispiele

Im folgenden Beispiel wird die Replikation am initiiert und die Änderungen repliziert`channel_1`, bis der Wert erreicht ist. GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`

```
call mysql.rds_start_replication_until_gtid_for_channel('3E11FA47-71CA-11E1-9E33-
C80AA9429562:23','channel_1');
```

mysql.rds_stop_replication_for_channel

Stoppt die Replikation von einer My SQL DB-Instance auf dem angegebenen Kanal.

Syntax

```
CALL mysql.rds_stop_replication_for_channel(channel_name);
```

Parameter

`channel_name`

Der Name des Replikationskanals auf dem Multiquellen-Replikat. Jeder Replikationskanal empfängt die binären Protokollereignisse von einer einzigen Quelle RDS für My SQL DB-Instance, die auf einem bestimmten Host und Port ausgeführt wird.

Nutzungshinweise

Die Prozedur `mysql.rds_stop_replication_for_channel` muss vom Hauptbenutzer ausgeführt werden.

Beispiele

Im folgenden Beispiel wird die Replikation `channel_1` eines Replikats mit mehreren Quellen beendet.

```
CALL mysql.rds_stop_replication_for_channel('channel_1');
```

Transaktionen replizieren mit GTIDs

Die folgenden gespeicherten Prozeduren steuern, wie Transaktionen repliziert werden, indem globale Transaktions-Identifikatoren (GTIDs) mit RDS für My verwendet werden. SQL Weitere Hinweise zur Replikation auf Basis von GTIDs with RDS für My finden Sie SQL unter. Verwenden der GTID basierten Replikation

Wenn Sie gespeicherte Prozeduren verwenden, um die Replikation mit einem Replikationsbenutzer zu verwaltencaching_sha2_password, müssen Sie bei der Konfiguration TLS Folgendes SOURCE_SSL=1 angeben: caching_sha2_passwordist das Standard-Authentifizierungs-Plugin RDS für My SQL 8.4.

Themen

•

•

Überspringt die Replikation einer Transaktion mit der angegebenen globalen Transaktions-ID (GTID) auf einer My SQL DB-Instance.

Sie können dieses Verfahren für die Notfallwiederherstellung verwenden, wenn bekannt ist, dass eine bestimmte GTID Transaktion ein Problem verursacht. Verwenden Sie diese gespeicherte Prozedur, um die problematische Transaktion zu überspringen. Problematisch sind beispielsweise Transaktionen, die die Replikation deaktivieren, wichtige Daten löschen oder dafür sorgen, dass die DB-Instance nicht mehr verfügbar ist.

Syntax

```
CALL mysql.rds_skip_transaction_with_gtid (
gtid_to_skip
);
```

Parameter

gtid_to_skip

Die GTID Replikationstransaktion, die übersprungen werden soll.

Nutzungshinweise

Die Prozedur `mysql.rds_skip_transaction_with_gtid` muss vom Hauptbenutzer ausgeführt werden.

Dieses Verfahren wird RDS für alle My SQL 5.7-Versionen, alle RDS für My SQL 8.0-Versionen und alle RDS für My SQL 8.4-Versionen unterstützt.

Beispiele

Im folgenden Beispiel wird die Replikation der Transaktion mit dem GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23` übersprungen.

```
CALL mysql.rds_skip_transaction_with_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

Initiiert die Replikation von einem aus und stoppt die Replikation unmittelbar nach der angegebenen globalen Transaktions-ID (GTID).

Syntax

```
CALL mysql.rds_start_replication_until_gtid(gtid);
```

Parameter

gtid

GTIDDanach soll die Replikation beendet werden.

Nutzungshinweise

Die Prozedur `mysql.rds_start_replication_until_gtid` muss vom Hauptbenutzer ausgeführt werden.

Dieses Verfahren wird RDS für alle My SQL 5.7-Versionen, alle RDS für My SQL 8.0-Versionen und alle RDS für My SQL 8.4-Versionen unterstützt.

Sie können diese Prozedur mit verzögerter Replikation für die Notfallwiederherstellung verwenden. Wenn Sie die verzögerte Replikation konfiguriert haben, können Sie diese Prozedur verwenden, um alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten

Lesereplikat wiederherzustellen. Nachdem diese Prozedur die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Sie können die verzögerte Replikation mit den folgenden gespeicherten Prozeduren konfigurieren:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay (für MariaDB und für meine Hauptversionen 8.0 und niedriger) RDS RDS SQL
- mysql.rds_set_external_source_with_delay (für meine Hauptversionen 8.4 und höher) RDS SQL
- mysql.rds_set_source_delay

Wenn der Parameter `gtid` eine Transaktion angibt, die bereits von dem Replikat ausgeführt wurde, wird die Replikation sofort gestoppt.

Beispiele

Im folgenden Beispiel wird die Replikation initiiert und die Änderungen repliziert, bis sie GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`

```
call mysql.rds_start_replication_until_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

Rotieren der Abfrage-Logs

Die folgenden gespeicherten Prozeduren rotieren Meine SQL Protokolle in Backup-Tabellen. Weitere Informationen finden Sie unter MySQL-Datenbank-Protokolldateien.

Themen

- mysql.rds_rotate_general_log
- mysql.rds_rotate_slow_log

mysql.rds_rotate_general_log

Rotiert die Tabelle `mysql.general_log` in eine Sicherungstabelle.

Syntax

```
CALL mysql.rds_rotate_general_log;
```

Nutzungshinweise

Sie können die Tabelle `mysql.general_log` in eine Sicherungstabelle rotieren, indem Sie die Prozedur `mysql.rds_rotate_general_log` aufrufen. Beim Rotieren von Protokolldateien wird die aktuelle Protokolltabelle in eine Sicherungsprotokolltabelle kopiert, und die Einträge in der aktuellen Protokolltabelle werden entfernt. Sofern bereits eine Sicherungsprotokolltabelle vorhanden ist, wird diese gelöscht, bevor die aktuelle Protokolltabelle in die Sicherungsprotokolltabelle kopiert wird. Sie können die Sicherungsprotokolltabelle abfragen, wenn dies nötig ist. Die Backup-Protokolltabelle für die `mysql.general_log`-Tabelle ist als `mysql.general_log_backup` benannt.

Sie können dieses Verfahren nur ausführen, wenn der Parameter `log_output` auf `TABLE` eingestellt ist.

mysql.rds_rotate_slow_log

Rotiert die Tabelle `mysql.slow_log` in eine Sicherungstabelle.

Syntax

```
CALL mysql.rds_rotate_slow_log;
```

Nutzungshinweise

Sie können die Tabelle `mysql.slow_log` in eine Sicherungstabelle rotieren, indem Sie die Prozedur `mysql.rds_rotate_slow_log` aufrufen. Beim Rotieren von Protokolldateien wird die aktuelle Protokolltabelle in eine Sicherungsprotokolltabelle kopiert, und die Einträge in der aktuellen Protokolltabelle werden entfernt. Sofern bereits eine Sicherungsprotokolltabelle vorhanden ist, wird diese gelöscht, bevor die aktuelle Protokolltabelle in die Sicherungsprotokolltabelle kopiert wird.

Sie können die Sicherungsprotokolltabelle abfragen, wenn dies nötig ist. Die Backup-Protokolltabelle für die `mysql.slow_log`-Tabelle ist als `mysql.slow_log_backup` benannt.

Konfiguration des Binärprotokolls einrichten und anzeigen

Die folgenden gespeicherten Prozeduren legen Konfigurationsparameter fest und zeigen sie an, z. B. für die Aufbewahrung binärer Protokolldateien.

Themen

- mysql.rds_set_configuration
- mysql.rds_show_configuration

mysql.rds_set_configuration

Gibt die Anzahl an Stunden an, für die die Binärprotokolle aufbewahrt werden sollen, oder die Anzahl Sekunden, um die die Replikation verzögert werden soll.

Syntax

```
CALL mysql.rds_set_configuration(name,value);
```

Parameter

name

Der Name des festzulegenden Konfigurationsparameters

value

Der Wert des Konfigurationsparameters

Nutzungshinweise

Die Prozedur `mysql.rds_set_configuration` unterstützt die folgenden Konfigurationsparameter:

- binlog retention hours
- Quellenverzögerung
- target delay

Die Konfigurationsparameter werden dauerhaft gespeichert und überstehen jeden Neustart oder Failover der DB-Instance.

binlog retention hours

Der Parameter `binlog retention hours` wird verwendet, um die Anzahl der Stunden anzugeben, die Binärprotokolldateien aufbewahrt werden sollen. Amazon RDS löscht normalerweise ein Binärprotokoll so schnell wie möglich, aber das Binärprotokoll ist möglicherweise immer noch für die Replikation mit einer externen SQL Datenbank von Meine Datenbank erforderlich. RDS

Der Standardwert von `binlog retention hours` ist `NULL`. RDSFor My `NULL` bedeutetSQL, dass Binärprotokolle nicht aufbewahrt werden (0 Stunden).

Um die Anzahl der Stunden zu bestimmen, für die Binärprotokolle auf einer/einem DB-Instance aufbewahrt werden sollen, verwenden Sie die gespeicherte Prozedur `mysql.rds_set_configuration` und geben Sie, wie in dem folgenden Beispiel gezeigt, einen ausreichend großen Zeitraum für die gewünschte Replikation an.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

> **Note**
>
> Sie können den Wert `0` nicht für `binlog retention hours` verwenden.

Für My SQL DB-Instances beträgt der `binlog retention hours` Höchstwert 168 (7 Tage).

Nachdem Sie den Aufbewahrungszeitraum festgelegt haben, überwachen Sie die Speichernutzung für die DB-Instance, um sicherzustellen, dass die aufbewahrten binären Protokolle nicht zu viel Speicherplatz beanspruchen.

Bei Multi-AZ-DB-Cluster-Bereitstellungen können Sie die Aufbewahrung von Binärprotokollen nur von der Writer-DB-Instance aus konfigurieren, und die Einstellung wird asynchron an alle Reader-DB-Instances weitergegeben. Wenn Binärprotokolle auf dem DB-Cluster die Hälfte des gesamten lokalen Speicherplatzes überschreiten, verschiebt Amazon RDS automatisch veraltete Protokolle auf das EBS Volume. Die neuesten Protokolle verbleiben jedoch im lokalen Speicher, sodass sie verloren gehen können, wenn ein Fehler auftritt, der einen Host-Austausch erfordert, oder wenn Sie die Datenbank nach oben oder unten skalieren.

Quellenverzögerung

Verwenden Sie den Parameter `source delay` in einem Lesereplikat zur Angabe der Sekundenzahl, um die die Replikation vom Lesereplikat zur entsprechenden Quelldatenbank-Instance verzögert werden soll. Amazon repliziert Änderungen RDS normalerweise so schnell wie möglich, aber Sie möchten möglicherweise, dass einige Umgebungen die Replikation verzögern. Wenn die Replikation verzögert wird, können Sie alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherstellen. Wenn eine Tabelle versehentlich entfernt wurde, können Sie sie aufgrund der verzögerten Replikation schnell wiederherstellen. Der Standardwert von `target delay` ist `0` (Replikation nicht verzögern).

Wenn Sie diesen Parameter verwenden, wird er ausgeführt mysql.rds_set_source_delay und verwendet CHANGE primary TO MASTER _ DELAY = Eingabewert. Bei Erfolg speichert die Prozedur den Parameter `source delay` in der Tabelle `mysql.rds_configuration`.

Um die Anzahl der Sekunden anzugeben, für die Amazon die Replikation RDS zu einer Quell-DB-Instance verzögert, verwenden Sie die `mysql.rds_set_configuration` gespeicherte Prozedur und geben Sie die Anzahl der Sekunden an, um die Replikation verzögert werden soll. Im folgenden Beispiel wird die Replikation um mindestens eine Stunde (3 600 Sekunden) verzögert.

```
call mysql.rds_set_configuration('source delay', 3600);
```

Die Prozedur führt dann `mysql.rds_set_source_delay(3600)` aus.

Die Obergrenze für den Parameter `source delay` beträgt einen Tag (86 400 Sekunden).

target delay

Verwenden Sie den `target delay` Parameter, um die Anzahl der Sekunden anzugeben, um die Replikation zwischen einer DB-Instance und allen future RDS verwalteten Read Replicas, die von dieser Instance erstellt wurden, verzögert werden soll. Dieser Parameter wird für non-RDS-managed Read Replicas ignoriert. Amazon repliziert Änderungen RDS normalerweise so schnell wie möglich, aber Sie möchten möglicherweise, dass einige Umgebungen die Replikation verzögern. Wenn die Replikation verzögert wird, können Sie alle Änderungen bis zu einem Zeitpunkt unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherstellen. Wenn eine Tabelle versehentlich entfernt wurde, können Sie sie mithilfe der verzögerten Replikation schnell wiederherstellen. Der Standardwert von `target delay` ist `0` (Replikation nicht verzögern).

Für die Notfallwiederherstellung können Sie diesen Konfigurationsparameter mit der gespeicherten Prozedur oder verwenden. Um alle Änderungen bis zu einem Zeitpunkt

unmittelbar vor Eintreten des Notfalls in einem verzögerten Lesereplikat wiederherzustellen, können Sie die Prozedur `mysql.rds_set_configuration` mit diesem Parametersatz ausführen. Nachdem die Prozedur `mysql.rds_start_replication_until` oder `mysql.rds_start_replication_until_gtid` die Replikation gestoppt hat, können Sie das Lesereplikat zur neuen primären DB-Instance hochstufen (siehe die Anleitung unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance).

Um das `mysql.rds_rds_start_replication_until_gtid` Verfahren verwenden zu können, muss die GTID basierte Replikation aktiviert sein. Um eine bestimmte GTID basierte Transaktion zu überspringen, von der bekannt ist, dass sie zu einem Notfall führt, können Sie die gespeicherte Prozedur verwenden. Weitere Informationen zum Arbeiten mit der GTID basierten Replikation finden Sie unterVerwenden der GTID basierten Replikation.

Um die Anzahl der Sekunden anzugeben, für die Amazon die Replikation RDS auf eine Read Replica verzögert, verwenden Sie die `mysql.rds_set_configuration` gespeicherte Prozedur und geben Sie die Anzahl der Sekunden an, um die Replikation verzögert werden soll. Das folgende Beispiel gibt an, dass die Replikation um mindestens eine Stunde (3600 Sekunden) verzögert wird.

```
call mysql.rds_set_configuration('target delay', 3600);
```

Die Obergrenze für den Parameter `target delay` beträgt einen Tag (86 400 Sekunden).

mysql.rds_show_configuration

Die Anzahl der Stunden, während der binäre Protokolldateien aufbewahrt werden sollen.

Syntax

```
CALL mysql.rds_show_configuration;
```

Nutzungshinweise

Verwenden Sie die `mysql.rds_show_configuration` gespeicherte Prozedur, um zu überprüfen, wie viele Stunden Amazon RDS Binärprotokolle aufbewahrt.

Beispiele

Nachfolgend sehen Sie ein Beispiel für die Anzeige des Aufbewahrungszeitraums:

```
call mysql.rds_show_configuration;
```

```
                name                          value     description
                binlog retention hours        24        binlog retention hours specifies
the duration in hours before binary logs are automatically deleted.
```

Wärmen des InnoDB-Caches

Die folgenden gespeicherten Prozeduren speichern, laden oder brechen das Laden des InnoDB-Pufferpools auf RDS-für-MySQL-DB-Instances ab. Weitere Informationen finden Sie unter InnoDB-Cache-Warming für MySQL in Amazon RDS.

Themen

- mysql.rds_innodb_buffer_pool_dump_now
- mysql.rds_innodb_buffer_pool_load_abort
- mysql.rds_innodb_buffer_pool_load_now

mysql.rds_innodb_buffer_pool_dump_now

Speichert den aktuellen Zustand des Pufferpools auf der Festplatte.

Syntax

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Nutzungshinweise

Die Prozedur `mysql.rds_innodb_buffer_pool_dump_now` muss vom Hauptbenutzer ausgeführt werden.

mysql.rds_innodb_buffer_pool_load_abort

Bricht das Laden des gespeicherten Zustands des Pufferpools ab, während der Vorgang läuft.

Syntax

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

Nutzungshinweise

Die Prozedur `mysql.rds_innodb_buffer_pool_load_abort` muss vom Hauptbenutzer ausgeführt werden.

mysql.rds_innodb_buffer_pool_load_now

Lädt den gespeicherten Zustand des Pufferpools von der Festplatte.

Syntax

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

Nutzungshinweise

Die Prozedur `mysql.rds_innodb_buffer_pool_load_now` muss vom Hauptbenutzer ausgeführt werden.

Amazon RDS für Oracle

Amazon RDS unterstützt DB-Instances, die die folgenden Versionen und Editionen von Oracle Database ausführen:

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)

> **Note**
>
> Oracle Database 11g, Oracle Database 12c und Oracle Database 18c sind Legacy-Versionen und werden nicht mehr unterstützt.

Führen Sie die Schritte im Abschnitt Einrichtung Ihrer Amazon RDS-Umgebung in diesem Handbuch durch, bevor Sie eine DB-Instance erstellen. Wenn Sie eine DB-Instance mit Ihrem Master-Konto erstellen, erhält das Konto DBA-Berechtigungen, mit einigen Einschränkungen. Verwenden Sie dieses Konto für administrative Aufgaben wie das Erstellen zusätzlicher Datenbankkonten. SYS, SYSTEM oder andere von Oracle bereitgestellte Administratorkonten können nicht verwendet werden.

Sie können das folgende erstellen:

- DB-Instances
- DB-Snapshots
- Point-in-time stellt wieder her
- Automatische Backups
- Manuelle Backups

Sie können DB-Instances verwenden, auf denen Oracle Database in einer VPC ausgeführt wird. Sie können Ihrer DB-Instance auch Funktionen hinzufügen, indem Sie verschiedene Optionen wie Oracle Spatial oder Oracle Statspack aktivieren. Amazon RDS unterstützt Multi-AZ-Bereitstellungen für Oracle als eine Lösung mit hoher Verfügbarkeit und Failover.

> **Important**
>
> Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Shell-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind. Sie können mit Standard-SQL-Clients wie Oracle SQL*Plus auf Ihre Datenbank zugreifen. Sie können jedoch nicht direkt auf den Host zugreifen, indem Sie Telnet oder Secure Shell (SSH) verwenden.

Themen

- Übersicht über Oracle on Amazon RDS
- Herstellen der Verbindung zu Ihrer Oracle-DB-Instance
- Sichern von Verbindungen von Oracle DB-Instances
- Arbeiten mit CDBs in RDS für Oracle
- Verwaltung Ihrer DB-Instance von RDS für Oracle
- Konfiguration erweiterter Funktionen von RDS für Oracle
- Daten in Oracle auf Amazon importieren RDS
- Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle
- Hinzufügen von Optionen zu Oracle DB-Instances
- Aktualisieren der DB-Engine von RDS für Oracle
- Verwenden von Drittanbietersoftware mit Ihrer RDS-for-Oracle-DB-Instance
- Versionshinweise zur Oracle-Datenbank-Engine

Übersicht über Oracle on Amazon RDS

In den folgenden Abschnitten erhalten Sie einen Überblick über RDS für Oracle.

Themen

- RDS for Oracle – Funktionen
- RDS für Oracle releases
- RDSfür Oracle-Lizenzoptionen
- RDS für Oracle-Benutzer und -Berechtigungen

- RDS für Oracle-DB-Instance-Klassen
- RDS für Oracle-Datenbankarchitektur
- Initialisierungsparameter für RDS für Oracle
- RDS for Oracle-Zeichensätze
- Beschränkungen von RDS for Oracle

RDS for Oracle – Funktionen

Amazon RDS for Oracle unterstützt die meisten Funktionen und Merkmale von Oracle Database. Einige Funktionen werden möglicherweise nur begrenzt unterstützt oder haben eingeschränkte Berechtigungen. Einige Funktionen sind nur in Enterprise Edition verfügbar und einige erfordern zusätzliche Lizenzen. Weitere Informationen zu Oracle Database-Funktionen für spezifische Oracle Database-Versionen finden Sie in dem Oracle Database Licensing Information User Manual für die von Ihnen verwendete Version.

Themen

- Neue Funktionen in RDS for Oracle
- Unterstützte Funktionen in RDS for Oracle
- Nicht unterstützte Funktionen in RDS for Oracle

Neue Funktionen in RDS for Oracle

Suchen Sie nach dem Schlüsselwort, um neue Funktionen in RDS Dokumentverlauf for Oracle zu sehen**`Oracle`**.

Unterstützte Funktionen in RDS for Oracle

Amazon RDS für Oracle unterstützt die folgenden Oracle Database-Funktionen:

> **Note**
>
> Die folgende Liste erhebt keinen Anspruch auf Vollständigkeit.

- Advanced Compression

- Oracle Application Express (APEX)

  Weitere Informationen finden Sie unter Oracle Application Express (APEX).
- Automatische Arbeitsspeicher-Verwaltung
- Automatische Verwaltung des Rückgängigmachens
- Automatic Workload Repository (AWR)

  Weitere Informationen finden Sie unter Generieren von Leistungsberichten mit Automatic Workload Repository (AWR).
- Active Data Guard mit maximaler Leistung in derselben AWS Region oder regionsübergreifend AWS

  Weitere Informationen finden Sie unter Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle.
- Blockchain-Tabellen (Oracle Database 21c und höher)

  Weitere Informationen finden Sie unter Verwalten von Blockchain-Tabellen in der Oracle-Database-Dokumentation.
- Kontinuierliche Abfragebenachrichtigung

  Weitere Informationen finden Sie unter Using Continuous Query Notification (CQN) in der Oracle-Dokumentation.
- Data Redaction
- Kontinuierliche Abfragebenachrichtigung

  Weitere Informationen finden Sie unter Database Change Notification in der Oracle-Dokumentation.
- In-Memory-Datenbank
- Verteilte Abfragen und Transaktionen
- Versionsbasierte Neudefinition

  Weitere Informationen finden Sie unter Einrichten der Standardversion für eine DB-Instance.
- EM Express (12c und höher)

  Weitere Informationen finden Sie unter Oracle Enterprise Manager.
- Detaillierte Überprüfung
- Flashback-Tabelle, Flashback-Abfrage und Flashback-Transaktionsabfrage
- Schrittweiser Passwort-Rollover für Anwendungen (Oracle Database 21c und höher)

Weitere Informationen finden Sie unter Verwalten des schrittweisen Datenbankpasswort-Rollovers für Anwendungen in der Oracle-Database-Dokumentation.

- HugePages

  Weitere Informationen finden Sie unter Aktivieren von HugePages für eine Instance von RDS für Oracle.

- Import/Export (Legacy und Data Pump) und SQL*Loader

  Weitere Informationen finden Sie unter Daten in Oracle auf Amazon importieren RDS.

- Java Virtual Machine (JVM)

  Weitere Informationen finden Sie unter Oracle Java Virtual Machine.

- JavaScript (Oracle Database 21c und höher)

  Weitere Informationen finden Sie unter DBMS_MLE in der Dokumentation zu Oracle Database.

- Label Security

  Weitere Informationen finden Sie unter Oracle Label Security.

- Ortung

  Weitere Informationen finden Sie unter Oracle Locator.

- Materialisierte Ansichten
- Multitenant

  Die Oracle-Multitenant-Architektur wird für alle Versionen von Oracle Database 19c und höher unterstützt. Weitere Informationen finden Sie unter Arbeiten mit CDBs in RDS für Oracle.

- Netzwerkverschlüsselung

  Weitere Informationen erhalten Sie unter Oracle Native Network Encryption und Oracle Secure Sockets Layer.

- Partitionierung
- Real Application Testing

  Um die vollen Aufnahme- und Wiedergabefunktionen nutzen zu können, müssen Sie Amazon Elastic File System (Amazon EFS) verwenden, um auf Dateien zuzugreifen, die von Oracle Real Application Testing generiert wurden. Weitere Informationen finden Sie unter EFSAmazon-

Integration und im Blogbeitrag Verwenden Sie die Funktionen von Oracle Real Application Testing mit Amazon RDS for Oracle.

- Sharding auf Anwendungsebene (aber nicht die Oracle Sharding-Funktion)
- Spatial and Graph

  Weitere Informationen finden Sie unter Oracle Spatial.
- Optimierung von Sternchen-Abfragen
- Streams und Advanced Queuing
- Verwaltung von Zusammenfassungen – Neuschreiben von materialisierten Ansichtsabfragen
- Text (Datei- und URL-Datastore-Typen werden nicht unterstützt)
- Total Recall
- Transparent Data Encryption (TDE)

  Weitere Informationen finden Sie unter Oracle Transparent Data Encryption.
- Unified Auditing, Mixed Mode

  Weitere Informationen finden Sie unter Mixed Mode Auditing in der Oracle-Dokumentation.
- XML-DB (ohne den XML-DB-Protokoll-Server)

  Weitere Informationen finden Sie unter Oracle XML DB.
- Virtual Private Database

Nicht unterstützte Funktionen in RDS for Oracle

Amazon RDS für Oracle unterstützt nicht die folgenden Oracle Database-Funktionen:

> **Note**
>
> Die folgende Liste erhebt keinen Anspruch auf Vollständigkeit.

- Automatic Storage Management (ASM)
- Database Vault
- Flashback Database

> **Note**
>
> Alternative Lösungen finden Sie im AWS Datenbank-Blogeintrag Alternativen zur Oracle-Flashback-Datenbankfunktion in Amazon RDS for Oracle.

- FTP und SFTP
- Hybride partitionierte Tabellen
- Messaging-Gateway
- Oracle Enterprise Manager Cloud Control Management Repository
- Real Application Clusters (Oracle RAC)
- Real Application Security (RAS)
- Unified Auditing, Pure Mode
- Workspace-Manager-Schema (WMSYS)

> **Warning**
>
> Im Allgemeinen hindert Sie Amazon RDS nicht daran, Schemata für nicht unterstützte Funktionen zu erstellen. Wenn Sie jedoch Schemata für Oracle-Funktionen und -Komponenten erstellen, die SYSDBA-Berechtigungen benötigen, können Sie das Data Dictionary beschädigen und die Verfügbarkeit Ihrer DB-Instance beeinträchtigen. Verwenden Sie nur unterstützte Funktionen und Schemata, die in Hinzufügen von Optionen zu Oracle DB-Instances verfügbar sind.

RDS für Oracle releases

RDS for Oracle for Oracle unterstützt mehrere Oracle Database-Versionen.

> **Note**
>
> Weitere Informationen zum Aktualisieren Ihrer Releases finden Sie unter Aktualisieren der DB-Engine von RDS für Oracle.

Themen

- Oracle Database 21c mit Amazon RDS
- Oracle Database 19c mit Amazon RDS

Oracle Database 21c mit Amazon RDS

Amazon RDS unterstützt Oracle Database 21c, einschließlich Oracle Enterprise Edition und Oracle Standard Edition 2. Die Oracle Database 21c (21.0.0.0) enthält viele neue Funktionen und Updates im Vergleich zur vorherigen Version. Eine wichtige Änderung besteht darin, dass Oracle Database 21c nur die mehrmandantenfähige Architektur unterstützt: Sie können eine Datenbank nicht mehr als traditionelle Nicht-CDB erstellen. Weitere Informationen zu den Unterschieden zwischen CDBs und Nicht-CDBs finden Sie unter Einschränkungen von RDS für Oracle CDBs.

In diesem Abschnitt finden Sie die wichtigsten Funktionen und Änderungen für die Verwendung von Oracle Database 21c (21.0.0.0) in Amazon RDS. Eine vollständige Liste der Änderungen finden Sie in der Dokumentation zu Oracle Database 21c. Eine vollständige Liste der von allen Oracle-Database-21c-Editionen unterstützten Funktionen finden Sie unter Permitted Features, Options, and Management Packs by Oracle Database Offering in der Oracle-Dokumentation.

Amazon RDS-Parameteränderungen für Oracle Database 21c (21.0.0.0)

Oracle Database 21c (21.0.0.0) enthält mehrere neue Parameter und Parameter mit neuen Bereichen und neuen Standardwerten.

Themen

- Neue Parameter
- Änderungen für den kompatiblen Parameter
- Parameter wurden entfernt

Neue Parameter

In der folgenden Tabelle werden die neuen Amazon-RDS-Parameter für Oracle Database 21c (21.0.0.0) dargestellt.

Name	Wertebereich	Standardwert	Anpass	Beschreibung
blockchain_table_max_no_drop	NONE | 0	NONE	Y	Ermöglicht es Ihnen, die maximale Leerlaufzeit zu

Name	Wertebereich	Standardwert	Anpassl	Beschreibung
				steuern, die beim Erstellen einer Blockchain-Tabelle angegeben werden kann.
dbnest_enable	NONE | CDB_RESOURCE_PDB_ALL	NONE	N	Ermöglicht es Ihnen, DBNest zu aktivieren oder zu deaktivieren. DbNest bietet Isolierung und Verwaltung von Betriebssystemressourcen, Dateisystemisolierung und sichere Datenverarbeitung für PDBs.
dbnest_pdb_fs_conf	NONE | *pathname*	NONE	N	Gibt die dbNest-Dateisystem-Konfigurationsdatei für eine PDB an.
diagnostics_control	ERROR | WARNING | IGNORE	IGNORE	Y	Ermöglicht es Ihnen, die Benutzer, die potenziell unsichere Datenbank diagnosevorgänge ausführen, zu kontrollieren und zu überwachen.
drcp_dedicated_opt	YES | NO	YES	Y	Aktiviert oder deaktiviert die Verwendung dedizierter Optimierung mit Database Resident Connection Pooling (DRCP).

Name	Wertebereich	Standardwert	Anpassl	Beschreibung
enable_per_pdb_drcp	`true | false`	`true`	N	Steuert, ob Database Resident Connection Pooling (DRCP) einen Verbindungspool für die gesamte CDB oder einen isolierten Verbindungspool für jede PDB konfiguriert.
inmemory_deep_vectorization	`true | false`	`true`	Y	Aktiviert oder deaktiviert das Deep Vectorization Framework.
mandatory_user_profile	*profile_name*	N/A	N	Gibt das obligatorische Benutzerprofil für eine CDB oder PDB an.
optimizer_capture_sql_quarantine	`true | false`	`false`	Y	Aktiviert oder deaktiviert das Deep Vectorization Framework.
optimizer_use_sql_quarantine	`true | false`	`false`	Y	Aktiviert oder deaktiviert die automatische Erstellung von SQL-Quarantine-Konfigurationen.
result_cache_execution_threshold	`0` auf `68719476736`	2	Y	Gibt an, wie oft eine PL/SQL-Funktion ausgeführt werden kann, bevor ihr Ergebnis im Ergebnis-Cache gespeichert wird.

Name	Wertebereich	Standardwert	Anpass	Beschreibung
result_cache_max_temp_result	0 auf 100	5	Y	Gibt den Prozentsatz von RESULT_CACHE_MAX_TEMP_SIZE an, den jedes einzelne, zwischengespeicherte Abfrageergebnis verbrauchen kann.
result_cache_max_temp_size	0 auf 2199023255552	RESULT_CACHE_SIZE * 10	Y	Gibt die maximale Menge an temporärem Tabellenraum (in Byte) an, die vom Ergebnis-Cache verbraucht werden kann.
sga_min_size	0 bis 2199023255552 (Maximalwert beträgt 50 % von sga_target)	0	Y	Gibt einen möglichen Mindestwert für die SGA-Verwendung einer steckbaren Datenbank (PDB) an.
tablespace_encryption_default_algorithm	GOST256 | SEED128 | ARIA256 | ARIA192 | ARIA128 | 3DES168 | AES256 | AES192 | AES128	AES128	Y	Gibt den Standardalgorithmus an, den die Datenbank beim Verschlüsseln eines Tabellenraums verwendet.

Änderungen für den kompatiblen Parameter

Der Parameter `compatible` hat einen neuen Maximalwert für Oracle Database 21c (21.0.0.0) in Amazon RDS. Die folgende Tabelle zeigt den neuen Standardwert.

Parametername	Oracle Database 21c (21.0.0.0) Maximalwert
compatible	21.0.0

Parameter wurden entfernt

Die folgenden Parameter wurden in Oracle Database 21c (21.0.0.0) entfernt:

- `remote_os_authent`
- `sec_case_sensitive_logon`
- `unified_audit_sga_queue_size`

Oracle Database 19c mit Amazon RDS

Amazon RDS unterstützt die Oracle Database 19c, die Oracle Enterprise Edition und Oracle Standard Edition Two umfasst.

Die Oracle Database 19c (19.0.0.0) enthält viele neue Funktionen und Updates im Vergleich zur vorherigen Version. In diesem Abschnitt finden Sie die wichtigsten Funktionen und Änderungen für die Verwendung von Oracle Database 19c (19.0.0.0) in Amazon RDS. Eine vollständige Liste der Änderungen finden Sie in der Dokumentation zu Oracle-Datenbank 19c. Eine vollständige Liste der von allen Oracle Database 19c-Editionen unterstützten Funktionen finden Sie unter Permitted Features, Options, and Management Packs by Oracle Database Offering in der Oracle-Dokumentation.

Amazon RDS-Parameteränderungen für Oracle Database 19c (19.0.0.0)

Oracle Database 19c (19.0.0.0) enthält mehrere neue Parameter und Parameter mit neuen Bereichen und neuen Standardwerten.

Themen

- Neue Parameter

- Änderungen am kompatiblen Parameter
- Parameter wurden entfernt

Neue Parameter

In der folgenden Tabelle werden die neuen Amazon RDS-Parameter für Oracle Database 19c (19.0.0.0) dargestellt.

Name	Werte	Anpas	Beschreibung
lob_signature_enable	TRUE, FALSE (Standard)	Y	Aktiviert oder deaktiviert die LOB Locator Signature-Funktion.
max_datapump_parallel_per_job	1 bis 1024 oder AUTO	Y	Gibt die maximale Anzahl zulässiger paralleler Prozesse für jeden Oracle Data Pump-Job an.

Änderungen am kompatiblen Parameter

Der Parameter `compatible` hat einen neuen Maximalwert für Oracle Database 19c (19.0.0.0) in Amazon RDS. Die folgende Tabelle zeigt den neuen Standardwert.

Parametername	Oracle Database 19c (19.0.0.0) Maximalwert
compatible	19.0.0

Parameter wurden entfernt

Die folgenden Parameter wurden in Oracle Database 19c (19.0.0.0) entfernt:

- `exafusion_enabled`
- `max_connections`
- `o7_dictionary_access`

RDSfür Oracle-Lizenzoptionen

Amazon RDS for Oracle bietet zwei Lizenzoptionen: License Included (LI) und Bring Your Own License (BYOL). Nachdem Sie eine Oracle-DB-Instance auf Amazon erstellt habenRDS, können Sie das Lizenzmodell ändern, indem Sie die DB-Instance ändern. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Important**
>
> Stellen Sie sicher, dass Sie über die entsprechende Oracle Database-Lizenz mit Software Update License and Support für Ihre DB-Instance-Klasse und Oracle Database-Edition verfügen. Stellen Sie außerdem sicher, dass Sie über Lizenzen für alle separat lizenzierten Funktionen von Oracle Database verfügen.

Themen

- Modell im Lieferumfang enthalten für SE2
- Bringen Sie Ihre eigene Lizenz (BYOL) für EE mit und SE2
- Lizenzieren von Oracle-Multi-AZ-Bereitstellungen

Modell im Lieferumfang enthalten für SE2

Beim Modell „Lizenz enthalten“ müssen Sie Oracle Database-Lizenzen nicht separat erwerben. AWS besitzt die Lizenz für die Oracle-Datenbanksoftware. Das Modell „Lizenz enthalten“ wird nur auf Amazon RDS für Oracle Database Standard Edition 2 (SE2) unterstützt.

Wenn Sie bei diesem Modell über ein AWS -Support Konto beim Fallsupport verfügen, wenden Sie sich Support für Serviceanfragen sowohl an Amazon RDS als auch an Oracle Database. Ihre Nutzung der LI-Option RDS für Oracle unterliegt Abschnitt 10.3.1 der AWS Servicebedingungen.

Bringen Sie Ihre eigene Lizenz (BYOL) für EE mit und SE2

In dem BYOL Modell können Sie Ihre vorhandenen Oracle Database-Lizenzen verwenden, um Datenbanken auf Amazon bereitzustellenRDS. Amazon RDS unterstützt das BYOL Modell nur für Oracle Database Enterprise Edition (EE) und Oracle Database Standard Edition 2 (SE2).

Stellen Sie sicher, dass Sie eine entsprechende Oracle-Datenbank-Lizenz für die DB-Instance-Klasse und die Oracle-Datenbank-Edition besitzen, die Sie ausführen möchten (mit der Lizenz für

Software-Updates und Support). Außerdem müssen Sie die Oracle-Richtlinien für die Lizenzierung von Oracle Database Software in der Cloud Computing-Umgebung befolgen. Weitere Informationen zur Lizenzpolitik von Oracle für Amazon EC2 finden Sie unter Lizenzierung von Oracle-Software in der Cloud-Computing-Umgebung.

In diesem Modell werden Sie Ihr aktives Oracle-Supportkonto weiter verwenden und für spezifische Serviceanfragen zu Oracle-Datenbanken, Oracle direkt kontaktieren. Wenn Sie ein AWS -Support Konto beim Fallsupport haben, können Sie sich Support bei RDS Problemen mit Amazon an uns wenden. Amazon Web Services und Oracle verfügen über ein Multi-Vendor-Support-Verfahren für Fälle, bei denen Unterstützung von beiden Organisationen benötigt wird.

Integration mit AWS License Manager

Um die Überwachung der Oracle-Lizenznutzung im BYOL Modell zu vereinfachen, ist es in Amazon RDS for Oracle AWS License Managerintegriert. License Manager unterstützt die Nachverfolgung von RDS Oracle-Engine-Editionen und Lizenzpaketen auf Basis virtueller Kerne (vCPUs). Sie können License Manager auch verwenden AWS Organizations , um alle Ihre Unternehmenskonten zentral zu verwalten.

Die folgende Tabelle zeigt die Produktinformationsfilter RDS für Oracle.

Filter	Name	Beschreibung
Engine-Edition	`oracle-ee`	Oracle Database Enterprise Edition (EE)
	`oracle-se2`	Oracle Database Standard Edition 2 (SE2)
Lizenzpaket	`data guard`	Siehe Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle (Oracle Active Data Guard)
	`olap`	Siehe Oracle OLAP
	`ols`	Siehe Oracle Label Security
	`diagnostic pack sqlt`	Siehe Orakel SQLT
	`tuning pack sqlt`	Siehe Orakel SQLT

Um die Lizenznutzung Ihrer Oracle-DB-Instances nachzuverfolgen, können Sie eine selbstverwaltete Lizenz mithilfe von AWS License Manager erstellen. In diesem Fall werden RDS für Oracle-Ressourcen, die dem Produktinformationsfilter entsprechen, automatisch der selbstverwalteten Lizenz zugeordnet. Die Erkennung von Oracle-DB-Instances kann bis zu 24 Stunden dauern. Sie können eine Lizenz auch kontenübergreifend verfolgen, indem Sie AWS Resource Access Manager

Konsole

Um eine selbstverwaltete Lizenz zu erstellen AWS License Manager , um die Lizenznutzung Ihrer RDS Oracle-DB-Instances nachzuverfolgen

1. Wechseln Sie zu https://console.aws.amazon.com/license-manager/.
2. Wählen Sie Selbstverwaltete Lizenz erstellen.

   Anweisungen finden Sie im AWS License Manager Benutzerhandbuch unter Eine selbstverwaltete Lizenz erstellen.

   Fügen Sie im Bereich RDSProduktinformationen eine Regel für einen Produktinformationsfilter hinzu.

   Weitere Informationen finden Sie ProductInformationin der AWS License Manager APIReferenz.
3. (Nur kontoübergreifendes Tracking) Verwenden Sie diese AWS Resource Access Manager Option, um Ihre selbstverwalteten Lizenzen mit oder über AWS ein beliebiges Konto zu teilen. AWS Organizations Weitere Informationen finden Sie unter Ressourcen teilen AWS.

AWS CLI

Rufen Sie den create-license-configurationBefehl auf AWS CLI, um mit dem eine selbstverwaltete Lizenz zu erstellen. Verwenden Sie die Parameter `--cli-input-json` oder `--cli-input-yaml`, um die Parameter an den Befehl zu übergeben.

Example

Im folgenden Beispiel wird eine selbstverwaltete Lizenz für Oracle Enterprise Edition erstellt.

```
aws license-manager create-license-configuration --cli-input-json file://rds-oracle-
ee.json
```

Im Folgenden finden Sie die Beispieldatei `rds-oracle-ee.json`, die im Beispiel verwendet wird.

```
{
    "Name": "rds-oracle-ee",
    "Description": "RDS Oracle Enterprise Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["oracle-ee"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Weitere Informationen zu Produktinformationen finden Sie unter Automatisiertes Erkennen des Ressourcenbestands im AWS License Manager -Benutzerhandbuch.

Weitere Informationen zu dem `--cli-input` Parameter finden Sie im AWS CLI User Guide unter Generieren von AWS CLI Skelett- und YAML Eingabeparametern aus einer JSON Oder-Eingabedatei.

Migration zwischen Oracle Database-Editionen

Wenn Sie über eine ungenutzte BYOL Oracle Database-Lizenz verfügen, die für die Edition und Klasse der DB-Instance, die Sie ausführen möchten, geeignet ist, können Sie von Standard Edition 2 (SE2) auf Enterprise Edition (EE) migrieren. Sie können nicht von EE zu anderen Editionen migrieren.

Um Ihre Oracle Database-Edition zu ändern und Ihre Daten beizubehalten

1. Erstellen Sie einen Snapshot der DB-Instance.

   Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
2. Stellen Sie den Snapshot auf einer neuen DB-Instance wieder her und wählen Sie die zu verwendende Edition der Oracle-Datenbank aus.

   Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.

3. (Optional) Löschen Sie die alte DB-Instance, sofern sie nicht weiter ausgeführt werden soll und Sie über die entsprechenden Oracle Datenbank-Lizenzen dafür verfügen.

   Weitere Informationen finden Sie unter Löschen einer DB-Instance.

Lizenzieren von Oracle-Multi-AZ-Bereitstellungen

Amazon RDS unterstützt Multi-AZ-Bereitstellungen für Oracle als hochverfügbare Failover-Lösung. Wir empfehlen für Produktions-Workloads Multi-AZ-Bereitstellungen. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Wenn Sie das Modell "Verwendung der eigenen Lizenz" verwenden, müssen Sie bei einer Multi-AZ-Bereitstellung sowohl für die primäre DB-Instance als auch für die Standby-DB-Instance eine Lizenz besitzen.

RDS für Oracle-Benutzer und -Berechtigungen

Wenn Sie eine DB-Instance von Amazon RDS für Oracle erstellen, hat der Standard-Hauptbenutzer die meisten maximalen Benutzerberechtigungen für die DB-Instance. Verwenden Sie das Hauptbenutzerkonto für alle administrativen Aufgaben, wie zum Beispiel das Erstellen von zusätzlichen Benutzerkonten in Ihrer Datenbank. Da es sich bei RDS um einen verwalteten Service handelt, dürfen Sie sich nicht als `SYS` und `SYSTEM` anmelden und verfügen daher nicht über `SYSDBA`-Berechtigungen.

Themen

- Beschränkungen für Oracle DBA-Berechtigungen
- So verwalten Sie Berechtigungen für SYS-Objekte

Beschränkungen für Oracle DBA-Berechtigungen

In der Datenbank ist eine Rolle eine Sammlung von Sonderrechten, die Sie einem Benutzer gewähren oder entziehen können. Eine Oracle-Datenbank verwendet Rollen, um Sicherheit zu gewährleisten. Weitere Informationen finden Sie unter Configuring Privilege and Role Authorization (Konfiguration von Berechtigungen und Rollenautorisierung) in der Oracle-Database-Dokumentation.

Die vordefinierte Rolle `DBA` erteilt normalerweise alle administrativen Rechte für eine Oracle-Datenbank-Engine. Wenn Sie eine DB-Instance erstellen, erhält das Hauptbenutzerkonto DBA-

Berechtigungen (mit einigen Einschränkungen). Um eine Verwaltung zu ermöglichen, bietet eine RDS für Oracle-Datenbank nicht die folgenden Berechtigungen für die DBA-Rolle:

- `ALTER DATABASE`
- `ALTER SYSTEM`
- `CREATE ANY DIRECTORY`
- `DROP ANY DIRECTORY`
- `GRANT ANY PRIVILEGE`
- `GRANT ANY ROLE`

Weitere Informationen zu Systemberechtigungen und Rollen von RDS für Oracle finden Sie unter Berechtigungen von Hauptbenutzerkonten.

So verwalten Sie Berechtigungen für SYS-Objekte

Sie können Berechtigungen für SYS-Objekte mithilfe des `rdsadmin.rdsadmin_util`-Pakets verwalten. Wenn Sie beispielsweise den Datenbankbenutzer `myuser` erstellen, könnten Sie das Verfahren `rdsadmin.rdsadmin_util.grant_sys_object` verwenden, um `myuser` `SELECT`-Berechtigungen für `V_$SQLAREA` zu erteilen. Weitere Informationen finden Sie unter den folgenden Themen:

- Gewährung SELECT von EXECUTE Rechten für SYS Objekte
- Widerrufen von SELECT EXECUTE Rechten für Objekte SYS
- Erteilen von Berechtigungen an Nicht-Hauptbenutzer

RDS für Oracle-DB-Instance-Klassen

Die Berechnungs- und Speicherkapazität einer RDS für Oracle-DB-Instance wird durch ihre Instance-Klasse bestimmt. Die benötigte DB-Instance-Klasse richtet sich nach Ihren Rechen- und Speicheranforderungen.

Unterstützte RDS für Oracle-DB-Instance-Klassen

Die unterstützten Oracle-Instance-Klassen sind eine Teilmenge der RDS-DB-Instance-Klassen. Eine vollständige Liste der RDS-Instance-Klassen finden Sie unteraus.

Vorkonfigurierte DB-Instance-Klassen von RDS für Oracle

RDS for Oracle bietet auch Instanzklassen, die für Workloads vorkonfiguriert sind, die zusätzlichen Arbeitsspeicher, Speicherplatz und I/O pro vCPU benötigen. Diese Instance-Klassen verwenden die folgende Namenskonvention:

```
db.r5b.instance_size.tpcthreads_per_core.memratio
db.r5.instance_size.tpcthreads_per_core.memratio
```

Im Folgenden finden Sie ein Beispiel für eine Instance-Klasse, die für zusätzlichen Speicher vorkonfiguriert ist:

```
db.r5b.4xlarge.tpc2.mem2x
```

Die Komponenten des vorhergehenden Instance-Klassennamens lauten wie folgt:

- `db.r5b.4xlarge`— Der Name der Instance-Klasse.
- `tpc2`— Die Threads pro Kern. Der Wert 2 bedeutet, dass Multithreading aktiviert ist. Ein Wert von 1 bedeutet, dass Multithreading ausgeschaltet ist.
- `mem2x`— Das Verhältnis von zusätzlichem Speicher zum Standardspeicher für die Instance-Klasse. In diesem Beispiel bietet die Optimierung doppelt so viel Speicher wie eine standardmäßige db.r5.4xlarge-DB-Instance.

Unterstützte Kombinationen aus Edition, Instance-Klasse und Lizenzen in RDS for Oracle

Wenn Sie die RDS-Konsole verwenden, können Sie herausfinden, ob eine bestimmte Kombination aus Edition, Instanzklasse und Lizenz unterstützt wird, indem Sie Datenbank erstellen wählen und eine andere Option angeben. In der AWS CLI können Sie den folgenden Befehl ausführen:

```
aws rds describe-orderable-db-instance-options --engine engine-type --license-
model license-type
```

In der folgenden Tabelle sind alle Editionen, Instanzklassen und Lizenztypen aufgeführt, die für RDS for Oracle unterstützt werden. Weitere Informationen zu den Speicherattributen der einzelnen Typen erhalten Sie unter RDS-für-Oracle-Instance-Typen. Informationen zur Preisgestaltung finden Sie unter Preismodelle von Amazon RDS for Oracle.

Oracle Edition	Oracle Database 19c und höher
Enterprise Edition (EE)	Standard-Instance-Klassen
Bring Your Own License (BYOL)	db.m7i.large — db.m7i.48xlarge db.m6in.large — db.m6in.32xlarge db.m6id.large — db.m6id.32xlarge db.m6i.large — db.m6i.32xlarge db.m5d.large–db.m5d.24xlarge db.m5.large–db.m5.24xlarge
	Arbeitsspeicheroptimierte Instance-Klassen
	db.r7i.large — db.r7i.48xlarge db.r6in.large — db.r6in.32xlarge db.r6id.large — db.r6id.32xlarge db.r6i.large–db.r6i.32xlarge db.r5d.large–db.r5d.24xlarge db.r5b.large–db.r5b.24xlarge db.r5.large–db.r5.24xlarge db.x2iedn.xlarge–db.x2iedn.32xlarge db.x2iezn.2xlarge–db.x2iezn.12xlarge db.x2idn.16xlarge–db.x2idn.32xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge — db.x1.32xlarge 1

Oracle Edition	Oracle Database 19c und höher
	db.z1d.large–db.z1d.12xlarge
	Speicheroptimierte vorkonfigurierte Instanzklassen

Oracle Edition	Oracle Database 19c und höher
	db.r6i.8xlarge.tpc2.mem4x
	db.r6i.8xlarge.tpc2.mem3x
	db.r6i.6xlarge.tpc2.mem4x
	db.r6i.4xgroß.tpc2.mem4x
	db.r6i.4xlarge.tpc2.mem3x
	db.r6i.4xlarge.tpc2.mem2x
	db.r6i.2xgroß.tpc2.mem8x
	db.r6i.2xlarge.tpc2.mem4x
	db.r6i.2xgroß.tpc1.mem2x
	db.r6i.xlarge.tpc2.mem4x
	db.r6i.xlarge.tpc2.mem2x
	db.r6i.large.tpc1.mem2x
	db.r5b.8xlarge.tpc2.mem3x
	db.r5b.6xlarge.tpc2.mem4x
	db.r5b.4xlarge.tpc2.mem4x
	db.r5b.4xlarge.tpc2.mem3x
	db.r5b.4xlarge.tpc2.mem2x
	db.r5b.2xlarge.tpc2.mem8x
	db.r5b.2xlarge.tpc2.mem4x
	db.r5b.2xlarge.tpc1.mem2x
	db.r5b.xlarge.tpc2.mem4x

Oracle Edition	Oracle Database 19c und höher
	db.r5b.xlarge.tpc2.mem2x
	db.r5b.large.tpc1.mem2x
	db.r5.12xlarge.tpc2.mem2x
	db.r5.8xlarge.tpc2.mem3x
	db.r5.6xlarge.tpc2.mem4x
	db.r5.4xlarge.tpc2.mem4x
	db.r5.4xlarge.tpc2.mem3x
	db.r5.4xlarge.tpc2.mem2x
	db.r5.2xlarge.tpc2.mem8x
	db.r5.2xlarge.tpc2.mem4x
	db.r5.2xlarge.tpc1.mem2x
	db.r5.xlarge.tpc2.mem4x
	db.r5.xlarge.tpc2.mem2x
	db.r5.large.tpc1.mem2x
	Instance-Klassen mit Spitzenleistung
	db.t3.small–db.t3.2xlarge

Oracle Edition	Oracle Database 19c und höher
Standardausgabe (2) SE2	Standard-Instance-Klassen
Bring Your Own License (BYOL)	db.m7i.large — db.m7i.4xlarge db.m6in.large — db.m6in.4xlarge db.m6id.large — db.m6id.4xlarge db.m6i.large — db.m6i.4xlarge db.m5d.large–db.m5d.4xlarge db.m5.large–db.m5.4xlarge
	Arbeitsspeicheroptimierte Instance-Klassen
	db.r7i.large — db.r7i.4xlarge db.r6in.large — db.r6in.4xlarge db.r6id.large — db.r6id.4xlarge db.r6i.large–db.r6i.4xlarge db.r5d.large–db.r5d.4xlarge db.r5b.large–db.r5b.4xlarge db.r5.large–db.r5.4xlarge db.x2iedn.xlarge–db.x2iedn.4xlarge db.x2iezn.2xlarge–db.x2iezn.4xlarge db.z1d.large–db.z1d.3xlarge
	Speicheroptimierte vorkonfigurierte Instanzklassen

Oracle Edition	Oracle Database 19c und höher
	db.r6i.4xlarge.tpc2.mem4x
	db.r6i.4xlarge.tpc2.mem3x
	db.r6i.4xlarge.tpc2.mem2x
	db.r6i.2xgroß.tpc2.mem8x
	db.r6i.2xlarge.tpc2.mem4x
	db.r6i.2xgroß.tpc1.mem2x
	db.r6i.xlarge.tpc2.mem4x
	db.r6i.xlarge.tpc2.mem2x
	db.r6i.large.tpc1.mem2x
	db.r5b.4xlarge.tpc2.mem4x
	db.r5b.4xlarge.tpc2.mem3x
	db.r5b.4xlarge.tpc2.mem2x
	db.r5b.2xlarge.tpc2.mem8x
	db.r5b.2xlarge.tpc2.mem4x
	db.r5b.2xlarge.tpc1.mem2x
	db.r5b.xlarge.tpc2.mem4x
	db.r5b.xlarge.tpc2.mem2x
	db.r5b.large.tpc1.mem2x
	db.r5.4xlarge.tpc2.mem4x
	db.r5.4xlarge.tpc2.mem3x
	db.r5.4xlarge.tpc2.mem2x

Oracle Edition	Oracle Database 19c und höher
	db.r5.2xlarge.tpc2.mem8x
	db.r5.2xlarge.tpc2.mem4x
	db.r5.2xlarge.tpc1.mem2x
	db.r5.xlarge.tpc2.mem4x
	db.r5.xlarge.tpc2.mem2x
	db.r5.large.tpc1.mem2x
	Instance-Klassen mit Spitzenleistung
	db.t3.small–db.t3.2xlarge
Standardausgabe (2) SE2	Standard-Instance-Klassen
Lizenz enthalten	db.m5.large–db.m5.4xlarge
	Arbeitsspeicheroptimierte Instance-Klassen
	db.r6i.large–db.r6i.4xlarge
	db.r5.large–db.r5.4xlarge
	Instance-Klassen mit Spitzenleistung
	db.t3.small–db.t3.2xlarge

[1] Sie können mit der X1.*-Instance-Klassenfamilie kein RDS mehr für Oracle-DB-Instances erstellen (X1E.*-Instance-Klassen werden weiterhin unterstützt). Wenn Sie derzeit X1.*-Klassen verwenden, wechseln Sie so bald wie möglich zu einer Instance-Klasse der neuen Generation. Ab dem 22. Januar 2025 beginnt RDS mit automatisierten Upgrades in Ihrem definierten Wartungsfenster. Während des Upgrades wählt RDS den entsprechenden X2IEDN-Instanztyp und aktualisiert ihn. Weitere Informationen finden Sie im re:Post-Artikel Amazon RDS for Oracle beendet den Support für X1-Datenbankinstanzen am 22. Januar 2025.

> **Note**
>
> Wir empfehlen allen Kunden, die eine eigene Lizenz verwenden, in ihrer Lizenzvereinbarung nachzulesen, welche Konsequenzen die Außerbetriebnahme von Amazon RDS for Oracle hat. Weitere Informationen zur Rechenkapazität der DB-Instance-Klassen, die RDS für Oracle unterstützt, finden Sie unter und Konfigurieren des Prozessors für eine DB-Instance-Klasse in RDS für Oracle.

> **Note**
>
> Wenn Sie DB-Snapshots von DB-Instances haben, die veraltete DB-Instance-Klassen verwendet haben, können Sie beim Wiederherstellen des DB-Snapshots eine DB-Instance-Klasse auswählen, die nicht veraltet ist. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.

Veraltetes RDS für Oracle-DB-Instance-Klassen

Im Folgenden finden Sie die DB-Instance-Klassen, die für RDS für Oracle veraltet sind:

- db.m1, db.m2, db.m3, db.m4
- db.t1, db.t2
- db.r1, db.r2, db.r3, db.r4

Diese DB-Instance-Klassen wurden durch bessere DB-Instance-Klassen ersetzt, die allgemein und zu geringeren Kosten verfügbar sind. Wenn Sie DB-Instances haben, die veraltete DB-Instance-Klassen verwenden, stehen Ihnen folgende Optionen zur Verfügung:

- Erlauben Sie Amazon RDS, jede DB-Instance automatisch zu ändern, um eine vergleichbare, nicht veraltete DB-Instance-Klasse zu verwenden. Zeitpläne für das Veralten von Versionen finden Sie unter DB-Instance-Klassenarten.
- Ändern Sie die DB-Instance-Klasse selbst, indem Sie die DB-Instance modifizieren. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Wenn Sie DB-Snapshots von DB-Instances haben, die veraltete DB-Instance-Klassen verwendet haben, können Sie beim Wiederherstellen des DB-Snapshots eine DB-Instance-Klasse auswählen, die nicht veraltet ist. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.

RDS für Oracle-Datenbankarchitektur

Die Oracle-Multitenant-Architektur, auch als CDB-Architektur bekannt, ermöglicht es einer Oracle-Datenbank, als Multitenant-Container-Datenbank (CDB) zu fungieren. Eine CDB kann vom Kunden erstellte austauschbare Datenbanken () enthalten. PDBs Eine Nicht-CDB ist eine Oracle-Datenbank, die die traditionelle Architektur verwendet, die nicht enthalten kann. PDBs Weitere Informationen über die Mandanten-Architektur finden Sie im Oracle Multitenant Administrator's Guide.

Für Oracle Database 19c und höher können Sie eine DB-Instance von RDS für Oracle erstellen, die die CDB-Architektur verwendet. In RDS für Oracle PDBs werden sie als Mandantendatenbanken bezeichnet. Ihre Client-Anwendungen stellen eine Verbindung auf Mandantendatenbank-Ebene (PDB) statt auf CDB-Ebene her. RDS für Oracle unterstützt die folgenden Konfigurationen der CDB-Architektur:

Multi-Tenant-Konfiguration

Amazon RDS ermöglicht es einer CDB-Instance, je nach Datenbankedition und erforderlichen Optionslizenzen zwischen 1 und 30 Tenant-Datenbanken zu enthalten. Sie können RDS verwenden, APIs um Mandantendatenbanken hinzuzufügen, zu ändern und zu entfernen. Die Mehrmandantenkonfiguration in RDS für Oracle unterstützt keine Anwendungen PDBs oder Proxys PDBs, bei denen es sich um spezielle Typen handelt. PDBs Weitere Informationen zu Anwendung PDBs und Proxy PDBs finden Sie PDBs in der Dokumentation zur Oracle-Datenbank unter Types von.

> **Note**
>
> Die Amazon RDS-Konfiguration wird als „Multi-Tenant“ und nicht als „Multitenant“ bezeichnet, da es sich um eine Funktion von Amazon RDS handelt, nicht nur um die Oracle-DB-Engine. In ähnlicher Weise bezieht sich der RDS-Begriff „Mandant“ auf jeden Mandanten in einer RDS-Konfiguration, nicht nur auf Oracle. PDBs In der RDS-Dokumentation bezieht sich der Begriff „Oracle Multitenant“ ohne Bindestrich ausschließlich auf die CDB-Architektur der Oracle-Datenbank, die sowohl mit lokalen als auch mit RDS-Bereitstellungen kompatibel ist.

Single-Tenant-Konfiguration

Amazon RDS beschränkt eine RDS for Oracle CDB-Instance auf eine Tenant-Datenbank (PDB). Mit RDS können Sie keine weiteren PDBs hinzufügen. APIs Die Single-Tenant-Konfiguration verwendet dasselbe RDS APIs wie die Nicht-CDB-Architektur. Daher ist die Erfahrung bei der Arbeit mit einer CDB in der Single-Tenant-Konfiguration weitgehend dieselbe wie bei der Arbeit mit einer Nicht-CDB.

Sie können eine CDB, die die Single-Tenant-Konfiguration verwendet, in eine Multi-Tenant-Konfiguration konvertieren, sodass Sie sie zu Ihrer CDB hinzufügen können. PDBs Diese Architekturänderung ist dauerhaft und irreversibel. Weitere Informationen finden Sie unter Konvertieren der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration.

> **Note**
>
> Sie können nicht auf die CDB selbst zugreifen.

In Oracle Database 21c und höher sind alle Datenbanken. CDBs Im Gegensatz dazu können Sie eine DB-Instance von Oracle Database 19c entweder als CDB oder als Nicht-CDB erstellen. Sie können eine Nicht-CDB nicht auf eine CDB aktualisieren. Sie können jedoch eine Nicht-CDB von Oracle Database 19c in eine CDB konvertieren und anschließend aktualisieren. Eine CDB können Sie nicht in eine Nicht-CDB konvertieren.

Weitere Informationen finden Sie in den folgenden Ressourcen:

- Arbeiten mit CDBs in RDS für Oracle
- Einschränkungen von RDS für Oracle CDBs
- Erstellen einer Amazon RDS-DB-Instance

Initialisierungsparameter für RDS für Oracle

In Amazon RDS verwalten Sie Parameter mithilfe einer DB-Parametergruppe. Mithilfe dieser Gruppe können Sie die Initialisierungsparameter anpassen. Sie können beispielsweise die Größe des Sortierbereichs mit `sort_area_size` konfigurieren. Alle RDS for Oracle DB-Instances, die einer bestimmten DB-Parametergruppe zugeordnet sind, verwenden dieselben Parametereinstellungen. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

Unterstützte Initialisierungsparameter in RDS for Oracle

Die unterstützten Parameter für Ihre DB-Instance hängen von Ihrer Oracle Database-Edition und -Version ab. Um die unterstützten Initialisierungsparameter für eine bestimmte Oracle Database-Edition und -Version anzuzeigen, führen Sie den AWS CLI Befehl `describe-engine-default-parameters`aus. Um beispielsweise die Namen der unterstützten Initialisierungsparameter für die Enterprise Edition von Oracle Database 19c aufzulisten, führen Sie den folgenden Befehl aus (einschließlich Beispielausgabe).

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family oracle-ee-19 \
    --output json | jq -r '.EngineDefaults.Parameters[].ParameterName'

_add_col_optim_enabled
_adg_parselock_timeout
_allow_insert_with_update_check
_allow_level_without_connect_by
_always_semi_join
_autotask_max_window
_awr_disabled_flush_tables
_awr_mmon_cpuusage
_awr_mmon_deep_purge_all_expired
_b_tree_bitmap_plans
_bct_bitmaps_per_file
_bloom_filter_enabled
_buffered_publisher_flow_control_threshold
_bug29394014_allow_triggers_on_vpd_table
_cleanup_rollback_entries
_client_enable_auto_unregister
_clusterwide_global_transactions
_complex_view_merging
_connect_by_use_union_all
_cost_equality_semi_join
_cursor_features_enabled
_cursor_obsolete_threshold
_datafile_write_errors_crash_instance
_db_block_buffers
...
```

Um einen einzelnen Initialisierungsparameter zu beschreiben, verwenden Sie den folgenden Befehl und `sga_max_size` ersetzen Sie ihn durch den Namen Ihres Parameters (einschließlich Beispielausgabe).

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family oracle-ee-19 \
    --query 'EngineDefaults.Parameters[?ParameterName==`sga_max_size`]' \
    --output json

[
    {
        "ParameterName": "sga_max_size",
        "Description": "max total SGA size",
        "Source": "engine-default",
        "ApplyType": "static",
        "DataType": "integer",
        "AllowedValues": "0-2199023255552",
        "IsModifiable": true
    }
]
```

Allgemeine Dokumentation zu den Initialisierungsparametern der Oracle-Datenbank finden Sie unter Initialisierungsparameter in der Oracle-Datenbank-Dokumentation. Beachten Sie, dass für den Parameter `ARCHIVE_LAG_TARGET` in RDS for Oracle besondere Überlegungen erforderlich sind. Dieser Parameter erzwingt einen Online-Redo-Log-Switch nach Ablauf der angegebenen Zeit. `ARCHIVE_LAG_TARGET`Ist in RDS für Oracle auf eingestellt, `300` weil das Recovery Point Objective (RPO) 5 Minuten beträgt. Um dieses Ziel zu erreichen, wechselt RDS for Oracle das Online-Redo-Log alle 5 Minuten und speichert es in einem Amazon S3 S3-Bucket.

Wenn die Häufigkeit des Online-Redo-Log-Switches die Leistung Ihrer RDS for Oracle-Datenbank beeinträchtigt, können Sie Ihre DB-Instance und Ihren Speicher skalieren, um höhere IOPS und einen höheren Durchsatz zu nutzen. Wenn Sie RDS Custom for Oracle verwenden oder eine Oracle-Datenbank auf Amazon bereitstellen EC2, können Sie alternativ die Einstellung des `ARCHIVE_LAG_TARGET` Initialisierungsparameters anpassen.

Gültige Parameterwerte in RDS für Oracle

In RDS for Oracle sind nur die folgenden Zeichen für Parameterwerte gültig:

- Buchstaben (`A-Zunda-z`)
- Zahlen (`0-9`)
- Leerzeichen (Leerzeichen, Tabulatoren und Zeilenumbrüche)
- Die folgenden Sonderzeichen: `_ / . : + = ( ) ' * , % $ -` (Bindestrich)

RDS for Oracle-Zeichensätze

RDS for Oracle unterstützt zwei Arten von Zeichensätzen: den DB-Zeichensatz und den nationalen Zeichensatz.

DB-Zeichensatz

Der Zeichensatz der Oracle-Datenbank wird in den Datentypen `CHAR`, `VARCHAR2` und `CLOB` verwendet. Die Datenbank verwendet diesen Zeichensatz auch für Metadaten wie Tabellennamen, Spaltennamen und SQL-Anweisungen. Der Zeichensatz der Oracle-Datenbank wird normalerweise als DB-Zeichensatz bezeichnet.

Sie können den Zeichensatz beim Erstellen einer DB-Instance einstellen. Sie können den DB-Zeichensatz nicht ändern, nachdem Sie die Datenbank erstellt haben.

Unterstützte DB-Zeichensätze

In der folgenden Tabelle werden die in Amazon RDS unterstützten Zeichensätze der Oracle-Datenbank aufgelistet. Sie können einen Wert aus dieser Tabelle mit dem Parameter `--character-set-name` des CLI-Befehls AWS CLI create-db-instance oder mit dem Parameter `CharacterSetName` der Amazon-RDS-API-Operation CreateDBInstance verwenden.

> **Note**
>
> Der Zeichensatz für eine CDB lautet immer AL32UTF8. Sie können nur für die PDB einen anderen Zeichensatz festlegen.

Value	Beschreibung
AL32UTF8	Unicode 5.0 UTF-8 Universeller Zeichensatz (Standard)
AR8ISO8859P6	ISO 8859-6 Lateinisch/Arabisch
AR8MSWIN1256	Microsoft Windows Code Page 1256 8-bit Lateinisch/Arabisch
BLT8ISO8859P13	ISO 8859-13 Baltisch

Value	Beschreibung
BLT8MSWIN1257	Microsoft Windows Code Page 1257 8-bit Baltisch
CL8ISO8859P5	ISO 88559-5 Lateinisch/Kyrillisch
CL8MSWIN1251	Microsoft Windows Code Page 1251 8-bit Lateinisch/Kyrillisch
EE8ISO8859P2	ISO 8859-2 Osteuropäisch
EL8ISO8859P7	ISO 8859-7 Lateinisch/Griechisch
EE8MSWIN1250	Microsoft Windows Code Page 1250 8-bit Osteuropäisch
EL8MSWIN1253	Microsoft Windows Code Page 1253 8-bit Lateinisch/Griechisch
IW8ISO8859P8	ISO 8859-8 Lateinisch/Hebräisch
IW8MSWIN1255	Microsoft Windows Code Page 1255 8-bit Lateinisch/Hebräisch
JA16EUC	EUC 24-bit Japanisch
JA16EUCTILDE	Entspricht JA16EUC, außer Mapping von Wave Dash und Tilde an und aus Unicode.
JA16SJIS	Shift-JIS 16-bit Japanisch
JA16SJISTILDE	Entspricht JA16SJIS, außer Mapping von Wave Dash und Tilde an und aus Unicode.
KO16MSWIN949	Microsoft Windows Code Page 949 Koreanisch
NE8ISO8859P10	ISO 8859-10 Nordeuropäisch
NEE8ISO8859P4	ISO 8859-4 Nord- und Nordosteuropäisch

Value	Beschreibung
TH8TISASCII	Thai Industrial Standard 620-2533-ASCII 8-bit
TR8MSWIN1254	Microsoft Windows Code Page 1254 8-bit Türkisch
US7ASCII	ASCII 7-bit Amerikanisch
UTF8	Unicode 3.0 UTF-8 Universeller Zeichensatz, CESU-8 konform
VN8MSWIN1258	Microsoft Windows Code Page 1258 8-bit Vietnamesisch
WE8ISO8859P1	Westeuropäisch 8-bit ISO 8859 Teil 1
WE8ISO8859P15	ISO 8859-15 Westeuropäisch
WE8ISO8859P9	ISO 8859-9 Westeuropäisch und Türkisch
WE8MSWIN1252	Microsoft Windows Code Page 1252 8-bit Westeuropäisch
ZHS16GBK	GBK 16-bit Vereinfachtes Chinesisch
ZHT16HKSCS	Microsoft Windows Code Page 950 mit Hong Kong; Ergänzender Zeichensatz HKSCS-200 1. Zeichensatzumwandlung basiert auf Unicode 3.0.
ZHT16MSWIN950	Microsoft Windows Code Page 950 Tradition elles Chinesisch
ZHT32EUC	EUC 32-bit Traditionelles Chinesisch

NLS_LANG-Umgebungsvariable

Ein Gebietsschema ist eine Reihe von Informationen, die sprachlichen und kulturellen Anforderungen für eine bestimmte Sprache und ein bestimmtes Land entsprechen. Der einfachste Weg, um das

Verhalten von Oracle zu bestimmen, erfolgt durch das Festlegen der Umgebungsvariable NLS_LANG in der Umgebung Ihres Kunden. Mit dieser Variablen werden die Sprache und die Region definiert, die von der Clientanwendung und dem Datenbankserver verwendet werden. Zudem wird damit der Zeichensatz des Clients bestimmt, der dem Zeichensatz von Daten entspricht, die in einer Client-Anwendung eingegeben oder von dieser angezeigt werden. Weitere Informationen zu NLS_LANG und Zeichensätzen finden Sie unter What is a Character set or Code Page? in der Oracle-Dokumentation.

NLS-Initialisierungsparameter

Sie können auch folgende National Language Support (NLS)-Initialisierungsparameter auf Instance-Ebene für eine Oracle-DB in Amazon RDS festlegen:

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_TIME_FORMAT
- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

Weitere Informationen zum Ändern von Instance-Parametern finden Sie unter Parametergruppen für Amazon RDS.

Sie können weitere NLS-Initialisierungsparameter in Ihrem SQL-Client festlegen. Folgende Anweisung legt die NLS_LANGUAGE-Initialisierungsparameter in einem SQL-Client, der mit einer Oracle-DB-Instance verbunden ist, auf GERMAN fest:

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

Weitere Information über das Verbinden mit einer Oracle-DB-Instance mit einem SQL-Client finden Sie unter Herstellen der Verbindung zu Ihrer Oracle-DB-Instance.

Nationaler Zeichensatz

Der nationale Zeichensatz wird in den Datentypen `NCHAR`, `NVARCHAR2` und `NCLOB` verwendet. Der nationale Zeichensatz wird normalerweise als NCHAR-Zeichensatz bezeichnet. Im Gegensatz zum DB-Zeichensatz wirkt sich der NCHAR-Zeichensatz nicht auf Datenbankmetadaten aus.

Der NCHAR-Zeichensatz unterstützt die folgenden Zeichensätze:

- AL16UTF16 (Standard)
- UTF8

Sie können einen der beiden Werte mit dem `--nchar-character-set-name`-Parameter des Befehls create-db-instance (AWS CLI nur Version 2) festlegen. Wenn Sie das Amazon RDS-API verwenden, spezifizieren Sie den `NcharCharacterSetName`-Parameter der Aktion CreateDBInstance. Sie können den nationalen Zeichensatz nicht ändern, nachdem Sie die Datenbank erstellt haben.

Weitere Informationen zu Unicode in Oracle-Datenbanken finden Sie unter Unterstützen von mehrsprachigen Datenbanken mit Unicode in der Oracle-Dokumentation.

Beschränkungen von RDS for Oracle

In den folgenden Abschnitten finden Sie wichtige Einschränkungen bei der Verwendung von RDS für Oracle. Spezifische Einschränkungen für finden Sie unterEinschränkungen von RDS für Oracle CDBs. CDBs

> **Note**
>
> Diese Liste ist nicht umfassend.

Themen

- Oracle-Dateigrößenbeschränkungen in Amazon RDS
- Blockgrößenbeschränkungen in RDS für Oracle
- Öffentliche Synonyme für Oracle-bereitgestellte Schemata
- Schemas für nicht unterstützte Funktionen in RDS for Oracle
- Einschränkungen für DBA-Rechte in RDS für Oracle
- TLS 1.0 und 1.1 Transport Layer Security in RDS für Oracle sind veraltet

Oracle-Dateigrößenbeschränkungen in Amazon RDS

Die maximale Dateigröße auf DB-Instances von RDS für Oracle beträgt 16 TiB (Tebibyte). Dieses Limit wird durch das ext4-Dateisystem festgelegt, das von der Instance verwendet wird. Daher sind

Oracle-Bigfile-Datendateien auf 16 TiB begrenzt. Wenn Sie versuchen, die Größe einer Datendatei in einem Bigfile-Tablespace in einen Wert zu ändern, der größer als der Grenzwert ist, erhalten Sie eine Fehlermeldung ähnlich der folgenden.

```
ORA-01237: cannot extend datafile 6
ORA-01110: data file 6: '/rdsdbdata/db/mydir/datafile/myfile.dbf'
ORA-27059: could not reduce file size
Linux-x86_64 Error: 27: File too large
Additional information: 2
```

Blockgrößenbeschränkungen in RDS für Oracle

Die maximale Größe eines einzelnen Blocks auf RDS für Oracle-DB-Instances beträgt 8 KB. Die Blockgröße von 16 KB wird nicht unterstützt.

Öffentliche Synonyme für Oracle-bereitgestellte Schemata

Erstellen oder ändern Sie keine öffentlichen Synonyme für von Oracle bereitgestellte Schemas, einschließlich `SYS`, `SYSTEM` und `RDSADMIN`. Solche Aktionen könnten zu einer Invalidierung der Komponenten der Kerndatenbank führen und sich auf die Verfügbarkeit der DB-Instance auswirken.

Sie können öffentliche Synonyme erstellen, die auf Objekte in Ihren eigenen Schemas verweisen.

Schemas für nicht unterstützte Funktionen in RDS for Oracle

Im Allgemeinen hindert Sie Amazon RDS nicht daran, Schemata für nicht unterstützte Funktionen zu erstellen. Wenn Sie jedoch Schemata für Oracle-Funktionen und -Komponenten erstellen, die SYS-Berechtigungen benötigen, können Sie das Data Dictionary beschädigen und Ihre Instance-Verfügbarkeit beeinträchtigen. Verwenden Sie nur unterstützte Funktionen und Schemata, die in verfügbar sin Hinzufügen von Optionen zu Oracle DB-Instances.

Einschränkungen für DBA-Rechte in RDS für Oracle

In der Datenbank ist eine Rolle eine Sammlung von Sonderrechten, die Sie einem Benutzer gewähren oder entziehen können. Eine Oracle-Datenbank verwendet Rollen, um Sicherheit zu gewährleisten.

Die vordefinierte Rolle `DBA` erteilt normalerweise alle administrativen Rechte für eine Oracle-Datenbank-Engine. Wenn Sie eine DB-Instance erstellen, erhält das Hauptbenutzerkonto DBA-

Berechtigungen (mit einigen Einschränkungen). Um eine Verwaltung zu ermöglichen, bietet eine RDS für Oracle-Datenbank nicht die folgenden Berechtigungen für die DBA-Rolle:

- ALTER DATABASE
- ALTER SYSTEM
- CREATE ANY DIRECTORY
- DROP ANY DIRECTORY
- GRANT ANY PRIVILEGE
- GRANT ANY ROLE

Verwenden Sie das Hauptbenutzerkonto für administrative Aufgaben, wie zum Beispiel das Erstellen von zusätzlichen Benutzerkonten in der Datenbank. Sie können SYS, SYSTEM und andere von Oracle bereitgestellte Administratorkonten nicht verwenden.

TLS 1.0 und 1.1 Transport Layer Security in RDS für Oracle sind veraltet

Die Versionen 1.0 und 1.1 (TLS 1.0 und TLS 1.1) des Transport Layer Security-Protokolls sind veraltet. Gemäß den bewährten Methoden für die Sicherheit hat Oracle die Verwendung von TLS 1.0 und TLS 1.1 als veraltet erklärt. Um Ihre Sicherheitsanforderungen zu erfüllen, empfehlen wir dringend, stattdessen TLS 1.2 zu verwenden.

Herstellen der Verbindung zu Ihrer Oracle-DB-Instance

Nachdem Amazon RDS Ihre Oracle-DB-Instance bereitgestellt hat, können Sie eine beliebige Standard-SQL-Client-Anwendung verwenden, um sich bei Ihrer DB-Instance anzumelden. Da es sich bei RDS um einen verwalteten Service handelt, können Sie sich nicht als SYS oder SYSTEM anmelden. Weitere Informationen finden Sie unter RDS für Oracle-Benutzer und -Berechtigungen.

In diesem Thema erfahren Sie, wie Sie Oracle SQL Developer oder SQL*Plus verwenden, um eine Verbindung mit einer DB-Instance von RDS für Oracle herzustellen. Ein Beispiel mit einer Anleitung zum Erstellen und Verbinden für eine Beispiel-DB-Instance finden Sie unter Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung.

Themen

- Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle
- Herstellen der Verbindung zu Ihrer DB-Instance mit Oracle SQL Developer
- Herstellen einer Verbindung mit Ihrer DB-Instance mithilfe von SQL*Plus
- Überlegungen für Sicherheitsgruppen
- Überlegungen zur Prozessarchitektur
- Fehlerbehebung bei Verbindungen mit Ihrer Oracle-DB-Instance
- Ändern von Verbindungseigenschaften mit sqlnet.ora-Parametern

Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle

Jede Amazon RDS-DB-Instance hat einen Endpunkt und jeder Endpunkt hat einen DNS-Namen und eine Portnummer für die DB-Instance. Um eine Verbindung mir Ihrer DB-Instance mit einer SQL-Client-Anwendung herzustellen, benötigen Sie den DNS-Namen und die Portnummer für Ihre DB-Instance.

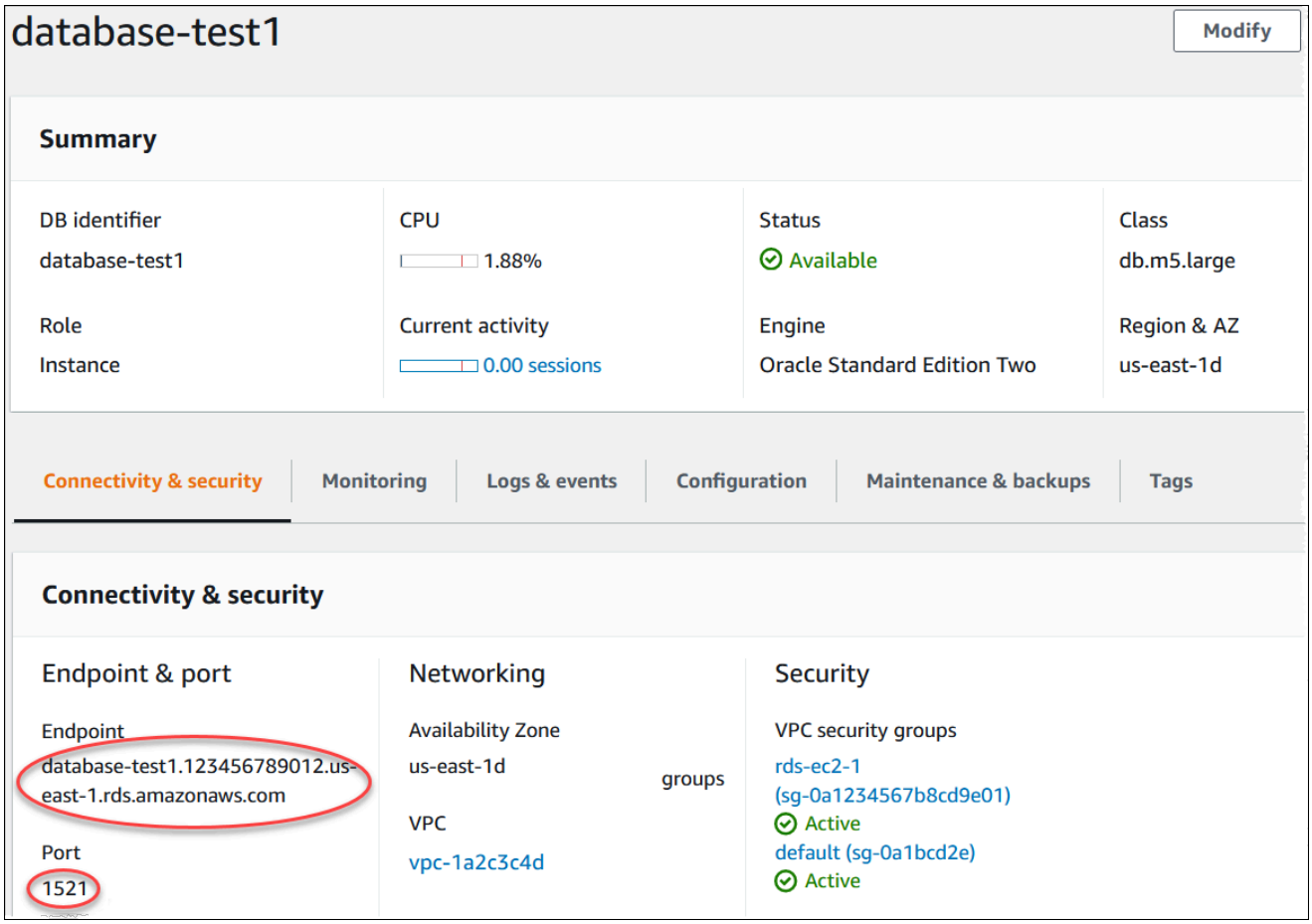
Sie können die Endpunkte für eine DB-Instance mithilfe der Amazon-RDS-Konsole oder der AWS CLI ermitteln.

> **Note**
>
> Wenn Sie die Kerberos-Authentifizierung verwenden, lesen Sie Herstellen einer Verbindung mit Oracle mithilfe der Kerberos-Authentifizierung.

Konsole

So ermitteln Sie den Endpunkt mit der Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie oben rechts in der Konsole die AWS-Region Ihrer DB-Instance aus.
3. Suchen Sie nach dem DNS-Namen und der Portnummer für Ihre DB-Instance.
   a. Wählen Sie Databases (Datenbanken) aus, um eine Liste Ihrer DB-Instances anzuzeigen.
   b. Wählen Sie den Namen der Oracle DB-Instance aus, um Details zur Instance anzuzeigen.
   c. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



AWS CLI

Führen Sie den Befehl describe-db-instance aus, um den Endpunkt einer Oracle-DB-Instance über die AWS CLI zu ermitteln.

Example So ermitteln Sie den Endpunkt mit der AWS CLI

```
aws rds describe-db-instances
```

Suchen Sie in der Ausgabe nach `Endpoint`, um den DNS-Namen und die Portnummer für Ihre DB-Instance zu ermitteln. Die Linie `Address` in der Ausgabe enthält den DNS-Namen. Nachstehend finden Sie ein Beispiel für die Ausgabe eines JSON-Endpunkts.

```
"Endpoint": {
    "HostedZoneId": "Z1PVIF0B656C1W",
    "Port": 3306,
    "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"
},
```

> **Note**
>
> Die Ausgabe kann Informationen für mehrere DB-Instances enthalten.

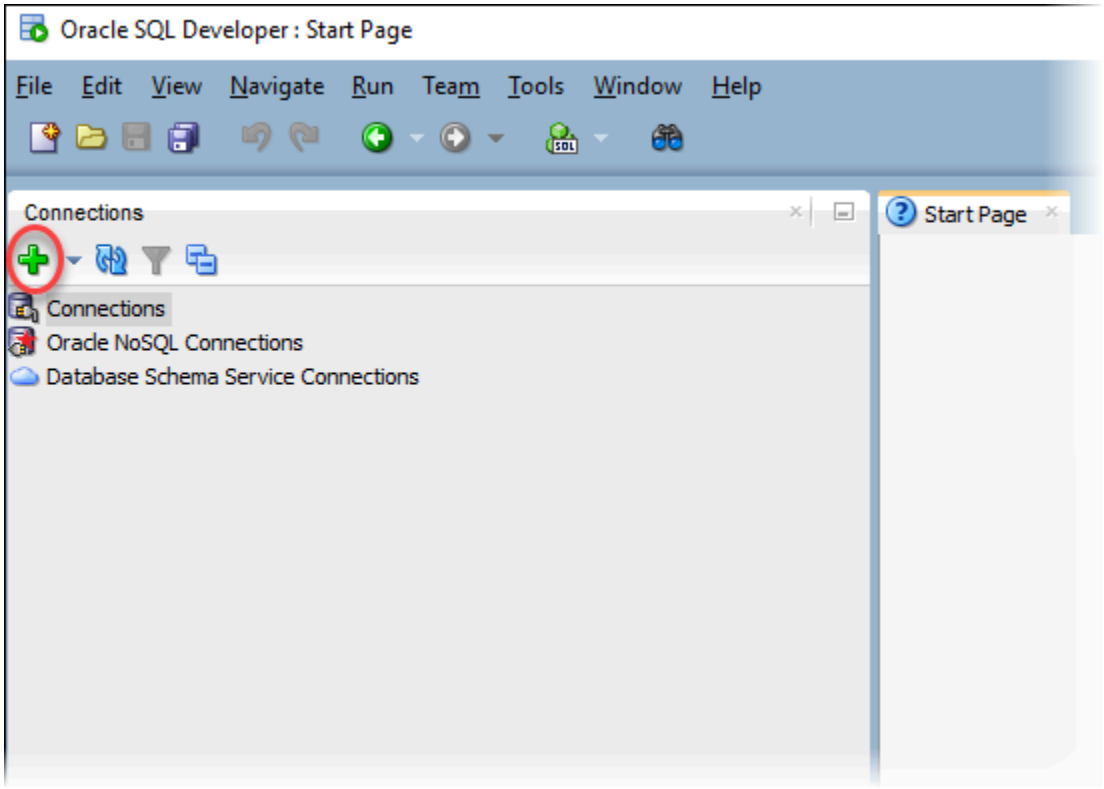
Herstellen der Verbindung zu Ihrer DB-Instance mit Oracle SQL Developer

Bei dieser Vorgehensweise verbinden Sie sich mit Ihrer DB-Instance mithilfe von Oracle SQL Developer. Eine eigenständige Version dieses Dienstprogramms zum Herunterladen finden Sie auf der Downloadseite für Oracle SQL Developer.

Sie benötigen den DNS-Namen und die Portnummer Ihrer DB-Instance, um sich mit ihr zu verbinden. Informationen zum Ermitteln des DNS-Namens und der Portnummer für eine DB-Instance finden Sie unter Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle.

So stellen Sie eine Verbindung mit einer DB-Instance mithilfe von SQL Developer her:

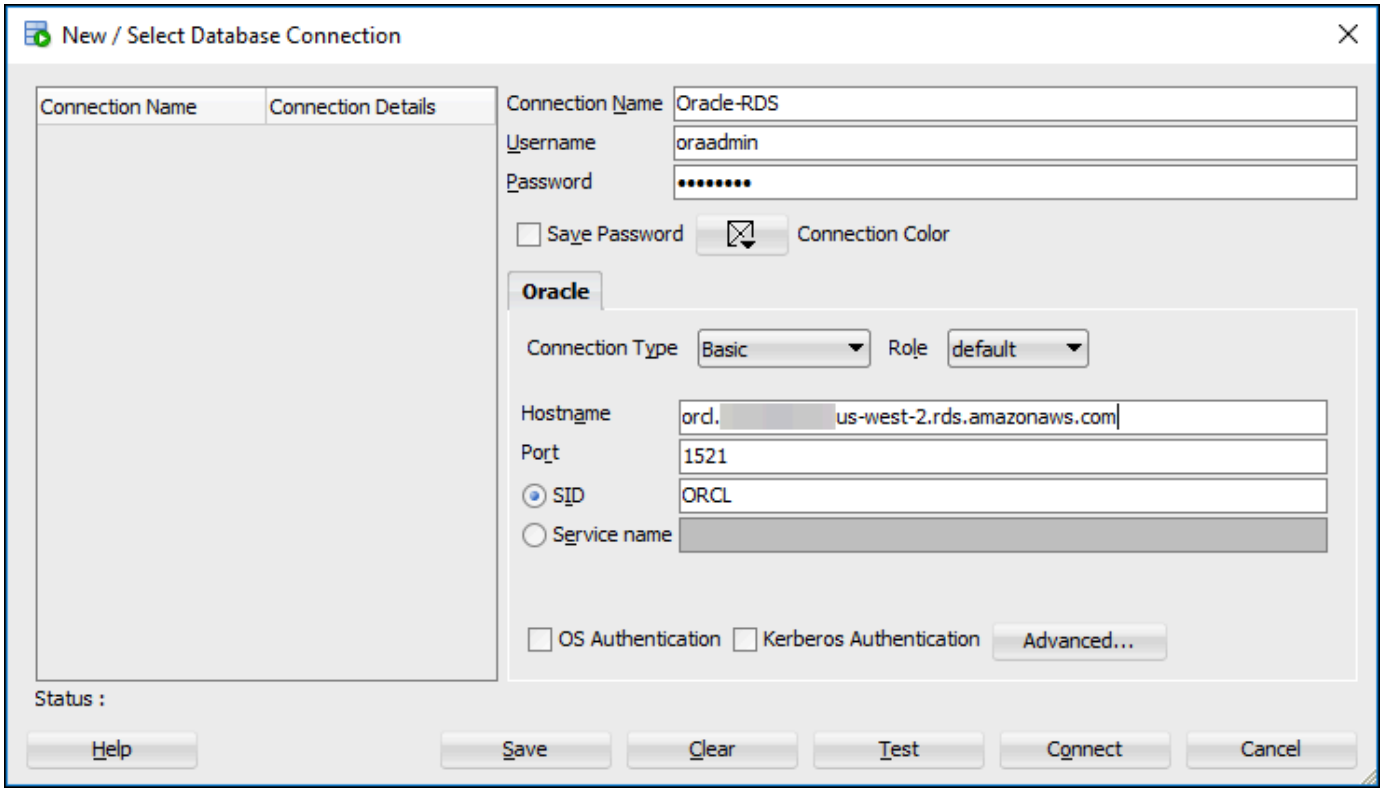
1. Starten Sie Oracle SQL Developer.
2. Wählen Sie auf der Registerkarte Connections (Verbindungen) die Option Hinzufügen (+) aus.



3. Geben Sie im Dialogfeld New/Select Database Connection (Neu/Datenbankverbindung auswählen) die Informationen für Ihre DB-Instance an:

   - Geben Sie unter Connection Name (Verbindungsname) einen Namen zur Beschreibung an, etwa `Oracle-RDS`.
   - Geben Sie in das Feld Username (Benutzername) den Namen des Datenbankadministrators für Ihre DB-Instance ein.
   - Geben Sie unter Password (Passwort) das Passwort des Datenbankadministrators an.
   - Geben Sie unterHostname den DNS-Namen der DB-Instance an.
   - Geben Sie unter Port die Portnummer ein.
   - Geben Sie für SID den DB-Namen ein. Sie finden den DB-Namen auf der Registerkarte Configuration (Konfiguration) Ihrer Seite mit den Datenbankdetails.

   Das vollständig ausgefüllte Dialogfeld sollte folgendermaßen aussehen.

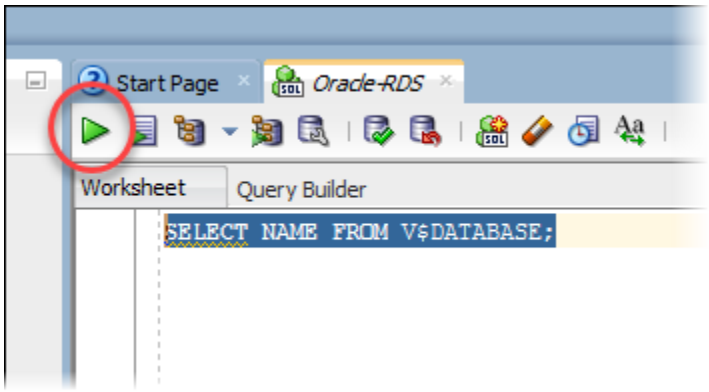


4. Wählen Sie Connect (Verbinden) aus.
5. Sie können nun wie üblich beginnen Ihre eigenen Datenbanken zu erstellen und Abfragen gegen Ihre DB-Instance und Datenbanken auszuführen. Gehen Sie wie folgt vor, um eine Testabfrage für die DB-Instance auszuführen:

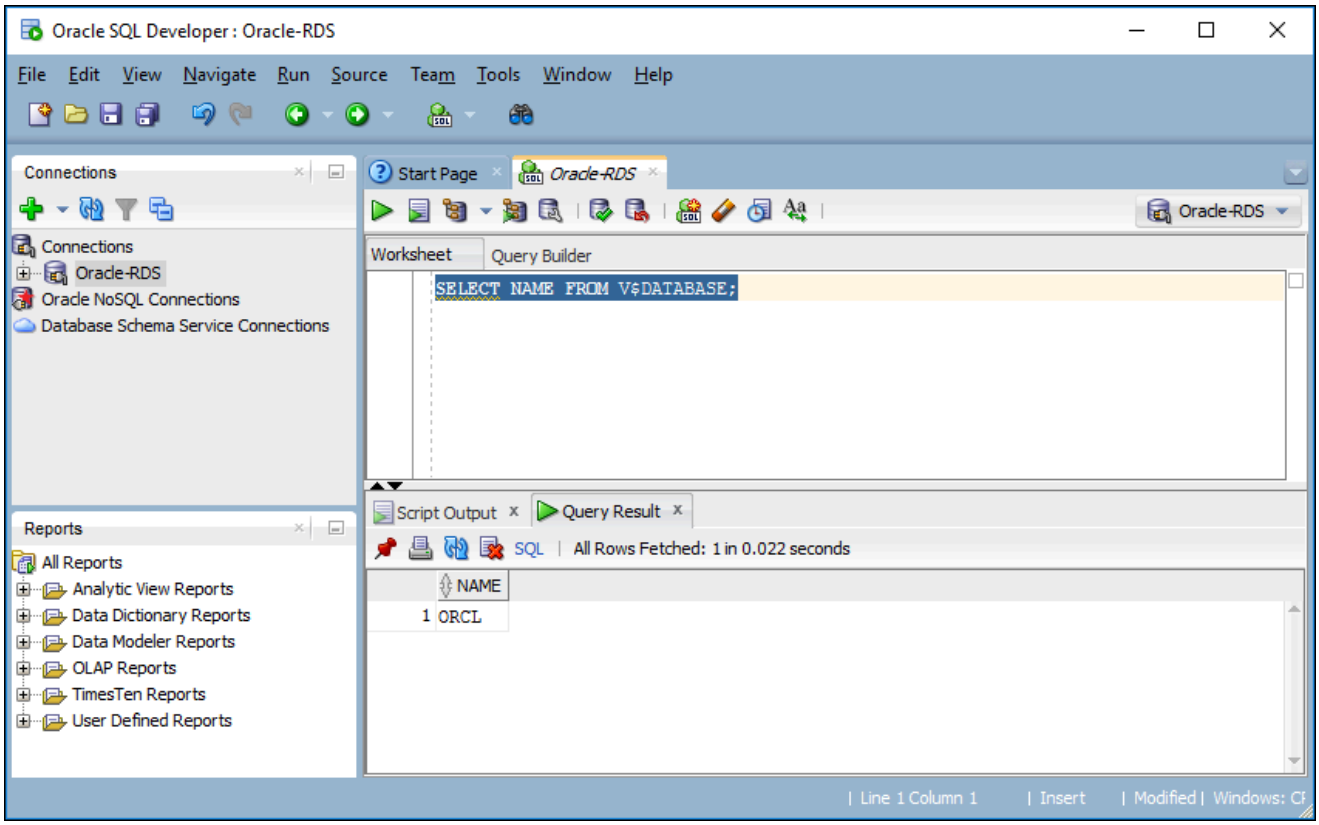
   a. Geben Sie auf dem Tab Worksheet (Arbeitsblatt) die nachfolgende SQL-Abfrage für Ihre Verbindung ein.

   ```
   SELECT NAME FROM V$DATABASE;
   ```

   b. Klicken Sie auf das Symbol zum Ausführen, um die Abfrage auszuführen.



SQL Developer gibt den Datenbanknamen zurück.



Herstellen einer Verbindung mit Ihrer DB-Instance mithilfe von SQL*Plus

Sie können ein Dienstprogramm wie SQL*Plus für die Verbindung zu einer Amazon RDS-DB-Instance unter Oracle verwenden. Informationen zum Herunterladen von Oracle Instant Client, der eine eigenständige Version von SQL*Plus enthält, finden Sie unter Oracle Instant Client – Downloads.

Sie benötigen den DNS-Namen und die Portnummer Ihrer DB-Instance, um sich mit ihr zu verbinden. Informationen zum Ermitteln des DNS-Namens und der Portnummer für eine DB-Instance finden Sie unter Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle.

Example So stellen Sie eine Verbindung mit einer Oracle-DB-Instance mithilfe von SQL*Plus her

Fügen Sie in den folgenden Beispielen den Benutzernamen des DB-Instance-Administrators ein. Geben Sie außerdem den DNS-Namen der DB-Instance und dann die Port-Nummer und die Oracle-SID an. Der SID-Wert ist der Datenbankname der DB-Instance, den Sie beim Anlegen der DB-Instance angegeben haben (nicht der Name der DB-Instance).

Für Linux, macOSoder Unix:

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))'
```

Windows:

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))
```

Die Ausgabe sollte in etwa wie folgt aussehen:

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, erscheint die SQL-Eingabeaufforderung.

```
SQL>
```

> **Note**
>
> Für die Verbindungszeichenfolge in kürzerem Format (EZ connect), z. B. `sqlplus USER/PASSWORD@longer-than-63-chars-rds-endpoint-here:1521/database-identifier`, kann eine maximale Zeichenanzahl gelten und sie sollte daher nicht für die Verbindung genutzt werden.

Überlegungen für Sicherheitsgruppen

Ihrer DB-Instance muss eine Sicherheitsgruppe zugeordnet sein, die die erforderlichen IP-Adressen und die Netzwerkkonfiguration enthält, um eine Verbindung mit Ihrer DB-Instance herzustellen. Ihre DB-Instance verwendet möglicherweise die Standardsicherheitsgruppe. Wenn Sie beim Erstellen der DB-Instance eine standardmäßige, nicht konfigurierte Sicherheitsgruppe zugewiesen haben, verhindert die Firewall Verbindungsversuche. Informationen zum Erstellen einer neuen Sicherheitsgruppe finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Nachdem Sie die neue Sicherheitsgruppe erstellt haben, ändern Sie Ihre DB-Instance, um ihr die Sicherheitsgruppe zuzuordnen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sie können die Sicherheitsstufe mithilfe von SSL erhöhen, um Verbindungen zu Ihrer DB-Instance zu verschlüsseln. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.

Überlegungen zur Prozessarchitektur

Die Benutzerverbindungen zu einer Oracle-DB-Instance werden von Serverprozessen gehandhabt. Grundsätzlich werden die Verbindungen zu einer Oracle-DB-Instance von dedizierten Serverprozessen gehandhabt. Bei dedizierten Serverprozessen bedient jeder Serverprozess nur jeweils einen Benutzerprozess. Sie können optional gemeinsam genutzte Serverprozesse konfigurieren. Bei gemeinsam genutzten Serverprozessen kann jeder Serverprozess mehrere Benutzerprozesse bedienen.

Sie sollten den Einsatz von gemeinsam genutzten Serverprozessen in Erwägung ziehen, wenn infolge einer hohen Anzahl von Benutzersitzungen zu viel Speicher auf dem Server in Anspruch genommen wird. Bei sehr häufigen Sitzungsanmeldungen und -abmeldungen, die zu Leistungsproblemen führen, sollten Sie den Einsatz von gemeinsam genutzten Serverprozessen in Erwägung ziehen. Der Einsatz gemeinsam genutzter Serverprozesse birgt auch Nachteile. Sie können beispielsweise CPU-Ressourcen belasten und die Konfiguration und Verwaltung gestaltet sich komplizierter.

Weitere Informationen zu dedizierten und gemeinsam genutzten Serverprozessen finden Sie unter About Dedicated and Shared Server Processes in der Oracle-Dokumentation. Weitere Informationen zum Konfigurieren von Shared Server-Prozessen auf einer RDS für Oracle-DB-Instance finden Sie unter Wie konfiguriere ich Amazon RDS for Oracle Database für die Arbeit mit freigegebenen Servern? im Wissenszentrum.

Fehlerbehebung bei Verbindungen mit Ihrer Oracle-DB-Instance

Die folgenden Probleme könnten auftreten, wenn Sie versuchen, eine Verbindung zu Ihrer Oracle-DB-Instance herzustellen.

Problem	Vorschläge für die Fehlerbehebung
Keine Verbindung zur DB-Instance	Bei einer neu erstellten DB-Instance lautet ihr Status creating (Wird erstellt), bis die DB-Instance bereit für die Verwendung ist. Wenn sich der Status in available (Verfügbar) ändert, können Sie die Verbindung zur DB-Instance herstellen. Je nach Klasse und

Problem	Vorschläge für die Fehlerbehebung
	Speicherort der DB-Instance kann es bis zu 20 Minuten dauern, bis die neue DB-Instance verfügbar ist.
Keine Verbindung zur DB-Instance	Wenn über den Port, den Sie beim Erstellen der DB-Instance angegeben haben, keine Daten gesendet oder empfangen werden, kann keine Verbindung zur DB-Instance hergestellt werden. Überprüfen Sie gemeinsam mit Ihrem Netzwerkadministrator, ob der festgelegte Port für die DB-Instance ein- und ausgehende Kommunikation zulässt.
Keine Verbindung zur DB-Instance	Die von Ihrer Firewall erzwungenen Zugriffsregeln und die IP-Adressen, die Sie für den Zugriff auf Ihre DB-Instance in der Sicherheitsgruppe für die DB-Instance autorisiert haben, könnten nicht übereinstimmen. Dieses Problem liegt in den meisten Fällen bei den Regeln für ein- oder ausgehenden Datenverkehr der Firewall. Sie können eine Regel für eingehenden Datenverkehr in der Sicherheitsgruppe hinzufügen oder ändern. Wählen Sie für Source (Quelle) die Option My IP (Meine IP) aus. Dies ermöglicht Zugriff auf die DB-Instance von der IP-Adresse, die in Ihrem Browser erkannt wird. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen. Ein Thema mit Anweisungen zum Einrichten von Regeln für Ihre Sicherheitsgruppe finden Sie unter Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur).

Problem	Vorschläge für die Fehlerbehebung
Die Verbindung ist fehlgeschlagen, weil der Zielhost bzw. das Zielobjekt nicht vorhanden ist. (Oracle-Fehler ORA-12545)	Vergewissern Sie sich, dass Sie den Servernamen und die Portnummer richtig angegeben haben. Geben Sie unter Server name (Servername) den DNS-Namen aus der Konsole ein. Informationen zum Ermitteln des DNS-Namens und der Portnummer für eine DB-Instance finden Sie unter Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle.
Benutzername/Passwort ungültig, Anmeldung verweigert. (Oracle-Fehler ORA-01017)	Sie konnten Ihre DB-Instance erreichen, jedoch wurde der Verbindungsversuch abgelehnt. Dies geschieht meistens bei der falschen Angabe des Benutzernamen oder Passworts. Überprüfen Sie den Benutzernamen und das Passwort und versuchen Sie es erneut.
TNS:Listener kennt derzeit keine SID, die im Connect-Deskriptor angegeben ist – Oracle, FEHLER: ORA-12505	Stellen Sie sicher, dass die richtige SID eingegeben wurde. Die SID entspricht Ihrem DB-Namen. Suchen Sie den DB-Namen auf der Registerkarte Configuration (Konfiguration) der Seite Databases (Datenbanken) für Ihre Instance. Sie können den DB-Namen auch mithilfe der AWS CLI finden: `aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier,DBName]' --output text`

Weitere Informationen zu Verbindungsproblemen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden.

Ändern von Verbindungseigenschaften mit sqlnet.ora-Parametern

Die Datei sqlnet.ora enthält Parameter, die Oracle Net-Funktionen auf Oracle-Datenbank-Servern und -Clients konfigurieren. Mit den Parametern in der Datei sqlnet.ora können Sie Eigenschaften für Verbindungen zur und von der Datenbank ändern.

Weitere Informationen zu den Gründen dafür, sqlnet.ora-Parameter zu ändern, finden Sie unter Configuring Profile Parameters in der Oracle-Dokumentation.

Festlegen von sqlnet.ora-Parametern

Amazon RDS für Oracle-Parametergruppen enthalten eine Teilmenge der sqlnet.ora-Parameter. Sie stellen sie auf die gleiche Weise wie andere Oracle-Parameter ein. sqlnet.ora-Parametern ist das Präfix `sqlnetora.` vorangestellt. In einer Oracle-Parametergruppe in Amazon RDS heißt der sqlnet.ora-Parameter `default_sdu_size` also beispielsweise `sqlnetora.default_sdu_size`.

Informationen zum Verwalten von Parametergruppen und zum Einstellen von Parameterwerten finden Sie unter Parametergruppen für Amazon RDS.

Unterstützte sqlnet.ora-Parameter

Amazon RDS unterstützt die folgenden sqlnet.ora-Parameter. Änderungen an dynamischen sqlnet.ora-Parametern werden sofort wirksam.

Parameter	Zulässig Werte	Statisch/ Dynamis	Beschreibung
`sqlnetora.default_sdu_size`	512 auf 20971	Dynamis	Die SDU-Größe (Session Data Unit, Sitzungsd ateneinheit) in Byte. Die SDU bezeichnet die Menge der Daten, die gleichzeitig in einen Puffer geschrieben und über das Netzwerk gesendet werden.
`sqlnetora.diag_adr_enabled`	`ON`, `OFF`	Dynamis	Ein Wert, der die ADR-Nachverfolgung (Automati sches Diagnose-Repository) aktiviert oder deaktiviert. `ON` gibt an, dass die ADR-Dateinachverfolgung verwendet wird.

Parameter	Zulässig Werte	Statisch/ Dynamis	Beschreibung
			OFF gibt an, dass eine ADR-fremde Dateinachverfolgung verwendet wird.
sqlnetora.recv_buf_size	8192 auf 26843	Dynamis	Das Pufferspeicherlimit für Empfangsoperationen von Sitzungen, die von den Protokollen TCP/IP, TCP/IP mit SSL und SDP unterstützt werden.
sqlnetora.send_buf_size	8192 auf 26843	Dynamis	Das Pufferspeicherlimit für Sendeoperationen von Sitzungen, die von den Protokollen TCP/IP, TCP/IP mit SSL und SDP unterstützt werden.
sqlnetora.sqlnet.allowed_logon_version_client	8, 10, 11, 12	Dynamis	Die minimale Authentifizierungsprotokollversion für Clients und als Clients agierende Server zur Herstellung einer Verbindung zu Oracle DB-Instances.
sqlnetora.sqlnet.allowed_logon_version_server	8, 9, 10, 11, 12, 12a	Dynamis	Die minimale Authentifizierungsprotokollversion zur Herstellung einer Verbindung zu Oracle DB-Instances.

Parameter	Zulässi Werte	Statisch/ Dynamis	Beschreibung
sqlnetora.sqlnet.expire_time	0 auf 1440	Dynamis	Zeitintervall in Minuten, in dem ein Prüfsignal gesendet wird, um zu verifizieren, dass Client-Server-Verbindungen aktiv sind.
sqlnetora.sqlnet.inbound_connect_timeout	0 oder 10 bis 7200	Dynamis	Zeit in Sekunden für einen Client, um die Verbindung zum Datenbank-Server herzustellen und die erforderlichen Authentifizierungsinformationen bereitzustellen.
sqlnetora.sqlnet.outbound_connect_timeout	0 oder 10 bis 7200	Dynamis	Zeit in Sekunden für einen Client, eine Oracle Net-Verbindung zur DB-Instance herzustellen.
sqlnetora.sqlnet.recv_timeout	0 oder 10 bis 7200	Dynamis	Zeit in Sekunden, die ein Datenbank-Server auf Client-Daten warten soll, nachdem eine Verbindung hergestellt wurde.
sqlnetora.sqlnet.send_timeout	0 oder 10 bis 7200	Dynamis	Zeit in Sekunden für einen Datenbank-Server, eine Sendeoperation an Clients abzuschließen, nachdem eine Verbindung hergestellt wurde.

Parameter	Zulässi Werte	Statisch/ Dynamis	Beschreibung
`sqlnetora.tcp.connect_timeout`	`0` oder `10` bis `7200`	Dynamis	Zeit in Sekunden für einen Client, eine TCP-Verbindung zum Datenbank-Server herzustellen.
`sqlnetora.trace_level_server`	`0`, `4`, `10`, `16`, `OFF`, `USER`, `ADMIN`, `SUPPOF`	Dynamis	Schaltet die Server-Na chverfolgung für eine ADR-fremde Nachverfolgung auf einer angegebenen Ebene ein oder aus.

Der Standardwert für jeden unterstützten sqlnet.ora-Parameter ist der Oracle-Datenbankstandard für die Version.

Anzeigen von sqlnet.ora-Parametern

Sie können die sqlnet.ora-Parameter und ihre Einstellungen mit dem AWS Management Console, oder einem SQL-Client anzeigen. AWS CLI

Anzeigen der sqlnet.ora-Parameter mit der Konsole

Weitere Informationen zum Anzeigen von Parametern in einer Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

In Oracle-Parametergruppen bezeichnet das Präfix `sqlnetora.` sqlnet.ora-Parameter.

Anzeigen der sqlnet.ora-Parameter mit der AWS CLI

Verwenden Sie den Befehl describe-db-parameters, um die sqlnet.ora-Parameter anzuzeigen, die in einer Oracle-Parametergruppe konfiguriert wurden. AWS CLI

Um alle sqlnet.ora-Parameter für eine Oracle-DB-Instance anzuzeigen, rufen Sie den Befehl download-db-log-file-portion auf. AWS CLI Geben Sie die DB-Instance-Kennung, den Protokolldateinamen und den Typ der Ausgabe an.

Example

Der folgende Code listet alle sqlnet.ora-Parameter für au `mydbinstance`.

Für, oder: Linux macOS Unix

```
aws rds download-db-log-file-portion \
    --db-instance-identifier mydbinstance \
    --log-file-name trace/sqlnet-parameters \
    --output text
```

Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier mydbinstance ^
    --log-file-name trace/sqlnet-parameters ^
    --output text
```

Anzeigen der sqlnet.ora-Parameter mit einem SQL-Client

Nachdem Sie die Verbindung zur Oracle-DB-Instance in einem SQL-Client hergestellt haben, können Sie mit folgender Abfrage die sqlnet.ora-Parameter auflisten.

```
SELECT * FROM TABLE
   (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'BDUMP',
        p_filename  => 'sqlnet-parameters'));
```

Information zum Herstellen einer Verbindung mit einer Oracle-DB-Instance in einem SQL-Client finden Sie unter Herstellen der Verbindung zu Ihrer Oracle-DB-Instance.

Sichern von Verbindungen von Oracle DB-Instances

Amazon RDS for Oracle unterstützt verschlüsselte SSL-/TLS-Verbindungen sowie die Option „Oracle Native Network Encryption (NNE)“ zum Verschlüsseln von Verbindungen zwischen Anwendungen und Ihrer Oracle-DB-Instance. Weitere Informationen über die Option Oracle Native Network Encryption finden Sie unter Oracle Native Network Encryption.

Themen

- Verwenden von SSL mit einer DB-Instance von RDS für Oracle
- Aktualisieren von Anwendungen, um Verbindungen mit Oracle-DB-Instances mithilfe neuer SSL/TLS-Zertifikate herzustellen
- Verwenden der nativen Netzwerkverschlüsselung mit einer DB-Instance von RDS für Oracle
- Konfigurieren der Kerberos-Authentifizierung für Amazon RDS for Oracle
- Konfigurieren des UTL_HTTP-Zugriffs mit Zertifikaten und einer Oracle Wallet

Verwenden von SSL mit einer DB-Instance von RDS für Oracle

Secure Sockets Layer (SSL) ist ein Branchen-Standardprotokoll, das für den Schutz von Netzwerkverbindungen zwischen Client und Server verwendet wird. Nach SSL-Version 3.0 wurde der Name in Transport Layer Security (TLS) geändert, aber wir bezeichnen das Protokoll häufig immer noch als SSL. Amazon RDS unterstützt SSL-Verschlüsselung für Oracle-DB-Instances. Durch die Verwendung von SSL können Sie eine Verbindung zwischen Ihrem Anwendungs-Client und Ihrer Oracle-DB-Instance herstellen. SSL-Support ist in allen AWS-Regionen für Oracle verfügbar.

Um die SSL-Verschlüsselung für eine Oracle-DB-Instance zu aktivieren, fügen Sie die Option "Oracle SSL" der Optionsgruppe hinzu, die der DB-Instance zugeordnet ist. Amazon RDS verwendet einen zweiten Port, wie von Oracle erforderlich, für SSL-Verbindungen. Auf diese Weise können sowohl Klartext- als auch SSL-verschlüsselte Kommunikation gleichzeitig zwischen einer DB-Instance und einem Oracle-Client erfolgen. Sie können z. B. den Port mit Klartext-Kommunikation verwenden, um mit anderen Ressourcen innerhalb einer VPC zu kommunizieren, und den Port mit SSL-verschlüsselter Kommunikation, um mit Ressourcen außerhalb der VPC zu kommunizieren.

Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.

> **Note**
>
> Sie können SSL und Oracle Native Network Encryption (NNE) nicht gleichzeitig in derselben DB-Instance verwenden. Bevor Sie die SSL-Verschlüsselung verwenden können, müssen Sie jede andere Verbindungsverschlüsselung deaktivieren.

Aktualisieren von Anwendungen, um Verbindungen mit Oracle-DB-Instances mithilfe neuer SSL/TLS-Zertifikate herzustellen

Am 13. Januar 2023 veröffentlichte Amazon RDS neue Zertifizierungsstellen-Zertifikate (Certificate Authority, CA) zum Herstellen von Verbindungen mit Ihren RDS-DB-Instances mithilfe von Secure Socket Layer oder Transport Layer Security (SSL/TLS). Im Folgenden finden Sie Informationen dazu, wie Sie Ihre Anwendungen aktualisieren, um die neuen Zertifikate verwenden zu können.

In diesem Thema finden Sie Informationen dazu, wie Sie ermitteln, ob Client-Anwendungen für die Herstellung von Verbindungen mit Ihren DB-Instances SSL/TLS verwenden.

> **Important**
>
> Wenn Sie das Zertifikat für eine Amazon RDS for Oracle DB-Instance ändern, wird nur der Datenbank-Listener neu gestartet. Die DB-Instance wird nicht neu gestartet. Bestehende Datenbankverbindungen sind davon nicht betroffen, aber bei neuen Verbindungen treten für eine kurze Zeitspanne Fehler auf, während der Listener neu gestartet wird.
> Wir empfehlen Ihnen, Ihre Oracle-DB neu zu starten, um Verbindungsfehler zu vermeiden.

> **Note**
>
> Für Client-Anwendungen, die SSL/TLS für Verbindungen zu Ihren DB-Instances verwenden, müssen Sie die Trust Stores Ihrer Client-Anwendung so aktualisieren, dass sie die neuen CA-Zertifikate enthalten.

Nach der Aktualisierung der CA-Zertifikate in den Trust Stores Ihrer Client-Anwendung können Sie die Zertifikate auf Ihren DB-Instances rotieren. Es wird nachdrücklich empfohlen, diese Verfahren vor der Implementierung in Produktionsumgebungen in einer Entwicklungs- oder Testumgebung zu testen.

Weitere Informationen zur Zertifikatrotation finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate. Weitere Informationen zum Herunterladen von Zertifikaten finden Sie unter . Informationen zum Verwenden von SSL/TLS mit Oracle-DB-Instances finden Sie unter Oracle Secure Sockets Layer.

Themen

- Prüfen der Verbindung von Anwendungen über SSL
- Aktualisieren des Trust Stores Ihrer Anwendung
- Java-Beispielcode für die Herstellung von SSL-Verbindungen

Prüfen der Verbindung von Anwendungen über SSL

Wenn Ihre Oracle-DB-Instance eine Optionsgruppe mit Hinzufügung der Option SSL verwendet, verwenden Sie möglicherweise SSL. Prüfen Sie dies anhand der Anweisungen in Auflisten der Optionen und Optionseinstellungen für eine Optionsgruppe. Weitere Informationen zur Option SSL finden Sie unter Oracle Secure Sockets Layer.

Prüfen Sie das Listener-Protokoll, um festzustellen, ob es SSL-Verbindungen gibt. Dies ist eine Beispielausgabe in einem Listener-Protokoll.

```
date time * (CONNECT_DATA=(CID=(PROGRAM=program)
(HOST=host)(USER=user))(SID=sid)) *
(ADDRESS=(PROTOCOL=tcps)(HOST=host)(PORT=port)) * establish * ORCL * 0
```

Wenn PROTOCOL den Wert `tcps` für einen Eintrag hat, zeigt dies eine SSL-Verbindung an. Wenn HOST jedoch `127.0.0.1` ist, können Sie den Eintrag ignorieren. Verbindungen von `127.0.0.1` sind ein lokaler Management Agent auf der DB-Instance. Dies sind keine externen SSL-Verbindungen. Daher haben Sie Anwendungen, die sich über SSL verbinden, wenn es Listener-Protokolleinträge gibt, bei denen PROTOCOL `tcps` ist und HOST nicht `127.0.0.1` ist.

Um das Listener-Protokoll zu überprüfen, können Sie das Protokoll in Amazon CloudWatch Logs veröffentlichen. Weitere Informationen finden Sie unter Oracle-Logs in Amazon CloudWatch Logs veröffentlichen.

Aktualisieren des Trust Stores Ihrer Anwendung

Sie können den Trust Store für Anwendungen aktualisieren, die SQL*Plus oder JDBC für SSL/TLS-Verbindungen verwenden.

Aktualisieren des Trust Stores Ihrer Anwendung für SQL*Plus

Sie können den Trust Store für Anwendungen aktualisieren, die SQL*Plus für SSL/TLS-Verbindungen verwenden.

> **Note**
>
> Wenn Sie den Trust Store aktualisieren, können Sie ältere Zertifikate beibehalten und die neuen Zertifikate einfach hinzufügen.

So aktualisieren Sie den Trust Store für SQL*Plus-Anwendungen:

1. Laden Sie das neue Stammzertifikat herunter, das für alle AWS Regionen funktioniert, und legen Sie die Datei im `ssl_wallet` Verzeichnis ab.

   Informationen zum Herunterladen des Stammverzeichnisses finden Sie unter .
2. Führen Sie den folgenden Befehl aus, um Das Oracle-Wallet zu aktualisieren:

   ```
   prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
         $ORACLE_HOME/ssl_wallet/ssl-cert.pem -auto_login_only
   ```

   Ersetzen Sie den Dateinamen durch den Namen der Datei, die Sie heruntergeladen haben.
3. Bestätigen Sie durch Ausführen des folgenden Befehls, dass das Wallet erfolgreich installiert wurde.

   ```
   prompt>orapki wallet display -wallet $ORACLE_HOME/ssl_wallet
   ```

   Die Ausgabe sollte Folgendes enthalten:

   ```
   Trusted Certificates:
   Subject: CN=Amazon RDS Root 2019 CA,OU=Amazon RDS,O=Amazon Web Services\,
    Inc.,L=Seattle,ST=Washington,C=US
   ```

Aktualisieren des Trust Stores Ihrer Anwendung für JDBC

Sie können den Trust Store für Anwendungen aktualisieren, die JDBC für SSL/TLS-Verbindungen verwenden.

Informationen zum Herunterladen des Stammverzeichnisses finden Sie unter .

Beispiele für Skripte, die Zertifikate importieren, finden Sie unter Beispielskript für den Import von Zertifikaten in Ihren Trust Store.

Java-Beispielcode für die Herstellung von SSL-Verbindungen

Das folgende Code-Beispiel zeigt, wie die SSL-Verbindung mit JDBC eingerichtet wird.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-
group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Important**
>
> Nachdem Sie festgestellt haben, dass Ihre Datenbankverbindungen SSL/TLS verwenden, und Ihren Application Trust Store aktualisiert haben, können Sie Ihre Datenbank so aktualisieren, dass sie die rds-ca-rsa 2048-g1-Zertifikate verwendet. Anweisungen hierzu finden Sie in Schritt 3 unter Aktualisierung Ihres CA-Zertifikats durch Änderung Ihrer DB-Instance oder Ihres Clusters.

Verwenden der nativen Netzwerkverschlüsselung mit einer DB-Instance von RDS für Oracle

Oracle Database bietet zwei Möglichkeiten, Daten über das Netzwerk zu verschlüsseln: native Netzwerkverschlüsselung (NNE) und Transport Layer Security (TLS). NNE ist eine proprietäre Sicherheitsfunktion von Oracle, wohingegen TLS ein Industriestandard ist. RDS für Oracle unterstützt NNE für alle Editionen von Oracle Database.

NNE bietet die folgenden Vorteile gegenüber TLS:

- Sie können NNE auf dem Client und Server mithilfe der Einstellungen in der Option NNE steuern:
  - `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` und `SQLNET.ALLOW_WEAK_CRYPTO`
  - `SQLNET.CRYPTO_CHECKSUM_CLIENT` und `SQLNET.CRYPTO_CHECKSUM_SERVER`
  - `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT` und `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
  - `SQLNET.ENCRYPTION_CLIENT` und `SQLNET.ENCRYPTION_SERVER`
  - `SQLNET.ENCRYPTION_TYPES_CLIENT` und `SQLNET.ENCRYPTION_TYPES_SERVER`
- In den meisten Fällen müssen Sie Ihren Client oder Server nicht konfigurieren. Im Gegensatz dazu erfordert TLS, dass Sie sowohl Client als auch Server konfigurieren.
- Es sind keine Zertifikate erforderlich. Bei TLS benötigt der Server ein Zertifikat (das letztendlich abläuft) und der Client benötigt ein vertrauenswürdiges Stammzertifikat für die Zertifizierungsstelle, die das Serverzertifikat ausgestellt hat.

Um die NNE-Verschlüsselung für eine Oracle-DB-Instance zu aktivieren, fügen Sie die Option „Oracle NNE“ der Optionsgruppe hinzu, die der DB-Instance zugeordnet ist. Weitere Informationen finden Sie unter Oracle Native Network Encryption.

> **Note**
>
> Sie können NNE und TLS nicht gleichzeitig in derselben DB-Instance verwenden.

Konfigurieren der Kerberos-Authentifizierung für Amazon RDS for Oracle

Sie können die Kerberos-Authentifizierung verwenden, um Benutzer zu authentifizieren, wenn diese sich mit Ihrer DB-Instance von Amazon RDS for Oracle verbinden. In dieser Konfiguration arbeitet Ihre DB-Instance mit AWS Directory Service for Microsoft Active Directory, auch als AWS Managed Microsoft AD bezeichnet. Wenn Benutzer sich mit einer DB-Instance von RDS for Oracle authentifizieren, die mit einer vertrauenswürdigen Domäne verbunden ist, werden die Authentifizierungsanfragen an das Verzeichnis weitergeleitet, das Sie mit AWS Directory Service erstellen.

Wenn Sie alle Ihre Anmeldeinformationen im selben Verzeichnis aufbewahren, können Sie Zeit und Mühe sparen. Sie verfügen über einen zentralen Ort zum Speichern und Verwalten von Anmeldeinformationen für mehrere Datenbank-Instances. Ein Verzeichnis kann auch Ihr allgemeines Sicherheitsprofil verbessern.

Verfügbarkeit von Regionen und Versionen

Verfügbarkeit von Funktionen und Support variiert zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Hinweise zur Versions- und Regionsverfügbarkeit von RDS for Oracle mit Kerberos-Authentifizierung finden Sie unter Unterstützte Regionen und DB-Engines für die Kerberos-Authentifizierung in Amazon RDS.

> **Note**
>
> Die Kerberos-Authentifizierung wird für DB-Instance-Klassen, die für DB-Instances von RDS for Oracle veraltet sind, nicht unterstützt. Weitere Informationen finden Sie unter RDS für Oracle-DB-Instance-Klassen.

Themen

- Kerberos für Oracle-DB-Instances einrichten
- Verwalten einer DB-Instance in einer Domäne

- Herstellen einer Verbindung mit Oracle mithilfe der Kerberos-Authentifizierung

Kerberos für Oracle-DB-Instances einrichten

Wird auch genannt AWS Directory Service for Microsoft Active Directory AWS Managed Microsoft AD, zum Einrichten der Kerberos-Authentifizierung für eine Oracle-DB-Instance verwendet. Führen Sie zum Einrichten der Kerberos-Authentifizierung die folgenden Schritte aus:

- Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD
- Schritt 2: Erstellen einer Vertrauensstellung
- Schritt 3: Konfigurieren von IAM-Berechtigungen für Amazon RDS
- Schritt 4: Anlegen und Konfigurieren von Benutzern
- Schritt 5: Aktivieren des VPC-übergreifenden Datenverkehrs zwischen dem Verzeichnis und der DB-Instance
- Schritt 6: Erstellen oder Ändern einer Oracle DB-Instance
- Schritt 7: Erstellen von Oracle-Anmeldeinformationen mit Kerberos-Authentifizierung
- Schritt 8: Konfigurieren eines Oracle-Clients

> **Note**
>
> Während der Einrichtung erstellt RDS einen Oracle-Datenbankbenutzer namens *managed_service_user* @ *example.com* mit der `CREATE SESSION` Berechtigung, wobei Ihr *example.com* Domainname steht. Dieser Benutzer entspricht dem Benutzer, den Directory Service in Ihrem verwalteten Active Directory erstellt. In regelmäßigen Abständen verwendet RDS die vom Directory Service bereitgestellten Anmeldeinformationen, um sich bei Ihrer Oracle-Datenbank anzumelden. Danach zerstört RDS sofort den Ticket-Cache.

Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD

AWS Directory Service erstellt ein vollständig verwaltetes Active Directory in der AWS Cloud. Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service erstellt in Ihrem Namen zwei Domänencontroller und DNS-Server (Domain Name System). Die Verzeichnisserver werden in verschiedenen Subnetzen in einer VPC erstellt. Diese Redundanz trägt dazu bei, dass Ihr Verzeichnis auch im Fehlerfall erreichbar bleibt.

Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service führt er in Ihrem Namen die folgenden Aufgaben aus:

- Einrichten eines Active Directory innerhalb der VPC.
- Erstellt ein Verzeichnisadministratorkonto mit dem Benutzernamen Admin und dem angegebenen Passwort. Mit diesem Konto verwalten Sie das Verzeichnis.

  > **Note**
  >
  > Achten Sie darauf, dieses Passwort zu speichern. AWS Directory Service speichert es nicht. Sie können es zurücksetzen, aber Sie können es nicht abrufen.

- Erstellt eine Sicherheitsgruppe für die Verzeichniscontroller.

Wenn Sie eine starten AWS Managed Microsoft AD, AWS erstellt eine Organisationseinheit (OU), die alle Objekte Ihres Verzeichnisses enthält. Diese OU hat den NetBIOS-Namen, den Sie bei der Erstellung Ihres Verzeichnisses angegeben haben. Sie befindet sich im Domänenstamm. Der Domänenstamm gehört und wird von diesem verwaltet AWS.

Das Administratorkonto, das mit Ihrem AWS Managed Microsoft AD Verzeichnis erstellt wurde, verfügt über Berechtigungen für die gängigsten Verwaltungsaktivitäten Ihrer Organisationseinheit:

- Erstellen, Aktualisieren oder Löschen von Benutzern
- Hinzufügen von Ressourcen zu Ihrer Domäne, etwa Datei- oder Druckserver, und anschließendes Gewähren der zugehörigen Ressourcenberechtigungen für Benutzer in der OU
- Zusätzliche OUs Container erstellen
- Delegieren von Befugnissen
- Wiederherstellen von gelöschten Objekten aus dem Active Directory-Papierkorb
- Führen Sie AD- und PowerShell DNS-Windows-Module im Active Directory-Webdienst aus

Das Admin-Konto hat außerdem die Rechte zur Durchführung der folgenden domänenweiten Aktivitäten:

- Verwalten von DNS-Konfigurationen (Hinzufügen, Entfernen oder Aktualisieren von Datensätzen, Zonen und Weiterleitungen)
- Aufrufen von DNS-Ereignisprotokollen

- Anzeigen von Sicherheitsereignisprotokollen

Verwenden Sie die, oder die AWS Directory Service API AWS Management Console, um das AWS CLI Verzeichnis zu erstellen. Stellen Sie sicher, dass Sie die relevanten ausgehenden Ports in der Verzeichnis-Sicherheitsgruppe öffnen, damit das Verzeichnis mit der Oracle-DB-Instance kommunizieren kann.

Um ein Verzeichnis zu erstellen mit AWS Managed Microsoft AD

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die AWS Directory Service Konsole unter https://console.aws.amazon.com/directoryservicev2/.
2. Wählen Sie im Navigationsbereich Directories (Verzeichnisse) aus. Wählen Sie denn Set up Directory (Verzeichnis einrichten) aus.
3. Wähle AWS Managed Microsoft AD. AWS Managed Microsoft AD ist die einzige Option, die Sie derzeit mit Amazon RDS verwenden können.
4. Geben Sie die folgenden Informationen ein:

   DNS-Name des Verzeichnisses

   Den vollständig qualifizierten Namen für das Verzeichnis, z. B. **`corp.example.com`**.

   NetBIOS-Name des Verzeichnisses

   Die kurzen Namen für das Verzeichnis, z. B. **`CORP`**.

   Verzeichnisbeschreibung

   (Optional) Eine Beschreibung für das Verzeichnis.

   Administratorpasswort

   Das Passwort für den Verzeichnisadministrator. Während des Verzeichniserstellungsprozesses wird ein Administratorkonto mit dem Benutzernamen Admin und diesem Passwort angelegt.

   Das Passwort für den Verzeichnisadministrator das nicht das Wort "admin" enthalten. Beachten Sie beim Passwort die Groß- und Kleinschreibung und es muss 8 bis 64 Zeichen lang sein. Zudem muss es mindestens ein Zeichen aus dreien der vier folgenden Kategorien enthalten:

   - Kleinbuchstaben (a–z)

- Großbuchstaben (A–Z)
- Zahlen (0–9)
- Nicht-alphanumerische Zeichen (~!@#$%^&*_-+=`|\(){}[]:;'"<>,.?/)

Passwort bestätigen

Das Administratorpasswort, das erneut eingegeben wurde.

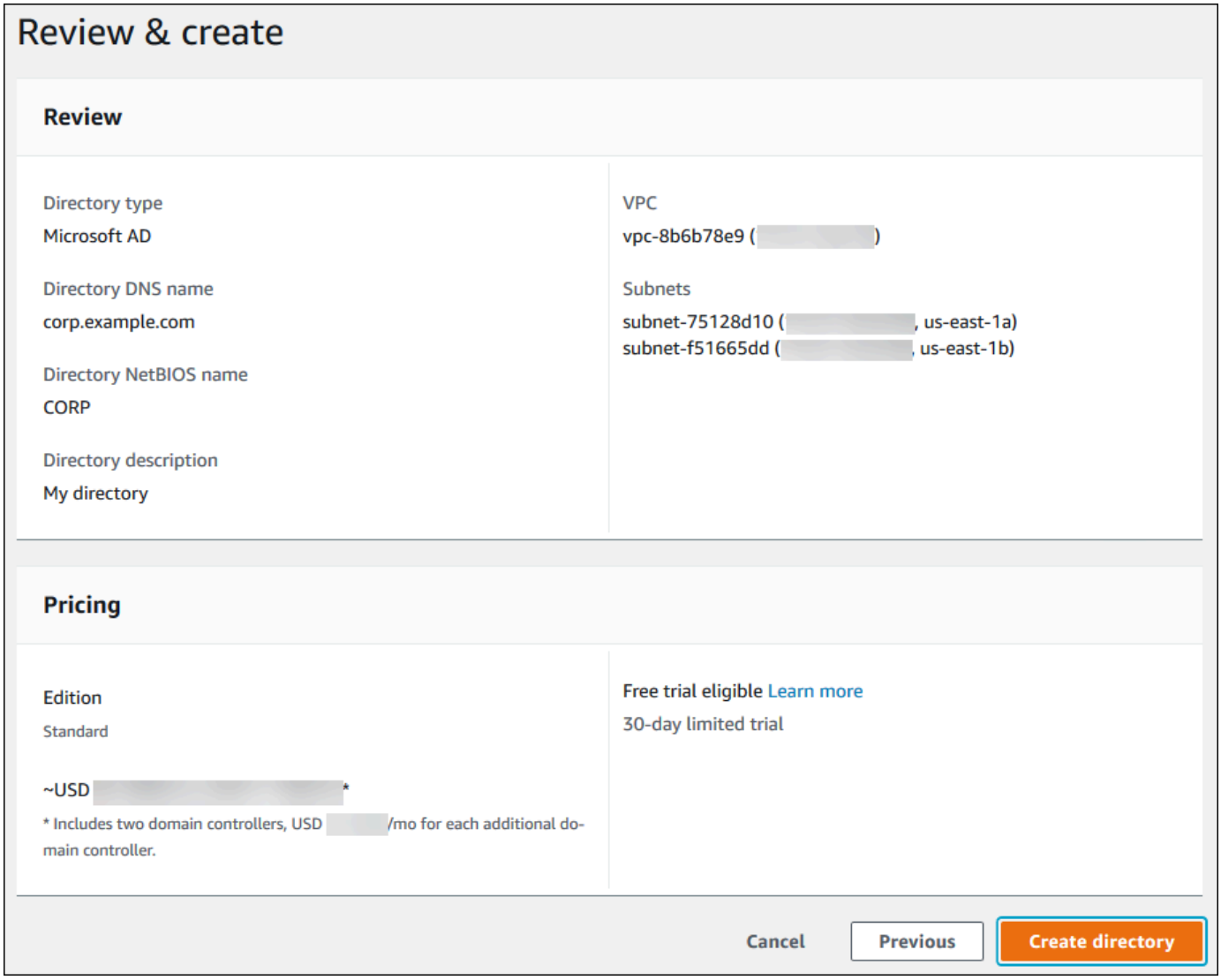
5. Wählen Sie Weiter aus.
6. Geben Sie die folgenden Informationen in den Abschnitt Networking ein. Wählen Sie dann Next (Weiter) aus:

   VPC

   Die VPC für das Verzeichnis. Erstellen Sie die Oracle-DB-Instance in derselben VPC.

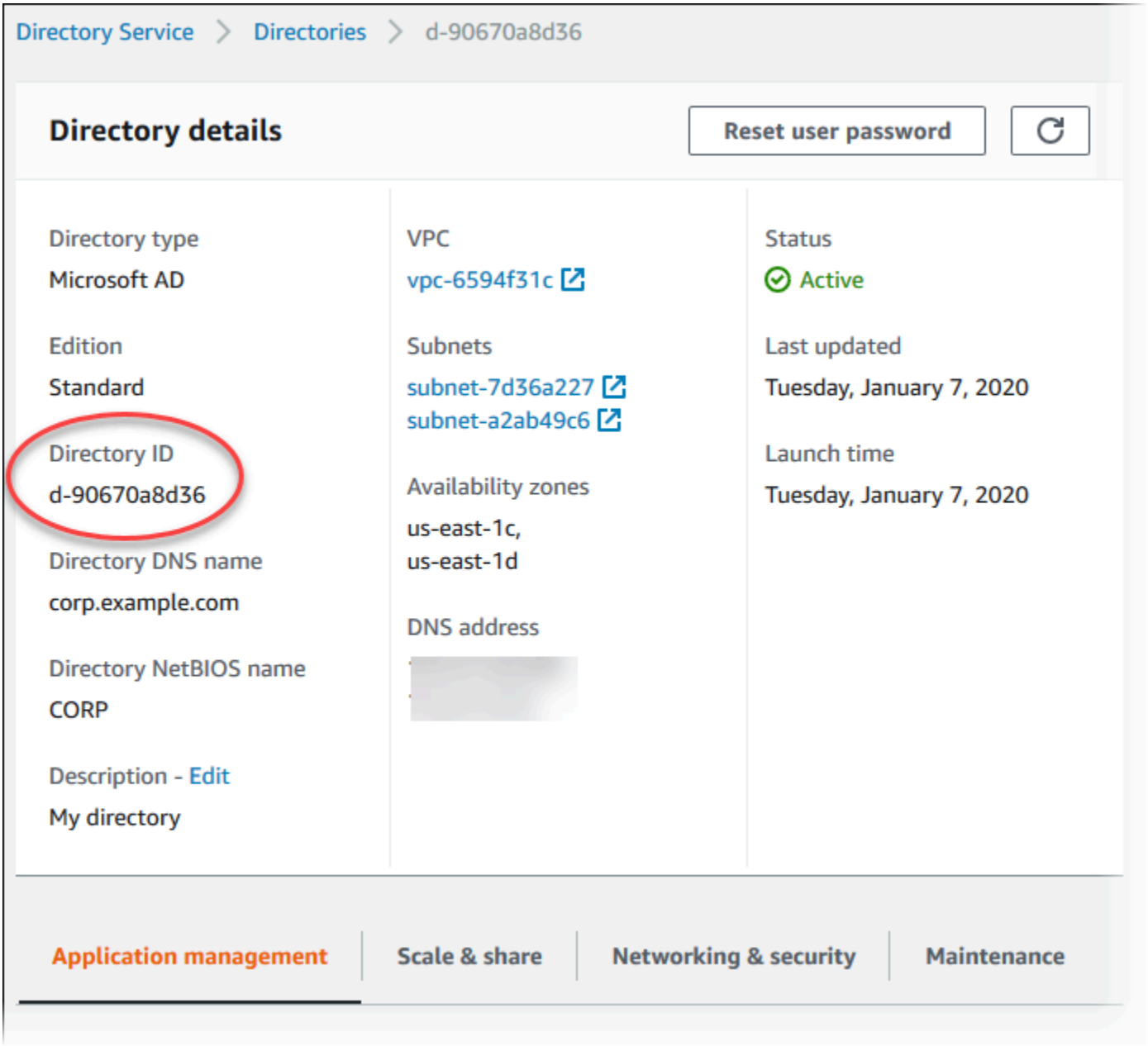
   Subnetze

   Subnetze für die Verzeichnisserver. Die beiden Subnetze müssen zu verschiedenen Availability-Zonen gehören.
7. Prüfen Sie die Verzeichnisinformationen und nehmen Sie ggf. Änderungen vor. Wenn die Informationen richtig sind, wählen Sie Create directory (Verzeichnis erstellen).



Es dauert einige Minuten, bis das Verzeichnis erstellt wurde. Wenn es erfolgreich erstellt wurde, ändert sich der Wert Status in Active (Aktiv).

Um Informationen über Ihr Verzeichnis anzuzeigen, wählen Sie den Verzeichnisnamen in der Verzeichnisliste aus. Notieren Sie den Wert Directory ID (Verzeichnis-ID), da Sie diesen Wert benötigen, wenn Sie Ihre Oracle-DB-Instance erstellen oder ändern.



Schritt 2: Erstellen einer Vertrauensstellung

Wenn Sie AWS Managed Microsoft AD nur verwenden möchten, fahren Sie mit fortSchritt 3: Konfigurieren von IAM-Berechtigungen für Amazon RDS.

Um die Kerberos-Authentifizierung mit Ihrem selbstverwalteten Active Directory zu aktivieren, müssen Sie eine Gesamtstruktur-Vertrauensstellung zwischen Ihrem selbstverwalteten Active Directory und dem im vorherigen Schritt AWS Managed Microsoft AD erstellten Active Directory einrichten. Die Vertrauensstellung kann einseitig sein, wobei das selbstverwaltete Active Directory als AWS Managed Microsoft AD vertrauenswürdig eingestuft wird. Die Vertrauensstellung kann auch

bidirektional erfolgen, wobei beide Active Directories einander vertrauen. Weitere Informationen zur Einrichtung von Gesamtvertrauensstellungen mithilfe von Cookies finden Sie unter Wann AWS Directory Service sollte eine Vertrauensstellung eingerichtet werden? im Administratorhandbuch.AWS Directory Service

Schritt 3: Konfigurieren von IAM-Berechtigungen für Amazon RDS

Um Sie anrufen AWS Directory Service zu können, benötigt Amazon RDS eine IAM-Rolle, die die verwaltete IAM-Richtlinie verwendet. `AmazonRDSDirectoryServiceAccess` Diese Rolle ermöglicht es Amazon RDS, Aufrufe von AWS Directory Service durchzuführen.

> **Note**
>
> Damit die Rolle Zugriff gewährt, muss der Endpunkt AWS Security Token Service (AWS STS) in der AWS-Region für Sie richtigen Weise aktiviert sein. AWS-Konto AWS STS Endpunkte sind standardmäßig in allen aktiv AWS-Regionen, und Sie können sie ohne weitere Aktionen verwenden. Weitere Informationen finden Sie unter Aktivierung und Deaktivierung AWS STSAWS-Region im IAM-Benutzerhandbuch.

Erstellen einer IAM-Rolle

Wenn Sie eine DB-Instance mit dem erstellen und der AWS Management Console Konsolenbenutzer über die `iam:CreateRole` entsprechende Berechtigung verfügt, wird die Konsole automatisch erstellt`rds-directoryservice-kerberos-access-role`. Andernfalls müssen Sie die IAM-Rolle manuell erstellen. Wenn Sie eine IAM-Rolle manuell erstellen`Directory Service`, wählen Sie die AWS verwaltete Richtlinie aus und hängen Sie sie `AmazonRDSDirectoryServiceAccess` an.

Weitere Informationen zum Erstellen von IAM-Rollen für einen Dienst finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

> **Note**
>
> Die für die Windows-Authentifizierung für RDS for Microsoft SQL Server verwendete IAM-Rolle kann nicht für RDS for Oracle verwendet werden.

Manuelles Erstellen einer IAM-Vertrauensrichtlinie

Optional können Sie Ressourcen-Richtlinien mit den erforderlichen Berechtigungen erstellen, anstatt die verwaltete IAM-Richtlinie zu verwenden `AmazonRDSDirectoryServiceAccess`. Geben Sie sowohl `directoryservice.rds.amazonaws.com` als auch `rds.amazonaws.com` als Prinzipale an.

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die Amazon RDS einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken. Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontextschlüssels `aws:SourceArn` mit dem vollständigen ARN einer Amazon-RDS-Ressource. Weitere Informationen finden Sie unter Vermeidung des dienstübergreifenden Confused-Deputy-Problems.

Das folgende Beispiel zeigt, wie Sie die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` für Amazon RDS verwenden können, um das Confused-Deputy-Problem zu verhindern.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
```

```
      }
    }
  ]
}
```

Für Opt-in-Regionen müssen Sie auch einen Service Principal für diese Region in der Form von angeben. `directoryservice.rds.region_name.amazonaws.com` Verwenden Sie in der Region Afrika (Kapstadt) beispielsweise die folgende Vertrauensrichtlinie:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "directoryservice.rds.af-south-1.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:af-south-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Die Rolle muss auch über die folgende IAM-Richtlinie verfügen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Schritt 4: Anlegen und Konfigurieren von Benutzern

Sie können Benutzer mit dem Active Directory-Tool "Benutzer und Computer" erstellen, das eines der Active Directory Domain Services und der Active Directory Lightweight Directory Services-Tools ist. In diesem Fall sind Benutzer Einzelpersonen oder Entitäten, die Zugriff auf Ihr Verzeichnis haben.

Um Benutzer in einem AWS Directory Service Verzeichnis zu erstellen, müssen Sie mit einer Windows-basierten EC2 Amazon-Instance verbunden sein, die Mitglied des AWS Directory Service Verzeichnisses ist. Gleichzeitig müssen Sie als Benutzer angemeldet sein, der über Rechte zum Erstellen von Benutzern verfügt. Weitere Informationen zum Erstellen von Benutzern in Ihrem Microsoft Active Directory finden Sie unter Verwalten von Benutzern und Gruppen in AWS Managed Microsoft AD im AWS Directory Service Administration Guide.

Schritt 5: Aktivieren des VPC-übergreifenden Datenverkehrs zwischen dem Verzeichnis und der DB-Instance

Wenn Sie beabsichtigen, das Verzeichnis und die DB-Instance in derselben VPC zu platzieren, überspringen Sie diesen Schritt und fahren Sie mit for Schritt 6: Erstellen oder Ändern einer Oracle DB-Instance.

Wenn Sie planen, das Verzeichnis und die DB-Instance in verschiedenen AWS Konten zu lokalisieren oder VPCs den VPC-übergreifenden Verkehr mithilfe von VPC-Peering oder Transit Gateway zu

konfigurieren.AWS Das folgende Verfahren ermöglicht den Datenverkehr zwischen VPCs mithilfe von VPC-Peering. Folgen Sie den Anweisungen unter Was ist VPC Peering? im Handbuch zu Amazon Virtual Private Cloud-Peering.

Aktivieren des VPC-übergreifenden Datenverkehrs mit VPC Peering

1. Richten Sie geeignete VPC-Routing-Regeln ein, um sicherzustellen, dass Netzwerk-Datenverkehr in beide Richtungen fließen kann.
2. Stellen Sie sicher, dass die Sicherheitsgruppe der DB-Instance eingehenden Datenverkehr von der Sicherheitsgruppe des Verzeichnisses empfangen kann. Weitere Informationen finden Sie unter Best Practices für AWS Managed Microsoft AD im AWS Directory Service Leitfaden für Administratoren.
3. Stellen Sie sicher, dass keine ACL-Regel (Network Access Control List) zum Blockieren des Datenverkehrs vorhanden ist.

Wenn ein anderes AWS Konto Eigentümer des Verzeichnisses ist, müssen Sie das Verzeichnis gemeinsam nutzen.

Um das Verzeichnis von mehreren AWS Konten gemeinsam zu nutzen

1. Beginnen Sie mit der gemeinsamen Nutzung des Verzeichnisses mit dem AWS Konto, unter dem die DB-Instance erstellt werden soll. Folgen Sie dazu den Anweisungen im AWS Directory Service Administratorhandbuch im Tutorial: AWS Managed Microsoft AD Verzeichnis teilen für einen nahtlosen EC2 Domänenbeitritt.
2. Melden Sie sich mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole an und stellen Sie sicher, dass die Domain den `SHARED` Status hat, bevor Sie fortfahren.
3. Notieren Sie sich den Wert der Verzeichnis-ID, während Sie mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole angemeldet sind. Sie verwenden diese Verzeichnis-ID, um die DB-Instance mit der Domäne zu verbinden.

Schritt 6: Erstellen oder Ändern einer Oracle DB-Instance

Erstellen oder ändern Sie eine Oracle DB-Instance für die Verwendung mit Ihrem Verzeichnis. Sie können die Konsole, CLI oder RDS-API verwenden, um eine DB-Instance einem Verzeichnis zuzuordnen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

- Erstellen Sie eine neue Oracle-DB-Instance mit der Konsole, dem create-db-instanceCLI-Befehl oder dem Vorgang Create DBInstance RDS API.

  Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine bestehende Oracle-DB-Instance mithilfe der Konsole, des modify-db-instanceCLI-Befehls oder der Operation Modify DBInstance RDS API.

  Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie eine Oracle-DB-Instance mithilfe der Konsole, des CLI-Befehls restore-db-instance-from-db-snapshot oder des API-Vorgangs Restore From DBSnapshot RDS DBInstanceaus einem DB-Snapshot wieder her.

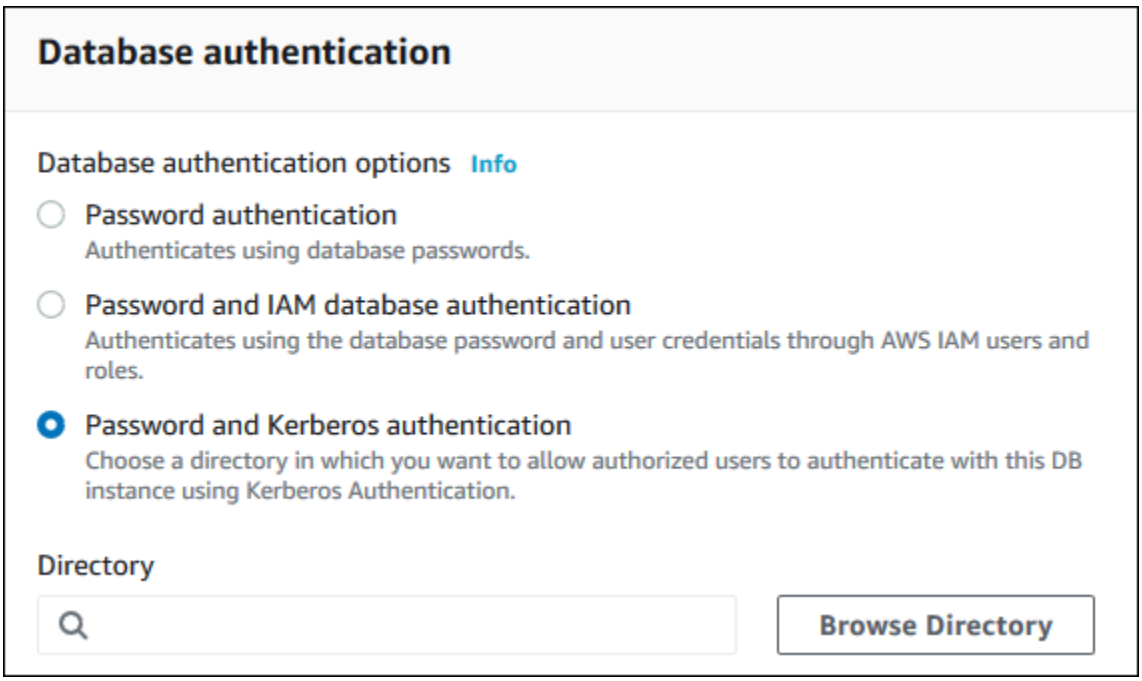
  Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.
- Stellen Sie eine Oracle-DB-Instance point-in-time mithilfe der Konsole, des Befehls restore-db-instance-to- point-in-time CLI oder der Operation Restore DBInstance ToPointInTime RDS API in einer wieder her.

  Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

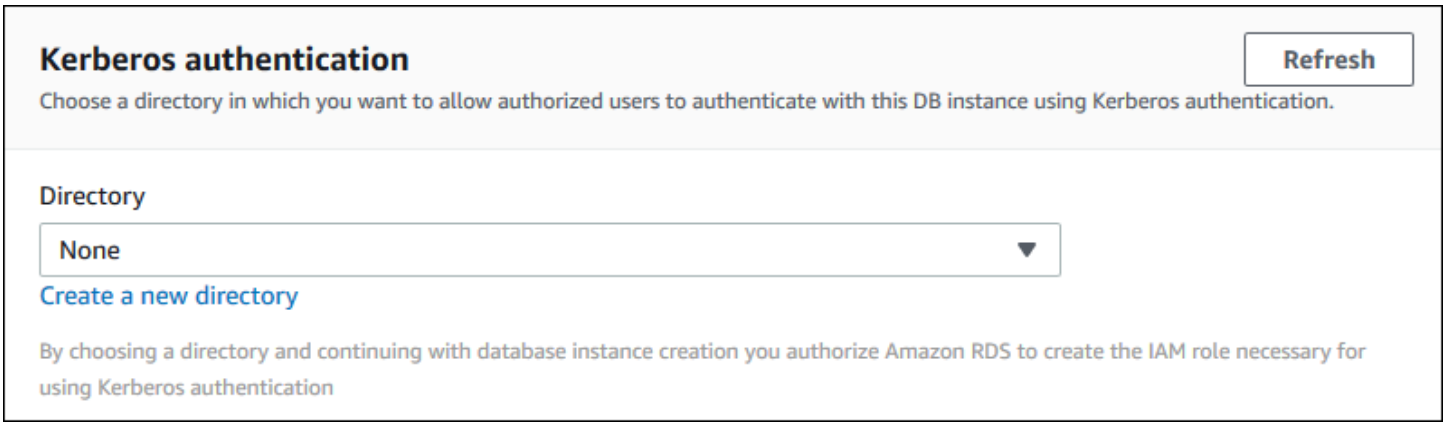
Die Kerberos-Authentifizierung wird nur für Oracle DB-Instances in einer VPC unterstützt. Die DB-Instance kann sich in derselben VPC wie das Verzeichnis oder in einer anderen VPC befinden. Wenn Sie die DB-Instance erstellen oder ändern, gehen Sie wie folgt vor:

- Geben Sie den Domänenbezeichner (`d-*`-Bezeichner) an, der beim Erstellen Ihres Verzeichnisses generiert wurde.
- Geben Sie außerdem den Namen der IAM-Rolle an, die Sie erstellt haben.
- Stellen Sie sicher, dass die Sicherheitsgruppe der DB-Instance eingehenden Datenverkehr von der Sicherheitsgruppe des Verzeichnisses empfangen und ausgehenden Datenverkehr an das Verzeichnis senden kann.

Wenn Sie die Konsole verwenden, um eine DB-Instance zu erstellen, wählen Sie im Abschnitt Datenbankauthentifizierung die Option Passwort- und Kerberos-Authentifizierung aus. Wählen Sie Verzeichnis durchsuchen und dann das Verzeichnis aus, oder klicken Sie auf Neues Verzeichnis erstellen.



Wenn Sie die Konsole zum Ändern oder Wiederherstellen einer DB-Instance verwenden, wählen Sie das Verzeichnis im Abschnitt Kerberos-Authentifizierung oder Neues Verzeichnis erstellen aus.



Wenn Sie den verwenden AWS CLI, sind die folgenden Parameter erforderlich, damit die DB-Instance das von Ihnen erstellte Verzeichnis verwenden kann:

- Für den `--domain`-Parameter verwenden Sie den Domänenbezeichner („d-*“-Bezeichner), der beim Erstellen des Verzeichnisses generiert wurde.
- Verwenden Sie für den `--domain-iam-role-name`-Parameter die von Ihnen erstellte Rolle, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet.

Beispielsweise ändert der folgende CLI-Befehl eine DB-Instance zur Verwendung eines Verzeichnisses.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

Important

Wenn Sie eine DB-Instance zur Aktivierung der Kerberos-Authentifizierung ändern, starten Sie die DB-Instance neu, nachdem Sie die Änderung vorgenommen haben.

Note

*MANAGED_SERVICE_USER*ist ein Dienstkonto, dessen Name zufällig von Directory Service for RDS generiert wird. Während der Einrichtung der Kerberos-Authentifizierung erstellt RDS for Oracle einen Benutzer mit demselben Namen und weist ihm die Berechtigung `CREATE SESSION` zu. Der Oracle-DB-Benutzer wird extern identifiziert als*MANAGED_SERVICE_USER@EXAMPLE.COM*, wo *EXAMPLE.COM* ist der Name Ihrer Domain. In regelmäßigen Abständen verwendet RDS die vom Directory Service bereitgestellten Anmeldeinformationen, um sich bei Ihrer Oracle-Datenbank anzumelden. Danach zerstört RDS sofort den Ticket-Cache.

Schritt 7: Erstellen von Oracle-Anmeldeinformationen mit Kerberos-Authentifizierung

Verwenden Sie die Anmeldeinformationen für den Amazon RDS-Hauptbenutzer, um eine Verbindung zur Oracle DB-Instance herzustellen, wie Sie es bei jeder anderen DB-Instance tun würden. Die DB-Instance ist mit der AWS Managed Microsoft AD Domain verbunden. Somit können Sie Oracle-Logins und -Benutzer von den Microsoft Active Directory-Benutzern in Ihrer Domain

bereitstellen. Zum Verwalten von Datenbankberechtigungen erteilen und widerrufen Sie Oracle-Standardberechtigungen für diese Anmeldungen.

Damit sich ein Microsoft Active Directory-Benutzer bei Oracle authentifizieren kann,

1. verwenden Sie die Anmeldeinformationen für den Amazon RDS-Hauptbenutzer, um eine Verbindung mit der Oracle DB-Instance herzustellen.
2. Erstellen Sie einen extern authentifizierten Benutzer in der Oracle-Datenbank.

   Im folgenden Beispiel ersetzen Sie *KRBUSER@CORP.EXAMPLE.COM* mit dem Benutzer- und Domänennamen.

   ```
   CREATE USER "KRBUSER@CORP.EXAMPLE.COM" IDENTIFIED EXTERNALLY;
   GRANT CREATE SESSION TO "KRBUSER@CORP.EXAMPLE.COM";
   ```

   Benutzer (sowohl Menschen als auch Anwendungen) aus Ihrer Domäne können sich nun von einem mit der Domäne verbundenen Client-Rechner aus mit Hilfe der Kerberos-Authentifizierung mit der Oracle-DB-Instance verbinden.

Schritt 8: Konfigurieren eines Oracle-Clients

Um einen Oracle-Client zu konfigurieren, müssen Sie die folgenden Voraussetzungen erfüllen:

- Erstellen Sie eine Konfigurationsdatei namens krb5.conf (Linux) oder krb5.ini (Windows), die auf die Domäne verweist. Konfigurieren Sie den Oracle-Client für die Verwendung dieser Konfigurationsdatei.
- Stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und AWS Directory Service über DNS-Port 53 über TCP/UDP, Kerberos-Ports (88 und 464 für verwaltet AWS Directory Service) über TCP und LDAP-Port 389 über TCP fließen kann.
- Stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und der DB-Instance über den Datenbank-Port fließen kann.

Im Folgenden finden Sie einen Beispielinhalt für. AWS Managed Microsoft AD

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
```

```
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = CORP.EXAMPLE.COM
 example.com = CORP.EXAMPLE.COM
```

Im Folgenden finden Sie Beispielinhalte für Microsoft AD vor Ort. Ersetzen Sie es in Ihrer Datei krb5.conf oder krb5.ini *on-prem-ad-server-name* durch den Namen Ihres lokalen AD-Servers.

```
[libdefaults]
 default_realm = ONPREM.COM
[realms]
 AWSAD.COM = {
  kdc = awsad.com
  admin_server = awsad.com
 }
 ONPREM.COM = {
  kdc = on-prem-ad-server-name
  admin_server = on-prem-ad-server-name
 }
[domain_realm]
 .awsad.com = AWSAD.COM
 awsad.com= AWSAD.COM
 .onprem.com = ONPREM.COM
 onprem.com= ONPREM.COM
```

> **Note**
>
> Nachdem Sie Ihre Datei krb5.ini oder krb5.conf konfiguriert haben, empfehlen wir Ihnen, den Server neu zu starten.

Im Folgenden finden Sie ein Beispiel für den Inhalt von sqlnet.ora für eine SQL*Plus-Konfiguration:

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5PRE,KERBEROS5)
SQLNET.KERBEROS5_CONF=path_to_krb5.conf_file
```

Ein Beispiel für eine SQL Developer-Konfiguration finden Sie unter Document 1609359.1 des Oracle-Supports.

Verwalten einer DB-Instance in einer Domäne

Sie können die Konsole, die CLI oder die RDS-API verwenden, um Ihre DB-Instance und ihre Beziehung zu Ihrem Microsoft Active Directory zu verwalten. Sie können z. B. ein Microsoft Active Directory zuordnen, um die Kerberos-Authentifizierung zu aktivieren. Sie können auch die Zuordnung eines Microsoft Active Directory trennen, um die Kerberos-Authentifizierung zu deaktivieren. Sie können auch eine DB-Instance verschieben, die von einem Microsoft Active Directory zu einem anderen extern authentifiziert wird.

Sie können z. B. mithilfe der CLI Folgendes tun:

- Um erneut zu versuchen, die Kerberos-Authentifizierung für eine fehlgeschlagene Mitgliedschaft zu aktivieren, verwenden Sie den CLI-Befehl modify-db-instance und geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft für die Option `--domain` an.
- Um die Kerberos-Authentifizierung auf einer DB-Instance zu deaktivieren, verwenden Sie den CLI-Befehl modify-db-instance und geben Sie `none` für die Option `--domain` an.
- Um eine DB-Instance von einer Domäne zu einer anderen zu verschieben, verwenden Sie den CLI-Befehl modify-db-instance und geben Sie den Domänenbezeichner der neuen Domäne für die Option `--domain` an.

Anzeigen des Status einer Domänen-Mitgliedschaft

Nachdem Sie Ihre DB-Instance erstellt oder modifiziert haben, wird die DB-Instance ein Mitglied der Domäne. Sie können den Status der Domänenmitgliedschaft für die DB-Instance in der Konsole anzeigen oder den CLI-Befehl describe-db-instances ausführen. Der Status der DB-Instance kann einer der folgenden sein:

- `kerberos-enabled` – Für die DB-Instance ist die Kerberos-Authentifizierung aktiviert.
- `enabling-kerberos` – AWS ist dabei, die Kerberos-Authentifizierung auf dieser DB-Instance zu aktivieren.
- `pending-enable-kerberos` – Das Aktivieren der Kerberos-Authentifizierung ist für diese DB-Instance ausstehend.
- `pending-maintenance-enable-kerberos` – AWS versucht, die Kerberos-Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu aktivieren.
- `pending-disable-kerberos` – Das Deaktivieren der Kerberos-Authentifizierung ist für diese DB-Instance ausstehend.

- `pending-maintenance-disable-kerberos` – AWS versucht, die Kerberos-Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu deaktivieren.
- `enable-kerberos-failed` – Ein Konfigurationsproblem hat AWS daran gehindert, die Kerberos-Authentifizierung auf der DB-Instance zu aktivieren. Beheben Sie das Konfigurationsproblem, bevor Sie den Befehl zum Ändern der DB-Instance erneut ausgeben.
- `disabling-kerberos` – AWS ist dabei, die Kerberos-Authentifizierung auf dieser DB-Instance zu deaktivieren.

Eine Anfrage zur Aktivierung der Kerberos-Authentifizierung kann wegen eines Netzwerkverbindungsproblems oder einer falschen IAM-Rolle fehlschlagen. Wenn der Versuch, die Kerberos-Authentifizierung zu aktivieren, fehlschlägt, wenn Sie eine DB-Instance erstellen oder ändern, stellen Sie sicher, dass Sie die richtige IAM-Rolle verwenden. Ändern Sie dann die DB-Instance, um der Domäne beizutreten.

> **Note**
>
> Nur die Kerberos-Authentifizierung mit Amazon RDS for Oracle sendet Datenverkehr an die DNS-Server der Domäne. Alle anderen DNS-Anforderungen werden als ausgehender Netzwerkzugriff auf Ihren DB-Instances behandelt, auf denen Oracle ausgeführt wird. Weitere Informationen zu ausgehendem Netzwerkzugriff mit Amazon RDS für Oracle finden Sie unter Einen benutzerdefinierten DNS Server einrichten.

Kerberos-Schlüssel mit erzwungener Rotation

Ein geheimer Schlüssel wird zwischen AWS Managed Microsoft AD und Amazon RDS for Oracle für die Oracle DB-Instance geteilt. Dieser Schlüssel wird alle 45 Tage automatisch rotiert. Sie können die folgende Amazon RDS-Prozedur verwenden, um die Rotation dieses Schlüssels zu erzwingen.

```
SELECT rdsadmin.rdsadmin_kerberos_auth_tasks.rotate_kerberos_keytab AS TASK_ID FROM
 DUAL;
```

> **Note**
>
> In einer Lesereplikat-Konfiguration ist diese Vorgehensweise nur auf der Quell-DB-Instance und nicht auf dem Lesereplikat verfügbar.

Die Anweisung SELECT gibt die ID der Aufgabe in einem VARCHAR2-Datentyp zurück. Sie können den Status einer laufenden Aufgabe in einer bdump-Datei einsehen. Die bdump-Dateien befinden sich im Verzeichnis /rdsdbdata/log/trace. Jeder bdump-Dateiname weist das folgende Format auf.

```
dbtask-task-id.log
```

Sie können das Ergebnis anzeigen, indem Sie die Ausgabedatei der Aufgabe anzeigen.

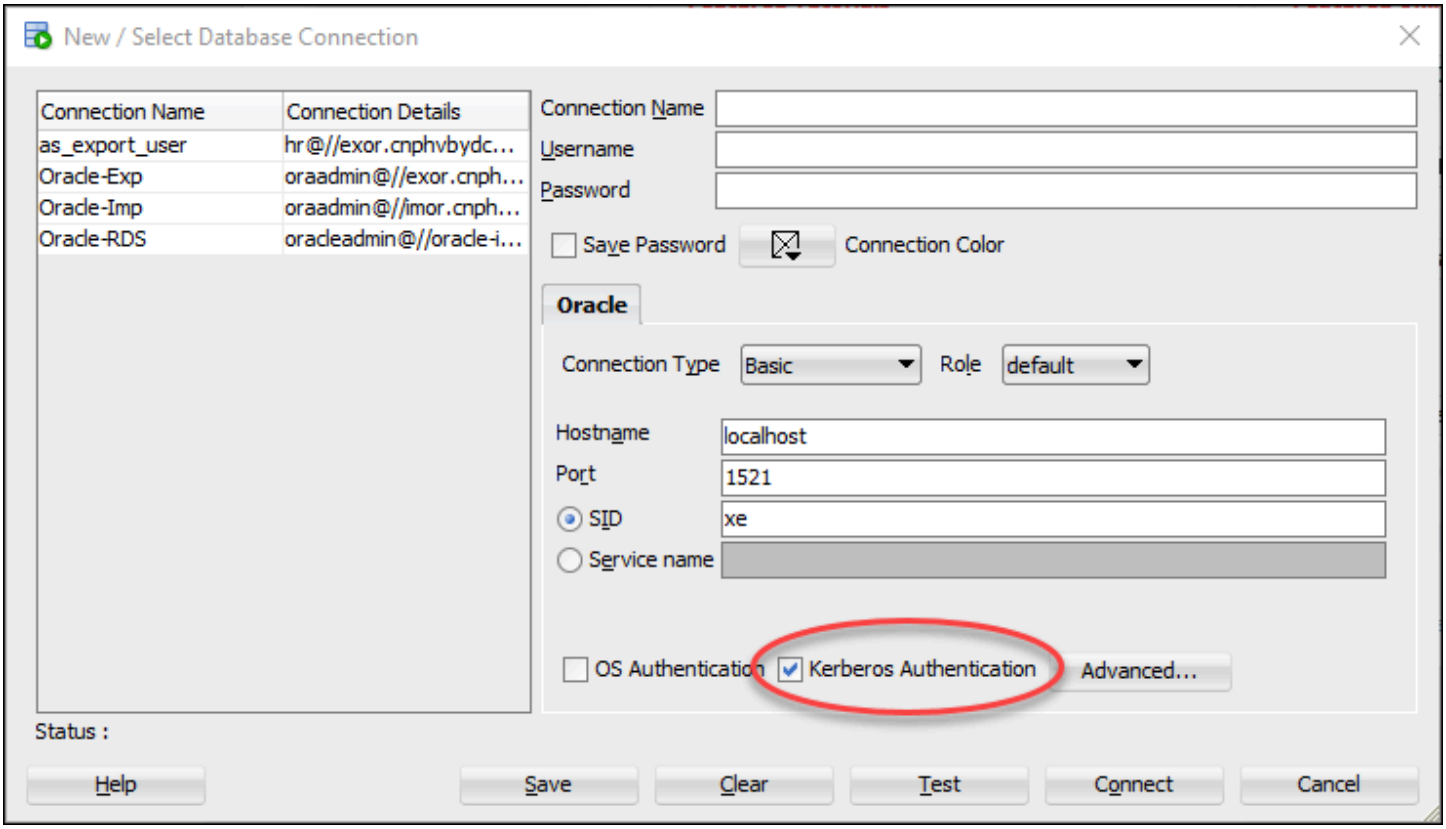
```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-
id.log'));
```

Ersetzen Sie *task-id* durch die von der Prozedur zurückgegebene Aufgaben-ID.

> **Note**
>
> Die Aufgaben werden asynchron ausgeführt.

Herstellen einer Verbindung mit Oracle mithilfe der Kerberos-Authentifizierung

In diesem Abschnitt wird davon ausgegangen, dass Sie Ihren Oracle-Client wie unter beschrieben eingerichtet habe Schritt 8: Konfigurieren eines Oracle-Clients. Um eine Verbindung mit der Oracle DB Kerberos-Authentifizierung herzustellen, melden Sie sich mit dem Kerberos-Authentifizierungstyp an. Wählen Sie beispielsweise nach dem Start von Oracle SQL Developer Kerberos Authentication als Authentifizierungstyp, wie im folgenden Beispiel gezeigt.



So stellen Sie eine Verbindung mit Oracle mit Kerberos-Authentifizierung mit SQL*Plus her:

1. Führen Sie über die Eingabeaufforderung den folgenden Befehl aus:

   ```
   kinit username
   ```

   Ersetzen Sie *username* durch den Benutzernamen und geben Sie über die Eingabeaufforderung das Passwort für den Benutzer ein, das im Microsoft Active Directory gespeichert ist.

2. Öffnen Sie SQL*Plus und stellen Sie eine Verbindung über den DNS-Namen und die Portnummer für die Oracle DB-Instance her.

   Weitere Informationen zum Herstellen einer Verbindung mit einer Oracle DB-Instance in SQL*Plus finden Sie unter Herstellen einer Verbindung mit Ihrer DB-Instance mithilfe von SQL*Plus.

> **Tip**
>
> Wenn Sie einen systemeigenen Windows-Cache verwenden, können Sie den `SQLNET.KERBEROS5_CC_NAME` Parameter auch auf `OSMSFT://` oder `MSLSA` in der Datei

sqlnet.ora setzen, um die im Microsoft Active Directory gespeicherten Anmeldeinformationen zu verwenden.

Konfigurieren des UTL_HTTP-Zugriffs mit Zertifikaten und einer Oracle Wallet

Amazon RDS unterstützt ausgehenden Netzwerkzugriff auf Ihre DB-Instances von RDS für Oracle. Verbinden Sie Ihre DB-Instance mithilfe der folgenden PL/SQL-Pakete mit dem Netzwerk:

`UTL_HTTP`

Dieses Paket macht HTTP-Aufrufe von SQL und PL/SQL. Sie können damit über HTTP auf Daten im Internet zugreifen. Weitere Informationen finden Sie unter UTL_HTTP in der Oracle-Dokumentation.

`UTL_TCP`

Dieses Paket bietet clientseitige TCP/IP-Zugriffsfunktionalität in PL/SQL. Dieses Paket ist nützlich für PL/SQL-Anwendungen, die Internetprotokolle und E-Mails verwenden. Weitere Informationen finden Sie unter UTL_TCP in der Oracle-Dokumentation.

`UTL_SMTP`

Dieses Paket bietet Schnittstellen zu den SMTP-Befehlen, die es einem Client ermöglichen, E-Mails an einen SMTP-Server zu senden. Weitere Informationen finden Sie unter UTL_SMTP in der Oracle-Dokumentation.

Wenn Sie die folgenden Aufgaben ausführen, können Sie `UTL_HTTP.REQUEST` konfigurieren, um mit Websites zu arbeiten, die während des SSL-Handshakes Clientauthentifizierungszertifikate benötigen. Sie können auch die Kennwortauthentifizierung für `UTL_HTTP`-Zugriff auf Websites konfigurieren, indem Sie die Befehle für das Generieren der Oracle Wallet und den Prozess `DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE` ändern. Weitere Informationen finden Sie unter DBMS_NETWORK_ACL_ADMIN in der Dokumentation zu Oracle Database.

> **Note**
>
> Sie können folgende Aufgaben für `UTL_SMTP` anpassen, mit denen Sie E-Mails über SSL/TLS senden (einschließlich Amazon Simple Email Service).

Themen

- Überlegungen zur Konfiguration des UTL_HTTP-Zugriffs
- Schritt 1: Abrufen des Stammzertifikats für eine Website
- Schritt 2: Erstellen einer Oracle Wallet
- Schritt 3: Herunterladen Ihrer Oracle Wallet auf Ihre RDS-for-Oracle-Instance
- Schritt 4: Erteilen Sie Benutzerberechtigungen für die Oracle Wallet
- Schritt 5: Konfigurieren des Zugriffs auf eine Website von Ihrer DB-Instance
- Schritt 6: Testen der Verbindungen Ihrer DB-Instance zu einer Website

Überlegungen zur Konfiguration des UTL_HTTP-Zugriffs

Berücksichtigen Sie Folgendes, bevor Sie den Zugriff konfigurieren:

- Sie können SMTP mit der Option UTL_MAIL verwenden. Weitere Informationen finden Sie unter Orakel UTL _ MAIL.
- Der DNS-Name (Domain Name Server) des Remote-Hosts kann einer der folgenden sein:
  - Öffentlich auflösbar.
  - Der Endpunkt einer Amazon RDS-DB-Instance.
  - Auflösbar über einen benutzerdefinierten DNS-Server. Weitere Informationen finden Sie unter Einen benutzerdefinierten DNS Server einrichten.
  - Der private DNS-Name einer Amazon EC2-Instance in derselben VPC oder einer gleichrangigen VPC. Stellen Sie in diesem Fall sicher, dass der Name über einen benutzerdefinierten DNS-Server auflösbar ist. Alternativ können Sie für die Verwendung des von Amazon bereitgestellten DNS das Attribut `enableDnsSupport` in den VPC-Einstellungen und den Support für DNS-Auflösung für die gleichrangige VPC-Verbindung aktivieren. Weitere Informationen finden Sie unter DNS-Unterstützung für Ihre VPC und Ändern Ihrer VPC-Peering-Verbindung.
  - Für eine sichere Verbindung mit Remote-SSL/TLS-Ressourcen empfehlen wir Ihnen, benutzerdefinierte Oracle Wallets zu erstellen und hochzuladen. Mittels der Funktion zur Amazon S3-Integration in Amazon RDS for Oracle können Sie ein Wallet von Amazon S3 auf Oracle-DB-Instances herunterladen. Weitere Informationen über die Amazon S3-Integration für Oracle finden Sie unter Amazon S3-Integration.
- Sie können Datenbankverbindungen zwischen Oracle-DB-Instances über einen SSL/TLS-Endpunkt einrichten, wenn die Oracle-SSL-Option für jede Instance konfiguriert ist. Weitere

Konfigurationseinstellungen sind nicht erforderlich. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.

Schritt 1: Abrufen des Stammzertifikats für eine Website

Damit die DB-Instance von RDS für Oracle sichere Verbindungen zu einer Website herstellen kann, fügen Sie das Stammzertifizierungsstellenzertifikat hinzu. Amazon RDS verwendet das Stammzertifikat, um das Website-Zertifikat für die Oracle Wallet zu signieren.

Sie können das Stammzertifikat auf verschiedene Arten abrufen. Sie können z. B. Folgendes tun:

1. Verwenden Sie einen Webserver, um die durch das Zertifikat gesicherte Website aufzurufen.
2. Laden Sie das Stammzertifikat herunter, das zum Signieren verwendet wurde.

Für AWS-Services werden die Zertifikate üblicherweise im Amazon Trust Services Repository hinterlegt.

Schritt 2: Erstellen einer Oracle Wallet

Erstellen Sie eine Oracle Wallet, die sowohl die Webserver-Zertifikate als auch die Clientauthentifizierungszertifikate enthält. Die RDS-Oracle-Instance verwendet das Webserver-Zertifikat, um eine sichere Verbindung zur Website herzustellen. Die Website benötigt das Clientzertifikat, um den Oracle-Datenbankbenutzer zu authentifizieren.

Möglicherweise möchten Sie sichere Verbindungen konfigurieren, ohne Clientzertifikate für die Authentifizierung zu verwenden. In diesem Fall können Sie die Java-Keystore-Schritte im folgenden Prozess überspringen.

Erstellen einer Oracle Wallet

1. Hinterlegen Sie die Root- und Clientzertifikate in einem einzelnen Verzeichnis und wechseln Sie dann in dieses Verzeichnis.
2. Konvertieren Sie das .p12-Clientzertifikat in den Java-Keystore.

> **Note**
>
> Wenn Sie keine Clientzertifikate für die Authentifizierung verwenden, können Sie diesen Schritt überspringen.

Im folgenden Beispiel wird das Clientzertifikat mit dem Namen *client_certificate.p12* zum Java-Keystore mit dem Namen *client_keystore.jks* konvertiert. Der Keystore wird dann in die Oracle Wallet integriert. Das Keystore-Passwort lautet *P12PASSWORT*.

```
orapki wallet pkcs12_to_jks -wallet ./client_certificate.p12 -
jksKeyStoreLoc ./client_keystore.jks -jksKeyStorepwd P12PASSWORD
```

3. Erstellen Sie ein Verzeichnis für Ihre Oracle Wallet, das sich vom Zertifikatsverzeichnis unterscheidet.

   Im folgenden Beispiel wird das Verzeichnis `/tmp/wallet` erstellt.

   ```
   mkdir -p /tmp/wallet
   ```

4. Erstellen Sie eine Oracle Wallet in Ihrem Wallet-Verzeichnis.

   Im folgenden Beispiel wird das Oracle-Wallet-Passwort auf *P12PASSWORT* festgelegt. Das ist das gleiche Passwort, das der Java-Keystore in einem vorherigen Schritt verwendet hat. Die Verwendung desselben Passworts ist praktisch, aber nicht notwendig. Der Parameter `-auto_login` aktiviert die automatische Anmeldefunktion, sodass Sie nicht jedes Mal ein Passwort angeben müssen, wenn Sie darauf zugreifen möchten.

   > **Note**
   >
   > Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

   ```
   orapki wallet create -wallet /tmp/wallet -pwd P12PASSWORD -auto_login
   ```

5. Fügen Sie den Java-Keystore zu Ihrer Oracle Wallet hinzu.

   > **Note**
   >
   > Wenn Sie keine Clientzertifikate für die Authentifizierung verwenden, können Sie diesen Schritt überspringen.

Im folgenden Beispiel wird der Keystore *client_keystore.jks* zur Oracle Wallet namens */tmp/wallet* hinzugefügt. In diesem Beispiel geben Sie das gleiche Passwort für den Java-Keystore und die Oracle Wallet an.

```
orapki wallet jks_to_pkcs12 -wallet /tmp/wallet -pwd P12PASSWORD -
keystore ./client_keystore.jks -jkspwd P12PASSWORD
```

6. Fügen Sie das Stammzertifikat für Ihre Ziel-Website der Oracle Wallet hinzu.

   Im folgenden Beispiel wird ein Zertifikat mit dem Namen *Root_CA.cer* hinzugefügt.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Root_CA.cer -
   pwd P12PASSWORD
   ```

7. Hinzufügen von Zwischenzertifikaten

   Im folgenden Beispiel wird ein Zertifikat mit dem Namen *Intermediate.cer* hinzugefügt. Wiederholen Sie diesen Schritt so oft wie nötig, bis Sie alle Zwischenzertifikate geladen haben.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Intermediate.cer -
   pwd P12PASSWORD
   ```

8. Bestätigen Sie, dass Ihre neu erstellte Oracle Wallet über die erforderlichen Zertifikate verfügt.

   ```
   orapki wallet display -wallet /tmp/wallet -pwd P12PASSWORD
   ```

Schritt 3: Herunterladen Ihrer Oracle Wallet auf Ihre RDS-for-Oracle-Instance

In diesem Schritt laden Sie Ihre Oracle Wallet auf Amazon S3 hoch und laden dann die Wallet von Amazon S3 auf Ihre RDS-for-Oracle-Instance herunter.

Laden Sie Ihre Oracle Wallet auf Ihre RDS-for-Oracle-Instance herunter wie folgt:

1. Erfüllen Sie die Voraussetzungen für die Amazon S3-Integration in Oracle und fügen Sie die Option `S3_INTEGRATION` zu Ihrer Oracle-DB-Instance hinzu. Stellen Sie sicher, dass die IAM-Rolle für die Option Zugriff auf den Amazon S3-Bucket hat, den Sie verwenden.

   Weitere Informationen finden Sie unter Amazon S3-Integration.

2. Melden Sie sich als Hauptbenutzer bei Ihrer DB-Instance an und erstellen Sie dann ein Oracle-Verzeichnis für die Oracle Wallet.

   Im folgenden Beispiel wird ein Oracle-Verzeichnis mit dem Namen *WALLET_DIR* erstellt.

   ```
   EXEC rdsadmin.rdsadmin_util.create_directory('WALLET_DIR');
   ```

   Weitere Informationen finden Sie unter Erstellen und Löschen von Verzeichnissen im Hauptdatenspeicherbereich.

3. Laden Sie die Oracle Wallet zu Ihrem Amazon S3 Bucket hoch.

   Sie können jede unterstützte Upload-Technik verwenden.

4. Wenn Sie eine Oracle Wallet erneut hochladen, löschen Sie die vorhandene Wallet. Andernfalls überspringen Sie diesen Schritt und gehen Sie direkt zum nächsten.

   Im folgenden Beispiel wird die vorhandene Wallet mit dem Namen *cwallet.sso* entfernt.

   ```
   EXEC UTL_FILE.FREMOVE ('WALLET_DIR','cwallet.sso');
   ```

5. Laden Sie die Oracle Wallet aus Ihrem Amazon S3 Bucket zur Oracle-DB-Instance herunter.

   Im folgenden Beispiel wird die Wallet mit dem Namen *cwallet.sso* aus dem Amazon S3 Bucket mit dem Namen *my_s3_bucket* in das DB-Instance-Verzeichnis mit dem Namen *WALLET_DIR* heruntergeladen.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'cwallet.sso',
         p_directory_name =>  'WALLET_DIR')
      AS TASK_ID FROM DUAL;
   ```

6. (Optional) Laden Sie eine kennwortgeschützte Oracle Wallet herunter.

   Laden Sie diese Wallet nur herunter, wenn Sie für jede Verwendung der Wallet eine Passworteingabe fordern. Im folgenden Beispiel wird die passwortgeschützte Wallet *ewallet.p12* heruntergeladen.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'ewallet.p12',
   ```

```
      p_directory_name =>  'WALLET_DIR')
   AS TASK_ID FROM DUAL;
```

7. Überprüfen Sie den Status Ihrer DB-Anfrage.

   Ersetzen Sie die von den vorangegangenen Schritten zurückgegebene Aufgaben-ID für *dbtask-1234567890123-4567.log* im folgenden Beispiel.

   ```
   SELECT TEXT FROM
    TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-4567.log'));
   ```

8. Überprüfen Sie den Inhalt des Verzeichnisses, das Sie zum Speichern der Oracle Wallet verwenden.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
   ```

   Weitere Informationen finden Sie unter Auflisten von Dateien in einem DB-Instance-Verzeichnis.

Schritt 4: Erteilen Sie Benutzerberechtigungen für die Oracle Wallet

Sie können entweder einen neuen Datenbankbenutzer erstellen oder einen vorhandenen Benutzer konfigurieren. In beiden Fällen müssen Sie den Benutzer so konfigurieren, dass er auf die Oracle Wallet zugreift, um sichere Verbindungen und Clientauthentifizierung mit Zertifikaten zu erhalten.

Erteilen Sie Benutzerberechtigungen für die Oracle Wallet wie folgt:

1. Melden Sie sich als Hauptnutzer bei Ihrer RDS-for-Oracle-DB-Instance an.
2. Wenn Sie keinen bestehenden Datenbankbenutzer konfigurieren möchten, erstellen Sie einen neuen Benutzer. Andernfalls überspringen Sie diesen Schritt und gehen Sie direkt zum nächsten.

   Im folgenden Beispiel wird ein Datenbankbenutzer mit dem Namen *my-user* erstellt.

   ```
   CREATE USER my-user IDENTIFIED BY my-user-pwd;
   GRANT CONNECT TO my-user;
   ```

3. Erteilen Sie Ihrem Datenbankbenutzer die Berechtigung für das Verzeichnis, das Ihre Oracle Wallet enthält.

   Im folgenden Beispiel wird dem Benutzer *my-user* Lesezugriff für das Verzeichnis *WALLET_DIR* gewährt.

```
GRANT READ ON DIRECTORY WALLET_DIR TO my-user;
```

4. Erteilen Sie Ihrem Datenbankbenutzer die Berechtigung zur Nutzung des Pakets UTL_HTTP.

   Das folgende PL/SQL-Programm gewährt dem Benutzer *my-user* UTL_HTTP-Zugriff.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
/
```

5. Erteilen Sie Ihrem Datenbankbenutzer die Berechtigung zur Nutzung des Pakets UTL_FILE.

   Das folgende PL/SQL-Programm gewährt dem Benutzer *my-user* UTL_FILE-Zugriff.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_FILE', UPPER('my-user'));
  END;
/
```

Schritt 5: Konfigurieren des Zugriffs auf eine Website von Ihrer DB-Instance

In diesem Schritt konfigurieren Sie Ihren Oracle Datenbankbenutzer so, dass er sich über UTL_HTTP, die hochgeladene Oracle Wallet und das Clientzertifikat mit Ihrer Zielwebsite verbinden kann. Weitere Informationen finden Sie unter Configuring Access Control to an Oracle Wallet in der Dokumentation zur Oracle Database.

Konfigurieren des Zugriffs auf eine Website von Ihrer Oracle-DB-Instance

1. Melden Sie sich als Hauptnutzer bei Ihrer RDS-for-Oracle-DB-Instance an.
2. Erstellen Sie einen Host Access Control Entry (ACE) für Ihren Benutzer und die Zielwebsite auf einem sicheren Port.

   Im folgenden Beispiel wird *my-user* so konfiguriert, dass er Zugriff auf *secret.encrypted-website.com* auf dem sicheren Port 443 erhält.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host       => 'secret.encrypted-website.com',
```

```
    lower_port => 443,
    upper_port => 443,
    ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                              principal_name => 'my-user',
                              principal_type => xs_acl.ptype_db));
                           -- If the program unit results in PLS-00201, set
                           -- the principal_type parameter to 2 as follows:
                           -- principal_type => 2));
END;
/
```

> **Important**
>
> Die vorhergehende Programmeinheit kann zu folgendem Fehler führen: `PLS-00201: identifier 'XS_ACL' must be declared`. Wenn dieser Fehler zurückgegeben wird, ersetzen Sie die Zeile, die einen Wert zuweist, `principal_type` durch die folgende Zeile und führen Sie dann die Programmeinheit erneut aus:
>
> ```
> principal_type => 2));
> ```
>
> Weitere Informationen zu Konstanten im PL/SQL-Paket finden Sie `XS_ACL`unter Administratorhandbuch für Real Application Security und Entwicklerhandbuch in der Oracle-Database-Dokumentation.

Weitere Informationen finden Sie unter Configuring Access Control for External Network Services in der Dokumentation zur Oracle Database.

3. (Optional) Erstellen Sie einen ACE für Ihren Benutzer und Ihre Zielwebsite am Standardport.

   Möglicherweise müssen Sie den Standardport verwenden, wenn einige Webseiten vom Standard-Webserver-Port (80) anstelle des sicheren Ports (443) bereitgestellt werden.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host       => 'secret.encrypted-website.com',
    lower_port => 80,
    upper_port => 80,
    ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                              principal_name => 'my-user',
                              principal_type => xs_acl.ptype_db));
```

```
                            -- If the program unit results in PLS-00201, set
                            -- the principal_type parameter to 2 as follows:
                            -- principal_type => 2));
END;
/
```

4. Bestätigen Sie, dass die Zugriffssteuerungseinträge existieren.

```
SET LINESIZE 150
COLUMN HOST FORMAT A40
COLUMN ACL FORMAT A50

SELECT HOST, LOWER_PORT, UPPER_PORT, ACL
  FROM DBA_NETWORK_ACLS
ORDER BY HOST;
```

5. Erteilen Sie Ihrem Datenbankbenutzer die Berechtigung zur Nutzung des Pakets UTL_HTTP.

   Das folgende PL/SQL-Programm gewährt dem Benutzer *my-user* UTL_HTTP-Zugriff.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
/
```

6. Bestätigen Sie, dass verwandte Zugriffssteuerungslisten existieren.

```
SET LINESIZE 150
COLUMN ACL FORMAT A50
COLUMN PRINCIPAL FORMAT A20
COLUMN PRIVILEGE FORMAT A10

SELECT ACL, PRINCIPAL, PRIVILEGE, IS_GRANT,
       TO_CHAR(START_DATE, 'DD-MON-YYYY') AS START_DATE,
       TO_CHAR(END_DATE, 'DD-MON-YYYY') AS END_DATE
  FROM DBA_NETWORK_ACL_PRIVILEGES
ORDER BY ACL, PRINCIPAL, PRIVILEGE;
```

7. Erteilen Sie Ihrem Datenbankbenutzer die Berechtigung, Zertifikate für die Clientauthentifizierung und Ihr Oracle Wallet für Verbindungen zu verwenden.

> **Note**
>
> Wenn Sie keine Clientzertifikate für die Authentifizierung verwenden, können Sie diesen Schritt überspringen.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE(
    wallet_path => 'file:/' || l_wallet_path,
    ace         =>  xs$ace_type(privilege_list => xs
$name_list('use_client_certificates'),
                                principal_name => 'my-user',
                                principal_type => xs_acl.ptype_db));
END;
/
```

Schritt 6: Testen der Verbindungen Ihrer DB-Instance zu einer Website

In diesem Schritt konfigurieren Sie Ihren Datenbankbenutzer so, dass er sich über UTL_HTTP, die hochgeladene Oracle Wallet und das Clientzertifikat mit der Website verbinden kann.

Konfigurieren des Zugriffs auf eine Website von Ihrer Oracle-DB-Instance

1. Melden Sie sich bei Ihrer RDS-for-Oracle-DB-Instance als Datenbankbenutzer mit UTL_HTTP-Berechtigungen an.
2. Bestätigen Sie, dass eine Verbindung zu Ihrer Zielwebsite die Hostadresse auflösen kann.

   Im folgenden Beispiel wird die Host-Adresse von *secret.encrypted-website.com* abgerufen.

```
SELECT UTL_INADDR.GET_HOST_ADDRESS(host => 'secret.encrypted-website.com')
  FROM DUAL;
```

3. Testen Sie eine fehlgeschlagene Verbindung.

   Die folgende Abfrage schlägt fehl, da UTL_HTTP den Speicherort der Oracle Wallet mit den Zertifikaten benötigt.

```
SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

4. Testen Sie den Website-Zugriff durch die Verwendung von UTL_HTTP.SET_WALLET wählen Sie DUAL aus.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  UTL_HTTP.SET_WALLET('file:/' || l_wallet_path);
END;
/

SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

5. (Optional) Testen Sie den Website-Zugriff, indem Sie Ihre Abfrage in einer Variable speichern und EXECUTE IMMEDIATE verwenden.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
  v_webpage_sql VARCHAR2(1000);
  v_results     VARCHAR2(32767);
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  v_webpage_sql := 'SELECT UTL_HTTP.REQUEST(''secret.encrypted-website.com'', '''',
 ''file:/' ||l_wallet_path||''') FROM DUAL';
  DBMS_OUTPUT.PUT_LINE(v_webpage_sql);
  EXECUTE IMMEDIATE v_webpage_sql INTO v_results;
  DBMS_OUTPUT.PUT_LINE(v_results);
```

```
END;
/
```

6. (Optional) Suchen Sie den Dateisystemspeicherort Ihres Oracle-Wallet-Verzeichnisses.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

Verwenden Sie die Ausgabe des vorherigen Befehls, um eine HTTP-Anfrage zu stellen. Lautet das Verzeichnis *rdsdbdata/userdirs/01*, führen Sie beispielsweise die folgende Abfrage aus.

```
SELECT UTL_HTTP.REQUEST('https://secret.encrypted-website.com/', '',
 'file://rdsdbdata/userdirs/01')
FROM   DUAL;
```

Arbeiten mit CDBs in RDS für Oracle

In der Oracle-Multi-Tenant-Architektur kann eine Container-Datenbank (CDB) vom Kunden erstellte Pluggable Databases (PDBs) enthalten. Weitere Informationen zu CDBs finden Sie unter Introduction to the Multitenant Architecture in der Dokumentation zu Oracle Database.

Themen

- Überblick über RDS for Oracle CDBs
- Konfiguration einer CDB von RDS für Oracle
- Sichern und Wiederherstellen einer CDB
- Konvertieren einer Nicht-CDB von RDS für Oracle in eine CDB
- Konvertieren der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration
- Hinzufügen einer RDS-für-Oracle-Tenant-Datenbank zu Ihrer CDB-Instance
- Ändern einer Tenant-Datenbank von RDS für Oracle
- Löschen einer Tenant-Datenbank von RDS für Oracle aus Ihrer CDB
- Anzeigen von Details zu einer Tenant-Datenbank
- Aktualisieren Ihrer CDB

Überblick über RDS for Oracle CDBs

Sie können eine DB-Instance von RDS für Oracle als Container-Datenbank (CDB) erstellen, wenn Sie Oracle Database 19c oder höher ausführen. Ab Oracle Database 21c sind CDBs es alle Datenbanken. Eine CDB unterscheidet sich von einer Nicht-CDB dadurch, dass sie austauschbare Datenbanken (PDBs) enthalten kann, die in RDS für Oracle als Mandantendatenbanken bezeichnet werden. Eine PDB ist eine portable Sammlung von Schemas und Objekten, die einer Anwendung als separate Datenbank angezeigt wird.

Sie erstellen Ihre Initial Tenant Database (PDB), wenn Sie Ihre CDB-Instance erstellen. In RDS for Oracle interagiert Ihre Client-Anwendung mit einer PDB und nicht mit der CDB. Ihre Erfahrung mit einer PDB ist größtenteils identisch mit Ihrer Erfahrung mit einer Nicht-CDB.

Themen

- Multi-Tenant-Konfiguration der CDB-Architektur
- Single-Tenant-Konfiguration der CDB-Architektur
- Erstellungs- und Konvertierungsoptionen für CDBs

- Benutzerkonten und Berechtigungen in einer CDB
- Parametergruppenfamilien in einer CDB
- Einschränkungen von RDS für Oracle CDBs

Multi-Tenant-Konfiguration der CDB-Architektur

RDS for Oracle unterstützt die Mehrmandantenkonfiguration der Oracle-Multitenant-Architektur, auch CDB-Architektur genannt. In dieser Konfiguration kann Ihre RDS for Oracle CDB-Instance je nach Datenbankedition und erforderlichen Optionslizenzen 1—30 Tenant-Datenbanken enthalten. In der Oracle-Datenbank ist eine Tenant-Datenbank eine PDB. Ihre DB-Instance muss die Oracle-Datenbank Version 19.0.0.0.ru-2022-01.rur-2022.r1 oder höher verwenden.

> **Note**
>
> Die Amazon RDS-Konfiguration wird als „Multi-Tenant“ und nicht als „Multitenant“ bezeichnet, da es sich um eine Funktion von Amazon RDS handelt, nicht nur um die Oracle-DB-Engine. In ähnlicher Weise bezieht sich der RDS-Begriff „Mandant“ auf jeden Mandanten in einer RDS-Konfiguration, nicht nur auf Oracle. PDBs In der RDS-Dokumentation bezieht sich der Begriff „Oracle Multitenant“ ohne Bindestrich ausschließlich auf die CDB-Architektur der Oracle-Datenbank, die sowohl mit lokalen als auch mit RDS-Bereitstellungen kompatibel ist.

Sie können die folgenden Einstellungen konfigurieren:

- Name der Tenant-Datenbank
- Hauptbenutzername für die Tenant-Datenbank
- Master-Passwort für die Mandantendatenbank (optional in Secrets Manager integriert)
- Zeichensatz für die Tenant-Datenbank
- Nationaler Zeichensatz für die Tenant-Datenbank

Sie können einen Zeichensatz für die Tenant-Datenbank wählen, der sich vom Zeichensatz der CDB unterscheidet. Dies gilt auch für den nationalen Zeichensatz. Nachdem Sie Ihre erste Tenant-Datenbank erstellt haben, können Sie Tenant-Datenbanken mit RDS erstellen, ändern oder löschen APIs. Der CDB-Name ist standardmäßig `RDSCDB` und kann nicht geändert werden. Weitere Informationen erhalten Sie unter Einstellungen für DB-Instances und Ändern einer Tenant-Datenbank von RDS für Oracle.

Single-Tenant-Konfiguration der CDB-Architektur

RDS für Oracle unterstützt die Single-Tenant-Konfiguration, eine ältere Konfiguration der Oracle-Multitenant-Architektur. In dieser Konfiguration kann eine CDB-Instance von RDS für Oracle nur einen Tenant (PDB) enthalten. Sie können PDBs später keine weiteren erstellen.

Erstellungs- und Konvertierungsoptionen für CDBs

Oracle Database 21c unterstützt nur CDBs, während Oracle Database 19c beide unterstützt CDBs und nicht. CDBs Alle RDS-für-Oracle-CDB-Instances unterstützen sowohl Multi-Tenant- als auch Single-Tenant-Konfigurationen.

Optionen für Erstellung, Konvertierung und Upgrade für die Oracle-Datenbankarchitektur

Die folgende Tabelle zeigt die verschiedenen Architekturoptionen für die Erstellung und Aktualisierung von RDS-für-Oracle-Datenbanken.

Veröffentlichung	Optionen zur Datenbankerstellung	Optionen für die Architekturkonvertierung	Upgrade-Ziele für die Hauptversion
Oracle Database 21c	Nur CDB-Architektur	N/A	N/A
Oracle Database 19c	CDB- oder Nicht-CDB-Architektur	Nicht-CDB zu CDB-Architektur (April 2021 RU oder höher)	Oracle Database 21c CDB

Wie in der vorherigen Tabelle gezeigt, können Sie eine Nicht-CDB nicht direkt auf eine CDB in einer neuen Haupt-Datenbankversion aktualisieren. Sie können jedoch eine Nicht-CDB von Oracle Database 19c in eine CDB von Oracle Database 19c konvertieren und die CDB von Oracle Database 19c dann auf eine CDB von Oracle Database 21c aktualisieren. Weitere Informationen finden Sie unter Konvertieren einer Nicht-CDB von RDS für Oracle in eine CDB.

Konvertierungsoptionen für CDB-Architekturkonfigurationen

Die folgende Tabelle zeigt die verschiedenen Optionen für die Konvertierung der Architekturkonfiguration einer DB-Instance von RDS für Oracle an.

Aktuelle Architektur und Konfiguration	Umstellung auf die Single-Tenant-Konfiguration der CDB-Architektur	Umstellung auf die Multi-Tenant-Konfiguration der CDB-Architektur	Umstellung auf die Nicht-CDB-Architektur
Nicht-CDB	Unterstützt	Unterstützt*	N/A
CDB mit Single-Tenant-Konfiguration	N/A	Unterstützt	Nicht unterstützt
CDB mit Multi-Tenant-Konfiguration	Nicht unterstützt	N/A	Nicht unterstützt

* Eine Nicht-CDB kann nicht in einem einzigen Vorgang in eine Multi-Tenant-Konfiguration konvertiert werden. Wenn Sie eine Nicht-CDB in eine CDB konvertieren, befindet sich die CDB in der Single-Tenant-Konfiguration. Anschließend können Sie die Single-Tenant-Konfiguration in einem separaten Vorgang in die Multi-Tenant-Konfiguration konvertieren.

Benutzerkonten und Berechtigungen in einer CDB

In der Oracle-Multi-Tenant-Architektur sind alle Benutzerkonten entweder allgemeine Benutzer oder lokale Benutzer. Ein allgemeiner CDB-Benutzer ist ein Datenbankbenutzer, dessen einheitliche Identität und Kennwort im CDB-Root und in jeder bestehenden und zukünftigen PDB bekannt sind. Im Gegensatz dazu existiert ein lokaler Benutzer nur in einer einzigen PDB.

Der RDS-Hauptbenutzer ist ein lokales Benutzerkonto in der PDB, das Sie beim Erstellen Ihrer DB-Instance benennen. Wenn Sie neue Benutzerkonten erstellen, sind diese Benutzer auch lokale Benutzer, die sich in der PDB befinden. Sie können keine Benutzerkonten verwenden, um eine neue PDB zu erstellen PDBs oder den Status der vorhandenen PDB zu ändern.

Der `rdsadmin`-Benutzer ist ein allgemeines Benutzerkonto. Sie können Pakete von RDS für Oracle ausführen, die in diesem Konto vorhanden sind, aber Sie können sich nicht als `rdsadmin` anmelden. Weitere Informationen finden Sie unter Informationen zu allgemeinen Benutzern und lokalen Benutzern in der Oracle-Dokumentation.

Für Masterbenutzer sowohl in der Mehrmandantenkonfiguration als auch in der Einzelmandantenkonfiguration können Sie Anmeldeinformationen verwenden, die selbst verwaltet werden oder von verwaltet werden. AWS Secrets Manager In der Single-Tenant-Konfiguration verwenden Sie CLI-Befehle auf Instanzebene, z. B. `create-db-instance` für verwaltete Master-Passwörter. In der Multi-Tenant-Konfiguration verwenden Sie Befehle für Mandantendatenbanken, z. B. `create-tenant-database` für verwaltete Master-Passwörter. Weitere Informationen zur Secrets Manager Manager-Integration finden Sie unterVerwaltung des Master-Benutzerpassworts für eine RDS for Oracle-Mandantendatenbank mit Secrets Manager.

Parametergruppenfamilien in einer CDB

CDBs haben ihre eigenen Parametergruppenfamilien und Standardparameterwerte. Die CDB-Parametergruppenfamilien lauten wie folgt:

- oracle-ee-cdb-21
- oracle-se2-cdb-21
- oracle-ee-cdb-19
- oracle-se2-cdb-19

Einschränkungen von RDS für Oracle CDBs

RDS für Oracle unterstützt eine Teilmenge der Funktionen, die in einer lokalen CDB verfügbar sind.

CDB-Einschränkungen

Die folgenden Einschränkungen gelten für RDS for Oracle auf CDB-Ebene:

- Sie können keine Verbindung zu einer CDB herstellen. Sie stellen immer eine Verbindung zur Tenant-Datenbank (PDB) statt zur CDB her. Geben Sie den Endpunkt für die PDB genau wie für eine Nicht-CDB an. Der einzige Unterschied besteht darin, dass Sie pdb_name als Datenbanknamen angeben, wobei pdb_name der Name ist, den Sie für Ihre PDB gewählt haben.
- Sie können eine CDB in der Multi-Tenant-Konfiguration nicht in eine CDB in der Single-Tenant-Konvertierung konvertieren. Die Umstellung auf die Multi-Tenant-Konfiguration ist unidirektional und irreversibel.
- Sie können die Multi-Tenant-Konfiguration nicht aktivieren oder in diese konvertieren, wenn Ihre DB-Instance eine Oracle-Datenbankversion verwendet, die niedriger als 19.0.0.0.ru-2022-01.rur-2022.r1 ist.

- Sie können einen RDS für Oracle CDB mit ORDS 22 und höher nicht verwenden. Als Problemumgehung können Sie entweder eine niedrigere Version von ORDS oder eine Nicht-CDB von Oracle Database 19c verwenden.
- Sie können Oracle Data Guard nicht in der Multi-Tenant-Konfiguration verwenden, aber Sie können es in der Single-Tenant-Konfiguration verwenden.
- Sie können Database Activity Streams nicht in einer CDB verwenden.
- Sie können die Überwachung nicht von innen heraus `CDB$ROOT` aktivieren. Sie müssen die Überwachung in jeder PDB einzeln aktivieren.

Tenant-Datenbank- (PDB) Einschränkungen

Die folgenden Einschränkungen gelten für Tenant-Datenbanken in der Multi-Tenant-Konfiguration von RDS für Oracle:

- Sie können Tenant-Datenbankoperationen nicht auf das Wartungsfenster verschieben. Alle Änderungen werden sofort wirksam.
- Sie können einer CDB, die die Single-Tenant-Konfiguration verwendet, keine Tenant-Datenbank hinzufügen.
- Sie können nicht mehrere Tenant-Datenbanken in einem einzigen Vorgang hinzufügen oder ändern. Sie können sie nur einzeln hinzufügen oder ändern.
- Sie können eine Tenant-Datenbank nicht so ändern, dass sie den Namen `CDB$ROOT` oder `PDB$SEED` erhält.
- Sie können eine Tenant-Datenbank nicht löschen, wenn sie der einzige Tenant in der CDB ist.
- Nicht alle DB-Instance-Klassentypen verfügen über ausreichende Ressourcen, um mehrere Typen PDBs in einer RDS for Oracle CDB-Instance zu unterstützen. Eine höhere PDB-Anzahl wirkt sich auf die Leistung und Stabilität der kleineren Instance-Klassen aus und verlängert die Dauer der meisten Operationen auf Instance-Ebene, z. B. Datenbank-Upgrades.
- Sie können nicht mehrere verwenden AWS-Konten , um PDBs in derselben CDB zu erstellen. PDBsmuss demselben Konto gehören wie die DB-Instance, auf der PDBs sie gehostet werden.
- Alle PDBs in einer CDB verwenden denselben Endpunkt und Datenbank-Listener.
- Die folgenden Operationen werden auf PDB-Ebene nicht unterstützt, aber auf CDB-Ebene:
  - Sicherung und Wiederherstellung
  - Datenbank-Upgrades
  - Wartungsoperationen

- Die folgenden Features werden auf PDB-Ebene nicht unterstützt, aber auf CDB-Ebene:
  - Optionsgruppen (Optionen sind auf allen PDBs auf Ihrer CDB-Instance installiert)
  - Parametergruppen (alle Parameter werden von der Parametergruppe abgeleitet, die Ihrer CDB-Instance zugeordnet ist)
- Zu den Vorgängen auf PDB-Ebene, die in der lokalen CDB-Architektur, aber nicht in einer RDS-für-Oracle-CDB unterstützt werden, gehören die folgenden:

  > **Note**
  >
  > Die folgende Liste ist nicht vollständig.

  - Anwendung PDBs
  - Proxy PDBs
  - Starten und Anhalten einer PDB
  - Ausstecken und einstecken PDBs

    Wenn Sie Daten in oder aus Ihrer CDB verschieben möchten, verwenden Sie dieselben Methoden wie bei einer Nicht-CDB. Weitere Informationen zur Migration von Daten finden Sie unter Daten in Oracle auf Amazon importieren RDS.
  - Einstellungsoptionen auf PDB-Ebene

    Die PDB übernimmt die Optionseinstellungen von der CDB-Optionsgruppe. Weitere Informationen zu den Einstellungsoptionen finden Sie unter Parametergruppen für Amazon RDS. Bewährte Methoden finden Sie unter Arbeiten mit DB-Parametergruppen.
  - Konfiguration von Parametern in einer PDB

    Die PDB erbt die Parametereinstellungen von der CDB. Weitere Informationen zur Einstellungsoption finden Sie unter Hinzufügen von Optionen zu Oracle DB-Instances.
  - Konfiguration verschiedener Listener für dieselbe CDB PDBs
  - Oracle-Flashback-Funktionen

Konfiguration einer CDB von RDS für Oracle

Eine CDB wird ähnlich konfiguriert wie eine Nicht-CDB.

Themen

- Erstellen einer CDB-Instance von RDS für Oracle
- Herstellen einer Verbindung mit einer PDB in Ihrer CDB von RDS für Oracle

Erstellen einer CDB-Instance von RDS für Oracle

In RDS for Oracle ist das Erstellen einer CDB-Instance fast identisch mit dem Erstellen einer Nicht-CDB-Instance. Der Unterschied besteht darin, dass Sie bei der Erstellung Ihrer DB-Instance die Multi-Tenant-Architektur von Oracle und eine Architekturkonfiguration auswählen: Multi-Tenant oder Single-Tenant. Wenn Sie beim Erstellen einer CDB in der Multi-Tenant-Konfiguration Tags erstellen, gibt RDS die Tags an die ursprüngliche Tenant-Datenbank weiter. Verwenden Sie die, die oder die AWS Management Console RDS-API, um eine CDB zu erstellen. AWS CLI

Konsole

So erstellen Sie eine CDB-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der oberen rechten Ecke der Amazon RDS-Konsole die aus, AWS-Region in der Sie die CDB-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.
5. Wählen Sie unter Choose a database creation method (Wählen Sie eine Datenbankerstellungsmethode aus) Standard Create (Standarderstellung) aus.
6. Wählen Sie unter Engine options (Engine-Optionen) die Option Oracle.
7. Wählen Sie für Datenbankverwaltungstyp die Option Amazon RDS aus.
8. Wählen Sie unter Architektureinstellungen die Option Multi-Tenant-Architektur aus.
9. Führen Sie für die Architekturkonfiguration einen der folgenden Schritte aus:
   - Wählen Sie Multi-Tenant-Konfiguration aus und fahren Sie mit dem nächsten Schritt fort.
   - Wählen Sie Single-Tenant-Konfiguration aus und fahren Sie mit Schritt 11 fort.
10. (Multi-Tenant-Konfiguration) Nehmen Sie für die Tenant-Datenbankeinstellungen die folgenden Änderungen vor:

- Geben Sie unter Tenant-Datenbankname den Namen Ihrer anfänglichen PDB ein. Der PDB-Name muss sich vom CDB-Namen unterscheiden, der standardmäßig `RDSCDB` lautet.
- Geben Sie für den Hauptbenutzernamen der Tenant-Datenbank den Hauptbenutzernamen Ihrer PDB ein. Sie können den Hauptbenutzernamen der Tenant-Datenbank nicht verwenden, um sich beim CDB selbst anzumelden.
- Wählen Sie für die Verwaltung von Anmeldeinformationen eine der folgenden Optionen für die Verwaltung von Anmeldeinformationen:
  - Verwaltet in AWS Secrets Manager

    Das verwaltete Passwort gilt für die ursprüngliche Tenant-Datenbank und nicht für die Instanz. Wählen Sie unter Verschlüsselungsschlüssel auswählen entweder einen KMS-Schlüssel aus, den Secrets Manager erstellt, oder einen Schlüssel, den Sie selbst erstellt haben.

    > **Note**
    >
    > Wir empfehlen AWS Secrets Manager dies als sicherste Methode zur Verwaltung von Anmeldeinformationen. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.

  - Selbstverwaltet

    Um ein Passwort anzugeben, deaktivieren Sie das Kontrollkästchen Passwort automatisch generieren, falls es aktiviert ist. Geben Sie dasselbe Passwort in die Felder Master-Passwort und Master-Passwort bestätigen ein.
- Wählen Sie unter Tenant-Datenbank-Zeichensatz einen Zeichensatz für die PDB aus. Sie können einen Zeichensatz für die Tenant-Datenbank wählen, der sich von dem Zeichensatz der CDB unterscheidet.

  Der Standard-PDB-Zeichensatz ist AL32UTF8. Wenn Sie einen nicht standardmäßigen PDB-Zeichensatz wählen, ist die CDB-Erstellung möglicherweise langsamer.

> **Note**
>
> Sie können beim Erstellungsvorgang nicht mehrere Mandantendatenbanken angeben. Die CDB hat eine PDB, wenn sie erstellt wird. Sie können einer vorhandenen CDB in einem separaten Vorgang etwas hinzufügen PDBs .

11. (Einzelmandantenkonfiguration) Wählen Sie die gewünschten Einstellungen auf der Grundlage der Optionen aus, die unter: Einstellungen für DB-Instances

    - Öffnen Sie im Abschnitt Settings (Einstellungen) die Option Credential Settings (Einstellungen zu Anmeldeinformationen). Führen Sie dann die folgenden Schritte aus:

      i. Geben Sie unter Hauptbenutzername den Namen eines lokalen Benutzers in Ihrer PDB ein. Sie können den Hauptbenutzernamen nicht verwenden, um sich beim CDB-Root anzumelden.

      ii. Wählen Sie für die Verwaltung von Anmeldeinformationen eine der folgenden Optionen für die Verwaltung von Anmeldeinformationen:

      - Verwaltet in AWS Secrets Manager

        Wählen Sie unter Verschlüsselungsschlüssel auswählen entweder einen KMS-Schlüssel aus, den Secrets Manager erstellt, oder einen Schlüssel, den Sie selbst erstellt haben.

        > **Note**
        >
        > Wir empfehlen AWS Secrets Manager dies als sicherste Methode zur Verwaltung von Anmeldeinformationen. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.

      - Selbstverwaltet

        Um ein Passwort anzugeben, deaktivieren Sie das Kontrollkästchen Passwort automatisch generieren, falls es aktiviert ist. Geben Sie dasselbe Passwort in die Felder Master-Passwort und Master-Passwort bestätigen ein.

12. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die DB-Instance an. Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.
13. Wählen Sie Datenbank erstellen aus.

AWS CLI

Verwenden Sie den create-db-instanceBefehl mit den folgenden Parametern, um eine CDB in der Multi-Tenant-Konfiguration zu erstellen:

- `--db-instance-identifier`
- `--db-instance-class`
- `--engine { oracle-ee-cdb | oracle-se2-cdb }`
- `--master-username`
- `--master-user-password` oder `--manage-master-user-password`
- `--multi-tenant` (Geben Sie für die Single-Tenant-Konfiguration entweder nicht `multi-tenant` an oder geben Sie `--no-multi-tenant` an)
- `--allocated-storage`
- `--backup-retention-period`

Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Im folgenden Beispiel wird eine RDS für Oracle-DB-Instance erstellt, die *`my-cdb-inst`* in der Multi-Tenant-Konfiguration benannt ist. Wenn Sie `--no-multi-tenant` oder nicht `--multi-tenant` angeben, ist die Standard-CDB-Konfiguration Single-Tenant. Die Engine ist `oracle-ee-cdb`: ein Befehl, der `oracle-ee` und `--multi-tenant` angibt und mit einem Fehler fehlschlägt. Die ursprüngliche Tenant-Datenbank ist benannt*`mypdb`*.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-db-instance \
    --engine oracle-ee-cdb \
    --db-instance-identifier my-cdb-inst \
    --multi-tenant \
    --db-name mypdb \
```

```
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --master-username pdb_admin \
    --manage-master-user-password \
    --backup-retention-period 3
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-db-instance ^
    --engine oracle-ee-cdb ^
    --db-instance-identifier my-cdb-inst ^
    --multi-tenant ^
    --db-name mypdb ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --master-username pdb_admin ^
    --manage-master-user-password \ ^
    --backup-retention-period 3
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Die Ausgabe dieses Befehls sieht etwa wie folgt aus. Der Datenbankname, der Zeichensatz, der nationale Zeichensatz, der Hauptbenutzer und das geheime Hauptbenutzergeheimnis sind nicht in der Ausgabe enthalten. Sie können diese Informationen mit dem CLI-Befehl `describe-tenant-databases` anzeigen.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-cdb-inst",
        "DBInstanceClass": "db.t3.large",
        "MultiTenant": true,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-ABCDEFGJIJKLMNOPQRSTUVWXYZ",
        "DBInstanceStatus": "creating",
        "AllocatedStorage": 250,
        "PreferredBackupWindow": "04:59-05:29",
        "BackupRetentionPeriod": 3,
        "DBSecurityGroups": [],
```

```
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0a1bcd2e",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.oracle-ee-cdb-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-1234567a",
            "SubnetGroupStatus": "Complete",
            ...
```

RDS-API

Um eine DB-Instance mithilfe der Amazon RDS-API zu erstellen, rufen Sie den DBInstance Vorgang Create auf.

Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Herstellen einer Verbindung mit einer PDB in Ihrer CDB von RDS für Oracle

Sie können ein Dienstprogramm wie SQL*Plus verwenden, um eine Verbindung mit einer PDB herzustellen. Informationen zum Herunterladen von Oracle Instant Client, der eine eigenständige Version von SQL*Plus enthält, finden Sie unter Oracle Instant Client – Downloads.

Sie benötigen die folgenden Informationen, um SQL*Plus mit Ihrer PDB zu verbinden:

- PDB-Name
- Datenbank-Benutzername und -Passwort
- Endpunkt für Ihre DB-Instance
- Port-Nummer

Informationen zum Auffinden der vorherigen Informationen finden Sie unter Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle.

Example So stellen Sie mit SQL*Plus eine Verbindung mit Ihrer PDB her

Ersetzen Sie in den folgenden Beispielen Ihren Master-Benutzer durch*master_user_name*. Geben Sie außerdem den Endpunkt für Ihre DB-Instance und dann die Port-Nummer und die Oracle-SID an. Der SID-Wert ist der Name der PDB, die Sie beim Erstellen Ihrer DB-Instance angegeben haben, und nicht die DB-Instance-Kennung.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
sqlplus 'master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
sqlplus master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))
```

Die Ausgabe sollte in etwa wie folgt aussehen:

```
SQL*Plus: Release 19.0.0.0.0 Production on Mon Aug 21 09:42:20 2021
```

Nachdem Sie das Passwort für den Benutzer eingegeben haben, erscheint die SQL-Eingabeaufforderung.

```
SQL>
```

> **Note**
>
> Der Verbindungsstring in kürzerem Format (Easy connect oder EZCONNECT), zum Beispiel `sqlplus username/password@LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE:1521/database-identifier`, kann die maximale Zeichenanzahl überschreiten und sollte nicht für die Verbindung genutzt werden.

Sichern und Wiederherstellen einer CDB

Sie können Ihre CDB mithilfe von RDS-DB-Snapshots oder unter Verwendung von Recovery Manager (RMAN) sichern und wiederherstellen.

Sichern und Wiederherstellen einer CDB mithilfe von DB-Snapshots

DB-Snapshots funktionieren in CDB- und Nicht-CDB-Architekturen ähnlich. Die wichtigsten Unterschiede:

- Wenn Sie einen DB-Snapshot einer CDB wiederherstellen, können Sie die CDB nicht umbenennen. Der Name der CDB lautet `RDSCDB` und kann nicht geändert werden.
- Wenn Sie einen DB-Snapshot einer CDB wiederherstellen, können Sie PDBs nicht umbenennen. Sie können den PDB-Namen mit dem Befehl modify-tenant-database ändern.
- Um Tenant-Datenbanken in einem Snapshot zu finden, verwenden Sie den CLI-Befehl describe-db-snapshot-tenant-databases.
- Sie können nicht direkt mit den Tenant-Datenbanken in einem CDB-Snapshot interagieren, der die Multi-Tenant-Architekturkonfiguration verwendet. Wenn Sie den DB-Snapshot wiederherstellen, stellen Sie alle zugehörigen Tenant-Datenbanken wieder her.
- RDS für Oracle kopiert Tags in einer Tenant-Datenbank implizit in die Tenant-Datenbank in einem DB-Snapshot. Wenn Sie eine Tenant-Datenbank wiederherstellen, werden die Tags in der wiederhergestellten Datenbank angezeigt.
- Wenn Sie einen DB-Snapshot wiederherstellen und mithilfe des Parameters `--tags` neue Tags angeben, überschreiben die neuen Tags alle vorhandenen Tags.
- Wenn Sie einen DB-Snapshot einer CDB-Instance mit Tags verwenden und `--copy-tags-to-snapshot` angeben, kopiert RDS für Oracle Tags aus den Tenant-Datenbanken in die Tenant-Datenbanken im Snapshot.

Weitere Informationen finden Sie unter Überlegungen zu Oracle Database.

Sichern und Wiederherstellen einer CDB unter Verwendung von RMAN

Informationen zum Sichern und Wiederherstellen einer CDB oder einzelnen Tenant-Datenbank mithilfe von RMAN finden Sie unter Allgemeine RMAN Aufgaben für Oracle-DB-Instances ausführen.

Konvertieren einer Nicht-CDB von RDS für Oracle in eine CDB

Mit dem Befehl können Sie die Architektur einer Oracle-Datenbank von der Nicht-CDB-Architektur auf die Oracle-Multitenant-Architektur, auch CDB-Architektur genannt, ändern. `modify-db-instance` In den meisten Fällen ist diese Technik dem Erstellen einer neuen CDB und dem Importieren von Daten vorzuziehen. Der Konvertierungsvorgang verursacht Ausfallzeiten.

Wenn Sie Ihre Datenbank-Engine-Version aktualisieren, können Sie die Datenbankarchitektur nicht im selben Vorgang ändern. Wenn Sie eine Nicht-CDB von Oracle Database 19c auf eine CDB von Oracle Database 21c aktualisieren möchten, müssen Sie daher zuerst in einem Schritt die Nicht-CDB in eine CDB konvertieren und dann in einem separaten Schritt die 19c-CDB auf eine 21c-CDB aktualisieren.

Für die Konvertierung der Nicht-CDB gelten die folgenden Anforderungen:

- Sie müssen `oracle-ee-cdb` oder `oracle-se2-cdb` als DB-Engine-Typ angeben. Es werden ausschließlich diese Werte unterstützt.
- Ihre DB-Engine muss Oracle Database 19c mit einem Release-Update (RU) von April 2021 oder später verwenden.

Für den Vorgang gelten folgende Einschränkungen:

- Eine CDB können Sie nicht in eine Nicht-CDB konvertieren. Sie können nur eine Nicht-CDB in eine CDB konvertieren.
- Eine Nicht-CDB kann nicht in einem einzigen `modify-db-instance`-Aufruf in die Multi-Tenant-Konfiguration konvertiert werden. Wenn Sie eine Nicht-CDB in eine CDB konvertiert haben, weist die CDB die Single-Tenant-Konfiguration auf. Führen Sie `modify-db-instance` erneut aus, um die Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration zu konvertieren. Weitere Informationen finden Sie unter Konvertieren der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration.
- Sie können eine Primär- oder Replikatdatenbank, für die Oracle Data Guard aktiviert ist, nicht konvertieren. Um eine Nicht-CDB mit Lesereplikaten zu konvertieren, löschen Sie zunächst alle Lesereplikate.
- Sie können die DB-Engine-Version nicht aktualisieren und im gleichen Vorgang eine Nicht-CDB in eine CDB konvertieren.

Bevor Sie Ihre Nicht-CDB-Dateien konvertieren, sollten Sie Folgendes beachten:

- Die Überlegungen für Options- und Parametergruppen sind dieselben wie bei der Aktualisierung der DB-Engine. Weitere Informationen finden Sie unter Überlegungen zu Oracle-Datenbank-Upgrades.
- Sie können bestehende Nicht-CDB-Instances, die verwaltete Master-Passwörter verwenden, in einem einzigen Vorgang in Single-Tenant-Instances konvertieren. Die Single-Tenant-Instanzen erben die verwalteten Passwörter.
- Wenn auf Ihrer DB-Instance die `OEMAGENT` Option installiert ist, empfiehlt es sich, diese Option zu entfernen, bevor Sie Ihre Nicht-CDB konvertieren. Nachdem Ihre Nicht-CDB in eine CDB konvertiert wurde, installieren Sie die Option erneut. Weitere Informationen finden Sie unter Oracle Management Agent für Enterprise Cloud Control.
- Während des Konvertierungsvorgangs setzt RDS die Größe des Online-Redo-Logs auf die Standardgröße von 128 MB zurück.

Konsole

So konvertieren Sie eine Nicht-CDB in eine CDB

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie oben rechts in der Amazon-RDS-Konsole die AWS-Region aus, in der sich Ihre DB-Instance befindet.
3. Wählen Sie im Navigationsbereich Datenbanken und dann die Nicht-CDB-Instance aus, die Sie in eine CDB-Instance konvertieren möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie unter Architektureinstellungen die Option Oracle-Multitenant-Architektur aus. Nach der Konvertierung weist Ihre CDB die Single-Tenant-Konfiguration auf.
6. (Optional) Wählen Sie für DB-Parametergruppe eine neue Parametergruppe für Ihre CDB-Instance aus. Bei der Konvertierung einer DB-Instance gelten dieselben Überlegungen zu Parametergruppen wie beim Aktualisieren einer DB-Instance. Weitere Informationen finden Sie unter Überlegungen zu Parametergruppen.
7. (Optional) Wählen Sie unter Optionsgruppe eine neue Optionsgruppe für Ihre CDB-Instance aus. Bei der Konvertierung einer DB-Instance gelten dieselben Überlegungen zu Optionsgruppen wie beim Aktualisieren einer DB-Instance. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.

8. (Optional) Wählen Sie für die Verwaltung von Anmeldeinformationen die Option Verwaltet in AWS Secrets Manager oder Selbstverwaltet aus. Weitere Informationen finden Sie unter Verwaltung des Hauptbenutzerpassworts für eine DB-Instance mit Secrets Manager.
9. Nachdem Sie die gewünschten Änderungen vorgenommen haben, wählen Sie Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.
10. (Optional) Klicken Sie auf Apply immediately (Sofort anwenden), um die Änderungen direkt zu übernehmen. Die Auswahl dieser Option kann in einigen Fällen Ausfallzeiten verursachen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.
11. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie DB-Instance ändern aus.

    Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Um die Nicht-CDB auf Ihrer DB-Instance in eine CDB in der Single-Tenant-Konfiguration zu konvertieren, legen Sie im Befehl `--engine` auf `oracle-ee-cdb` oder fest. `oracle-se2-cdb` AWS CLI modify-db-instance Weitere Informationen finden Sie unter Einstellungen für DB-Instances.

Das folgende Beispiel konvertiert die benannte DB-Instance *my-non-cdb* und gibt eine benutzerdefinierte Optionsgruppe und Parametergruppe an. Der Befehl ermöglicht auch die Passwortverwaltung mit Secrets Manager.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-non-cdb \
    --engine oracle-ee-cdb \
    --option-group-name custom-option-group \
    --db-parameter-group-name custom-parameter-group \
    --manage-master-user-password
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-non-cdb ^
```

```
    --engine oracle-ee-cdb ^
    --option-group-name custom-option-group ^
    --db-parameter-group-name custom-parameter-group ^
    --manage-master-user-password
```

RDS-API

Um eine Nicht-CDB in eine CDB zu konvertieren, geben Sie `Engine` in der RDS-API-Operation Modify an. DBInstance

Konvertieren der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration

Sie können die Architektur einer CDB von RDS für Oracle von der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration ändern. Vor und nach der Konvertierung enthält Ihre CDB eine Single-Tenant-Datenbank (PDB). Tags für die DB-Instance werden an die ursprüngliche Tenant-Datenbank weitergegeben, die während der Konvertierung erstellt wurde.

Bevor Sie beginnen, stellen Sie sicher, dass Ihre IAM-Richtlinie über die Berechtigung zum Erstellen einer Tenant-Datenbank verfügt. Während der Konvertierung migriert RDS für Oracle die folgenden Metadaten in die neue Tenant-Datenbank:

- Den Masterbenutzernamen
- Das verwaltete Master-Passwort (wenn die Quell-CDB in Secrets Manager integriert ist)
- Den Datenbanknamen
- Den Zeichensatz
- Den nationalen Zeichensatz

Vor der Konvertierung können Sie die vorherigen Informationen mithilfe des `describe-db-instances` Befehls anzeigen. Nach der Konvertierung verwenden Sie zum Anzeigen der Informationen den Befehl `describe-tenant-database`.

Die Umstellung von Single-Tenant auf Multi-Tenant hat die folgenden Einschränkungen:

- Sie können später nicht mehr zur Einzelmandantenkonfiguration zurückkehren, nachdem Sie zur Mehrmandantenkonfiguration konvertiert haben. Die Konvertierung ist irreversibel.
- Sie können eine Primär- oder Replikatdatenbank, für die Oracle Data Guard aktiviert ist, nicht konvertieren.

- Sie können die DB-Engine-Version nicht aktualisieren und in demselben Vorgang zur Multi-Tenant-Konfiguration konvertieren.
- Sie können verwaltete Masterbenutzerkennwörter während der Konvertierung nicht aktivieren oder deaktivieren.

Konsole

So konvertieren Sie eine CDB, die die Single-Tenant-Konfiguration verwendet, in die Multi-Tenant-Konfiguration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie oben rechts in der Amazon-RDS-Konsole die AWS-Region aus, in der sich Ihre DB-Instance befindet.
3. Wählen Sie im Navigationsbereich Datenbanken und dann die Nicht-CDB-Instance aus, die Sie in eine CDB-Instance konvertieren möchten.
4. Wählen Sie Ändern aus.
5. Wählen Sie unter Architektureinstellungen die Option Oracle-Multitenant-Architektur aus.
6. Wählen Sie für Konfiguration der Architektur die Option Konfiguration für mehrere Mandanten aus.
7. (Optional) Wählen Sie für DB-Parametergruppe eine neue Parametergruppe für Ihre CDB-Instance aus. Bei der Konvertierung einer DB-Instance gelten dieselben Überlegungen zu Parametergruppen wie beim Aktualisieren einer DB-Instance.
8. (Optional) Wählen Sie unter Optionsgruppe eine neue Optionsgruppe für Ihre CDB-Instance aus. Bei der Konvertierung einer DB-Instance gelten dieselben Überlegungen zu Optionsgruppen wie beim Aktualisieren einer DB-Instance.
9. Nachdem Sie die gewünschten Änderungen vorgenommen haben, wählen Sie Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.
10. Wählen Sie Apply immediately (Sofort anwenden) aus. Diese Option ist erforderlich, wenn Sie zu einer Multi-Tenant-Konfiguration wechseln. Diese Option kann in einigen Fällen Ausfallzeiten verursachen.
11. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie DB-Instance ändern aus.

Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Um eine CDB, die die Single-Tenant-Konfiguration verwendet, in die Multi-Tenant-Konfiguration zu konvertieren, geben Sie `--multi-tenant` im Befehl Folgendes an. AWS CLI modify-db-instance

Im folgenden Beispiel wird die DB-Instance `my-st-cdb` von der Single-Tenant-Konfiguration in die Multi-Tenant-Konfiguration konvertiert. Die Option `--apply-immediately` ist erforderlich.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance --region us-east-1\
    --db-instance-identifier my-st-cdb \
    --multi-tenant \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance --region us-east-1 ^
    --db-instance-identifier my-st-cdb ^
    --multi-tenant ^
    --apply-immediately
```

Die Ausgabe sieht ungefähr wie folgt aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-st-cdb",
        "DBInstanceClass": "db.r5.large",
        "MultiTenant": false,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-AB1CDE2FGHIJK34LMNOPRLXTXU",
        "DBInstanceStatus": "modifying",
        "MasterUsername": "admin",
        "DBName": "ORCL",
```

```
        ...
        "EngineVersion": "19.0.0.0.ru-2022-01.rur-2022-01.r1",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "bring-your-own-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:oracle-ee-cdb-19",
                "Status": "in-sync"
            }
        ],
        ...
        "PendingModifiedValues": {
            "MultiTenant": "true"
        }
    }
}
```

Hinzufügen einer RDS-für-Oracle-Tenant-Datenbank zu Ihrer CDB-Instance

In der Multi-Tenant-Konfiguration von RDS für Oracle ist eine Tenant-Datenbank eine PDB. Überprüfen Sie, ob die folgenden Voraussetzungen erfüllt sind, um eine Tenant-Datenbank hinzuzufügen:

- In Ihrer CDB ist die Mehrmandantenkonfiguration aktiviert. Weitere Informationen finden Sie unter Multi-Tenant-Konfiguration der CDB-Architektur.
- Sie verfügen über die erforderlichen IAM-Berechtigungen zum Erstellen der Tenant-Datenbank.

Sie können eine Tenant-Datenbank mithilfe der AWS Management Console AWS CLI, der oder der RDS-API hinzufügen. Sie können nicht mehrere Tenant-Datenbanken in einem einzigen Vorgang hinzufügen: Sie müssen sie einzeln hinzufügen. Wenn für die CDB die Aufbewahrung von Backups aktiviert ist, sichert Amazon RDS die DB-Instance vor und nach dem Hinzufügen einer neuen Tenant-Datenbank.

Konsole

So fügen Sie Ihrer DB-Instance eine Tenant-Datenbank hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie rechts oben in der Amazon-RDS-Konsole die AWS-Region aus, in der Sie die Tenant-Datenbank erstellen möchten.
3. Wählen Sie im Navigationsbereich Datenbanken aus.
4. Wählen Sie die CDB-Instance aus, der Sie eine Tenant-Datenbank hinzufügen möchten. Ihre DB-Instance muss die Multi-Tenant-Konfiguration der CDB-Architektur verwenden.
5. Wählen Sie Aktionen und dann Tenant-Datenbank hinzufügen.
6. Führen Sie für Globale Datenbankeinstellungen die folgenden Schritte aus:

   - Geben Sie unter Name der Tenant-Datenbank den Namen Ihrer neuen PDB ein.
   - Geben Sie unter Hauptbenutzername der Tenant-Datenbank den Namen des Hauptbenutzers für Ihre PDB ein.
   - Wählen Sie eine der folgenden Optionen zur Verwaltung von Anmeldeinformationen:
     - Verwaltet in AWS Secrets Manager

       Wählen Sie unter Verschlüsselungsschlüssel auswählen entweder einen KMS-Schlüssel aus, den Secrets Manager erstellt, oder einen Schlüssel, den Sie selbst erstellt haben.

       > **Note**
       >
       > Wir empfehlen AWS Secrets Manager dies als sicherste Methode zur Verwaltung von Anmeldeinformationen. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.

     - Selbstverwaltet

       Um ein Passwort anzugeben, deaktivieren Sie das Kontrollkästchen Passwort automatisch generieren, falls es aktiviert ist. Geben Sie dasselbe Passwort in die Felder Master-Passwort und Master-Passwort bestätigen ein.

     - Geben Sie unter Zusätzliche Konfiguration den Namen Ihrer PDB als Ursprünglicher Datenbankname ein. Sie können die CDB nicht benennen, da diese den Standardnamen `RDSCDB` hat.
   - Wählen Sie unter Tenant-Datenbank-Zeichensatz einen Zeichensatz für die PDB aus. Der Standardwert ist AL32UTF8. Sie können einen PDB-Zeichensatz wählen, der sich vom Zeichensatz der CDB unterscheidet.

- Wählen Sie unter Länderspezifischer Tenant-Datenbank-Zeichensatz einen länderspezifischen Zeichensatz für die PDB aus. Der Standardwert ist AL32UTF8. Der nationale Zeichensatz spezifiziert die Kodierung nur für Spalten, die den NCHAR-Datentyp (NCHAR, NVARCHAR2 und NCLOB) verwenden, und wirkt sich nicht auf Datenbank-Metadaten aus.

Weitere Informationen zu diesen Einstellungen finden Sie unter Einstellungen für DB-Instances.

7. Wählen Sie Tenant hinzufügen.

AWS CLI

Um Ihrer CDB mit dem eine Tenant-Datenbank hinzuzufügen AWS CLI, verwenden Sie den Befehl create-tenant-databasemit den folgenden erforderlichen Parametern:

- `--db-instance-identifier`
- `--tenant-db-name`
- `--master-username`
- `--master-user-password`

Das folgende Beispiel erstellt eine Tenant-Datenbank mit dem Namen *mypdb2* der CDB-Instance RDS for Oracle mit dem Namen. *my-cdb-inst* Der PDB-Zeichensatz ist UTF-16.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name mypdb2 \
    --master-username mypdb2-admin \
    --master-user-password mypdb2-pwd \
    --character-set-name UTF-16
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name mypdb2 ^
    --master-username mypdb2-admin ^
```

```
    --master-user-password mypdb2-pwd ^
    --character-set-name UTF-16
```

Die Ausgabe sieht folgendermaßen oder ähnlich aus.

```
...}
    "TenantDatabase" :
         {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:mypdb2",
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "mypdb2",
            "Status" : "creating",
            "MasterUsername" : "mypdb2",
            "CharacterSetName" : "UTF-16",
            ...
        }
}...
```

Ändern einer Tenant-Datenbank von RDS für Oracle

Sie können nur den PDB-Namen und das Master-Benutzerpasswort einer Tenant-Datenbank in Ihrer CDB ändern. Beachten Sie die folgenden Anforderungen und Einschränkungen:

- Damit Sie die Einstellungen einer Tenant-Datenbank in Ihrer DB-Instance ändern können, muss die Tenant-Datenbank vorhanden sein.
- Sie können nicht mehrere Tenant-Datenbanken in einem einzigen Vorgang ändern. Sie können jeweils nur eine Tenant-Datenbank ändern.
- Sie können den Namen einer Tenant-Datenbank nicht in `CDB$ROOT` oder `PDB$SEED` ändern.

Sie können PDBs über die AWS Management Console, die AWS CLI oder die RDS-API ändern.

Konsole

So ändern Sie den PDB-Namen oder das Master-Passwort einer Tenant-Datenbank

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie rechts oben in der Amazon-RDS-Konsole die AWS-Region aus, in der Sie die Tenant-Datenbank erstellen möchten.
3. Wählen Sie im Navigationsbereich Datenbanken aus.
4. Wählen Sie die Tenant-Datenbank aus, deren Datenbanknamen oder Master-Benutzerpasswort Sie ändern möchten.
5. Wählen Sie Ändern aus.
6. Führen Sie für Tenant-Datenbankeinstellungen die folgenden Schritte aus:
   - Geben Sie unter Name der Tenant-Datenbank den neuen Namen Ihrer neuen PDB ein.
   - Geben Sie für Master-Passwort der Tenant-Datenbank ein neues Passwort ein.
7. Wählen Sie Tenant ändern aus.

AWS CLI

Um eine Tenant-Datenbank mit der zu ändernAWS CLI, rufen Sie den modify-tenant-database Befehl mit den folgenden Parametern auf:

- `--db-instance-identifier` *value (Wert)*
- `--tenant-db-name` *value*
- `[--new-tenant-db-name` *value*`]`
- `[--master-user-password` *value*`]`

Im folgenden Beispiel wird die Tenant-Datenbank `pdb1` in der DB-Instance `my-cdb-inst` in `pdb-hr` umbenannt.

Example

Für Linux, macOSoder Unix:

```
aws rds modify-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb1 \
    --new-tenant-db-name pdb-hr
```

Windows:

```
aws rds modify-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb1 ^
    --new-tenant-db-name pdb-hr
```

Die Ausgabe dieses Befehls sieht etwa wie folgt aus.

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac567",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb1",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb1",
        "Status" : "modifying",
        "MasterUsername" : "tenant-admin-user"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "pdb1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "pdb1-options",
                "Status": "in-sync"
            }
        ],
        "PendingModifiedValues": {
            "TenantDBName": "pdb-hr"
        }
    }
}
```

Löschen einer Tenant-Datenbank von RDS für Oracle aus Ihrer CDB

Sie können eine Tenant-Datenbank (PDB) mithilfe der AWS Management Console, der AWS CLI oder der RDS-API löschen. Berücksichtigen Sie die folgenden Voraussetzungen und Einschränkungen:

- Die Tenant-Datenbank und die DB-Instance müssen vorhanden sein.
- Für einen erfolgreichen Löschvorgang muss eine der folgenden Situationen zutreffen:
  - Die Tenant-Datenbank und die DB-Instance sind verfügbar.

    > **Note**
    >
    > Sie können einen endgültigen Snapshot erstellen, jedoch nur, wenn sich die Tenant-Datenbank und die DB-Instance vor der Ausführung des Befehls `delete-tenant-database` in einem verfügbaren Zustand befanden.

  - Die Tenant-Datenbank wird gerade erstellt.
  - Die DB-Instance ändert die Tenant-Datenbank.
- Sie können nicht mehrere Tenant-Datenbanken in einem einzigen Vorgang löschen.
- Sie können eine Tenant-Datenbank nicht löschen, wenn sie der einzige Tenant in der CDB ist.

Konsole

So löschen Sie eine Tenant-Datenbank

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die Tenant-Datenbank aus, die Sie löschen möchten.
3. Klicken Sie bei Actions auf Delete.
4. Um einen endgültigen DB-Snapshot für die DB-Instance zu erstellen, aktivieren Sie Create final snapshot? (Endgültigen Snapshot erstellen?).
5. Wenn Sie einen endgültigen Snapshot erstellen möchten, geben Sie den Final snapshot name (Name des endgültigen Snapshots) ein.
6. Geben Sie **`delete me`** in das Feld ein.
7. Wählen Sie Löschen aus.

AWS CLI

Um eine Tenant-Datenbank mit der zu löschenAWS CLI, rufen Sie den delete-tenant-database Befehl mit den folgenden Parametern auf:

- `--db-instance-identifier value`
- `--tenant-db-name value`
- `[--skip-final-snapshot | --no-skip-final-snapshot]`
- `[--final-snapshot-identifier value]`

Im folgenden Beispiel wird die Tenant-Datenbank namens *pdb-test* von der CDB namens gelöscht*my-cdb-inst*. Standardmäßig wird bei dem Vorgang ein endgültiger Snapshot erstellt.

Example

Für Linux, macOSoder Unix:

```
aws rds delete-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb-test \
    --final-snapshot-identifier final-snap-pdb-test
```

Windows:

```
aws rds delete-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb-test ^
    --final-snapshot-identifier final-snap-pdb-test
```

Die Ausgabe dieses Befehls sieht etwa wie folgt aus.

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac456",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb-
test",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb-test",
        "Status" : "deleting",
        "MasterUsername" : "pdb-test-admin"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
```

```
        "ParameterGroups": [
            {
                "ParameterGroupName": "tenant-1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "tenant-1-options",
                "Status": "in-sync"
            }
        ]
    }
}
```

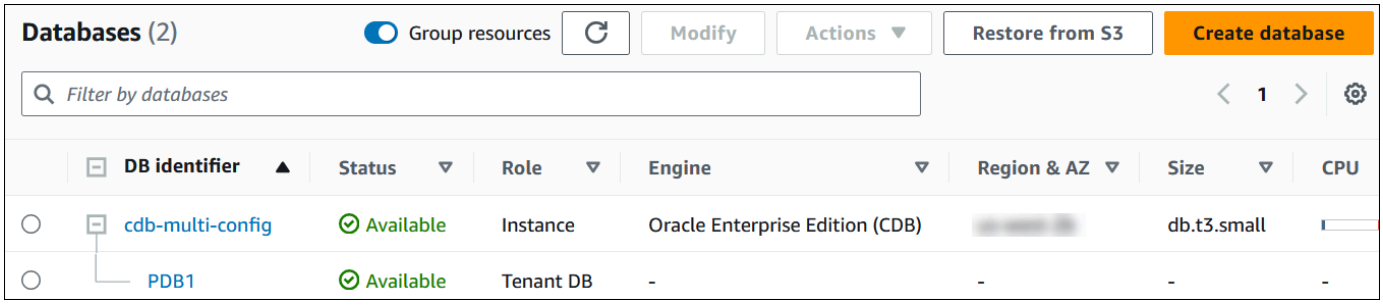
Anzeigen von Details zu einer Tenant-Datenbank

Sie können Details zu einer Tenant-Datenbank genauso anzeigen wie Details zu Non-CDB oder CDB.

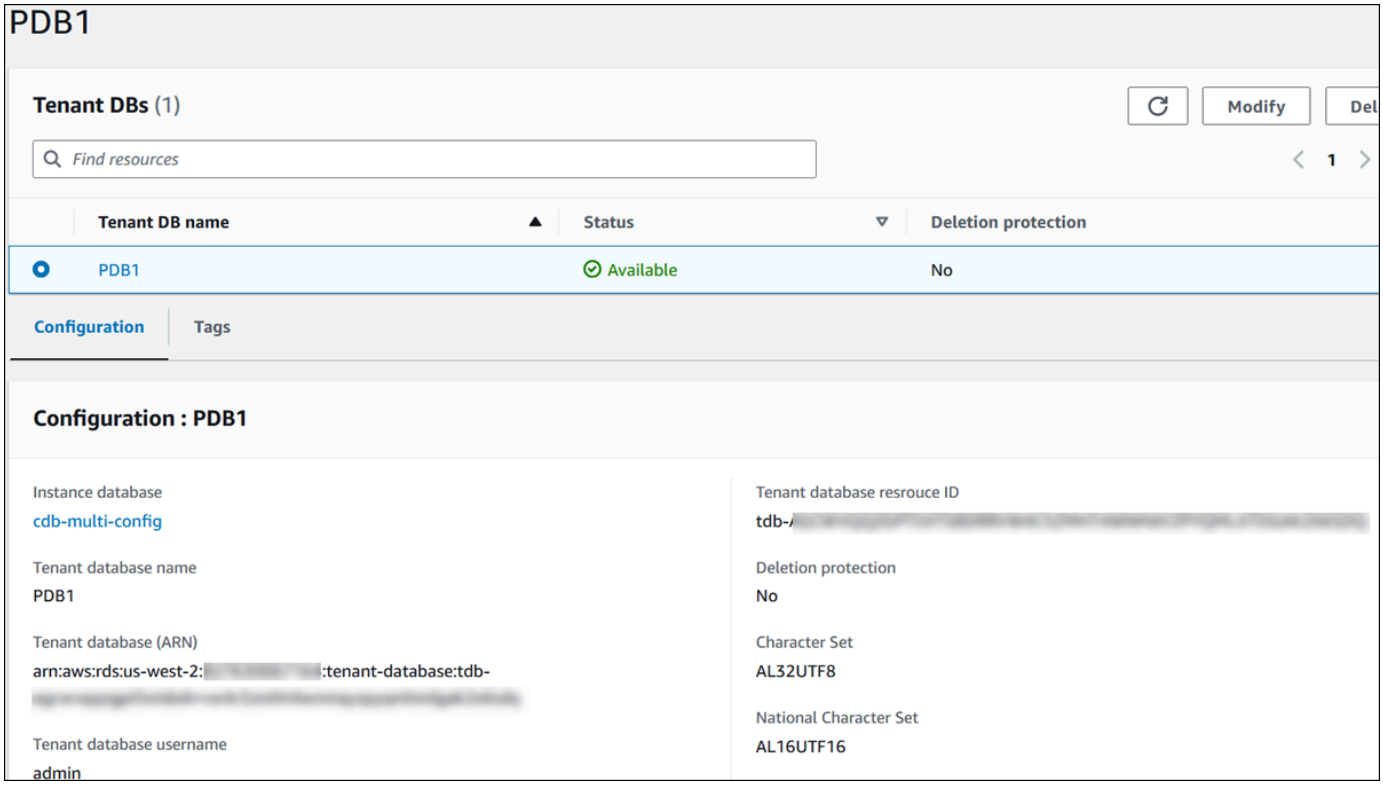
Konsole

So zeigen Sie Details zu einer Tenant-Datenbank an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie oben rechts in der Amazon-RDS-Konsole die AWS-Region aus, in der sich Ihre DB-Instance befindet.
3. Wählen Sie im Navigationsbereich Datenbanken aus.



   In der vorherigen Abbildung wird die Sole-Tenant-Datenbank (PDB) als untergeordnetes Element der DB-Instance angezeigt.
4. Wählen Sie den Namen einer Tenant-Datenbank aus.



AWS CLI

Um Details zu Ihren PDBs anzuzeigen, verwenden Sie den AWS CLI Befehl describe-tenant-databases.

Im folgenden Beispiel werden alle Tenant-Datenbanken in der angegebenen Region beschrieben.

Example

Für Linux, macOSoder Unix:

```
aws rds describe-tenant-databases --region us-east-1
```

Windows:

```
aws rds describe-tenant-databases --region us-east-1
```

Die Ausgabe dieses Befehls sieht etwa wie folgt aus.

```
    "TenantDatabases" : [
```

```
        {
           "DBInstanceIdentifier" : "my-cdb-inst",
           "TenantDBName" : "pdb-test",
           "Status" : "available",
           "MasterUsername" : "pdb-test-admin",
           "DbiResourceId" : "db-abc123",
           "TenantDatabaseResourceId" : "tdb-bac456",
           "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb-test",
           "CharacterSetName": "AL32UTF8",
           "NcharCharacterSetName": "AL16UTF16",
           "DeletionProtection": false,
           "PendingModifiedValues": {
                "MasterUserPassword": "****"
           },
           "TagList": []
        },
        {

           "DBInstanceIdentifier" : "my-cdb-inst2",
           "TenantDBName" : "pdb-dev",
           "Status" : "modifying",
           "MasterUsername" : "masterrdsuser"
           "DbiResourceId" : "db-xyz789",
           "TenantDatabaseResourceId" : "tdb-ghp890",
           "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst2:pdb-dev",
           "CharacterSetName": "AL32UTF8",
           "NcharCharacterSetName": "AL16UTF16",
           "DeletionProtection": false,
           "PendingModifiedValues": {
                "MasterUserPassword": "****"
           },
           "TagList": []
        },
        ... other truncated data
```

Im folgenden Beispiel werden die Tenant-Datenbanken der DB-Instance `my-cdb-inst` in der angegebenen Region beschrieben.

Example

Für Linux, macOSoder Unix:

```
aws rds describe-tenant-databases --region us-east-1 \
    --db-instance-identifier my-cdb-inst
```

Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
    --db-instance-identifier my-cdb-inst
```

Die Ausgabe dieses Befehls sieht etwa wie folgt aus.

```
{
    "TenantDatabase": {
        "TenantDatabaseCreateTime": "2023-10-19T23:55:30.046Z",
        "DBInstanceIdentifier": "my-cdb-inst",
        "TenantDBName": "pdb-hr",
        "Status": "creating",
        "MasterUsername": "tenant-admin-user",
        "DbiResourceId": "db-abc123",
        "TenantDatabaseResourceId": "tdb-bac567",
        "TenantDatabaseARN": "arn:aws:rds:us-west-2:579508833180:pdb-hr:tdb-
abcdefghi1jklmno2p3qrst4uvw5xy6zabc7defghi8jklmn90op",
        "CharacterSetName": "AL32UTF8",
        "NcharCharacterSetName": "AL16UTF16",
        "DeletionProtection": false,
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
        "TagList": [
            {
                "Key": "TEST",
                "Value": "testValue"
            }
        ]
    }
}
```

Im folgenden Beispiel wird die Tenant-Datenbank `pdb1` der DB-Instance `my-cdb-inst` in der Region USA Ost (Nord-Virginia) beschrieben.

Example

Für Linux, macOSoder Unix:

```
aws rds describe-tenant-databases --region us-east-1 \
--db-instance-identifier my-cdb-inst \
--tenant-db-name pdb1
```

Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
--db-instance-identifier my-cdb-inst ^
--tenant-db-name pdb1
```

Die Ausgabe dieses Befehls sieht etwa wie folgt aus.

```
{
    "TenantDatabases" : [
        {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb1"
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "pdb1",
            "Status" : "ACTIVE",
            "MasterUsername" : "masterawsuser"
            "Port" : "1234",
            "CharacterSetName": "UTF-8",
            "ParameterGroups": [
                {
                    "ParameterGroupName": "tenant-custom-pg",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            {
            "OptionGroupMemberships": [
                {
                    "OptionGroupName": "tenant-custom-og",
                    "Status": "in-sync"
                }
            ]
         }
    ]
}
```

Aktualisieren Ihrer CDB

Sie können eine CDB auf eine andere Oracle-Database-Version aktualisieren. Sie können beispielsweise eine CDB von Oracle Database 19c auf eine CDB von Oracle Database 21c aktualisieren. Sie können die Datenbankarchitektur während eines Upgrades nicht ändern. Daher können Sie eine Nicht-CDB nicht auf eine CDB oder eine CDB auf eine Nicht-CDB aktualisieren.

Das Verfahren für das Upgrade einer CDB auf eine CDB ist dasselbe wie für das Aktualisieren einer Nicht-CDB auf eine Nicht-CDB. Weitere Informationen finden Sie unter Aktualisieren der DB-Engine von RDS für Oracle.

Verwaltung Ihrer DB-Instance von RDS für Oracle

Nachfolgend finden Sie die üblichen Verwaltungsaufgaben, die Sie mit einer DB-Instance von RDS für Oracle ausführen. Einige Aufgaben sind für alle RDS-DB-Instances gleich. Andere Aufgaben sind spezifisch für RDS for Oracle.

Die folgenden Aufgaben sind allen RDS-Datenbanken gemeinsam, Oracle Database hat jedoch spezielle Aspekte. Sie stellen beispielsweise mithilfe der Oracle-Clients SQL*Plus und SQL Developer eine Verbindung zu einer Oracle-Datenbank her.

Aufgabenbereich	Relevante Dokumentation
Instance-Klassen, Speicher und PIOPS Wenn Sie eine Produktionsinstance erstellen, erfahren Sie, wie Instance-Klassen, Speichertypen und bereitgestellte IOPS in Amazon RDS funktionieren.	RDS für Oracle-DB-Instance-Klassen Amazon RDS-Speichertypen
Multi-AZ-Bereitstellungen Bei einer DB-Instance für die Produktion sollten Multi-AZ-Bereitstellungen eingesetzt werden. Multi-AZ-Bereitstellungen bieten eine erhöhte Verfügbarkeit, eine längere Lebensdauer von Daten sowie eine höhere Fehlertoleranz für DB-Instances.	Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS
Amazon VPC Wenn Ihr AWS-Konto über eine Standard-Virtual Private Cloud (VPC) verfügt, wird Ihre DB-Instance automatisch in dieser Standard-VPC erstellt. Wenn Ihr Konto über keine Standard-VPC verfügt und Sie die DB-Instance in einer VPC erstellen möchten, müssen Sie zunächst die VPC und Subnetz-Gruppen erstellen, bevor Sie die Instance erstellen können.	Arbeiten mit einer DB-Instance in einer VPC
Sicherheitsgruppen Standardmäßig verwenden DB-Instances eine Firewall, die den Zugriff verhindert. Stellen Sie sicher, dass Sie eine Sicherhei	Zugriffskontrolle mit Sicherheitsgruppen

Aufgabenbereich	Relevante Dokumentation
tsgruppe mit den korrekten IP-Adressen und Netzwerkkonfigurationen erstellen, um auf die DB-Instance zugreifen zu können.	
Parametergruppen Wenn Ihre DB-Instance spezifische Datenbankparameter erfordert, erstellen Sie vor der DB-Instance eine Parametergruppe.	Parametergruppen für Amazon RDS
Optionsgruppen Wenn Ihre DB-Instance bestimmte Datenbankoptionen erfordert, erstellen Sie eine Optionsgruppe, bevor Sie die DB-Instance erstellen.	Hinzufügen von Optionen zu Oracle DB-Instances
Herstellen einer Verbindung mit einer DB-Instance Nachdem Sie eine Sicherheitsgruppe erstellt und diese einer DB-Instance zugeordnet haben, können Sie mithilfe einer beliebigen Standard-SQL-Client-Anwendung, zum Beispiel Oracle SQL*Plus, eine Verbindung mit dieser DB-Instance herstellen.	Herstellen der Verbindung zu Ihrer Oracle-DB-Instance
Backup und Wiederherstellung Sie können Ihre DB-Instance so konfigurieren, dass sie automatische Backups oder manuelle Snapshots vornimmt. Aus diesen Backups oder Snapshots können Sie dann Instances wiederherstellen.	Daten sichern, wiederherstellen und exportieren
Überwachung Sie können eine Oracle-DB-Instance mithilfe von CloudWatch Amazon RDS-Metriken, -Ereignissen und verbesserter Überwachung überwachen.	Metriken in der RDS Amazon-Konsole anzeigen RDSAmazon-Ereignisse anzeigen

Aufgabenbereich	Relevante Dokumentation
Protokolldateien Sie können auf die Protokolldateien für Ihre Orale-DB-Instance zugreifen.	Überwachung von Amazon RDS Amazon

Nachfolgend finden Sie eine Beschreibung für Amazon-RDS-spezifische Implementierungen von häufigen DBA-Aufgaben für RDS Oracle. Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Shell-Zugriff auf DB-Instances. Außerdem schränkt RDS den Zugriff auf bestimmte Systemprozeduren und -tabellen ein, die erweiterte Berechtigungen erfordern. Bei vielen Aufgaben führen Sie das `rdsadmin`-Paket aus, das ist ein Amazon-RDS-spezifisches Tool ist, mit dem Sie Ihre Datenbank verwalten können.

Nachfolgend sehen Sie häufige DBA-Aufgaben für DB-Instances, in denen Oracle ausgeführt wird:

- Systemaufgaben

Trennen einer Sitzung	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.disconnect` Oracle-Methode: `alter system disconnect session`
Beenden einer Sitzung	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.kill` Oracle-Methode: `alter system kill session`
Eine SQL Aussage in einer Sitzung stornieren	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.cancel` Oracle-Methode: `alter system cancel sql`
Aktivieren und Deaktivieren von beschränkten Sitzungen	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.restricted_session` Oracle-Methode: `alter system enable restricted session`
Bereinigen des freigegebenen Pools	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.flush_shared_pool`

	Oracle-Methode: `alter system flush shared_pool`
Bereinigen des Buffer-Cache	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.flush_buffer_cache` Oracle-Methode: `alter system flush buffer_cache`
Gewährung SELECT von EXECUTE Rechten für SYS Objekte	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.grant_sys_object` Oracle-Methode: `grant`
Widerrufen von SELECT EXECUTE Rechten für Objekte SYS	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.revoke_sys_object` Oracle-Methode: `revoke`
Verwaltung von RDS_X$-Ansichten für Oracle-DB-Instances	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.create_sys_x$_view` Oracle-Methode: `CREATE VIEW`
Erteilen von Berechtigungen an Nicht-Hauptbenutzer	Amazon RDS-Methode: `grant`
Erstellen von benutzerdefinierten Funktionen für das Überprüfen von Passwörtern	Amazon RDS-Methode: `rdsadmin.rdsadmin_password_verify.create_verify_function` Amazon RDS-Methode: `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`
Einen benutzerdefinierten DNS Server einrichten	—

Auflisten zulässiger Systemdiagnoseereignisse	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.list_allowed_system_events` Oracle-Methode: &endash;
Festlegen von Systemdiagnoseereignissen	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.set_allowed_system_events` Oracle-Methode: `ALTER SYSTEM SET EVENTS 'set_event_clause'`
Auflisten der festgesetzten Systemdiagnoseereignisse	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.list_set_system_events` Oracle-Methode: `ALTER SESSION SET EVENTS 'IMMEDIATE EVENTDUMP(SYSTEM)'`
Aufheben von Systemdiagnoseereignissen	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.unset_system_event` Oracle-Methode: `ALTER SYSTEM SET EVENTS 'unset_event_clause'`

- Datenbankaufgaben

Ändern des globalen Namens einer Datenbank	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.rename_global_name` Oracle-Methode: `alter database rename`
Erstellen und Größenfestlegung von Tabellenräumen	Amazon RDS-Methode: `create tablespace` Oracle-Methode: `alter database`

Einrichten des Standard-Tabellenraums	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.alter_default_tablespace` Oracle-Methode: `alter database default tablespace`
Einrichten des temporären Standard-Tabellenraums	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.alter_default_temp_tablespace` Oracle-Methode: `alter database default temporary tablespace`
Erstellen eines temporären Tabellenraums im Instance-Speicher	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace` Oracle-Methode: `create temporary tablespace`
Überprüfung einer Datenbank	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.checkpoint` Oracle-Methode: `alter system checkpoint`
Einstellen der verteilten Wiederherstellung	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.enable_distr_recovery` Oracle-Methode: `alter system enable distributed recovery`
Einstellen der Datenbank-Zeitzone	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.alter_db_time_zone` Oracle-Methode: `alter database set time_zone`
Arbeiten mit externen Oracle-Tabellen	—
Generieren von Leistungsberichten mit Automatic Workload Repository (AWR)	Amazon RDS-Methode: `rdsadmin.rdsadmin_diagnostic_util` -Prozeduren Oracle-Methode: `dbms_workload_repository` -Paket

Anpassen von Datenbank-Links für die Verwendung mit DB-Instances in einem VPC	—
Einrichten der Standardversion für eine DB-Instance	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.alter_default_edition` Oracle-Methode: `alter database default edition`
Aktivieren der Überwachung für dieSYS. AUD$-Tabelle	Amazon RDS-Methode: `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` Oracle-Methode: `audit`
Deaktivierung der Überwachung für dieSYS. AUD$-Tabelle	Amazon RDS-Methode: `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` Oracle-Methode: `noaudit`
Bereinigen unterbrochener Online-Index-Builds	Amazon RDS-Methode: `rdsadmin.rdsadmin_dbms_repair.online_index_clean` Oracle-Methode: &enda: `dbms_repair.online_index_clean`
Überspringen von beschädigten Blöcken	Amazon RDS-Methode: Mehrere `rdsadmin.rdsadmin_dbms_repair` -Verfahren Oracle-Methode: `dbms_repair` -Paket
Ändern der Größe von Tabellenbereichen, Datendateien und temporären Dateien	Amazon-RDS-Methode: Prozeduren `rdsadmin.rdsadmin_util.resize_temp_tablespace` , `rdsadmin.rdsadmin_util.resize_tempfile` oder `rdsadmin.rdsadmin_util.autoextend_tempfile` Prozedur `rdsadmin.rdsadmin_util.resize_datafile` oder `rdsadmin.rdsadmin_util.autoextend_datafile` Oracle-Methode: &endash;

Bereinigen des Papierkorbs	Amazon RDS-Methode: `EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin` Oracle-Methode: `purge dba_recyclebin`
Festlegen der angezeigten Standardwerte für vollständige Redaktion	Amazon RDS-Methode: `EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` Oracle-Methode: `exec dbms_redact.UPDATE_FULL_REDACTION_VALUES`

- Protokollaufgaben

Einstellen der erzwungenen Protokollierung	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.force_logging` Oracle-Methode: `alter database force logging`
Einstellen der ergänzenden Protokollierung	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.alter_supplemental_logging` Oracle-Methode: `alter database add supplemental log`
Wechseln zwischen Online-Protokolldateien	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.switch_logfile` Oracle-Methode: `alter system switch logfile`

Hinzufügen von Online-Redo-Log-Dateien	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.add_logfile`
Löschen von Online-Redo-Log-Dateien	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.drop_logfile`
Anpassen der Größe von Online-Redo-Log-Dateien	—
Beibehaltung von archivierten Redo-Log-Dateien	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.set_configuration`
Herunterladen von archivierten Redo-Protokolle aus Amazon S3	Amazon RDS-Methode: `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum` Amazon RDS-Methode: `rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range`
Zugriff auf Online- oder archivierte Redo-Protokolle	Amazon RDS-Methode: `rdsadmin.rdsadmin_master_util.create_archivelog_dir` Amazon RDS-Methode: `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`

- RMANAufgaben

Datenbankdateien RDS für Oracle validieren	Amazon RDS-Methode: `rdsadmin_rman_util. procedure` Oracle-Methode: `RMAN VALIDATE`
Aktivieren und Deaktivieren der Nachverfolgung von Blockänderungen	Amazon RDS-Methode: `rdsadmin_rman_util. procedure` Oracle-Methode: `ALTER DATABASE`
Gegenprüfen archivierter Redo-Logs	Amazon RDS-Methode: `rdsadmin_rman_util.crosscheck_archivelog` Oracle-Methode: `RMAN BACKUP`
Archivierte Redo-Log-Dateien sichern	Amazon RDS-Methode: `rdsadmin_rman_util. procedure` Oracle-Methode: `RMAN BACKUP`
Durchführen einer vollständigen Datenbanksicherung	Amazon RDS-Methode: `rdsadmin_rman_util.backup_database_full` Oracle-Methode: `RMAN BACKUP`

Durchführen einer inkrementellen Datenbanksicherung	Amazon RDS-Methode: `rdsadmin_rman_util.backup_database_incremental` Oracle-Methode: `RMAN BACKUP`
Sichern eines Tablespace	Amazon RDS-Methode: `rdsadmin_rman_util.backup_database_tablespace` Oracle-Methode: `RMAN BACKUP`

- Oracle-Scheduler-Aufgaben

DBMS_ SCHEDULER Jobs ändern	Amazon RDS-Methode: `dbms_scheduler.set_attribute` Oracle-Methode: `dbms_scheduler.set_attribute`
AutoTask Wartungsfenster ändern	Amazon RDS-Methode: `dbms_scheduler.set_attribute` Oracle-Methode: `dbms_scheduler.set_attribute`

Festlegen der Zeitzone für Oracle Scheduler-Aufgaben	Amazon RDS-Methode: `dbms_scheduler.set_scheduler_attribute` Oracle-Methode: `dbms_scheduler.set_scheduler_attribute`
Ausschalten von Oracle Scheduler-Jobs, deren Eigentümer SYS	Amazon RDS-Methode: `rdsadmin.rdsadmin_dbms_scheduler.disable` Oracle-Methode: `dbms_scheduler.disable`
Oracle Scheduler-Jobs einschalten, deren Eigentümer SYS	Amazon RDS-Methode: `rdsadmin.rdsadmin_dbms_scheduler.enable` Oracle-Methode: `dbms_scheduler.enable`
Ändern des Oracle Scheduler-Wiederholungsintervalls für Jobs des Typs CALENDAR	Amazon RDS-Methode: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Oracle-Methode: `dbms_scheduler.set_attribute`

Ändern des Oracle Scheduler-Wiederholungsintervalls für Jobs des Typs NAMED	Amazon RDS-Methode: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Oracle-Methode: `dbms_scheduler.set_attribute`
Deaktivieren von Autocommit für die Erstellung von Oracle-Scheduler-Aufgaben	Amazon RDS-Methode: `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` Oracle-Methode: `dbms_isched.set_no_commit_flag`

- Probleme diagnostizieren

Auflistung von Vorfällen	Amazon RDS-Methode: `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` Oracle-Methode: ADRCI-Befehl `show incident`
Probleme mit der Auflistung	Amazon RDS-Methode: `rdsadmin.rdsadmin_adrci_util.list_adrci_problem` Oracle-Methode: ADRCI-Befehl `show problem`

Erstellen von Vorfallpaketen	Amazon RDS-Methode: `rdsadmin.rdsadmin_adrci_util.create_adrci_package` Oracle-Methode: ADRCI-Befehl `ips create package`
Anzeigen von Trace-Dateien	Amazon RDS-Methode: `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` Oracle-Methode: ADRCI-Befehl `show tracefile`

- Weitere Aufgaben

Erstellen und Löschen von Verzeichnissen im Hauptdatenspeicherbereich	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.create_directory` Oracle-Methode: `CREATE DIRECTORY` Amazon RDS-Methode: `rdsadmin.rdsadmin_util.drop_directory` Oracle-Methode: `DROP DIRECTORY`

Auflisten von Dateien in einem DB-Instance-Verzeichnis	Amazon RDS-Methode: `rdsadmin.rds_file_util.listdir` Oracle-Methode: &endash;
Lesen von Dateien in einem DB-Instance-Verzeichnis	Amazon RDS-Methode: `rdsadmin.rds_file_util.read_text_file` Oracle-Methode: &endash;
Zugreifen auf Opatch-Dateien	Amazon RDS-Methode: `rdsadmin.rds_file_util.read_text_file` oder `rdsadmin.tracefile_listing` Oracle-Methode: `opatch`
Festlegen von Parametern für Beraterаufgaben	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.advisor_task_set_parameter` Oracle-Methode: Verschiedene gespeicherte Prozeduren im Paket
Deaktivieren von AUTO_STATS_ADVISOR_TASK	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.advisor_task_drop` Oracle-Methode: &endash;

Erneutes Aktivieren von AUTO_STATS_ADVISOR_TASK	Amazon RDS-Methode: `rdsadmin.rdsadmin_util.dbms_stats_init` Oracle-Methode:

Sie können auch Amazon RDS-Verfahren für die Amazon S3-Integration mit Oracle und für die Ausführung von OEM Management-Agent-Datenbankaufgaben verwenden. Weitere Informationen finden Sie unter Amazon S3-Integration und Ausführen von Datenbankaufgaben mit dem Management Agent.

Durchführen allgemeiner Systemaufgaben für Oracle DB-Instances

Im Folgenden erfahren Sie, wie Sie bestimmte allgemeine DBA Aufgaben im Zusammenhang mit dem System auf Ihren RDS Amazon-DB-Instances ausführen, auf denen Oracle ausgeführt wird. Um ein Managed-Service-Erlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances und schränkt den Zugriff auf bestimmte Systemprozeduren und Tabellen ein, für die erweiterte Rechte erforderlich sind.

Themen

- Trennen einer Sitzung
- Beenden einer Sitzung
- Eine SQL Aussage in einer Sitzung stornieren
- Aktivieren und Deaktivieren von beschränkten Sitzungen
- Bereinigen des freigegebenen Pools
- Gewährung SELECT von EXECUTE Rechten für SYS Objekte
- Widerrufen von SELECT EXECUTE Rechten für Objekte SYS
- Verwaltung von RDS_X$-Ansichten für Oracle-DB-Instances
- Erteilen von Berechtigungen an Nicht-Hauptbenutzer
- Erstellen von benutzerdefinierten Funktionen für das Überprüfen von Passwörtern
- Einen benutzerdefinierten DNS Server einrichten
- Festlegen und Aufheben von Systemdiagnoseereignissen

Trennen einer Sitzung

Verwenden Sie das RDS Amazon-Verfahren, um die aktuelle Sitzung zu trennen, indem Sie den dedizierten Serverprozess beenden`rdsadmin.rdsadmin_util.disconnect`. Die Prozedur `disconnect` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`sid`	Zahl	—	Ja	Die Sitzungskennung
`serial`	Zahl	—	Ja	Die Seriennummer der Sitzung
`method`	varchar	'IMMEDIATE'	Nein	Gültige Werte sind `'IMMEDIATE'` oder `'POST_TRANSACTION'` .

Im folgenden Beispiel wird die Verbindung mit einer Sitzung getrennt.

```
begin
    rdsadmin.rdsadmin_util.disconnect(
        sid    => sid,
        serial => serial_number);
end;
/
```

Stellen Sie an die Ansicht `V$SESSION` eine Abfrage, um die Sitzungskennung und die -seriennummer zu erhalten. Im folgenden Beispiel werden alle Sitzungen für den Benutzer abgerufen `AWSUSER`.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

Die Datenbank muss offen sein, um diese Methode zu verwenden. Weitere Informationen zum Trennen einer Sitzung finden Sie ALTERSYSTEMin der Oracle-Dokumentation.

Beenden einer Sitzung

Verwenden Sie das RDS Amazon-Verfahren, um eine Sitzung zu beenden`rdsadmin.rdsadmin_util.kill`. Die Prozedur `kill` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`sid`	Zahl	—	Ja	Die Sitzungskennung
`serial`	Zahl	—	Ja	Die Seriennummer der Sitzung
`method`	varchar	Null	Nein	Gültige Werte sind `'IMMEDIATE'` oder `'PROCESS'`. Wenn Sie `IMMEDIATE` angeben, hat dies den gleichen Effekt wie das Ausführen der folgenden Anweisung: `ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE` Wenn Sie `PROCESS` angeben, beenden Sie die mit einer Sitzung verbundenen Prozesse. Geben Sie `PROCESS` nur an, wenn Sie die Sitzung mit `IMMEDIATE` nicht beenden konnten.

Stellen Sie an die Ansicht `V$SESSION` eine Abfrage, um die Sitzungskennung und die -seriennummer zu erhalten. Im folgenden Beispiel werden alle Sitzungen für den Benutzer abgerufen *AWSUSER*.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

Im folgenden Beispiel wird eine Sitzung beendet.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'IMMEDIATE');
END;
/
```

Im folgenden Beispiel werden die mit einer Sitzung verbundenen Prozesse beendet.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'PROCESS');
END;
/
```

Eine SQL Aussage in einer Sitzung stornieren

Verwenden Sie das RDS Amazon-Verfahren, um einen SQL Kontoauszug in einer Sitzung zu stornieren`rdsadmin.rdsadmin_util.cancel`.

> **Note**
>
> Dieses Verfahren wird für Oracle Database 19c (19.0.0) und alle höheren Haupt- und Nebenversionen von RDS für Oracle unterstützt.

Die Prozedur `cancel` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`sid`	Zahl	—	Ja	Die Sitzungskennung

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`serial`	Zahl	—	Ja	Die Seriennummer der Sitzung
`sql_id`	varchar2	Null	Nein	Der SQL Bezeichner der SQL Anweisung.

Im folgenden Beispiel wird eine SQL Anweisung in einer Sitzung storniert.

```
begin
    rdsadmin.rdsadmin_util.cancel(
        sid    => sid,
        serial => serial_number,
        sql_id => sql_id);
end;
/
```

Um die Sitzungs-ID, die Sitzungs-Seriennummer und die SQL ID einer SQL Anweisung abzurufen, fragen `V$SESSION` Sie die Ansicht ab. Im folgenden Beispiel werden alle Sitzungen und SQL Identifikatoren für den Benutzer `AWSUSER` abgerufen.

```
select SID, SERIAL#, SQL_ID, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

Aktivieren und Deaktivieren von beschränkten Sitzungen

Verwenden Sie das RDS Amazon-Verfahren, um eingeschränkte Sitzungen zu aktivieren und zu deaktivieren`rdsadmin.rdsadmin_util.restricted_session`. Die Prozedur `restricted_session` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Ja	Beschreibung
`p_enable`	Boolean	true	Nein	Setzen Sie diesen Parameter auf `true`, um beschränkte Sitzungen zu aktivieren, oder auf `false`, um beschränkte

Parametername	Datentyp	Standard	Ja	Beschreibung
				Sitzungen zu deaktivieren.

Im folgenden Beispiel wird gezeigt, wie beschränkte Sitzungen aktiviert und deaktiviert werden können.

```
/* Verify that the database is currently unrestricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED

/* Enable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => true);


/* Verify that the database is now restricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
----------
RESTRICTED


/* Disable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => false);


/* Verify that the database is now unrestricted again. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
```

```
ALLOWED
```

Bereinigen des freigegebenen Pools

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.flush_shared_pool`, um den gemeinsam genutzten Pool zu leeren. Die Prozedur `flush_shared_pool` hat keine Parameter.

Im folgenden Beispiel wird der geteilte Pool bereinigt.

```
EXEC rdsadmin.rdsadmin_util.flush_shared_pool;
```

Bereinigen des Buffer-Cache

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.flush_buffer_cache`, um den Puffer-Cache zu leeren. Die Prozedur `flush_buffer_cache` hat keine Parameter.

Im folgenden Beispiel wird der Buffer-Cache bereinigt.

```
EXEC rdsadmin.rdsadmin_util.flush_buffer_cache;
```

Leeren des Smart-Flash-Caches der Datenbank

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.flush_flash_cache`, um den SmartFlash-Cache der Datenbank zu leeren. Die Prozedur `flush_flash_cache` hat keine Parameter. Im folgenden Beispiel wird der Smart-Flash-Cache der Datenbank geleert.

```
EXEC rdsadmin.rdsadmin_util.flush_flash_cache;
```

Weitere Hinweise zur Verwendung des Datenbank-Smart-Flash-Cache mit RDS für Oracle finden Sie unterSpeichern temporärer Daten in einem Instance-Speicher von RDS für Oracle.

Gewährung SELECT von EXECUTE Rechten für SYS Objekte

Im Normalfall werden Sonderrechte mithilfe von Rollen übertragen, die viele Objekte beinhalten können. Verwenden Sie das RDS Amazon-Verfahren, um einem einzelnen Objekt Berechtigungen zu gewährenrdsadmin.rdsadmin_util.grant_sys_object. Die Prozedur gewährt nur Berechtigungen, die dem Masterbenutzer bereits über eine Rolle oder direkte Erteilung gewährt wurden.

Die Prozedur `grant_sys_object` hat die folgenden Parameter.

> **Important**
>
> Verwenden Sie für alle Parameterwerte Großbuchstaben, es sei denn, Sie haben den Benutzer mit einer Kennung mit bedeutsamer Groß- und Kleinschreibung erstellt. Wenn Sie z. B. `CREATE USER myuser` oder `CREATE USER MYUSER` ausführen , wird im Datenwörterbuch `MYUSER` gespeichert. Wenn Sie jedoch doppelte Anführungszeichen in `CREATE USER "MyUser"` verwenden, speichert das Datenwörterbuch `MyUser`.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_obj_name`	varchar2	—	Ja	Der Name des Objekts dem die Sonderrechte erteilt werden sollen. Das Objekt kann ein Verzeichnis, eine Funktion, ein Paket, eine Prozedur, eine Sequenz, eine Tabelle oder eine Ansicht sein. Objektnamen müssen genauso angegeben werden, wie sie in sin `DBA_OBJECTS` . Die meisten Systemobj ekte sind in Großbuchs taben definiert. Daher empfehlen wir Ihnen, zuerst diese Schreibwe ise zu verwenden.
`p_grantee`	varchar2	—	Ja	Der Name des Objekts, dem die Sonderrechte erteilt werden sollen. Das Objekt kann ein Schema oder eine Rolle sein.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_privilege	varchar2	Null	Ja	—
p_grant_option	Boolean	false	Nein	Setzen Sie diesen Wert auf true, um ihn mit der Genehmigungsoption zu verwenden.

Im folgenden Beispiel werden einem Objekt mit dem Namen V_$SESSION für einen Benutzer mit dem Namen USER1 ausgewählte Berechtigungen erteilt.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_grantee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

Im folgenden Beispiel werden einem Objekt mit dem Namen V_$SESSION für einen Benutzer mit dem Namen USER1 mit der Erteilungsoption ausgewählte Berechtigungen erteilt.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name     => 'V_$SESSION',
        p_grantee      => 'USER1',
        p_privilege    => 'SELECT',
        p_grant_option => true);
end;
/
```

Um Sonderrechte für ein Objekt erteilen zu können, muss Ihr Konto über diese Sonderrechte verfügen, die ihm entweder direkt, mithilfe der Genehmigungsoption oder einer Rolle mithilfe von erteilt wurde with admin option. In den meisten Fällen möchten Sie die Gewährung möglicherweise für eine DBA Ansicht gewährenSELECT, die der SELECT_CATALOG_ROLE Rolle zugewiesen wurde. Wenn diese Rolle Ihrem Benutzer nicht bereits direkt mithilfe von with admin option erteilt wurde, können Sie die Berechtigung nicht übertragen. Wenn Sie über die DBA

entsprechende Berechtigung verfügen, können Sie die Rolle direkt einem anderen Benutzer zuweisen.

Im folgenden Beispiel wird die SELECT_CATALOG_ROLE und EXECUTE_CATALOG_ROLE an USER1 übertragen. Da der verwendet with admin option wird, USER1 kann jetzt Zugriff auf SYS Objekte gewährt werden, denen eine Genehmigung erteilt wurdeSELECT_CATALOG_ROLE.

```
GRANT SELECT_CATALOG_ROLE TO USER1 WITH ADMIN OPTION;
GRANT EXECUTE_CATALOG_ROLE to USER1 WITH ADMIN OPTION;
```

Objekte, die bereits an PUBLIC erteilt wurden, müssen nicht erneut erteilt werden. Wenn Sie die Prozedur grant_sys_object verwenden, um erneut Zugriffsrechte zu erteilen, ist der Prozeduraufruf erfolgreich.

Widerrufen von SELECT EXECUTE Rechten für Objekte SYS

Verwenden Sie das RDS Amazon-Verfahren, um Rechte für ein einzelnes Objekt zu entziehenrdsadmin.rdsadmin_util.revoke_sys_object. Das Verfahren widerruft nur Privilegien, die dem Masterkonto bereits über eine Rolle oder direkte Erteilung gewährt wurden.

Die Prozedur revoke_sys_object hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_obj_name	varchar2	—	Ja	Der Name des Objekts, für das Berechtigungen widerrufen werden sollen. Das Objekt kann ein Verzeichnis, eine Funktion, ein Paket, eine Prozedur, eine Sequenz, eine Tabelle oder eine Ansicht sein. Objektnamen müssen genauso angegeben werden, wie sie in sin DBA_OBJECTS . Die meisten Systemobjekte sind in Großbuchstaben

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				definiert, von daher empfehlen wir Ihnen, diese Schreibweise zuerst auszuprobieren.
p_revokee	varchar2	—	Ja	Der Name des Objekts, für das Berechtigungen widerrufen werden sollen. Das Objekt kann ein Schema oder eine Rolle sein.
p_privilege	varchar2	Null	Ja	—

Im folgenden Beispiel werden ausgewählte Berechtigungen für ein Objekt mit dem Namen V_$SESSION von einem Benutzer mit dem Namen USER1 widerrufen.

```
begin
    rdsadmin.rdsadmin_util.revoke_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_revokee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

Verwaltung von RDS_X$-Ansichten für Oracle-DB-Instances

Möglicherweise müssen Sie auf SYS.X$ feste Tabellen zugreifen, auf die nur über zugegriffen werden kannSYS. Verwenden Sie die im rdsadmin.rdsadmin_util Paket enthaltenen Verfahren, um SYS.RDS_X$ Ansichten für geeignete X$ Tabellen zu erstellen. Ihrem Hauptbenutzer wird automatisch die Berechtigung für die SELECT … WITH GRANT OPTION RDS_X$ Ansichten erteilt.

Die rdsadmin.rdsadmin_util Verfahren sind in den folgenden Fällen verfügbar:

- Bestehende DB-Instances, die noch nie aktualisiert wurden und die folgenden Versionen verwenden:
  - 21.0.0.0.ru-2023-10.rur-2023-10.r1und höhere 2.1c-Versionen

  - `19.0.0.0.ru-2023-10.rur-2023-10.r1`und höhere 19c-Versionen
- Jede neue DB-Instance, die Sie erstellen
- Jede bestehende DB-Instance, die Sie aktualisiert haben

> **Important**
>
> Intern erstellt das `rdsadmin.rdsadmin_util` Paket Ansichten für `X$` Tabellen. Die `X$` Tabellen sind interne Systemobjekte, die in der Oracle Database-Dokumentation nicht beschrieben werden. Wir empfehlen, dass Sie bestimmte Ansichten in Ihrer Nicht-Produktionsdatenbank testen und nur Ansichten in Ihrer Produktionsdatenbank unter Anleitung des Oracle-Supports erstellen.

Listet feste X$-Tabellen auf, die für die Verwendung in RDS_X$-Ansichten geeignet sind

Verwenden Sie das RDS-Verfahren, um X$-Tabellen aufzulisten, die für die Verwendung in `RDS_X$` Ansichten in Frage kommen. `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views` Dieses Verfahren akzeptiert keine Parameter. In den folgenden Anweisungen werden alle geeigneten `X$` Tabellen aufgeführt (einschließlich Beispielausgabe).

```
SQL> SET SERVEROUTPUT ON
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_allowed_sys_x$_views);

'X$BH'
'X$K2GTE'
'X$KCBWBPD'
'X$KCBWDS'
'X$KGLLK'
'X$KGLOB'
'X$KGLPN'
'X$KSLHOT'
'X$KSMSP'
'X$KSPPCV'
'X$KSPPI'
'X$KSPPSV'
'X$KSQEQ'
'X$KSQRS'
'X$KTUXE'
'X$KQRFP'
```

Die Liste der in Frage kommenden X$ Tabellen kann sich im Laufe der Zeit ändern. Um sicherzustellen, dass Ihre Liste der in Frage kommenden X$ festen Tische aktuell ist, sollten Sie sie `list_allowed_sys_x$_views` regelmäßig wiederholen.

SYS.RDS_X$-Ansichten werden erstellt

Verwenden Sie das RDS-Verfahren, um eine RDS_X$ Ansicht für eine geeignete X $ Tabelle zu erstellen. `rdsadmin.rdsadmin_util.create_sys_x$_view` Sie können nur Ansichten für die Tabellen erstellen, die in der Ausgabe von aufgeführt sind`rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. Das `create_sys_x$_view`-Verfahren akzeptiert die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_x$_tbl`	varchar2	Null	Ja	Ein gültiger X$ Tabellenname. Bei dem Wert muss es sich um eine der von gemeldeten X$ Tabellen handeln`list_allowed_sys_x$_views` .
`p_force_creation`	Boolesch	`FALSE`	Nein	Ein Wert, der angibt, ob die Erstellung einer RDS_X$ Ansicht erzwungen werden soll, die bereits für eine X$ Tabelle existiert. Standardmäßig erstellt RDS keine Ansicht, wenn sie bereits vorhanden ist. Um die Erstellung zu erzwingen, setzen Sie diesen Parameter auf`TRUE`.

Im folgenden Beispiel wird die `SYS.RDS_X$KGLOB` Ansicht für die Tabelle erstellt`X$KGLOB`. Das Format für den Namen der Ansicht ist`RDS_`*`X$tablename`*.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.create_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

Die folgende Datenwörterbuchabfrage listet die Ansicht auf SYS.RDS_X$KGLOB und zeigt ihren Status an. Ihrem Masterbenutzer wird automatisch die Berechtigung SELECT ... WITH GRANT OPTION für diese Ansicht gewährt.

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

OWNER                          OBJECT_NAME                    STATUS
------------------------------ ------------------------------
 ------------------------------
SYS                            RDS_X$KGLOB                    VALID
```

> **Important**
>
> X$Es kann nicht garantiert werden, dass Tabellen vor und nach einem Upgrade unverändert bleiben. RDS for Oracle löscht die RDS_X$ Ansichten der X$ Tabellen während eines Engine-Upgrades und erstellt sie neu. Anschließend wird das SELECT ... WITH GRANT OPTION Privileg dem Masterbenutzer gewährt. Erteilen Sie Datenbankbenutzern nach einem Upgrade nach Bedarf Berechtigungen für die entsprechenden RDS_X$ Ansichten.

SYS.RDS_X$-Ansichten auflisten

Verwenden Sie das RDS-Verfahren, um vorhandene RDS_X$ Ansichten aufzulisten. rdsadmin.rdsadmin_util.list_created_sys_x$_views Die Prozedur listet nur Ansichten auf, die mit der Prozedur erstellt wurdencreate_sys_x$_view. Das folgende Beispiel listet X$ Tabellen RDS_X$ mit entsprechenden Ansichten auf (einschließlich Beispielausgabe).

```
SQL> SET SERVEROUTPUT ON
```

```
SQL> COL XD_TBL_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_created_sys_x$_views);

XD_TBL_NAME                    STATUS
------------------------------ ------------------------------
X$BH                           VALID
X$K2GTE                        VALID
X$KCBWBPD                      VALID

3 rows selected.
```

RDS_X$-Ansichten werden gelöscht

Verwenden Sie das RDS-Verfahren, um eine SYS.RDS_X$ Ansicht zu löschen. `rdsadmin.rdsadmin_util.drop_sys_x$_view` Sie können nur Ansichten löschen, die in der Ausgabe von aufgeführt sind`rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. Das `drop_sys_x$_view`-Verfahren akzeptiert den folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_x$_tbl`	varchar2	Null	Ja	Ein gültiger X$ fester Tabellenname. Bei dem Wert muss es sich um eine der X$ festen Tabellen handeln, die von gemeldet wurden`list_crea ted_sys_x$_views` .

Im folgenden Beispiel wird die RDS_X$KGLOB Ansicht gelöscht, die für die Tabelle erstellt wurdeX $KGLOB.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.drop_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

Das folgende Beispiel zeigt, dass die Ansicht gelöscht SYS.RDS_X$KGLOB wurde (einschließlich Beispielausgabe).

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

no rows selected
```

Erteilen von Berechtigungen an Nicht-Hauptbenutzer

Sie können Auswahl-Sonderrechte für viele Objekte im SYS-Schema mithilfe der SELECT_CATALOG_ROLE-Rolle erteilen. Die Rolle SELECT_CATALOG_ROLE gibt Benutzern SELECT-Sonderrechte für Datenverzeichnisansichten. Im folgenden Beispiel wird die Rolle SELECT_CATALOG_ROLE einem Benutzer mit dem Namen user1 erteilt.

```
GRANT SELECT_CATALOG_ROLE TO user1;
```

Sie können EXECUTE-Sonderrechte für viele Objekte im SYS-Schema mithilfe der EXECUTE_CATALOG_ROLE-Rolle erteilen. Die Rolle EXECUTE_CATALOG_ROLE gibt Benutzern EXECUTE-Sonderrechte für Pakete und Prozeduren im Datenverzeichnis. Im folgenden Beispiel wird die Rolle EXECUTE_CATALOG_ROLE einem Benutzer mit dem Namen user1 erteilt.

```
GRANT EXECUTE_CATALOG_ROLE TO user1;
```

Im folgenden Beispiel werden die Berechtigungen abgerufen, die durch die Rollen SELECT_CATALOG_ROLE und EXECUTE_CATALOG_ROLE gewährt werden.

```
  SELECT *
    FROM ROLE_TAB_PRIVS
   WHERE ROLE IN ('SELECT_CATALOG_ROLE','EXECUTE_CATALOG_ROLE')
ORDER BY ROLE, TABLE_NAME ASC;
```

Im folgenden Beispiel wird ein Nicht-Masterbenutzer mit dem Namen `user1` erstellt, die Berechtigung `CREATE SESSION` gewährt und die Berechtigung `SELECT` für eine Datenbank mit dem Namen sh.sales erteilt.

```
CREATE USER user1 IDENTIFIED BY PASSWORD;
GRANT CREATE SESSION TO user1;
GRANT SELECT ON sh.sales TO user1;
```

Erstellen von benutzerdefinierten Funktionen für das Überprüfen von Passwörtern

Es gibt verschiedene Möglichkeiten, eine benutzerdefinierte Funktion für die Passwortüberprüfung zu erstellen:

- Um die Standardüberprüfungslogik zu verwenden und Ihre Funktion im SYS-Schema zu speichern, verwenden Sie die Prozedur `create_verify_function`.
- Um eine benutzerdefinierte Überprüfungslogik zu verwenden und Ihre Funktion nicht im SYS-Schema zu speichern, verwenden Sie die Prozedur `create_passthrough_verify_fcn`.

Die Prozedur create_verify_function

Sie können mithilfe des RDS Amazon-Verfahrens eine benutzerdefinierte Funktion zur Überprüfung von Passwörtern erstellen`rdsadmin.rdsadmin_password_verify.create_verify_function`. Das `create_verify_function` Verfahren wird für alle Versionen von RDS for Oracle unterstützt.

Die Prozedur `create_verify_function` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_verify_function_name`	varchar2	—	Ja	Der Name für Ihre benutzerdefinierte Funktion. Diese Funktion wird für Sie im SYS Schema erstellt. Sie teilen diese Funktion den Benutzerprofilen zu.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_min_length	Zahl	8	Nein	Die erforderliche Mindestzeichenanzahl.
p_max_length	Zahl	256	Nein	Die maximale Anzahl der erlaubten Zeichen
p_min_letters	Zahl	1	Nein	Die Mindestanzahl der erforderlichen Buchstaben
p_min_uppercase	Zahl	0	Nein	Die Mindestanzahl der erforderlichen Großbuchstaben
p_min_lowercase	Zahl	0	Nein	Die Mindestanzahl der erforderlichen Kleinbuchstaben
p_min_digits	Zahl	1	Nein	Die Mindestanzahl der erforderlichen Zahlen
p_min_special	Zahl	0	Nein	Die Mindestanzahl der erforderlichen Sonderzeichen
p_min_different_chars	Zahl	3	Nein	Die Mindestanzahl der zwischen dem alten und dem neuen Passwort erforderlichen unterschiedlichen Zeichen.
p_disallow_username	Boolean	true	Nein	Setzen Sie diesen Wert auf `true`, um den Benutzernamen im Passwort nicht zu erlauben.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_disallow_reverse	Boolean	true	Nein	Auf true festlegen, um die Umkehrung des Benutzernamens im Passwort zu verbieten.
p_disallow_db_name	Boolean	true	Nein	Setzen Sie diesen Wert auf true, um den Datenbank- oder Servernamen im Passwort nicht zu erlauben.
p_disallow_simple_strings	Boolean	true	Nein	Setzen Sie diesen Wert auf true, um einfache Zeichenfolgen im Passwort nicht zu erlauben.
p_disallow_whitespace	Boolean	false	Nein	Setzen Sie diesen Wert auf true, um Leerzeichen im Passwort nicht zu erlauben.
p_disallow_at_sign	Boolean	false	Nein	Setzen Sie diesen Wert auf true, um das @-Zeichen im Passwort nicht zu erlauben.

Sie können mehrere benutzerdefinierte Funktionen für die Passwortüberprüfung erstellen.

Es gibt Einschränkungen im Hinblick auf den Namen für Ihre benutzerdefinierte Funktion. Der Name Ihrer benutzerdefinierten Funktion darf nicht mit dem Namen eines vorhandenen Systemobjekts identisch sein. Der Name darf nicht mehr als 30 Zeichen lang sein. Der Name muss außerdem eine der folgenden Zeichenfolgen enthalten: PASSWORD, VERIFY, COMPLEXITY, ENFORCE oder STRENGTH.

Im folgenden Beispiel wird die Funktion mit dem Namen erstell CUSTOM_PASSWORD_FUNCTION. Die Funktion erfordert ein Passwort mit mindestens 12 Zeichen, 2 Großbuchstaben, 1 Zahl und 1 Sonderzeichen. Außerdem ist das @-Zeichen im Passwort nicht erlaubt.

```
begin
    rdsadmin.rdsadmin_password_verify.create_verify_function(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_min_length           => 12,
        p_min_uppercase        => 2,
        p_min_digits           => 1,
        p_min_special          => 1,
        p_disallow_at_sign     => true);
end;
/
```

Tätigen Sie die Abfrage DBA_SOURCE, um den Text Ihrer Überprüfungsfunktion anzusehen. Im folgenden Beispiel wird der Text einer benutzerdefinierten Passwortfunktion mit dem Namen erhalte CUSTOM_PASSWORD_FUNCTION.

```
COL TEXT FORMAT a150

  SELECT TEXT
    FROM DBA_SOURCE
   WHERE OWNER = 'SYS'
     AND NAME = 'CUSTOM_PASSWORD_FUNCTION'
ORDER BY LINE;
```

Verwenden Sie `alter profile`, um Ihre Überprüfungsfunktion einem Benutzerprofil zuzuordnen. Verwenden Sie DEFAULT, um Ihre Überprüfungsfunktion dem Benutzerprofil zuzuordnen.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Stellen Sie die Abfrage , um zu sehen, welchen Benutzerprofilen welche Überprüfungsfunktionen zugeordnet sin DBA_PROFILES. Im folgenden Beispiel wird das Profil erhalten, dem die benutzerdefinierte Überprüfungsfunktion mit dem Namen zugehörig is CUSTOM_PASSWORD_FUNCTION.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD' AND LIMIT =
 'CUSTOM_PASSWORD_FUNCTION';
```

```
PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
```

Im folgenden Beispiel werden alle Profile und Passwortüberprüfungsfunktionen erhalten, die miteinander verknüpft sind.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION';

PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
RDSADMIN                   PASSWORD_VERIFY_FUNCTION         PASSWORD  NULL
```

Die Prozedur create_passthrough_verify_fcn

Das `create_passthrough_verify_fcn` Verfahren wird für alle Versionen von RDS for Oracle unterstützt.

Sie können mithilfe des RDS Amazon-Verfahrens eine benutzerdefinierte Funktion zur Überprüfung von Passwörtern erstellen`rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`. Die Prozedur `create_passthrough_verify_fcn` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_verify_function_name`	varchar2	—	Ja	Der Name für Ihre benutzerdefinierte Überprüfungsfunktion. Dies ist eine Wrapper-Funktion, die für Sie im SYS Schema erstellt wird und keine Bestätigungslogik enthält. Sie

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				teilen diese Funktion den Benutzerprofilen zu.
p_target_owner	varchar2	—	Ja	Der Schemabesitzer für Ihre benutzerdefinierte Überprüfungsfunktion
p_target_function_name	varchar2	—	Ja	Der Name für Ihre bestehende benutzerdefinierte Funktion, der die Überprüfungslogik beinhaltet. Ihre benutzerdefinierte Funktion muss einen Booleschen Wert zurückgeben. Ihre Funktion sollte `true`zurückgeben, wenn das Passwort gültig ist, und `false`, wenn das Passwort ungültig ist.

Im folgenden Beispiel wird eine Passwortüberprüfungsfunktion erstellt, die die Logik aus der Funktion mit dem Namen verwende `PASSWORD_LOGIC_EXTRA_STRONG`.

```
begin
    rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_target_owner         => 'TEST_USER',
        p_target_function_name => 'PASSWORD_LOGIC_EXTRA_STRONG');
end;
/
```

Verwenden Sie , um die Überprüfungsfunktion Ihrem Benutzerprofil zuzuordne `alter profile`. Im folgenden Beispiel wird die Überprüfungsfunktion mit dem `DEFAULT`-Benutzerprofil verknüpft.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Einen benutzerdefinierten DNS Server einrichten

Amazon RDS unterstützt ausgehenden Netzwerkzugriff auf Ihre DB-Instances, auf denen Oracle ausgeführt wird. Weitere Informationen zu ausgehendem Netzwerkzugriff, einschließlich Voraussetzungen, finden Sie unter Konfigurieren des UTL_HTTP-Zugriffs mit Zertifikaten und einer Oracle Wallet.

Amazon RDS Oracle ermöglicht die Auflösung von Domain Name Service (DNS) von einem benutzerdefinierten DNS Server aus, der dem Kunden gehört. Sie können nur vollständig qualifizierte Domainnamen aus Ihrer Amazon RDS DB-Instance über Ihren benutzerdefinierten DNS Server auflösen.

Nachdem Sie Ihren benutzerdefinierten DNS Nameserver eingerichtet haben, dauert es bis zu 30 Minuten, bis die Änderungen an Ihrer DB-Instance weitergegeben werden. Nachdem die Änderungen an Ihre DB-Instance weitergegeben wurden, fragt der gesamte ausgehende Netzwerkverkehr, für den ein DNS Lookup erforderlich ist, Ihren DNS Server über Port 53 ab.

Gehen Sie wie folgt vor, um einen benutzerdefinierten DNS Server für Ihre Amazon RDS for Oracle-DB-Instance einzurichten:

- Stellen Sie aus dem DHCP Optionssatz, der Ihrer virtuellen privaten Cloud (VPC) zugeordnet ist, die `domain-name-servers` Option auf die IP-Adresse Ihres DNS Nameservers ein. Weitere Informationen finden Sie unter DHCPOptionssätze.

  > **Note**
  >
  > Die `domain-name-servers` Option akzeptiert bis zu vier Werte, aber Ihre Amazon RDS DB-Instance verwendet nur den ersten Wert.

- Stellen Sie sicher, dass Ihr DNS Server alle Suchanfragen lösen kann, einschließlich öffentlicher DNS Namen, EC2 privater DNS Amazon-Namen und kundenspezifischer DNS Namen. Wenn der ausgehende Netzwerkverkehr DNS Suchvorgänge enthält, die Ihr DNS Server nicht verarbeiten kann, müssen für Ihren DNS Server die entsprechenden DNS Upstream-Anbieter konfiguriert sein.
- Konfigurieren Sie Ihren DNS Server so, dass er User Datagram Protocol (UDP) -Antworten von 512 Byte oder weniger erzeugt.
- Konfigurieren Sie Ihren DNS Server so, dass er Transmission Control Protocol (TCP) -Antworten mit einer Größe von 1024 Byte oder weniger erzeugt.

- Konfigurieren Sie Ihren DNS Server so, dass eingehender Datenverkehr von Ihren Amazon RDS DB-Instances über Port 53 zugelassen wird. Wenn sich Ihr DNS Server in einem Amazon befindetVPC, VPC muss er über eine Sicherheitsgruppe verfügen, die Regeln für eingehenden TCP Datenverkehr auf Port 53 enthält. UDP Wenn sich Ihr DNS Server nicht in einem Amazon befindetVPC, muss er über eine entsprechende Firewall-Zulassungsliste verfügen, um TCP eingehenden Datenverkehr auf Port 53 zuzulassenUDP.

  Weitere Informationen finden Sie unter Sicherheitsgruppen für Sie VPC und Regeln hinzufügen und entfernen.

- Konfigurieren Sie Ihre Amazon RDS DB-Instance so, dass ausgehender Datenverkehr über Port 53 zugelassen wird. VPC Ihre Sicherheitsgruppe VPC muss über eine Sicherheitsgruppe verfügen, die Regeln für ausgehenden TCP Datenverkehr auf Port 53 enthält. UDP

  Weitere Informationen finden Sie unter Sicherheitsgruppen für Sie VPC und Regeln hinzufügen und entfernen.

- Der Routing-Pfad zwischen der RDS Amazon-DB-Instance und dem DNS Server muss korrekt konfiguriert sein, um DNS Datenverkehr zuzulassen.
  - Wenn sich die RDS Amazon-DB-Instance und der DNS Server nicht in derselben befindenVPC, muss eine Peering-Verbindung zwischen ihnen eingerichtet werden. Weitere Informationen finden Sie unter Was ist VPC Peering?

Festlegen und Aufheben von Systemdiagnoseereignissen

Um Diagnoseereignisse auf Sitzungsebene festzulegen und zu deaktivieren, können Sie die SQL Oracle-Anweisung `ALTER SESSION SET EVENTS` verwenden. Um Ereignisse auf Systemebene festzulegen, können Sie Oracle SQL jedoch nicht verwenden. Verwenden Sie stattdessen die Systemereignisprozesse im `rdsadmin.rdsadmin_util`-Paket. Die Systemereignisprozesse sind in den folgenden Engine-Versionen verfügbar:

- Alle Versionen von Oracle Database 21c
- 19.0.0.0.ru-2020-10.rur-2020-10.r1 oder höhere Versionen von Oracle Database 19c

  Weitere Informationen finden Sie unter Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 in den Versionshinweisen zu Amazon for Oracle RDS

> **Important**
>
> Intern legt das `rdsadmin.rdsadmin_util`-Paket Ereignisse mithilfe der `ALTER SYSTEM SET EVENTS`-Anweisung fest. Diese `ALTER SYSTEM`-Aussage ist nicht in der Oracle Database-Dokumentation dokumentiert. Einige Systemdiagnoseereignisse können große Mengen an Rückverfolgungsinformationen generieren, Konflikte verursachen oder die Datenbankverfügbarkeit beeinträchtigen. Wir empfehlen, dass Sie bestimmte Diagnoseereignisse in Ihrer Nicht-Produktionsdatenbank testen und Ereignisse in Ihrer Produktionsdatenbank nur unter Anleitung von Oracle Support festlegen.

Auflisten zulässiger Systemdiagnoseereignisse

Verwenden Sie das RDS Amazon-Verfahren, um die Systemereignisse aufzulisten, die Sie festlegen können`rdsadmin.rdsadmin_util.list_allowed_system_events`. Dieses Verfahren akzeptiert keine Parameter.

Im folgenden Beispiel werden alle Systemereignisse aufgeführt, die Sie festlegen können.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_allowed_system_events;
```

Die folgende Beispielausgabe listet Ereignisnummern und ihre Beschreibungen auf. Verwenden Sie die RDS Amazon-Verfahren`set_system_event`, um diese Ereignisse festzulegen und `unset_system_event` zu deaktivieren.

```
604   - error occurred at recursive SQL level
942   - table or view does not exist
1401  - inserted value too large for column
1403  - no data found
1410  - invalid ROWID
1422  - exact fetch returns more than requested number of rows
1426  - numeric overflow
1427  - single-row subquery returns more than one row
1476  - divisor is equal to zero
1483  - invalid length for DATE or NUMBER bind variable
1489  - result of string concatenation is too long
1652  - unable to extend temp segment by  in tablespace
1858  - a non-numeric character was found where a numeric was expected
4031  - unable to allocate  bytes of shared memory ("","","","")
6502  - PL/SQL: numeric or value error
```

```
10027 - Specify Deadlock Trace Information to be Dumped
10046 - enable SQL statement timing
10053 - CBO Enable optimizer trace
10173 - Dynamic Sampling time-out error
10442 - enable trace of kst for ORA-01555 diagnostics
12008 - error in materialized view refresh path
12012 - error on auto execute of job
12504 - TNS:listener was not given the SERVICE_NAME in CONNECT_DATA
14400 - inserted partition key does not map to any partition
31693 - Table data object  failed to load/unload and is being skipped due to error:
```

> **Note**
>
> Die Liste der erlaubten Systemereignisse kann sich im Laufe der Zeit ändern. Um sicherzustellen, dass Sie die aktuellste Liste der berechtigten Ereignisse haben, verwenden Sie `rdsadmin.rdsadmin_util.list_allowed_system_events`.

Festlegen von Systemdiagnoseereignissen

Verwenden Sie das RDS Amazon-Verfahren, um ein Systemereignis festzulegen`rdsadmin.rdsadmin_util.set_system_event`. Sie können nur Ereignisse festlegen, die in der Ausgabe von aufgeführt sin `rdsadmin.rdsadmin_util.list_allowed_system_events`. Das `set_system_event`-Verfahren akzeptiert die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_event`	Zahl	—	Ja	Die Systemereignisnummer. Der Wert muss eine der von gemeldeten Ereignisnummern sei `list_allowed_system_events` .
`p_level`	Zahl	—	Ja	Das Event-Level. In der Oracle Database-Dokumentation oder bei Oracle Support finden

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				Sie Beschreibungen von Werten verschiedener Ebenen.

Das Verfahren `set_system_event` konstruiert und führt die erforderlichen `ALTER SYSTEM SET EVENTS`-Aussagen nach folgenden Grundsätzen aus:

- Der Ereignistyp (`context` oder `errorstack`) wird automatisch bestimmt.
- Eine Anweisung im Formular `ALTER SYSTEM SET EVENTS 'event LEVEL event_level'` legt die Kontextereignisse fest. Diese Notation entspricht `ALTER SYSTEM SET EVENTS 'event TRACE NAME CONTEXT FOREVER, LEVEL event_level'`.
- Eine Anweisung im Formular `ALTER SYSTEM SET EVENTS 'event ERRORSTACK (event_level)'` legt die Fehler-Stack-Ereignisse fest. Diese Notation entspricht `ALTER SYSTEM SET EVENTS 'event TRACE NAME ERRORSTACK LEVEL event_level'`.

Im folgenden Beispiel werden Ereignis 942 auf Ebene 3 und Ereignis 10 442 auf Ebene 10 festgelegt. Die Beispielausgabe ist enthalten.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(942,3);
Setting system event 942 with: alter system set events '942 errorstack (3)'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(10442,10);
Setting system event 10442 with: alter system set events '10442 level 10'

PL/SQL procedure successfully completed.
```

Auflisten der festgesetzten Systemdiagnoseereignisse

Verwenden Sie das RDS Amazon-Verfahren, um die aktuell festgelegten Systemereignisse aufzulistenrdsadmin.rdsadmin_util.list_set_system_events. Dieses Verfahren meldet nur Ereignisse, die auf Systemebene von festgelegt wurde `set_system_event`.

Im folgenden Beispiel werden die aktiven Systemereignisse aufgeführt.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_set_system_events;
```

Die folgende Beispielausgabe zeigt die Liste der Ereignisse, den Ereignistyp, die Ebene, auf der die Ereignisse derzeit festgelegt sind, und den Zeitpunkt, zu dem das Ereignis festgelegt wurde.

```
942 errorstack (3) - set at 2020-11-03 11:42:27
10442 level 10 - set at 2020-11-03 11:42:41

PL/SQL procedure successfully completed.
```

Aufheben von Systemdiagnoseereignissen

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.unset_system_event`, um ein Systemereignis rückgängig zu machen. Sie können nur die in der Ausgabe von aufgelisteten Ereignisse aufhebe `rdsadmin.rdsadmin_util.list_allowed_system_events`. Das `unset_system_event`-Verfahren akzeptiert den folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_event`	Zahl	—	Ja	Die Systemereignisnummer. Der Wert muss eine der von gemeldeten Ereignisnummern sei `list_allowed_system_events` .

Im folgenden Beispiel werden die Ereignisse 942 und 10 442 aufgezeichnet. Die Beispielausgabe ist enthalten.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(942);
Unsetting system event 942 with: alter system set events '942 off'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(10442);
Unsetting system event 10442 with: alter system set events '10442 off'
```

```
PL/SQL procedure successfully completed.
```

Ausführen allgemeiner Datenbank-Aufgaben für Oracle DB-Instances

Im Folgenden erfahren Sie, wie Sie bestimmte allgemeine DBA Aufgaben im Zusammenhang mit Datenbanken auf Ihren RDS Amazon-DB-Instances ausführen, auf denen Oracle ausgeführt wird. Um ein Managed-Serviceerlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances. Amazon schränkt RDS auch den Zugriff auf einige Systemprozeduren und Tabellen ein, für die erweiterte Rechte erforderlich sind.

Themen

- Ändern des globalen Namens einer Datenbank
- Tablespaces verwenden
- Tempfiles verwenden
- Überprüfung einer Datenbank
- Einstellen der verteilten Wiederherstellung
- Einstellen der Datenbank-Zeitzone
- Arbeiten mit externen Oracle-Tabellen
- Generieren von Leistungsberichten mit Automatic Workload Repository (AWR)
- Anpassen von Datenbank-Links für die Verwendung mit DB-Instances in einem VPC
- Einrichten der Standardversion für eine DB-Instance
- Aktivieren der Überwachung für dieSYS. AUD$-Tabelle
- Deaktivierung der Überwachung für dieSYS. AUD$-Tabelle
- Bereinigen unterbrochener Online-Index-Builds
- Überspringen von beschädigten Blöcken
- Ändern der Größe von Tabellenbereichen, Datendateien und temporären Dateien
- Bereinigen des Papierkorbs
- Festlegen der angezeigten Standardwerte für vollständige Redaktion

Ändern des globalen Namens einer Datenbank

Verwenden Sie das RDS Amazon-Verfahren, um den globalen Namen einer Datenbank zu ändern`rdsadmin.rdsadmin_util.rename_global_name`. Die Prozedur `rename_global_name` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_new_global_name	varchar2	—	Ja	Der neue globale Name für die Datenbank

Die Datenbank muss geöffnet sein, damit die Änderungen übernommen werden. Weitere Informationen zum Ändern des globalen Namens einer Datenbank finden Sie ALTERDATABASEin der Oracle-Dokumentation.

Im folgenden Beispiel wird der globale Name einer Datenbank geänder `new_global_name`.

```
EXEC rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

Tablespaces verwenden

Sie können Tablespaces mit RDS for Oracle verwenden, einer logischen Speichereinheit, die die Daten der Datenbank speichert.

Themen

- Erstellen und Größenfestlegung von Tabellenräumen
- Einrichten des Standard-Tabellenraums
- Einrichten des temporären Standard-Tabellenraums
- Erstellen eines temporären Tabellenraums im Instance-Speicher

Erstellen und Größenfestlegung von Tabellenräumen

Amazon unterstützt Oracle Managed Files (OMF) RDS nur für Datendateien, Protokolldateien und Kontrolldateien. Wenn Sie Dateien oder Protokolldateien erstellen, können Sie die physikalischen Dateinamen nicht festlegen.

Wenn Sie keine Datendateigröße angeben, werden Tabellenbereiche standardmäßig mit der Standardeinstellung `AUTOEXTEND ON` und ohne maximal zulässige Größe erstellt. Im folgenden Beispiel der Tablespace *`users1`* ist automatisch erweiterbar.

```
CREATE TABLESPACE users1;
```

Durch diese Standardeinstellungen können Tabellenräume so sehr anwachsen, dass sie den gesamten zugewiesenen Speicherplatz verwenden. Wir empfehlen, dass Sie eine angemessene Maximalgröße für permanente und temporäre Tabellenräume festlegen, und dass Sie die Speicherverwendung sorgfältig überwachen.

Im folgenden Beispiel wird ein Tablespace mit dem Namen erstellt *users2* mit einer Startgröße von 1 Gigabyte. Da zwar die Datendateigröße, nicht aber AUTOEXTEND ON angegeben ist, ist der Tabellenbereich nicht automatisch erweiterbar.

```
CREATE TABLESPACE users2 DATAFILE SIZE 1G;
```

Im folgenden Beispiel wird ein Tablespace mit dem Namen erstellt *users3* mit einer Startgröße von 1 Gigabyte, aktivierter Autoextension und einer maximalen Größe von 10 Gigabyte.

```
CREATE TABLESPACE users3 DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE 10G;
```

Im folgenden Beispiel wird ein temporärer Tablespace mit dem Namen erstellt *temp01*.

```
CREATE TEMPORARY TABLESPACE temp01;
```

Sie können die Größe eines Tabellenraums mit großen Dateien mithilfe von änder ALTER TABLESPACE. Sie können die Größe in Kilobytes (KB), Megabytes (MB), Gigabytes (GB) oder Terabytes (TB) festlegen. Im folgenden Beispiel wird die Größe eines BigFile-Tablespace mit dem Namen geändert *users_bf* auf 200 MB.

```
ALTER TABLESPACE users_bf RESIZE 200M;
```

Im folgenden Beispiel wird einem Tablespace für kleine Dateien eine zusätzliche Datendatei mit dem Namen hinzugefügt *users_sf*.

```
ALTER TABLESPACE users_sf ADD DATAFILE SIZE 100000M AUTOEXTEND ON NEXT 250m
 MAXSIZE UNLIMITED;
```

Einrichten des Standard-Tabellenraums

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.alter_default_tablespace`, um den Standard-Tablespace festzulegen. Die Prozedur `alter_default_tablespace` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
tablespace_name	varchar	—	Ja	Der Name des Standard-Tabellenraums

Im folgenden Beispiel wird der Standard-Tablespace auf festgelegt *users2*:

```
EXEC rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

Einrichten des temporären Standard-Tabellenraums

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.alter_default_temp_tablespace`, um den standardmäßigen temporären Tablespace festzulegen. Die Prozedur `alter_default_temp_tablespace` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
tablespace_name	varchar	—	Ja	Der Name des temporären Standard-Tabellenraums

Im folgenden Beispiel wird der standardmäßige temporäre Tablespace auf festgelegt *temp01*.

```
EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

Erstellen eines temporären Tabellenraums im Instance-Speicher

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace`, um einen temporären Tablespace im Instance-Speicher zu erstellen. Die Prozedur `create_inst_store_tmp_tblspace` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_tablespace_name	varchar	—	Ja	Der Name des temporären Tabellenraums.

Im folgenden Beispiel wird der temporäre Tablespace erstellt *temp01* im Instanzspeicher.

```
EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace(p_tablespace_name =>
 'temp01');
```

> **Important**
>
> Beim Ausführen von rdsadmin_util.create_inst_store_tmp_tblspace wird der neu erstellte temporäre Tabellenraum nicht automatisch als temporärer Standardtabellenraum festgelegt. Informationen zum Festlegen als Standard finden Sie unter Einrichten des temporären Standard-Tabellenraums.

Weitere Informationen finden Sie unter Speichern temporärer Daten in einem Instance-Speicher von RDS für Oracle.

Tempfiles verwenden

Hinzufügen einer temporären Datei zum Instance-Speicher auf einer Read Replica

Wenn Sie einen temporären Tabellenraum auf einer primären DB-Instance erstellen, werden von der Read Replica keine temporären Dateien erstellt. Nehmen Sie an, dass aus einem der folgenden Gründe ein leerer temporärer Tabellenraum in Ihrer Read Replica vorhanden ist:

- Sie haben eine temporäre Datei aus dem Tabellenraum Ihrer Read Replica gelöscht. Weitere Informationen finden Sie unter Löschen von temporären Dateien auf einer Read Replica.
- Sie haben einen neuen temporären Tabellenraum auf der primären DB-Instance erstellt. In diesem Fall synchronisiert RDS for Oracle die Metadaten mit der Read Replica.

Sie können ein temporäre Datei dem leeren temporären Tabellenraum hinzufügen und die temporäre Dateie im Instance-Speicher ablegen. Verwenden Sie das RDS Amazon-

Verfahren`rdsadmin.rdsadmin_util.add_inst_store_tempfile`, um eine temporäre Datei im Instance-Speicher zu erstellen. Sie können dieses Verfahren nur für eine Read Replica verwenden. Die Prozedur hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_tablespace_name`	varchar	—	Ja	Der Name des temporären Tabellenraums auf Ihrer Read Replica.

Im folgenden Beispiel der leere temporäre Tablespace *`temp01`* ist auf Ihrer Read Replica vorhanden. Führen Sie den folgenden Befehl aus, um eine temporäre Datei für diesen Tabellenraum zu erstellen und sie im Instance-Speicher abzulegen.

```
EXEC rdsadmin.rdsadmin_util.add_inst_store_tempfile(p_tablespace_name => 'temp01');
```

Weitere Informationen finden Sie unter Speichern temporärer Daten in einem Instance-Speicher von RDS für Oracle.

Löschen von temporären Dateien auf einer Read Replica

Sie können einen vorhandenen temporären Tabellenraum auf einer Read Replica nicht löschen. Sie können den Tempfile-Speicher auf einer Read Replica von Amazon EBS zum Instance-Speicher oder vom Instance-Speicher zu Amazon ändern. EBS Gehen Sie wie folgt vor, um diese Ziele zu erreichen:

1. Löschen Sie die aktuellen temporären Dateien im temporären Tabellenraum auf der Read Replica.
2. Erstellen Sie neue temporäre Dateien in einem anderen Speicher.

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.drop_replica_tempfiles`, um die temporären Dateien zu löschen. Sie können dieses Verfahren nur für Read Replicas verwenden. Die Prozedur `drop_replica_tempfiles` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_tablespace_name	varchar	—	Ja	Der Name des temporären Tabellenraums auf Ihrer Read Replica.

Gehen Sie davon aus, dass ein temporärer Tablespace mit dem Namen *temp01* befindet sich im Instanzspeicher auf Ihrer Read Replica. Löschen Sie alle temporären Dateien in diesem Tabellenraum, indem Sie den folgenden Befehl ausführen.

```
EXEC rdsadmin.rdsadmin_util.drop_replica_tempfiles(p_tablespace_name => 'temp01');
```

Weitere Informationen finden Sie unter Speichern temporärer Daten in einem Instance-Speicher von RDS für Oracle.

Überprüfung einer Datenbank

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.checkpoint`, um die Datenbank zu überprüfen. Die Prozedur `checkpoint` hat keine Parameter.

Im folgenden Beispiel wird die Datenbank stichprobenartig kontrolliert.

```
EXEC rdsadmin.rdsadmin_util.checkpoint;
```

Einstellen der verteilten Wiederherstellung

Verwenden Sie die RDS Amazon-Verfahren `rdsadmin.rdsadmin_util.enable_distr_recovery` und, um die verteilte Wiederherstellung einzurichten`disable_distr_recovery`. Die Prozeduren haben keine Parameter.

Im folgenden Beispiel wird die verteilte Wiederherstellung aktiviert.

```
EXEC rdsadmin.rdsadmin_util.enable_distr_recovery;
```

Im folgenden Beispiel wird die verteilte Wiederherstellung deaktiviert.

```
EXEC rdsadmin.rdsadmin_util.disable_distr_recovery;
```

Einstellen der Datenbank-Zeitzone

Sie können die Zeitzone Ihrer Amazon RDS Oracle-Datenbank auf folgende Weise festlegen:

- Die Option `Timezone`

  Die Option `Timezone` ändert die Zeitzone auf Host-Ebene und wirkt sich auf alle Datumsspalten und -Werte aus, wie zum Beispiel auf `SYSDATE`. Weitere Informationen finden Sie unter Oracle-Zeitzone.
- Das RDS Amazon-Verfahren `rdsadmin.rdsadmin_util.alter_db_time_zone`

  Die Prozedur `alter_db_time_zone` ändert die Zeitzone nur für bestimmte Datentypen und ändert nicht `SYSDATE`. Beim Einstellen der Zeitzone bestehen weitere Beschränkungen, die Sie in der Oracle-Dokumentation nachlesen.

> **Note**
>
> Sie können auch die Standardzeitzone für Oracle Scheduler festlegen. Weitere Informationen finden Sie unter Festlegen der Zeitzone für Oracle Scheduler-Aufgaben.

Die Prozedur `alter_db_time_zone` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_new_tz`	varchar2	—	Ja	Die neue Zeitzone als benannte Region oder als absolute Abweichung von der koordinierten Weltzeit (UTC). Gültige Offsets liegen im Bereich von -12:00 bis +14:00.

Im folgenden Beispiel wird die Zeitzone auf UTC plus drei Stunden geändert.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

Im folgenden Beispiel wird die Zeitzone auf die Zeitzone „Afrika/Algerien“ geändert.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

Nachdem Sie die Zeitzone mithilfe der Prozedur `alter_db_time_zone` geändert haben, starten Sie Ihre DB-Instance neu, damit die Änderung übernommen wird. Weitere Informationen finden Sie unter Neustarten einer DB-Instance. Weitere Informationen zum Aktualisieren von Zeitzonen finden Sie unter Überlegungen zur Zeitzone.

Arbeiten mit externen Oracle-Tabellen

Externe Oracle-Tabellen sind Tabellen mit Daten, die sich nicht in der Datenbank befinden. Stattdessen befinden sich die Daten in externen Dateien, auf die die Datenbank zugreifen kann. Durch die Verwendung externer Tabellen können Sie auf die Daten zugreifen, ohne sie in die Datenbank zu laden. Weitere Informationen zu externen Tabellen finden Sie unter Managing External Tables in der Oracle-Dokumentation.

Mit Amazon RDS können Sie externe Tabellendateien in Verzeichnisobjekten speichern. Sie können ein Verzeichnisobjekt erstellen oder eines verwenden, das in der Oracle-Datenbank vordefiniert ist, z. B. das DIR Verzeichnis DATA PUMP _ _. Weitere Informationen zum Erstellen von Verzeichnisobjekten finden Sie unter Erstellen und Löschen von Verzeichnissen im Hauptdatenspeicherbereich. Sie können die DIRECTORIES Ansicht ALL _ abfragen, um die Verzeichnisobjekte für Ihre Amazon RDS Oracle-DB-Instance aufzulisten.

> **Note**
>
> Verzeichnisobjekte verweisen auf den Hauptdatenspeicher (EBSAmazon-Volume), der von Ihrer Instance verwendet wird. Der belegte Speicherplatz – mit Datendateien, Redo-Logs, Audit- und Nachverfolgungsdateien sowie anderen Dateien – wird auf den zugewiesenen Speicherplatz angerechnet.

Sie können eine externe Datendatei von einer Oracle-Datenbank in eine andere verschieben, indem Sie das TRANSFER Paket DBMSFILE_ _ oder das FILE Paket UTL_ verwenden. Die externe Datendatei wird von einem Verzeichnis auf der Quelldatenbank in das angegebene Verzeichnis auf der Zieldatenbank verschoben. Hinweise zur Verwendung von DBMS _ FILE _ TRANSFER finden Sie unterImportieren mit Oracle Data Pump.

Nachdem Sie die externe Datendatei verschoben haben, können Sie damit eine externe Tabelle anlegen. Im folgenden Beispiel wird eine externe Tabelle erstellt, die die `emp_xt_file1.txt` Datei im DIR1 Verzeichnis USER _ verwendet.

```
CREATE TABLE emp_xt (
  emp_id      NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  user_name   VARCHAR2(20)
)
ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY USER_DIR1
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL
    (emp_id,first_name,last_name,user_name)
  )
  LOCATION ('emp_xt_file1.txt')
)
PARALLEL
REJECT LIMIT UNLIMITED;
```

Angenommen, Sie möchten Daten, die sich in einer Amazon RDS Oracle-DB-Instance befinden, in eine externe Datendatei verschieben. In diesem Fall können Sie die externe Datendatei füllen, indem Sie eine externe Tabelle anlegen und die Daten aus der Tabelle in der Datenbank auswählen. Die folgende SQL Anweisung erstellt beispielsweise die `orders_xt` externe Tabelle, indem sie die `orders` Tabelle in der Datenbank abfragt.

```
CREATE TABLE orders_xt
  ORGANIZATION EXTERNAL
   (
     TYPE ORACLE_DATAPUMP
     DEFAULT DIRECTORY DATA_PUMP_DIR
     LOCATION ('orders_xt.dmp')
   )
   AS SELECT * FROM orders;
```

In diesem Beispiel werden die Daten in die `orders_xt.dmp` Datei im DIR Verzeichnis DATA _ PUMP _ eingefügt.

Generieren von Leistungsberichten mit Automatic Workload Repository (AWR)

Um Leistungsdaten zu sammeln und Berichte zu generieren, empfiehlt Oracle Automatic Workload Repository (AWR). AWRerfordert Oracle Database Enterprise Edition und eine Lizenz für die Diagnostics and Tuning Packs. Um dies zu aktivierenAWR, setzen Sie den `CONTROL_MANAGEMENT_PACK_ACCESS` Initialisierungsparameter entweder auf `DIAGNOSTIC` oder`DIAGNOSTIC+TUNING`.

Arbeiten mit AWR Berichten in RDS

Um AWR Berichte zu generieren, können Sie Skripte wie ausführen`awrrpt.sql`. Diese Skripte werden auf dem Datenbankhostserver installiert. Bei Amazon RDS haben Sie keinen direkten Zugriff auf den Host. Sie können jedoch Kopien von SQL Skripten aus einer anderen Installation von Oracle Database abrufen.

Sie können dies auch verwenden, AWR indem Sie Prozeduren im `SYS.DBMS_WORKLOAD_REPOSITORY` SQL PL/-Paket ausführen. Sie können dieses Paket zum Verwalten von Baselines und Snapshots sowie zum Anzeigen und Berichten verwenden. ASH AWR Um beispielsweise einen AWR Bericht im Textformat zu generieren, führen Sie das Verfahren aus. `DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT` Sie können diese AWR Berichte jedoch nicht über den erreichen AWS Management Console.

Bei der Arbeit mit AWR empfehlen wir die Verwendung der `rdsadmin.rdsadmin_diagnostic_util` Verfahren. Sie können diese Verfahren verwenden, um Folgendes zu generieren:

- AWRberichtet
- Berichte zum aktiven Sitzungsverlauf (ASH)
- Automatische Berichte des Datenbankdiagnosemonitors (ADDM)
- Oracle Data Pump exportiert Dumpdateien mit Daten AWR

Die `rdsadmin_diagnostic_util`-Verfahren speichern die Berichte im DB-Instance-Dateisystem. Sie können über die Konsole auf diese Berichte zugreifen. Sie können auch mithilfe der `rdsadmin.rds_file_util`-Verfahren auf Berichte zugreifen, und Sie können auf Berichte zugreifen, die mit der Option „S3-Integration“ in Amazon S3 kopiert werden. Weitere Informationen erhalten Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis und Amazon S3-Integration.

Sie können die `rdsadmin_diagnostic_util` Verfahren in den folgenden Versionen der Amazon RDS for Oracle-DB-Engine verwenden:

- Alle Versionen von Oracle Database 21c
- 19.0.0.0.ru-2020-04.rur-2020-04.r1 und höhere Versionen von Oracle Database 19c

Einen Blog, der erklärt, wie Sie mit Diagnoseberichten in einem Replikationsszenario arbeiten, finden Sie unter Generieren von AWR Berichten für Amazon RDS for Oracle-Read Replicas.

Geläufige Parameter für das Diagnose-Utility-Paket

In der Regel verwenden Sie bei der Verwaltung AWR und ADDM Verwendung des `rdsadmin_diagnostic_util` Pakets die folgenden Parameter.

Parameter	Datenty	Standa	Erford ich	Beschreibung
`begin_snap_id`	`NUMBEI`	—	Ja	Die ID des beginnenden Snapshots.
`end_snap_id`	`NUMBEI`	—	Ja	Die ID des abschließenden Snapshots.
`dump_directory`	`VARCHA`	`BDUMI`	Nein	Das Verzeichnis, in das der Bericht oder die Exportdatei geschrieben werden soll. Wenn Sie ein nicht standardmäßiges Verzeichnis angeben, muss der Benutzer, der die `rdsadmin_diagnostic_util` -Verfahren ausführt, über Schreibberechtigungen für das Verzeichnis verfügen.
`p_tag`	`VARCHA`	—	Nein	Eine Zeichenfolge, die verwendet werden kann, um zwischen Backups zu unterscheiden, um den Zweck oder die Verwendung von Backups anzugeben, wie beispielsweise `incremental` oder `daily`. Sie können bis zu 30 Zeichen angeben. Gültige Zeichen sind `a-z`, `A-Z`, `0-9`, ein Unterstrich (_), ein Bindestrich (-) und ein Punkt (.). Bei einem Tag wird die Groß- und Kleinschreibung nicht beachtet. RMANspeichert Tags

Parameter	Datenty	Standa	Erforderlich	Beschreibung
				immer in Großbuchstaben, unabhängig von der Groß- und Kleinschreibung, die bei der Eingabe verwendet wurde. Tags müssen nicht eindeutig sein, daher können mehrere Backups das gleiche Tag haben. Wenn Sie kein Tag angeben, RMAN weist es automatisch ein Standard-Tag unter Verwendung des folgenden `TAG`*`YYYYMMDDTHHMMSS`* Formats zu *`YYYY`* ist das Jahr, *`MM`* ist der Monat, *`DD`* ist der Tag, *`HH`* ist die Stunde (im 24-Stunden-Format), *`MM`* ist die Minute und *`SS`* sind die Sekunden. Datum und Uhrzeit geben an, wann das Backup RMAN gestartet wurde. Ein Backup mit dem Standardtag `TAG20190927T214517` weist beispielsweise auf ein Backup hin, das am 27.09.2019 um 21:45:17 Uhr gestartet wurde. Der `p_tag` Parameter wird für die folgenden RDS Versionen der Oracle-DB-Engine unterstützt: • Oracle Database 21c (21.0.0) • Oracle Datenbank 19c (19.0.0) mit 19.0.0.0.ru-2021-10.rur-2021-10.r1 und höher
`report_type`	`VARCHA`	`HTML`	Nein	Das Format des Berichts. Gültige Werte sind `TEXT` und `HTML`.
`dbid`	`NUMBER`	—	Nein	Eine gültige Datenbank-ID (DBID), die in der `DBA_HIST_DATABASE_INSTANCE` Ansicht für Oracle angezeigt wird. Wenn dieser Parameter nicht angegeben ist, wird der aktuelle Wert RDS verwendetDBID, der in der `V$DATABASE.DBID` Ansicht angezeigt wird.

Bei der Verwaltung ASH mit dem Paket rdsadmin_diagnostic_util verwenden Sie in der Regel die folgenden Parameter.

Parameter	Datentyp	Standa	Erforde ich	Beschreibung
`begin_tim e`	`DATE`	—	Ja	Die ASH Startzeit der Analyse.
`end_time`	`DATE`	—	Ja	Die Endzeit der ASH Analyse.
`slot_widt h`	`NUMBER`	`0`	Nein	Die Dauer der Slots (in Sekunden), die im Abschnitt „Wichtigste Aktivitäten“ des ASH Berichts verwendet wurden. Wenn dieser Parameter nicht angegeben ist, verwendet das Zeitintervall zwischen `begin_time` und `end_time` nicht mehr als 10 Slots.
`sid`	`NUMBER`	Null	Nein	Die Sitzungs-ID.
`sql_id`	`VARCHAR2`	Null	Nein	Die SQL ID.
`wait_clas s`	`VARCHAR2`	Null	Nein	Der Name der Warteklasse.
`service_h ash`	`NUMBER`	Null	Nein	Der Servicenamen-Hash.
`module_na me`	`VARCHAR2`	Null	Nein	Der Modulname.
`action_na me`	`VARCHAR2`	Null	Nein	Die Aktionsname.
`client_id`	`VARCHAR2`	Null	Nein	Die anwendungsspezifische ID der Datenbank sitzung.
`plsql_ent ry`	`VARCHAR2`	Null	Nein	Der SQL PL/Einstiegspunkt.

Einen Bericht generieren AWR

Gehen Sie wie `rdsadmin.rdsadmin_diagnostic_util.awr_report` folgt vor, um einen AWR Bericht zu generieren.

Im folgenden Beispiel wird ein AWR Bericht für den Snapshot-Bereich 101—106 generiert. Die Ausgabetextdatei heißt `awrrpt_101_106.txt`. Sie können auf diesen Bericht in der zugreife AWS Management Console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(101,106,'TEXT');
```

Im folgenden Beispiel wird ein HTML Bericht für den Snapshot-Bereich 63—65 generiert. Die HTML Ausgabedatei ist benannt. `awrrpt_63_65.html` Das Verfahren schreibt den Bericht in das nicht standardmäßige Datenbankverzeichnis namens `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(63,65,'HTML','AWR_RPT_DUMP');
```

Extrahieren von AWR Daten in eine Dump-Datei

Gehen Sie wie folgt vor, um AWR Daten in eine Dump-Datei zu extrahieren. `rdsadmin.rdsadmin_diagnostic_util.awr_extract`

Im folgenden Beispiel wird der Snapshot-Bereich 101–106 extrahiert. Die Ausgabe-Dump-Datei heißt `awrextract_101_106.dmp`. Sie können über die Konsole auf diese Datei zugreifen.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(101,106);
```

Im folgenden Beispiel wird der Snapshot-Bereich 63–65 extrahiert. Die Ausgabe-Dump-Datei heißt `awrextract_63_65.dmp`. Die Datei wird im nicht standardmäßigen Datenbankverzeichnis namens gespeicher `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(63,65,'AWR_RPT_DUMP');
```

Einen Bericht generieren ADDM

Gehen Sie wie `rdsadmin.rdsadmin_diagnostic_util.addm_report` folgt vor, um einen ADDM Bericht zu generieren.

Im folgenden Beispiel wird ein ADDM Bericht für den Snapshot-Bereich 101—106 generiert. Die Ausgabetextdatei heißt `addmrpt_101_106.txt`. Sie können über die Konsole auf den Bericht zugreifen.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(101,106);
```

Im folgenden Beispiel wird ein ADDM Bericht für den Snapshot-Bereich 63—65 generiert. Die Ausgabetextdatei heißt `addmrpt_63_65.txt`. Die Datei wird im nicht standardmäßigen Datenbankverzeichnis namens gespeicher `ADDM_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(63,65,'ADDM_RPT_DUMP');
```

Ein Bericht wird generiert ASH

Gehen Sie wie `rdsadmin.rdsadmin_diagnostic_util.ash_report` folgt vor, um einen ASH Bericht zu generieren.

Im folgenden Beispiel wird ein ASH Bericht generiert, der die Daten von vor 14 Minuten bis zur aktuellen Uhrzeit enthält. Der Name der Ausgabedatei verwendet das Format `ashrpt`*`begin_time`*`end_time`.`txt`, wobei *`begin_time`* und *`end_time`* das Format `YYYYMMDDHH24MISS` verwenden. Sie können über die Konsole auf die Datei zugreifen.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>     SYSDATE-14/1440,
        end_time       =>     SYSDATE,
        report_type    =>     'TEXT');
END;
/
```

Im folgenden Beispiel wird ein ASH Bericht generiert, der die Daten vom 18. November 2019 um 18:07 Uhr bis zum 18. November 2019 um 18:15 Uhr enthält. Der Name des HTML Ausgabeberichts lautet. `ashrpt_20190918180700_20190918181500.html` Der Bericht wird im nicht standardmäßigen Datenbankverzeichnis mit dem Namen gespeicher `AWR_RPT_DUMP`.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>    TO_DATE('2019-09-18 18:07:00', 'YYYY-MM-DD HH24:MI:SS'),
        end_time       =>    TO_DATE('2019-09-18 18:15:00', 'YYYY-MM-DD HH24:MI:SS'),
```

```
        report_type    =>    'html',
        dump_directory =>    'AWR_RPT_DUMP');
END;
/
```

Zugriff auf AWR Berichte von der Konsole aus oder CLI

Um auf AWR Berichte zuzugreifen oder Dumpdateien zu exportieren, können Sie das AWS Management Console oder AWS CLI verwenden. Weitere Informationen finden Sie unter Herunterladen einer Datenbank-Protokolldatei.

Anpassen von Datenbank-Links für die Verwendung mit DB-Instances in einem VPC

Um Oracle-Datenbanklinks mit RDS Amazon-DB-Instances innerhalb derselben Virtual Private Cloud (VPC) oder Peered zu verwendenVPCs, sollten die beiden DB-Instances über eine gültige Route zwischen ihnen verfügen. Überprüfen Sie die gültige Route zwischen den DB-Instances anhand Ihrer VPC Routing-Tabellen und der Netzwerkzugriffskontrollliste (ACL).

Die Sicherheitsgruppe jeder DB-Instance muss den Eintritt und den Austritt von einer zur anderen DB-Instance erlauben. Die Regeln für eingehenden und ausgehenden Datenverkehr können sich auf Sicherheitsgruppen derselben Gruppe VPC oder einer Peering-Gruppe beziehen. VPC Weitere Informationen finden Sie unter Aktualisieren Ihrer Sicherheitsgruppen mit Bezug auf VPC Peering-Sicherheitsgruppen.

Wenn Sie einen benutzerdefinierten DNS Server mit den DHCP Optionssätzen in Ihrem konfiguriert habenVPC, muss Ihr benutzerdefinierter DNS Server in der Lage sein, den Namen des Datenbank-Link-Ziels aufzulösen. Weitere Informationen finden Sie unter Einen benutzerdefinierten DNS Server einrichten.

Weitere Informationen über die Verwendung von Datenbank-Links mit Oracle Data Pump finden Sie unter Importieren mit Oracle Data Pump.

Einrichten der Standardversion für eine DB-Instance

Sie können Datenbankobjekte in einer privaten Umgebung neu definieren, einer so genannten Version. Mit der versionsbasierten Neudefinition können Sie die Datenbankobjekte einer Anwendung mit minimalen Ausfallzeiten aktualisieren.

Sie können die Standardedition einer Amazon RDS Oracle-DB-Instance mithilfe des RDS Amazon-Verfahrens festlegen`rdsadmin.rdsadmin_util.alter_default_edition`.

Im folgenden Beispiel wird die Standardedition für die Amazon RDS Oracle-DB-Instance auf festgelegtRELEASE_V1.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

Im folgenden Beispiel wird die Standardedition für die Amazon RDS Oracle-DB-Instance auf den Oracle-Standard zurückgesetzt.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('ORA$BASE');
```

Weitere Informationen über die versionsbasierte Neudefinition von Oracle finden Sie unter About Editions and Edition-Based Redefinition in der Oracle-Dokumentation.

Aktivieren der Überwachung für dieSYS. AUD$-Tabelle

Verwenden Sie das RDS Amazon-VerfahrenSYS.AUD$, um die Audit-Trail-Tabelle der Datenbank zu aktivierenrdsadmin.rdsadmin_master_util.audit_all_sys_aud_table. Die einzige unterstützte Audit-Eigenschaft ist ALL. Sie können einzelne Anweisungen und Operationen (nicht) prüfen.

Die Aktivierung der Prüfung wird für Oracle DB-Instances unterstützt, die die folgenden Versionen ausführen:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Die Prozedur audit_all_sys_aud_table hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_by_access	Boolean	true	Nein	Auf true festlegen, um BY ACCESS zu prüfen. Auf false festlegen, um BY SESSION zu prüfen.

Die folgende Abfrage gibt die aktuelle Audit-Konfiguration für SYS.AUD$ für eine Datenbank zurück.

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

Die folgenden Befehle aktivieren die Prüfung von `ALL` auf `SYS.AUD$ BY ACCESS`.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;

EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

Der folgende Befehl aktiviert die Prüfung von `ALL` auf `SYS.AUD$ BY SESSION`.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

Weitere Informationen finden Sie unter AUDIT(herkömmliches Auditing) in der Oracle-Dokumentation.

Deaktivierung der Überwachung für dieSYS. AUD$-Tabelle

Verwenden Sie das RDS Amazon-VerfahrenSYS.AUD$, um das Auditing in der Datenbank-Audit-Trail-Tabelle zu deaktivierenrdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table. Diese Prozedur verwendet keine Parameter.

Die folgende Abfrage gibt die aktuelle Audit-Konfiguration für `SYS.AUD$` für eine Datenbank zurück:

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

Der folgende Befehl deaktiviert die Prüfung von `ALL` auf `SYS.AUD$`.

```
EXEC rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

Weitere Informationen finden Sie unter NOAUDIT(herkömmliches Auditing) in der Oracle-Dokumentation.

Bereinigen unterbrochener Online-Index-Builds

Verwenden Sie das RDS Amazon-Verfahren, um fehlgeschlagene Online-Indexerstellungen zu bereinigenrdsadmin.rdsadmin_dbms_repair.online_index_clean.

Die Prozedur `online_index_clean` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`object_id`	binary_integer	`ALL_INDEX_ID`	Nein	Die Objekt-ID des Index. In der Regel können Sie die Objekt-ID aus dem Fehlertext ORA -08104 verwenden.
`wait_for_lock`	binary_integer	`rdsadmin.rdsadmin_dbms_repair.lock_wait`	Nein	Geben Sie `rdsadmin.rdsadmin_dbms_repair.lock_wait` an (Standardeinstellung), um zu versuchen, eine Sperre für das zugrunde liegende Objekt zu erhalten und den Vorgang zu wiederholen, bis ein interner Grenzwert erreicht ist, wenn die Sperre fehlschlägt. Geben Sie `rdsadmin.rdsadmin_dbms_repair.lock_nowait` an, um zu versuchen, eine Sperre für das zugrunde liegende Objekt zu erhalten, den Vorgang jedoch nicht zu wiederholen, wenn die Sperre fehlschlägt.

Das folgende Beispiel bereinigt einen fehlgeschlagenen Online-Index-Build:

```
declare
  is_clean boolean;
begin
  is_clean := rdsadmin.rdsadmin_dbms_repair.online_index_clean(
    object_id     => 1234567890,
    wait_for_lock => rdsadmin.rdsadmin_dbms_repair.lock_nowait
  );
end;
/
```

Weitere Informationen finden Sie unter ONLINE_ INDEX _ CLEAN function in der Oracle-Dokumentation.

Überspringen von beschädigten Blöcken

Zum Überspringen von beschädigten Blöcken während Index- und Tabellenscans verwenden Sie das `rdsadmin.rdsadmin_dbms_repair`-Paket.

Die folgenden Verfahren umschließen die Funktionalität der `sys.dbms_repair.admin_table`-Prozedur und verwenden keine Parameter:

- `rdsadmin.rdsadmin_dbms_repair.create_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table`

Die folgenden Verfahren verwenden dieselben Parameter wie ihre Gegenstücke im `DBMS_REPAIR`-Paket für Oracle-Datenbanken:

- `rdsadmin.rdsadmin_dbms_repair.check_object`
- `rdsadmin.rdsadmin_dbms_repair.dump_orphan_keys`
- `rdsadmin.rdsadmin_dbms_repair.fix_corrupt_blocks`
- `rdsadmin.rdsadmin_dbms_repair.rebuild_freelists`
- `rdsadmin.rdsadmin_dbms_repair.segment_fix_status`
- `rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks`

Weitere Informationen zum Umgang mit Datenbankbeschädigungen finden Sie unter DBMS_REPAIR in der Oracle-Dokumentation.

Example Reaktion auf beschädigte Blöcke

Dieses Beispiel zeigt den grundlegenden Workflow für die Reaktion auf beschädigte Blöcke. Ihre Schritte hängen vom Ort und der Art Ihrer Blockbeschädigung ab.

> **Important**
>
> Bevor Sie versuchen, beschädigte Blöcke zu reparieren, lesen Sie sich die REPAIR Dokumentation zu DBMS_ sorgfältig durch.

So überspringen Sie beschädigte Blöcke bei Index- und Tabellenscans

1. Führen Sie die folgenden Verfahren aus, um Reparaturtabellen zu erstellen, wenn sie noch nicht vorhanden sind.

```
EXEC rdsadmin.rdsadmin_dbms_repair.create_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table;
```

2. Führen Sie die folgenden Verfahren aus, um nach vorhandenen Datensätzen zu suchen und diese ggf. zu löschen.

```
SELECT COUNT(*) FROM SYS.REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.ORPHAN_KEY_TABLE;
SELECT COUNT(*) FROM SYS.DBA_REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.DBA_ORPHAN_KEY_TABLE;

EXEC rdsadmin.rdsadmin_dbms_repair.purge_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table;
```

3. Führen Sie das folgende Verfahren aus, um nach beschädigten Blöcken zu suchen.

```
SET SERVEROUTPUT ON
DECLARE v_num_corrupt INT;
BEGIN
  v_num_corrupt := 0;
  rdsadmin.rdsadmin_dbms_repair.check_object (
    schema_name => '&corruptionOwner',
```

```
    object_name => '&corruptionTable',
    corrupt_count =>  v_num_corrupt
  );
  dbms_output.put_line('number corrupt: '||to_char(v_num_corrupt));
END;
/

COL CORRUPT_DESCRIPTION FORMAT a30
COL REPAIR_DESCRIPTION FORMAT a30

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
       CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION
FROM   SYS.REPAIR_TABLE;

SELECT SKIP_CORRUPT
FROM   DBA_TABLES
WHERE  OWNER = '&corruptionOwner'
AND    TABLE_NAME = '&corruptionTable';
```

4. Führen Sie das Verfahren `skip_corrupt_blocks` aus, um das Überspringen von Beschädigungen für betroffene Tabellen zu aktivieren oder zu deaktivieren. Abhängig von der Situation müssen Sie möglicherweise auch Daten in eine neue Tabelle extrahieren und dann die Tabelle löschen, die den beschädigten Block enthält.

   Führen Sie das folgende Verfahren aus, um das Überspringen von Beschädigungen für betroffene Tabellen zu aktivieren.

   ```
   begin
     rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
       schema_name => '&corruptionOwner',
       object_name => '&corruptionTable',
       object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
       flags => rdsadmin.rdsadmin_dbms_repair.skip_flag);
   end;
   /
   select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
    = '&corruptionTable';
   ```

   Führen Sie das folgende Verfahren aus, um das Überspringen von Beschädigungen zu deaktivieren.

   ```
   begin
   ```

```
  rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
    flags => rdsadmin.rdsadmin_dbms_repair.noskip_flag);
end;
/

select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
 = '&corruptionTable';
```

5. Wenn Sie alle Reparaturarbeiten abgeschlossen haben, führen Sie die folgenden Verfahren aus, um die Reparaturtabellen zu löschen.

```
EXEC rdsadmin.rdsadmin_dbms_repair.drop_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table;
```

Ändern der Größe von Tabellenbereichen, Datendateien und temporären Dateien

Standardmäßig werden Oracle-Tabellenbereiche mit aktivierter automatischen Erweiterbarkeit und ohne Obergrenze für die Größe erstellt. Durch diese Standardeinstellungen können Tabellenbereiche bisweilen zu groß werden. Wir empfehlen, dass Sie eine angemessene Maximalgröße für permanente und temporäre Tabellenräume festlegen, und dass Sie die Speicherverwendung sorgfältig überwachen.

Ändern der Größe von permanenten Tabellenbereichen

Verwenden Sie eines der folgenden RDS Amazon-Verfahren, um die Größe eines permanenten Tablespace in einer RDS Oracle-DB-Instance zu ändern:

- `rdsadmin.rdsadmin_util.resize_datafile`
- `rdsadmin.rdsadmin_util.autoextend_datafile`

Die Prozedur `resize_datafile` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_data_file_id	Zahl	—	Ja	Der Bezeichner der Datendatei, deren Größe geändert werden soll.
p_size	varchar2	—	Ja	Die Größe der Datendatei. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an.

Die Prozedur autoextend_datafile hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_data_file_id	Zahl	—	Ja	Der Bezeichner der Datendatei, deren Größe geändert werden soll.
p_autoextend_state	varchar2	—	Ja	Der Status der automatischen Erweiterungsfunktion. Geben Sie ON an, um die Datendatei automatisch erweitern zu lassen, und OFF, wenn die automatische Erweiterung deaktiviert werden soll.
p_next	varchar2	—	Nein	Die Größe des nächsten Datendateiinkrements. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_maxsize	varchar2	—	Nein	Der maximale Festplatt enspeicher, der für die automatische Erweiteru ng zulässig ist. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an. Sie können UNLIMITED angeben, um die Dateigrößenbeschrä nkung aufzuheben.

Im folgenden Beispiel wird die Größe der Datendatei 4 auf 500 MB geändert.

```
EXEC rdsadmin.rdsadmin_util.resize_datafile(4,'500M');
```

Im folgenden Beispiel wird die automatische Erweiterung für die Datendatei 4 deaktiviert. Außerdem wird die automatische Erweiterung für die Datendatei 5 mit einem Inkrement von 128 MB und ohne maximal zulässige Größe aktiviert.

```
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(4,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(5,'ON','128M','UNLIMITED');
```

Ändern der Größe temporärer Tabellenbereiche

Verwenden Sie eines der folgenden Amazon-Verfahren, um die Größe temporärer Tablespaces in einer RDS Oracle-DB-Instance, einschließlich einer Read Replica, zu ändern: RDS

- rdsadmin.rdsadmin_util.resize_temp_tablespace
- rdsadmin.rdsadmin_util.resize_tempfile
- rdsadmin.rdsadmin_util.autoextend_tempfile

Die Prozedur resize_temp_tablespace hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_temp_tablespace_name	varchar2	—	Ja	Der Name des temporären Tabellenraums für die Größenanpassung.
p_size	varchar2	—	Ja	Die Größe des Tabellenbereichs. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an.

Die Prozedur `resize_tempfile` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_temp_file_id	Zahl	—	Ja	Die Kennung der temporären Datei, deren Größe geändert werden soll.
p_size	varchar2	—	Ja	Die Größe der temporären Datei. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an.

Die Prozedur `autoextend_tempfile` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_temp_file_id	Zahl	—	Ja	Die Kennung der temporären Datei, deren

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				Größe geändert werden soll.
p_autoextend_state	varchar2	—	Ja	Der Status der automatischen Erweiterungsfunktion. Geben Sie ON an, um die temporäre Datei automatisch erweitern zu lassen, und OFF, wenn die automatische Erweiterung deaktiviert werden soll.
p_next	varchar2	—	Nein	Die Größe des nächsten temporären Dateiinkrements. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an.
p_maxsize	varchar2	—	Nein	Der maximale Festplattenspeicher, der für die automatische Erweiterung zulässig ist. Geben Sie die Größe in Byte (Standard), Kilobyte (KB), Megabyte (MB) oder Gigabyte (GB) an. Sie können UNLIMITED angeben, um die Dateigrößenbeschränkung aufzuheben.

In den folgenden Beispielen wird die Größe eines temporären Tabellenbereichs mit dem Namen TEMP auf 4 GB geändert.

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4G');
```

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4096000000');
```

Im folgenden Beispiel wird die Größe eines temporären Tabellenbereichs auf der Grundlage der temporären Datei mit der Datei-ID 1 auf 2 MB geändert.

```
EXEC rdsadmin.rdsadmin_util.resize_tempfile(1,'2M');
```

Im folgenden Beispiel wird die automatische Erweiterung für die temporäre Datei 1 deaktiviert. Außerdem wird die maximale Größe der automatischen Erweiterung der temporären Datei 2 auf 10 GB mit einem Inkrement von 100 MB festgelegt.

```
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(1,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(2,'ON','100M','10G');
```

Weitere Informationen zu Lesereplikaten für Oracle-DB-Instances finden Sie unter Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle.

Bereinigen des Papierkorbs

Wenn Sie eine Tabelle löschen, entfernt Ihre Oracle-Datenbank nicht sofort ihren Speicherplatz. Die Datenbank benennt die Tabelle um und platziert sie und alle zugehörigen Objekte in einem Papierkorb. Durch das Bereinigen des Papierkorbs werden diese Elemente entfernt und der Speicherplatz freigegeben.

Verwenden Sie das RDS Amazon-Verfahren, um den gesamten Papierkorb zu leeren. `rdsadmin.rdsadmin_util.purge_dba_recyclebin` Dieser Vorgang kann jedoch den Papierkorb von SYS- und RDSADMIN-Objekten nicht bereinigen. Wenn Sie diese Objekte löschen müssen, wenden Sie sich an den AWS Support.

Im folgenden Beispiel wird der gesamte Papierkorb bereinigt.

```
EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin;
```

Festlegen der angezeigten Standardwerte für vollständige Redaktion

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val`, um die angezeigten Standardwerte für die vollständige Schwärzung auf Ihrer Amazon RDS Oracle-Instance zu ändern. Beachten Sie, dass Sie mit dem `DBMS_REDACT` SQL PL/-Paket eine Schwärzungsrichtlinie erstellen, wie in der Oracle Database-Dokumentation beschrieben. Das Verfahren `dbms_redact_upd_full_rdct_val` gibt die Zeichen an, die für verschiedene Datentypen angezeigt werden sollen, auf die sich eine bestehende Richtlinie auswirkt.

Die Prozedur `dbms_redact_upd_full_rdct_val` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_number_val`	Zahl	Null	Nein	Ändert den Standardwert für Spalten des Datentyps `NUMBER`.
`p_binfloat_val`	binary_float	Null	Nein	Ändert den Standardwert für Spalten des Datentyps `BINARY_FLOAT` .
`p_bindouble_val`	binary_double	Null	Nein	Ändert den Standardwert für Spalten des Datentyps `BINARY_DOUBLE` .
`p_char_val`	char	Null	Nein	Ändert den Standardwert für Spalten des Datentyps `CHAR`.
`p_varchar_val`	varchar2	Null	Nein	Ändert den Standardwert für Spalten des Datentyps `VARCHAR2`.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_nchar_val	nchar	Null	Nein	Ändert den Standardwert für Spalten des Datentyps NCHAR.
p_nvarchar_val	nvarchar2	Null	Nein	Ändert den Standardwert für Spalten des Datentyps NVARCHAR2 .
p_date_val	date	Null	Nein	Ändert den Standardwert für Spalten des Datentyps DATE.
p_ts_val	Zeitstempel	Null	Nein	Ändert den Standardwert für Spalten des Datentyps TIMESTAMP .
p_tswtz_val	timestamp with time zone	Null	Nein	Ändert den Standardwert für Spalten des Datentyps TIMESTAMP WITH TIME ZONE.
p_blob_val	blob	Null	Nein	Ändert den Standardwert für Spalten des Datentyps BLOB.
p_clob_val	clob	Null	Nein	Ändert den Standardwert für Spalten des Datentyps CLOB.
p_nclob_val	nclob	Null	Nein	Ändert den Standardwert für Spalten des Datentyps NCLOB.

Im folgenden Beispiel wird der standardmäßige redigierte Wert für den Datentyp CHAR auf * geändert:

```
EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(p_char_val => '*');
```

Im folgenden Beispiel wird der standardmäßige redigierte Wert für die Datentypen NUMBER, DATE und CHAR geändert:

```
BEGIN
rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(
    p_number_val=>1,
    p_date_val=>to_date('1900-01-01','YYYY-MM-DD'),
    p_varchar_val=>'X');
END;
/
```

Nachdem Sie die Standardwerte für vollständige Redaktion mit dem Prozess dbms_redact_upd_full_rdct_val geändert haben, starten Sie Ihre DB-Instance neu, damit die Änderung übernommen wird. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Ausführen allgemeiner Protokoll-bezogener Aufgaben für Oracle DB-Instances

Im Folgenden erfahren Sie, wie Sie bestimmte allgemeine DBA-Aufgaben durchführen können, die mit der Protokollierung Ihrer Amazon RDS-DB-Instances in Oracle zusammenhängen. Um eine verwaltete Service-Erfahrung zu bieten, stellt Amazon RDS keinen Shell-Zugriff zu DB-Instances bereit und beschränkt den Zugriff auf bestimmte Systemprozeduren und -tabellen, die erweiterte Sonderrechte erfordern.

Weitere Informationen finden Sie unter Datenbankprotokolldateien von Amazon RDS für Oracle.

Themen

- Einstellen der erzwungenen Protokollierung
- Einstellen der ergänzenden Protokollierung
- Wechseln zwischen Online-Protokolldateien
- Hinzufügen von Online-Redo-Log-Dateien
- Löschen von Online-Redo-Log-Dateien
- Anpassen der Größe von Online-Redo-Log-Dateien
- Beibehaltung von archivierten Redo-Log-Dateien
- Zugriff auf Online- oder archivierte Redo-Protokolle
- Herunterladen von archivierten Redo-Protokolle aus Amazon S3

Einstellen der erzwungenen Protokollierung

Im Modus für erzwungene Protokollierung protokolliert Oracle alle Änderungen in einer Datenbank, außer Änderung in temporären Tabellenräumen und temporären Segmenten (`NOLOGGING` Klauseln werden ignoriert). Weitere Informationen finden Sie unter Specifying FORCE LOGGING Mode in der Oracle-Dokumentation.

Um die erzwungene Protokollierung festzulegen, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.force_logging`. Die Prozedur `force_logging` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Ja	Beschreibung
`p_enable`	Boolean	true	Nein	Setzen Sie diesen Wert auf `true`, um die Datenbank in den Modus für erzwungene Protokollierung zu setzen, oder auf `false`, um Datenbanken aus diesem Modus zu entfernen.

Im folgenden Beispiel wird eine Datenbank in den Modus für erzwungene Protokollierung gesetzt.

```
EXEC rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Einstellen der ergänzenden Protokollierung

Wenn Sie die zusätzliche Protokollierung aktivieren, LogMiner verfügt es über die erforderlichen Informationen, um verkettete Zeilen und geclusterte Tabellen zu unterstützen. Weitere Informationen finden Sie unter Supplemental Logging in der Oracle-Dokumentation.

In Oracle Database ist die ergänzende Protokollierung standardmäßig deaktiviert. Um die erzwungene Protokollierung festzulegen, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.alter_supplemental_logging`. Weitere Informationen über die Verwaltung der Aufbewahrung von archivierten Redo-Log-Dateien für Oracle-DB-Instances in Amazon RDS finden Sie unter Beibehaltung von archivierten Redo-Log-Dateien.

Die Prozedur `alter_supplemental_logging` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_action`	varchar2	—	Ja	`'ADD'`, um ergänzende Protokollierung hinzuzufügen, `'DROP'` um ergänzende Protokollierung zu verwerfen.
`p_type`	varchar2	Null	Nein	Der Typ der ergänzenden Protokollierung. Gültige Werte sind `'ALL'`, `'FOREIGN KEY'`, `'PRIMARY KEY'`, `'UNIQUE'` oder `PROCEDURAL` .

Im folgenden Beispiel wird die ergänzende Protokollierung aktiviert.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD');
end;
/
```

Im folgenden Beispiel wird die ergänzende Protokollierung für alle Spalten mit fester Länge für Maximalgröße aktiviert.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'ALL');
end;
/
```

Im folgenden Beispiel wird die ergänzende Protokollierung für primäre Schlüsselspalten aktiviert.

```
begin
```

```
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'PRIMARY KEY');
end;
/
```

Wechseln zwischen Online-Protokolldateien

Um die Protokolldateien zu wechseln, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.switch_logfile`. Die Prozedur `switch_logfile` hat keine Parameter.

Im folgenden Beispiel wird zwischen Protokolldateien gewechselt.

```
EXEC rdsadmin.rdsadmin_util.switch_logfile;
```

Hinzufügen von Online-Redo-Log-Dateien

Eine Amazon RDS-DB-Instance, die Oracle ausführt, beginnt mit vier Online-Redo-Log-Dateien, jede 128 MB groß. Um weitere Wiederherstellungsprotokolle hinzuzufügen, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.add_logfile`.

Die Prozedur `add_logfile` hat die folgenden Parameter.

> **Note**
>
> Die Parameter schließen sich gegenseitig aus.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`bytes`	positiv	Null	Nein	Die Größe der Protokoll datei in Bytes Verwenden Sie diesen Parameter nur, wenn die Größe des Protokolls unter 2147483648 Byte (2 GiB) liegt. Andernfal

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				ls gibt RDS einen Fehler aus. Verwenden Sie für Log-Größen, die über diesem Bytewert liegen, stattdessen den `p_size` Parameter.
`p_size`	varchar2	—	Ja	Die Größe der Protokoll datei in Kilobyte (K), Megabyte (M) oder Gigabyte (G).

Mit dem folgenden Befehl wird eine Protokolldatei der Größe 100 MB hinzugefügt.

```
EXEC rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

Löschen von Online-Redo-Log-Dateien

Um Wiederherstellungsprotokolle zu löschen, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.drop_logfile`. Die Prozedur `drop_logfile` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`grp`	positiv	—	Ja	Die Gruppennummer des Protokolls

Im folgenden Beispiel wird das Protokoll mit der Gruppennummer 3 verworfen.

```
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

Sie können nur Protokolle verwerfen, die den Status "unbenutzt" oder "inaktiv" haben. Im folgenden Beispiel werden die Stati der Protokolle abgerufen.

```
SELECT GROUP#, STATUS FROM V$LOG;
```

```
GROUP#     STATUS
---------- ----------------
1          CURRENT
2          INACTIVE
3          INACTIVE
4          UNUSED
```

Anpassen der Größe von Online-Redo-Log-Dateien

Eine RDS Amazon-DB-Instance, auf der Oracle ausgeführt wird, beginnt mit vier Online-Redo-Logs mit jeweils 128 MB. Das folgende Beispiel zeigt, wie Sie die RDS Amazon-Verfahren verwenden können, um die Größe Ihrer Protokolle von jeweils 128 MB auf jeweils 512 MB zu ändern.

```
/* Query V$LOG to see the logs.          */
/* You start with 4 logs of 128 MB each. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE


/* Add four new logs that are each 512 MB */

EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);


/* Query V$LOG to see the logs. */
/* Now there are 8 logs.        */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
```

```
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Drop each inactive log using the group number. */

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 1);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 4);


/* Query V$LOG to see the logs. */
/* Now there are 5 logs.        */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  CURRENT
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Switch logs so that group 2 is no longer current. */

EXEC rdsadmin.rdsadmin_util.switch_logfile;


/* Query V$LOG to see the logs.        */
/* Now one of the new logs is current. */

SQL>SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  ACTIVE
5          536870912  CURRENT
```

```
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* If the status of log 2 is still "ACTIVE", issue a checkpoint to clear it to
 "INACTIVE".  */

EXEC rdsadmin.rdsadmin_util.checkpoint;


/* Query V$LOG to see the logs.            */
/* Now the final original log is inactive. */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  INACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


# Drop the final inactive log.

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 2);


/* Query V$LOG to see the logs.    */
/* Now there are four 512 MB logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED
```

Beibehaltung von archivierten Redo-Log-Dateien

Sie können archivierte Redo-Logs lokal auf Ihrer DB-Instance speichern, um sie mit Produkten wie Oracle LogMiner (`DBMS_LOGMNR`) zu verwenden. Nachdem Sie die Redo-Logs aufbewahrt haben, können Sie sie LogMiner zur Analyse der Logs verwenden. Weitere Informationen finden Sie in der LogMiner Oracle-Dokumentation unter Verwendung zur Analyse von Redo-Log-Dateien.

Um archivierte Wiederherstellungsprotokolle zu erhalten, verwenden Sie die Amazon RDS--Prozedur `rdsadmin.rdsadmin_util.set_configuration`. Wenn Sie dieses Verfahren auf einer primären Instance in Oracle Data Guard anwenden, ändert RDS die Einstellung für die Aufbewahrung von Archiv-Logs auf der primären Instance und öffnet Read Replicas, aber nicht auf gemounteten Replicas. RDS speichert die neuesten Archiv-Redo-Logs auf bereitgestellten Replikaten für einen kurzen Zeitraum. RDS löscht automatisch ältere Protokolle, die auf gemountete Replikate heruntergeladen wurden.

Die Prozedur `set_configuration` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`name`	varchar	—	Ja	Der Name für die zu aktualisierende Konfiguration Um die Aufbewahrungszeiten für archivierte Redo-Logs zu ändern, setzen Sie den Namen auf. `archivelog retention hours`
`value`	varchar	—	Ja	Der Wert für die Konfiguration Geben Sie als Wert die Anzahl der Stunden an, für die die Protokolle aufbewahrt werden sollen.

Im folgenden Beispiel werden 24 Stunden an Redo-Logs aufbewahrt.

```
begin
```

```
    rdsadmin.rdsadmin_util.set_configuration(
        name  => 'archivelog retention hours',
        value => '24');
end;
/
commit;
```

> **Note**
>
> Das Commit muss durchgeführt werden, damit die Änderungen wirksam werden.

Um zu sehen, wie lange archivierte Wiederherstellungsprotokolle für Ihre DB-Instance aufbewahrt werden, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.show_configuration`.

Im folgenden Beispiel wird die Protokoll-Aufbewahrungszeit angezeigt.

```
set serveroutput on
EXEC rdsadmin.rdsadmin_util.show_configuration;
```

Der Ausgang zeigt die aktuelle Einstellung für `archivelog retention hours`. Die folgende Ausgabe zeigt, dass archivierte Wiederherstellungsprotokolle 48 Stunden lang aufbewahrt werden.

```
NAME:archivelog retention hours
VALUE:48
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo
 log files are automatically deleted.
```

Da die archivierten Redo-Log-Dateien in Ihrer DB-Instance aufbewahrt werden, stellen Sie sicher, dass Ihre DB-Instance genügend zugewiesenen Speicherplatz für die aufbewahrten Protokolle bietet. Um festzustellen, wie viel Speicherplatz Ihre DB-Instance in den letzten X Stunden belegt hat, können Sie die folgende Abfrage ausführen. Dabei ersetzen Sie X durch die Anzahl von Stunden.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) bytes
  FROM V$ARCHIVED_LOG
 WHERE FIRST_TIME >= SYSDATE-(X/24) AND DEST_ID=1;
```

RDS for Oracle generiert nur dann archivierte Redo-Protokolle, wenn die Backup-Aufbewahrungsdauer Ihrer DB-Instance größer als null ist. Standardmäßig ist der Aufbewahrungszeitraum für Backups größer als null.

Wenn der Aufbewahrungszeitraum für archivierte Protokolle abläuft, entfernt RDS for Oracle die archivierten Redo-Protokolle aus Ihrer DB-Instance. Um die Backup Ihrer DB-Instance zu einem bestimmten Zeitpunkt zu unterstützen, bewahrt Amazon RDS die archivierten Redo-Protokolle außerhalb Ihrer DB-Instance basierend auf dem Aufbewahrungszeitraum für Backups auf. Informationen zum Ändern des Aufbewahrungszeitraums für Backups finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Note**
>
> In einigen Fällen kann es vorkommen, das beim Verwenden von JDBC unter Linux zum Herunterladen von Redo-Logs lange Latenzzeiten auftreten und Verbindungen zurückgesetzt werden. Das Problem wird in solchen Fällen vielleicht durch die Standardeinstellung des Zufallszahlengenerators in Ihrem Java-Client verursacht. Wir empfehlen, dass Sie für die JDBC-Treiber die Verwendung eines blockierungsfreien Zufallszahlengenerators festlegen.

Zugriff auf Online- oder archivierte Redo-Protokolle

Möglicherweise möchten Sie mit externen Tools wie Attunity, Informatica und anderen auf Ihre Online- und archivierten Redo-Log-Dateien für das Mining zugreifen. GoldenGate Gehen Sie folgendermaßen vor, um auf diese Dateien zuzugreifen:

1. Erstellen Sie Verzeichnisobjekte, die schreibgeschützten Zugriff auf die physischen Dateipfade bieten.

   Verwendung von `rdsadmin.rdsadmin_master_util.create_archivelog_dir` und `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`.

2. Lesen Sie die Dateien mit PL/. SQL

   Sie können die Dateien mit SQL PL/ lesen. Weitere Informationen über das Lesen von Dateien in Verzeichnisobjekten finden Sie unter Auflisten von Dateien in einem DB-Instance-Verzeichnis und Lesen von Dateien in einem DB-Instance-Verzeichnis.

Der Zugriff auf Transaktionsprotokolle wird für die folgenden Versionen unterstützt:

- Oracle Database 21c
- Oracle Database 19c

Der folgende Code erstellt Verzeichnisse, die schreibgeschützten Zugriff auf Ihre Online- und archivierte Redo-Log-Dateien bieten:

> **Important**
>
> Dieser Code widerruft das Sonderrecht `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.create_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

Der folgende Code verwirft die Verzeichnisse für Ihre Online- und archivierten Redo-Log-Dateien.

```
EXEC rdsadmin.rdsadmin_master_util.drop_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

Der folgende Code erteilt oder widerruft die Berechtigung `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;
EXEC rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

Herunterladen von archivierten Redo-Protokolle aus Amazon S3

Sie können archivierte Redo-Protokolle auf Ihrer DB-Instance mit dem `rdsadmin.rdsadmin_archive_log_download`-Paket. Wenn sich archivierte Redo-Protkolle nicht mehr auf Ihrer DB-Instance befinden, sollten Sie sie möglicherweise erneut von Amazon S3 herunterladen. Dann können Sie die Protokolle durchsuchen oder sie verwenden, um Ihre Datenbank wiederherzustellen oder zu replizieren.

> **Note**
>
> Sie können keine archivierten Redo-Logs auf Read Replica-Instanzen herunterladen.

Archivierte Redo-Protokolle herunterladen: Grundlegende Schritte

Die Verfügbarkeit Ihrer archivierten Redo-Protokolle hängt von den folgenden Aufbewahrungsrichtlinien ab:

- Backup-Aufbewahrungsrichtlinie – Protokolle innerhalb dieser Richtlinie sind in Amazon S3 verfügbar. Protokolle außerhalb dieser Richtlinie werden entfernt.
- Aufbewahrungsrichtlinie für archivierte Protokolle – Protokolle innerhalb dieser Richtlinie sind auf Ihrer DB-Instance verfügbar. Protokolle außerhalb dieser Richtlinie werden entfernt.

Wenn sich Protokolle nicht auf Ihrer Instance befinden, aber durch Ihre Backup-Aufbewahrungsdauer geschützt sind, verwenden Sie `rdsadmin.rdsadmin_archive_log_download`, um sie erneut herunterzuladen. RDSfür Oracle speichert die Logs im `/rdsdbdata/log/arch` Verzeichnis auf Ihrer DB-Instance.

Herunterladen von archivierten Redo-Protokolle aus Amazon S3

1. Konfigurieren Sie Ihren Aufbewahrungszeitraum, um sicherzustellen, dass Ihre heruntergeladenen archivierten Redo-Protokolle so lange aufbewahrt werden, wie Sie sie benötigen. Stellen Sie sicher, dass Sie die Änderungen `COMMIT`

   RDSbewahrt Ihre heruntergeladenen Protokolle gemäß der Aufbewahrungsrichtlinie für archivierte Protokolle auf, und zwar ab dem Zeitpunkt, zu dem die Protokolle heruntergeladen wurden. Informationen zum Festlegen der Aufbewahrungsrichtlinie finden Sie unter Beibehaltung von archivierten Redo-Log-Dateien.
2. Warten Sie bis zu 5 Minuten, bis die Änderung der Aufbewahrungsrichtlinie für archivierte Protokolle wirksam wird.
3. Laden Sie die archivierten Redo-Protokolle von Amazon S3 mit`rdsadmin.rdsadmin_archive_log_download` herunter.

   Weitere Informationen erhalten Sie unter Herunterladen eines einzelnen archivierten Redo-Protokoll und Herunterladen einer Reihe archivierter Redo-Protokolle.

> **Note**
>
> RDSüberprüft vor dem Herunterladen automatisch den verfügbaren Speicherplatz. Wenn die angeforderten Protokolle einen hohen Anteil an Speicherplatz verbrauchen, erhalten Sie eine Warnung.

4. Bestätigen Sie, dass die Protokolle erfolgreich von Amazon S3 heruntergeladen wurden.

   Sie können den Status Ihrer Download-Aufgabe in einer bdump-Datei anzeigen. Die bdump-Dateien haben den Pfadnamen `/rdsdbdata/log/trace/dbtask-`*`task-id`*`.log`. Im vorherigen Download-Schritt führen Sie eine `SELECT`-Anweisung aus, die die Aufgaben-ID im Datentyp `VARCHAR2` zurückgibt. Weitere Informationen finden Sie in ähnlichen Beispielen in Überwachen des Status einer Dateiübertragung.

Herunterladen eines einzelnen archivierten Redo-Protokoll

Um ein einzelnes archiviertes Redo-Protkoll in das `/rdsdbdata/log/arch`-Verzeichnis herunterzuladen, verwenden Sie `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum`. Dieses Verfahren hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`seqnum`	Zahl	—	Ja	Die Sequenznummer des archivierten Redo-Protokolls.

Im folgenden Beispiel wird das Protokoll mit der Sequenznummer 20 heruntergeladen.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum(seqnum => 20)
       AS TASK_ID
FROM   DUAL;
```

Herunterladen einer Reihe archivierter Redo-Protokolle

Um eine Reihe archivierter Redo-Protokolle in das `/rdsdbdata/log/arch`-Verzeichnis herunterzuladen, verwenden Sie `download_logs_in_seqnum_range`. Ihr Download ist auf 300

Protokolle pro Anfrage beschränkt. Die Prozedur `download_logs_in_seqnum_range` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`start_seq`	Zahl	—	Ja	Die Anfangsse quenznummer für die Reihe.
`end_seq`	Zahl	—	Ja	Die Endsequenznummer für die Reihe.

Im folgenden Beispiel werden die Protokolle von Sequenz 50 auf 100 heruntergeladen.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range(start_seq
 => 50, end_seq => 100)
       AS TASK_ID
FROM   DUAL;
```

Allgemeine RMAN Aufgaben für Oracle-DB-Instances ausführen

Im folgenden Abschnitt erfahren Sie, wie Sie Oracle Recovery Manager (RMAN) DBA -Aufgaben auf Ihren RDS Amazon-DB-Instances ausführen können, auf denen Oracle ausgeführt wird. Um ein Managed-Serviceerlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind.

Verwenden Sie das RDS Amazon-Paket`rdsadmin.rdsadmin_rman_util`, um RMAN Backups Ihrer Amazon RDS for Oracle-Datenbank auf Festplatte durchzuführen. Das `rdsadmin.rdsadmin_rman_util`-Paket unterstützt vollständige und inkrementelle Sicherungen von Datenbankdateien, Tabellenräumen und archivierten Redo-Protokollen.

Nach Abschluss einer RMAN Sicherung können Sie die Sicherungsdateien vom Amazon RDS for Oracle DB-Instance-Host kopieren. Sie können dies tun, um Backups auf einem anderen RDS Server wiederherzustellen oder um Backups langfristig zu speichern. Sie können die Sicherungsdateien beispielsweise in einen Amazon S3-Bucket kopieren. Weitere Informationen finden Sie unter "Verwendung von Amazon S3-Integration.

Die Sicherungsdateien für RMAN Backups verbleiben auf dem RDS Amazon-DB-Instance-Host, bis Sie sie manuell entfernen. Sie können mithilfe des Oracle-Verfahrens `UTL_FILE.FREMOVE` Dateien aus einem Verzeichnis entfernen. Weitere Informationen finden Sie in der Dokumentation zur Oracle-Datenbank unter FREMOVEVerfahren.

Sie können das nicht RMAN RDS zur Wiederherstellung von Oracle-DB-Instances verwenden. Sie können es jedoch verwenden, RMAN um ein Backup auf einer lokalen oder EC2 Amazon-Instance wiederherzustellen. Weitere Informationen finden Sie im Blogartikel Restore an Amazon RDS for Oracle Instance to a Self-Managed Instance.

> **Note**
>
> Zum Sichern und Wiederherstellen auf einer anderen Amazon RDS for Oracle-DB-Instance können Sie weiterhin die RDS Sicherungs- und Wiederherstellungsfunktionen von Amazon verwenden. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

Themen

- Voraussetzungen für RMAN Backups
- Allgemeine Parameter für RMAN Verfahren
- Datenbankdateien RDS für Oracle validieren
- Aktivieren und Deaktivieren der Nachverfolgung von Blockänderungen
- Gegenprüfen archivierter Redo-Logs
- Archivierte Redo-Log-Dateien sichern
- Durchführen einer vollständigen Datenbanksicherung
- Durchführen einer vollständigen Sicherung einer Tenant-Datenbank
- Durchführen einer inkrementellen Datenbanksicherung
- Durchführen einer inkrementellen Sicherung einer Tenant-Datenbank
- Sichern eines Tablespace
- Sichern einer Steuerdatei
- Blockmedienwiederherstellung durchführen

Voraussetzungen für RMAN Backups

Bevor Sie Ihre Datenbank mit dem `rdsadmin.rdsadmin_rman_util`-Paket sichern, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen:

- Stellen Sie sicher, dass RDS sich Ihre Oracle-Datenbank im `ARCHIVELOG` Modus befindet. Um diesen Modus zu aktivieren, legen Sie den Aufbewahrungszeitraum für Backups auf einen Wert ungleich Null fest.
- Wenn Sie archivierte Redo-Protokolle sichern oder ein vollständiges oder inkrementelles Backup durchführen, das archivierte Redo-Protokolle umfasst, muss für die Beibehaltung von Redo-Logs ein Wert ungleich Null festgelegt werden. Archivierte Redo-Protokolle sind erforderlich, um die Datenbankdateien während der Wiederherstellung konsistent zu halten. Weitere Informationen finden Sie unter Beibehaltung von archivierten Redo-Log-Dateien.
- Stellen Sie sicher, dass Ihre DB-Instance über ausreichend freien Speicherplatz für die Backups verfügt. Wenn Sie ein Backup für Ihre Datenbank durchführen, geben Sie als Parameter im Prozeduraufruf ein Oracle-Verzeichnisobjekt an. RMANplatziert die Dateien im angegebenen Verzeichnis. Sie können Standardverzeichnisse, wie z. B. `DATA_PUMP_DIR`, verwenden oder ein neues Verzeichnis erstellen. Weitere Informationen finden Sie unter Erstellen und Löschen von Verzeichnissen im Hauptdatenspeicherbereich.

  Sie können den aktuellen freien Speicherplatz in einer RDS Oracle-Instanz mithilfe der CloudWatch Metrik überwachen`FreeStorageSpace`. Wir empfehlen, dass Ihr freier Speicherplatz die aktuelle Größe der Datenbank überschreitet. Es werden jedoch RMAN nur formatierte Blöcke gesichert und die Komprimierung unterstützt.

Allgemeine Parameter für RMAN Verfahren

Sie können die Verfahren im RDS Amazon-Paket verwenden`rdsadmin.rdsadmin_rman_util`, um Aufgaben auszuführenRMAN. Den Verfahren im Paket sind mehrere Parameter gemeinsam. Das Paket besitzt die folgenden geläufigen Parameter.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
`p_directory_name`	varchar	Ein gültiger Datenbank	—	Ja	Der Name des Verzeichnisses, das die Sicherungsdateien enthalten soll.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
		verzeichn isname.			
`p_label`	varchar	`a-z`, `A-Z`, `0-9`, `'_'`, `'-'`, `'.'`	—	Nein	Eine eindeutige Zeichenfolge, die in die Sicherungsdateinamen eingeschlossen wird. **Note** Das Limit ist 30 Zeichen.
`p_owner`	varchar	Ein gültiger Eigentüme r des in angegeben en Verzeichn isse `p_directo ry_name` .	—	Ja	Der Eigentümer des Verzeichnisses, das die Sicherungsdateien enthalten soll.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
`p_tag`	varchar	`a-z`, `A-Z`, `0-9`, `'_'`, `'-'`, `'.'`	NULL	Nein	Eine Zeichenfolge, die verwendet werden kann, um zwischen Backups zu unterscheiden, um den Zweck oder die Verwendung von Backups anzugeben , wie beispielsweise tägliche, wöchentli che oder inkrementelle Backups. Das Limit ist 30 Zeichen. Bei einem Tag wird die Groß- und Kleinschr eibung nicht beachtet. Tags werden immer in Großbuchstaben gespeichert, unabhängig davon, ob bei der Eingabe Groß- oder Kleinschreibung verwendet wird. Tags müssen nicht eindeutig sein, daher können mehrere Backups das gleiche Tag haben. Wenn Sie kein Tag angeben, RMAN weist es automatisch ein Standard-Tag zu. Dabei wird das folgende Format `TAG`*`YYYYMMDDTHHMMSS`* verwendet *`YYYY`* ist das Jahr, *`MM`* ist der Monat, *`DD`* ist der Tag, *`HH`* ist die Stunde (im 24-Stunden-Format), *`MM`* ist die Minute und *`SS`* sind die Sekunden. Datum und Uhrzeit beziehen sich darauf, wann das Backup RMAN gestartet wurde. Beispielsweise könnte ein Backup das Tag `TAG20190927T214517` für ein Backup erhalten, das am 27.09.2019 um 21:45:17 Uhr gestartet wurde.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
					Der `p_tag` Parameter wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt: • Oracle Database 21c (21.0.0) • Oracle Datenbank 19c (19.0.0) mit 19.0.0.0.ru-2021-10.rur-2021-10.r1 und höher
`p_compress`	boolesc	`TRUE`, `FALSE`	`FALSE`	Nein	Geben Sie `TRUE` an, ob die BASIC Backup-Komprimierung aktiviert werden soll. Geben Sie `FALSE` an, ob die BASIC Backup-Komprimierung deaktiviert werden soll.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
p_include_archive_logs	boolesc	TRUE, FALSE	FALSE	Nein	Geben Sie TRUE an, um archivierte Redo-Logs in das Backup einzuschl ießen. Geben Sie FALSE an, um archivierte Redo-Logs aus dem Backup auszuschl ießen. Wenn Sie archivierte Redo-Logs in das Backup einschließen, legen Sie mit dem Verfahren rdsadmin.rdsadmin_ util.set_configuration als Aufbewahrungszeitraum eine Stunde oder länger fest. Rufen Sie zudem das Verfahren rdsadmin.rdsadmin_ rman_util.crossche ck_archivelog unverzüglich auf, bevor Sie das Backup ausführen . Andernfalls schlägt die Sicherung möglicherweise fehl, da archivierte Redo-Log-Dateien fehlen, die durch die RDS Amazon-Verwaltungsverfahren gelöscht wurden.
p_include_controlf ile	boolesc	TRUE, FALSE	FALSE	Nein	Geben Sie TRUE an, um die Kontrolld atei in das Backup einzuschließen. Geben Sie FALSE an, um die Kontrolld atei aus dem Backup auszuschließen.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
p_optimize	Boolean	TRUE, FALSE	TRUE	Nein	Geben Sie TRUE zum Aktivieren der Sicherungsoptimierung an, wenn archivierte Redo-Logs eingeschlossen sind, um die Sicherungsgröße zu reduzieren. Geben Sie FALSE an, um die Sicherung soptimierung zu deaktivieren.
p_parallel	Zahl	Eine gültige Ganzzahl zwischen 1 und 254 für Oracle Database Enterpris e Edition (EE) 1 für andere Oracle Datenbank versionen	1	Nein	Anzahl von Channels.

Parameter name	Datenty	Zulässige Werte	Standa	Erforde ich	Beschreibung
p_rman_to_dbms_output	Boolean	TRUE, FALSE	FALSE	Nein	WennTRUE, wird die RMAN Ausgabe zusätzlich zu einer Datei im BDUMP Verzeichnis an das DBMS_OUTPUT Paket gesendet. Verwenden Sie in SQL *Plus, SET SERVEROUTPUT ON um die Ausgabe zu sehen. WennFALSE, wird die RMAN Ausgabe nur an eine Datei im BDUMP Verzeichnis gesendet.
p_section_size_mb	Zahl	Eine gültige Ganzzahl	NULL	Nein	Die Abschnittsgröße in Megabyte (MB). Validiert parallel, indem jede Datei in die angegebene Abschnittsgröße aufgeteilt wird. Bei NULL wird der Parameter ignoriert.
p_validation_type	varchar	'PHYSICAL' , 'PHYSICAL+LOGICAL'	'PHYS '	Nein	Der Level der Korruptionserkennung. Geben Sie 'PHYSICAL' an, um auf physikalische Beschädigung zu überprüfen. Ein Beispiel für physische Korruption oder Beschädigung ist ein Block mit einer Diskrepanz zwischen Kopf- und Fußzeile. Geben Sie 'PHYSICAL+LOGICAL' an, um zusätzlich zur physischen Korruption auf logische Inkonsistenzen zu prüfen. Ein Beispiel für eine logische Beschädigung ist ein korrupter Block.

Datenbankdateien RDS für Oracle validieren

Sie können das RDS Amazon-Paket verwenden, `rdsadmin.rdsadmin_rman_util` um Amazon RDS for Oracle-Datenbankdateien wie Datendateien, Tablespaces, Steuerdateien und Serverparameterdateien () SPFILEs zu validieren.

Weitere Informationen zur Validierung finden Sie unter RMAN Validierung von Datenbankdateien und Backups und VALIDATEin der Oracle-Dokumentation.

Themen

- Eine Datenbank validieren
- Validieren einer Tenant-Datenbank
- Validieren eines Tablespaces
- Validieren einer Steuerdatei
- Validieren eines SPFILE
- Eine Oracle-Datendatei validieren

Eine Datenbank validieren

Verwenden Sie das RDS Amazon-Verfahren, um alle relevanten Dateien, die von einer Oracle-Datenbank verwendet werden, RDS für Oracle zu validieren`rdsadmin.rdsadmin_rman_util.validate_database`.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Im folgenden Beispiel wird die Datenbank anhand der Standardwerte für die Parameter validiert.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_database;
```

Im folgenden Beispiel wird die Datenbank anhand der angegebenen Werte für die Parameter validiert.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_database(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Wenn der `p_rman_to_dbms_output` Parameter auf gesetzt ist`FALSE`, wird die RMAN Ausgabe in eine Datei im `BDUMP` Verzeichnis geschrieben.

Um die Dateien im Verzeichnis `BDUMP` anzuzeigen, führen Sie die folgende `SELECT`-Anweisung aus.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Um den Inhalt einer Datei im Verzeichnis `BDUMP` anzuzeigen, führen Sie die folgende `SELECT`-Anweisung aus.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

Ersetzen Sie den Dateinamen durch den Namen der Datei, die Sie anzeigen möchten.

Validieren einer Tenant-Datenbank

Verwenden Sie das RDS Amazon-Verfahren, um die Datendateien der Tenant-Datenbank in einer Container-Datenbank (CDB) zu validieren`rdsadmin.rdsadmin_rman_util.validate_tenant`.

Dieses Verfahren gilt nur für die aktuelle Tenant-Datenbank und verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren. Dieses Verfahren wird für die folgenden DB-Engine-Versionen unterstützt:

- Oracle-Datenbank 21c (21.0.0) CDB
- Oracle-Datenbank 19c (19.0.0) CDB

Im folgenden Beispiel wird die aktuelle Tenant-Datenbank unter Verwendung der Standardwerte für die Parameter validiert.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_tenant;
```

Im folgenden Beispiel wird die aktuelle Tenant-Datenbank unter Verwendung der angegebenen Werte für die Parameter validiert.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_tenant(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Wenn der `p_rman_to_dbms_output` Parameter auf gesetzt ist`FALSE`, wird die RMAN Ausgabe in eine Datei im `BDUMP` Verzeichnis geschrieben.

Um die Dateien im Verzeichnis `BDUMP` anzuzeigen, führen Sie die folgende `SELECT`-Anweisung aus.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Um den Inhalt einer Datei im Verzeichnis `BDUMP` anzuzeigen, führen Sie die folgende `SELECT`-Anweisung aus.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

Ersetzen Sie den Dateinamen durch den Namen der Datei, die Sie anzeigen möchten.

Validieren eines Tablespaces

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.validate_tablespace`, um die mit einem Tablespace verknüpften Dateien zu validieren.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren wird außerdem der folgende zusätzliche Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_tablespace_name`	varchar2	Ein gültiger Tabellenraumname	—	Ja	Der Name des Tabellenraums.

Validieren einer Steuerdatei

Verwenden Sie das RDS Amazon-Verfahren, um nur die von einer Amazon RDS Oracle-DB-Instance verwendete Kontrolldatei zu validieren`rdsadmin.rdsadmin_rman_util.validate_current_controlfile`.

Dieses Verfahren verwendet den folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_validation_type`
- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Validieren eines SPFILE

Verwenden Sie das RDS Amazon-Verfahren, um nur die Serverparameterdatei (SPFILE) zu validieren, die von einer Amazon RDS Oracle-DB-Instance verwendet wird`rdsadmin.rdsadmin_rman_util.validate_spfile`.

Dieses Verfahren verwendet den folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_validation_type`
- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Eine Oracle-Datendatei validieren

Verwenden Sie das RDS Amazon-Verfahren, um eine Datendatei zu validieren`rdsadmin.rdsadmin_rman_util.validate_datafile`.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren werden außerdem die folgenden zusätzlichen Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_datafile`	varchar2	Eine gültige ID-Nummer der Datendatei oder ein	—	Ja	Die ID-Nummer der Datendatei (aus v `$datafile.file#`) oder der vollständige Datendatename einschließlich

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
		gültiger Datendateiname mit vollständigem Pfad			des Pfades (aus v `$datafile.name`).
`p_from_block`	Zahl	Eine gültige Ganzzahl	`NULL`	Nein	Nummer des Blocks, in dem die Validierung innerhalb der Datendatei beginnt. Bei `NULL` wird 1 verwendet.
`p_to_block`	Zahl	Eine gültige Ganzzahl	`NULL`	Nein	Nummer des Blocks, in dem die Validierung innerhalb der Datendatei endet. Bei `NULL` wird der maximale Block in der Datendatei verwendet .

Aktivieren und Deaktivieren der Nachverfolgung von Blockänderungen

Block, der Nachverfolgungs-Datensätze ändert, hat Blöcke in einer Nachverfolgungsdatei geändert. Diese Technik kann die Leistung RMAN inkrementeller Backups verbessern. Weitere Informationen finden Sie unter Using Block Change Tracking to Improve Incremental Backup Performance (Verwenden der Nachverfolgung von Blockänderungen zur Verbesserung der inkrementellen Sicherungs-Performance) in der Oracle Database-Dokumentation.

RMANFunktionen werden in einer Read Replica nicht unterstützt. Im Rahmen Ihrer Hochverfügbarkeitsstrategie können Sie sich jedoch dafür entscheiden, die Blocknachverfolgung mithilfe des Verfahrens `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`

in einem schreibgeschützten Replikat zu aktivieren. Wenn Sie dieses schreibgeschützte Replikat zu einer Quell-DB-Instance heraufstufen, wird die Nachverfolgung von Blockänderungen für die neue Quell-Instance aktiviert. Somit kann Ihre Instance von schnellen inkrementellen Backups profitieren.

Verfahren zur Nachverfolgung von Blockänderungen werden nur in der Enterprise Edition für die folgenden DB-Engine-Versionen unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

> **Note**
>
> In einem Single-Tenant CDB funktionieren die folgenden Operationen, aber kein für den Kunden sichtbarer Mechanismus kann den aktuellen Status der Vorgänge ermitteln. Siehe auch Einschränkungen von RDS für Oracle CDBs.

Verwenden Sie das RDS Amazon-Verfahren, um die Nachverfolgung von Blockänderungen für eine DB-Instance zu aktivieren`rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`.
Sie können die Nachverfolgung von Blockänderungen mithilfe von deaktivier `disable_block_change_tracking`. Diese Verfahren haben keine Parameter.

Um festzustellen, ob die Verfolgung von Blockänderungen für Ihre DB-Instance aktiviert ist, führen Sie die folgenden Abfrage durch.

```
SELECT STATUS, FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

Im folgenden Beispiel wird die Verfolgung von Blockänderungen für eine DB-Instance aktiviert.

```
EXEC rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

Im folgenden Beispiel wird die Verfolgung von Blockänderungen für eine DB-Instance deaktiviert.

```
EXEC rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

Gegenprüfen archivierter Redo-Logs

Sie können archivierte Redo-Logs mit dem Amazon-Verfahren überprüfen. RDS `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog`

Mit diesem Verfahren können Sie archivierte Redo-Protokolle, die in der Kontrolldatei registriert sind, gegenprüfen und abgelaufene Protokolldatensätze auf Wunsch löschen. Wenn ein RMAN Backup erstellt wird, wird ein Datensatz in der Kontrolldatei erstellt. Im Laufe der Zeit vergrößert sich die Kontrolldatei aufgrund dieser Einträge. Es wird empfohlen, abgelaufene Datensätze regelmäßig zu entfernen.

> **Note**
>
> RDSStandard-Amazon-Backups verwenden keine Datensätze RMAN und erstellen daher keine Datensätze in der Kontrolldatei.

Dieses Verfahren verwendet den gemeinsamen Parameter `p_rman_to_dbms_output` für RMAN Aufgaben.

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren wird außerdem der folgende zusätzliche Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_delete_expired`	Boolean	`TRUE`, `FALSE`	`TRUE`	Nein	Bei `TRUE` werden abgelaufene, archivierte Redo-Protokolle aus der Kontrolldatei gelöscht. Bei `FALSE` werden abgelaufene, archivierte Redo-Protokolle in der Kontrolldatei beibehalten.

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Im folgenden Beispiel werden archivierte Redo-Protokoll-Datensätze in der Kontrolldatei als abgelaufen markiert, die Datensätze jedoch nicht gelöscht.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => FALSE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Im folgenden Beispiel werden abgelaufene archivierte Redo-Protokolle aus der Kontrolldatei entfernt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => TRUE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Archivierte Redo-Log-Dateien sichern

Sie können das RDS Amazon-Paket verwenden`rdsadmin.rdsadmin_rman_util`, um archivierte Redo-Logs für eine Amazon RDS Oracle-DB-Instance zu sichern.

Die Verfahren zum Sichern archivierter Redo-Logs werden für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Themen

- Sichern aller archivierten Redo-Logs

- Sichern eines archivierten Redo-Logs aus einem Datumsbereich
- Ein archiviertes Redo-Log aus einem SCN Bereich sichern
- Sichern eines archivierten Redo-Logs aus einem Sequenznummernbereich

Sichern aller archivierten Redo-Logs

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.backup_archivelog_all`, um alle archivierten Redo-Logs für eine Amazon RDS Oracle-DB-Instance zu sichern.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Das folgende Beispiel sichert alle archivierten Redo-Logs für die DB-Instance.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Sichern eines archivierten Redo-Logs aus einem Datumsbereich

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.backup_archivelog_date`, um bestimmte archivierte Redo-Logs für eine Amazon RDS Oracle-DB-Instance durch Angabe eines Datumsbereichs zu sichern. Der Datumsbereich gibt an, welche archivierten Redo-Logs gesichert werden sollen.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren werden außerdem die folgenden zusätzlichen Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_from_date`	date	Ein Datum zwischen dem `start_date` und `next_date` eines archivierten Redo-	—	Ja	Das Anfangsdatum für die archivierten Protokollsicherungen.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
		Logs auf der Festplatte. Der Wert muss kleiner oder gleich dem für angegebenen Wert sei `p_to_date`.			

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
p_to_date	date	Ein Datum zwischen dem start_date und next_date eines archivierten Redo-Logs auf der Festplatte. Der Wert muss größer oder gleich dem für angegebenen Wert sei p_from_date .	—	Ja	Das Enddatum für die archivierten Protokollsicherungen.

Das folgende Beispiel sichert archivierte Redo-Logs im Datumsbereich für die DB-Instance.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
```

```
        p_from_date           => '03/01/2019 00:00:00',
        p_to_date             => '03/02/2019 00:00:00',
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Ein archiviertes Redo-Log aus einem SCN Bereich sichern

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.backup_archivelog_scn`, um bestimmte archivierte Redo-Logs für eine Amazon RDS Oracle-DB-Instance zu sichern, indem Sie einen Bereich für die Systemänderungsnummer (SCN) angeben. Der SCN Bereich gibt an, welche archivierten Redo-Logs gesichert werden sollen.

Bei diesem Verfahren werden die folgenden allgemeinen Parameter für RMAN Aufgaben verwendet:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren werden außerdem die folgenden zusätzlichen Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_from_scn`	Zahl	Ein Teil SCN eines archivier	—	Ja	Der Startpunkt SCN für die archivierten Log-Backups.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
		ten Redo-Logs, das auf der Festplatte vorhanden ist. Der Wert muss kleiner oder gleich dem für angegebenen Wert sei `p_to_scn`.			

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
p_to_scn	Zahl	Ein Teil SCN eines archivierten Redo-Logs, das auf der Festplatte vorhanden ist. Der Wert muss größer oder gleich dem für angegebenen Wert sei p_from_scn.	—	Ja	Das Ende SCN für die archivierten Log-Backups.

Im folgenden Beispiel werden archivierte Redo-Logs im SCN Bereich für die DB-Instance gesichert.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_from_scn             => 1533835,
        p_to_scn               => 1892447,
```

```
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Sichern eines archivierten Redo-Logs aus einem Sequenznummernbereich

Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence`, um bestimmte archivierte Redo-Logs für eine Amazon RDS Oracle-DB-Instance zu sichern, indem Sie einen Sequenznummernbereich angeben. Der Sequenznummernbereich gibt an, welche archivierten Redo-Logs gesichert werden sollen.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren werden außerdem die folgenden zusätzlichen Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_from_sequence`	Zahl	Die Sequenznummer eines archivier	—	Ja	Die Anfangssequenznummer für die archivierten Protokollsicherungen.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
		ten Redo-Logs, das auf der Festplatte vorhanden ist. Der Wert muss kleiner oder gleich dem für angegebenen Wert sei `p_to_sequence`.			

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
p_to_sequence	Zahl	Die Sequenznummer eines archivierten Redo-Logs, das auf der Festplatte vorhanden ist. Der Wert muss größer oder gleich dem für angegebenen Wert sei p_from_sequence .	—	Ja	Die Endsequenznummer für die archivierten Protokollsicherungen.

Das folgende Beispiel sichert archivierte Redo-Logs im Sequenznummernbereich für die DB-Instance.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_from_sequence       => 11160,
```

```
        p_to_sequence         => 11160,
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Durchführen einer vollständigen Datenbanksicherung

Sie können mithilfe des RDS Amazon-Verfahrens eine Sicherung aller in der Sicherung enthaltenen Datenblöcke durchführen`rdsadmin.rdsadmin_rman_util.backup_database_full`.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Im folgenden Beispiel wird ein vollständiges Backup der DB-Instance mit den angegebenen Werten für die Parameter durchgeführt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Durchführen einer vollständigen Sicherung einer Tenant-Datenbank

Sie können eine Sicherung aller Datenblöcke durchführen, die in einer Tenant-Datenbank enthalten sind, in einer Container-Datenbank (CDB). Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.backup_tenant_full`. Dieses Verfahren gilt nur für die aktuelle Datenbanksicherung und verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Das `rdsadmin_rman_util.backup_tenant_full` Verfahren wird für die folgenden RDS Versionen der Oracle-DB-Engine unterstützt:

- Oracle-Datenbank 21c (21.0.0) CDB

- Oracle-Datenbank 19c (19.0.0) CDB

Im folgenden Beispiel wird eine vollständige Sicherung der aktuellen Tenant-Datenbank mit den angegebenen Werten für die Parameter durchgeführt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_TENANT_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Durchführen einer inkrementellen Datenbanksicherung

Sie können mithilfe des RDS Amazon-Verfahrens ein inkrementelles Backup Ihrer DB-Instance durchführen`rdsadmin.rdsadmin_rman_util.backup_database_incremental`.

Weitere Informationen über inkrementelle Sicherungen finden Sie unter Incremental Backups in der Oracle-Dokumentation.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`

- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Bei diesem Verfahren wird außerdem der folgende zusätzliche Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_level`	Zahl	`0`, `1`	`0`	Nein	Geben Sie `0` an, um ein vollständiges inkrementelles Backup zu aktivieren. Geben Sie `1` an, um ein nicht kumulatives inkrementelles Backup zu aktivieren.

Im folgenden Beispiel wird ein inkrementelles Backup der DB-Instance mit den angegebenen Werten für die Parameter durchgeführt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_incremental(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_level                => 1,
        p_parallel             => 4,
        p_section_size_mb      => 10,
        p_tag                  => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
```

```
/
```

Durchführen einer inkrementellen Sicherung einer Tenant-Datenbank

Sie können eine inkrementelle Sicherung der aktuellen Tenant-Datenbank in Ihrem CDB durchführen. Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_rman_util.backup_tenant_incremental`.

Weitere Informationen über inkrementelle Sicherungen finden Sie unter Incremental Backups in der Oracle-Datenbankdokumentation.

Dieses Verfahren gilt nur für die aktuelle Tenant-Datenbank und verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle-Datenbank 21c (21.0.0) CDB
- Oracle-Datenbank 19c (19.0.0) CDB

Bei diesem Verfahren wird außerdem der folgende zusätzliche Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_level`	Zahl	`0`, `1`	`0`	Nein	Geben Sie `0` an, um ein vollständiges inkrementelles Backup zu aktivieren. Geben Sie `1` an, um ein nicht kumulatives inkrementelles Backup zu aktivieren.

Im folgenden Beispiel wird eine inkrementelle Sicherung der aktuellen Tenant-Datenbank mit den angegebenen Werten für die Parameter durchgeführt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_incremental(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_level               => 1,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Sichern eines Tablespace

Sie können einen Tablespace mit dem RDS Amazon-Verfahren `rdsadmin.rdsadmin_rman_util.backup_tablespace` sichern.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`

- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Bei diesem Verfahren wird außerdem der folgende zusätzliche Parameter verwendet.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
p_tablespace_name	varchar2	Ein gültiger Tabellenraumname.	—	Ja	Der Name des zu sichernden Tabellenraums.

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Im folgenden Beispiel wird ein Tabellenraum-Backup mit den angegebenen Werten für die Parameter durchgeführt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tablespace(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tablespace_name     => 'MYTABLESPACE',
```

```
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MYTABLESPACE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Sichern einer Steuerdatei

Sie können eine Kontrolldatei mit dem RDS Amazon-Verfahren sichern`rdsadmin.rdsadmin_rman_util.backup_current_controlfile`.

Dieses Verfahren verwendet die folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Im folgenden Beispiel wird ein Steuerdatei-Backup mit den angegebenen Werten für die Parameter durchgeführt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_current_controlfile(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tag                 => 'CONTROL_FILE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
```

```
/
```

Blockmedienwiederherstellung durchführen

Sie können einzelne Datenblöcke mithilfe der RDS Amazon-Verfahren wiederherstellen, was als Blockmedienwiederherstellung bezeichnet wird`rdsadmin.rdsadmin_rman_util.recover_datafile_block`. Sie können dieses überladene Verfahren verwenden, um entweder einen einzelnen Datenblock oder eine Reihe von Datenblöcken wiederherzustellen.

Dieses Verfahren verwendet den folgenden allgemeinen Parameter für RMAN Aufgaben:

- `p_rman_to_dbms_output`

Weitere Informationen finden Sie unter Allgemeine Parameter für RMAN Verfahren.

Dieses Verfahren verwendet die folgenden zusätzlichen Parameter.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`p_datafile`	`NUMBER`	Eine gültige Datendatei-ID-Nummer.	—	Ja	Die Datendatei, die die beschädigten Blöcke enthält. Geben Sie die Datendatei auf eine der folgenden Arten an: • Die ID-Nummer der Datendatei, die sich in befindet `V$DATAFILE.FILE#` • Der vollständige Name der Datendatei, einschließlich des Pfads, befindet sich in `V$DATAFILE.NAME`

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
p_block	NUMBER	Eine gültige Ganzzahl.	—	Ja	Die Nummer eines einzelnen Blocks, der wiederhergestellt werden soll. Die folgenden Parameter schließen sich gegenseitig aus: • p_block • p_from_block und p_to_block
p_from_block	NUMBER	Eine gültige Ganzzahl.	—	Ja	Die erste Blocknummer in einem Bereich von Blöcken, die wiederhergestellt werden sollen. Die folgenden Parameter schließen sich gegenseitig aus: • p_block • p_from_block und p_to_block

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
p_to_block	NUMBER	Eine gültige Ganzzahl.	—	Ja	Die letzte Blocknummer in einem Bereich von Blöcken, die wiederhergestellt werden sollen. Die folgenden Parameter schließen sich gegenseitig aus: • p_block • p_from_block und p_to_block

Dieses Verfahren wird für die folgenden Versionen der Amazon RDS for Oracle-DB-Engine unterstützt:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Das folgende Beispiel stellt Block 100 in Datendatei 5 wieder her.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_block               => 100,
        p_rman_to_dbms_output => TRUE);
END;
/
```

Im folgenden Beispiel werden die Blöcke 100 bis 150 in der Datendatei 5 wiederhergestellt.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_from_block          => 100,
```

```
        p_to_block            => 150,
        p_rman_to_dbms_output => TRUE);
END;
/
```

Ausführen allgemeiner Planungsaufgaben für Oracle DB-Instances

Einige SYS-eigene Scheduler-Aufträge können den normalen Datenbankbetrieb stören. In solchen Fällen empfiehlt Oracle Support, den Zeitplan zu ändern. Wenn Sie SYS Jobs aktivieren oder deaktivieren müssen, testen Sie den Vorgang an geplanten Jobs in einer Testumgebung, bevor Sie ihn in einer Produktionsumgebung implementieren. Verwenden Sie das RDS Amazon-Paket`rdsadmin.rdsadmin_dbms_scheduler`, um Aufgaben für Oracle Scheduler-Jobs auszuführenSYS, die Eigentum von sind.

Die `rdsadmin.rdsadmin_dbms_scheduler` Verfahren werden für die in der folgenden Tabelle aufgeführten Versionen der Amazon RDS for Oracle-DB-Engine unterstützt. Wenn Sie dieses Paket verwenden, können Sie die in der Tabelle aufgeführten SYS Jobs angeben.

Datenbank version	Jobs sind standardmäßig aktiviert	Jobs sind standardmäßig deaktiviert
Oracle Database 19c	BSLN_MAINTAIN_STATS_JOB CLEANUP_NON_EXIST_OBJ CLEANUP_ONLINE_IND_BUILD CLEANUP_ONLINE_PMO CLEANUP_TAB_IOT_PMO CLEANUP_TRANSIENT_PKG CLEANUP_TRANSIENT_TYPE DRA_REEVALUATE_OPEN_FAILU RES FILE_SIZE_UPD ORA$AUTOTASK_CLEAN PMO_DEFERRED_GIDX_MAINT_JO B PURGE_LOG RSE$CLEAN_RECOVERABLE_SC RIPT SM$CLEAN_AUTO_SPLIT_MERGE	FGR$AUTOPURGE_JOB FILE_WATCHER HM_CREATE_OFFLINE_DICTIONARY LOAD_OPATCH_INVENTORY ORA$PREPLUGIN_BACKUP_JOB XMLDB_NFS_CLEANUP_JOB

Datenbankversion	Jobs sind standardmäßig aktiviert	Jobs sind standardmäßig deaktiviert
Oracle Database 21c	BSLN_MAINTAIN_STATS_JOB CLEANUP_NON_EXIST_OBJ CLEANUP_ONLINE_IND_BUILD CLEANUP_ONLINE_PMO CLEANUP_TAB_IOT_PMO CLEANUP_TRANSIENT_PKG CLEANUP_TRANSIENT_TYPE DRA_REEVALUATE_OPEN_FAILURES FILE_SIZE_UPD ORA$AUTOTASK_CLEAN PMO_DEFERRED_GIDX_MAINT_JOB PURGE_LOG	FGR$AUTOPURGE_JOB FILE_WATCHER HM_CREATE_OFFLINE_DICTIONARY LOAD_OPATCH_INVENTORY ORA$PREPLUGIN_BACKUP_JOB ORA$_ATSK_AUTOSTS XMLDB_NFS_CLEANUP_JOB

Geläufige Parameter für Oracle Scheduler-Prozeduren

Verwenden Sie die Verfahren im RDS Amazon-Paket`rdsadmin.rdsadmin_dbms_scheduler`, um Aufgaben mit Oracle Scheduler auszuführen. Den Verfahren im Paket sind mehrere Parameter gemeinsam. Das Paket besitzt die folgenden geläufigen Parameter.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`name`	varchar2	Die Verfahren sind in der Tabelle unter aufgeführt [Ausführen allgemeiner	—	Ja	Der Name des zu ändernden Jobs.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
		Planungsaufgaben für Oracle DB-Instances			
`attribute`	varchar2	`'REPEAT_INTERVAL'` `_NAME'`	–	Ja	Zu änderndes Attribut. Um das Wiederholungsintervall für den Job zu ändern, geben Sie a `'REPEAT_INTERVAL'` . Um den Zeitplannamen für den Job zu ändern, geben Sie a `'SCHEDULE_NAME'` .
`value`	varchar2	Ein gültiges Zeitintervall oder ein gültiger Zeitplanname, abhängig vom verwendeten Attribut.	–	Ja	Der neue Wert des Attributs.

DBMS_ SCHEDULER Jobs ändern

Sie können die Oracle-Prozedur `dbms_scheduler.set_attribute` verwenden, um bestimmte Komponenten des Oracle Schedulers zu ändern. Weitere Informationen finden Sie unter DBMS_ SCHEDULER und SET_ ATTRIBUTE procedure in der Oracle-Dokumentation.

Wenn Sie mit RDS Amazon-DB-Instances arbeiten, stellen Sie den Schemanamen `SYS` dem Objektnamen voran. Im folgenden Beispiel wird ein Ressourcenplan-Attribut für das Fensterobjekt "Monday" festgelegt.

```
BEGIN
    DBMS_SCHEDULER.SET_ATTRIBUTE(
        name      => 'SYS.MONDAY_WINDOW',
        attribute => 'RESOURCE_PLAN',
        value     => 'resource_plan_1');
END;
/
```

AutoTask Wartungsfenster ändern

Amazon RDS for Oracle-Instances werden mit Standardeinstellungen für Wartungsfenster erstellt. Automatisierte Wartungsaufgaben wie die Erfassung von Optimierungsstatistiken werden in diesen Zeitfenstern ausgeführt. Standardmäßig aktivieren die Wartungsfenster Oracle Database Resource Manager.

Sie können das `DBMS_SCHEDULER`-Paket verwenden, um ein Wartungsfenster zu ändern. Möglicherweise müssen Sie die Einstellungen für Wartungsfenster aus den folgenden Gründen ändern:

- Sie möchten, dass Wartungsaufträge zu einem anderen Zeitpunkt, mit anderen Einstellungen oder gar nicht ausgeführt werden. Sie können beispielsweise die Dauer des Wartungsfensters ändern oder die Wiederholungszeit und das Intervall ändern.
- Sie möchten die Leistungsbeeinträchtigung durch die Aktivierung von Resource Manager während der Wartung vermeiden. Wenn beispielsweise der Standard-Wartungsplan angegeben ist und das Wartungsfenster beginnt, während die Datenbank ausgelastet ist, können Warteereignisse wie `resmgr:cpu quantum` ausgegeben werden. Dieses Warteereignis bezieht sich auf Database Resource Manager. Ihnen stehen folgende Optionen zur Verfügung:
  - Stellen Sie sicher, dass Wartungsfenster außerhalb der Spitzenzeiten für Ihre DB-Instance aktiv sind.

- Deaktivieren Sie den Standard-Wartungsplan, indem Sie für das Attribut `resource_plan` eine leere Zeichenfolge angeben.
- Legen Sie den Parameter `resource_manager_plan` in der Parametergruppe auf `FORCE:` fest. Wenn Ihre Instance die Enterprise Edition verwendet, wird durch diese Einstellung verhindert, dass Database-Resource-Manager-Pläne aktiviert werden.

Ändern Sie die Einstellungen eines Wartungsfensters wie folgt:

1. Stellen Sie mit einem SQL Oracle-Client eine Connect zu Ihrer Datenbank her.
2. Fragen Sie die aktuelle Konfiguration für ein Scheduler-Wartungsfenster ab.

   Im folgenden Beispiel wird die Konfiguration für `MONDAY_WINDOW` abgefragt.

   ```
   SELECT ENABLED, RESOURCE_PLAN, DURATION, REPEAT_INTERVAL
   FROM   DBA_SCHEDULER_WINDOWS
   WHERE  WINDOW_NAME='MONDAY_WINDOW';
   ```

   Die folgende Ausgabe zeigt, dass für das Wartungsfenster die Standardwerte verwendet werden.

   ```
   ENABLED         RESOURCE_PLAN                  DURATION         REPEAT_INTERVAL
   --------------- ------------------------------ ----------------
    ------------------------------
   TRUE            DEFAULT_MAINTENANCE_PLAN       +000 04:00:00
    freq=daily;byday=MON;byhour=22
                                                                    ;byminute=0;
    bysecond=0
   ```

3. Ändern Sie das Wartungsfenster mit dem `DBMS_SCHEDULER`-Paket.

   Im folgenden Beispiel wird der Ressourcenplan auf null festgelegt, damit Resource Manager nicht während des Wartungsfensters ausgeführt wird.

   ```
   BEGIN
     -- disable the window to make changes
     DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);

     -- specify the empty string to use no plan
     DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
    attribute=>'RESOURCE_PLAN', value=>'');
   ```

```
  -- re-enable the window
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Im folgenden Beispiel wird die maximale Dauer des Wartungsfensters auf 2 Stunden eingestellt.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'DURATION', value=>'0 2:00:00');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Im folgenden Beispiel wird das Wiederholungsintervall auf montags 10 Uhr festgelegt.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'REPEAT_INTERVAL',
 value=>'freq=daily;byday=MON;byhour=10;byminute=0;bysecond=0');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Festlegen der Zeitzone für Oracle Scheduler-Aufgaben

Um die Zeitzone für Oracle Scheduler zu ändern, können Sie die Oracle-Prozedur verwende `dbms_scheduler.set_scheduler_attribute`. Weitere Informationen über das `dbms_scheduler` Paket finden Sie unter DBMS_ SCHEDULER und _ SET SCHEDULER_ ATTRIBUTE in der Oracle-Dokumentation.

So ändern Sie die Einstellung für die aktuelle Zeitzone

1. Connect Sie mithilfe eines Clients wie SQL Developer eine Verbindung zur Datenbank her. Weitere Informationen finden Sie unter Herstellen der Verbindung zu Ihrer DB-Instance mit Oracle SQL Developer.

2. Legen Sie die Standardzeitzone wie folgt fest, indem Sie durch Ihre Zeitzone ersetze *time_zone_name*.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'time_zone_name'
  );
END;
/
```

Im folgenden Beispiel ändern Sie die Zeitzone in Asia/Shanghai.

Beginnen Sie, indem Sie die aktuelle Zeitzone abfragen, wie im Folgenden gezeigt.

```
SELECT VALUE FROM DBA_SCHEDULER_GLOBAL_ATTRIBUTE WHERE
 ATTRIBUTE_NAME='DEFAULT_TIMEZONE';
```

Die Ausgabe zeigt, dass die aktuelle ZeitzoneETC/istUTC.

```
VALUE
-------
Etc/UTC
```

Dann stellen Sie die Zeitzone auf „Asien/Shanghai“ ein.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'Asia/Shanghai'
  );
END;
/
```

Weitere Informationen zum Ändern der Systemzeitzone finden Sie unter Oracle-Zeitzone.

Ausschalten von Oracle Scheduler-Jobs, deren Eigentümer SYS

Um einen Oracle Scheduler-Job zu deaktivieren, der dem SYS Benutzer gehört, verwenden Sie das `rdsadmin.rdsadmin_dbms_scheduler.disable` Verfahren.

Diese Vorgehensweise verwendet den allgemeinen Parameter `name` für Oracle Scheduler-Tasks. Weitere Informationen finden Sie unter Geläufige Parameter für Oracle Scheduler-Prozeduren.

Das folgende Beispiel deaktiviert den Oracle Scheduler-Job `SYS.CLEANUP_ONLINE_IND_BUILD`.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.disable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Oracle Scheduler-Jobs einschalten, deren Eigentümer SYS

Gehen Sie wie folgt vor, um einen Oracle Scheduler-Job zu aktivierenSYS, der `rdsadmin.rdsadmin_dbms_scheduler.enable` Eigentümer ist.

Diese Vorgehensweise verwendet den allgemeinen Parameter `name` für Oracle Scheduler-Tasks. Weitere Informationen finden Sie unter Geläufige Parameter für Oracle Scheduler-Prozeduren.

Das folgende Beispiel aktiviert den Oracle Scheduler-Job `SYS.CLEANUP_ONLINE_IND_BUILD`.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.enable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Ändern des Oracle Scheduler-Wiederholungsintervalls für Jobs des Typs CALENDAR

Um das Wiederholungsintervall so zu ändern, dass ein Oracle Scheduler-Job `CALENDAR` vom Typ SYS -Eigentümer geändert wird, verwenden Sie das Verfahren. `rdsadmin.rdsadmin_dbms_scheduler.disable`

Diese Vorgehensweise verwendet die folgenden allgemeinen Parameter für Oracle Scheduler-Tasks:

- `name`
- `attribute`
- `value`

Weitere Informationen finden Sie unter Geläufige Parameter für Oracle Scheduler-Prozeduren.

Das folgende Beispiel ändert das Wiederholungsintervall des Oracle Scheduler-Jobs `SYS.CLEANUP_ONLINE_IND_BUILD`.

```
BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute(
          name      => 'SYS.CLEANUP_ONLINE_IND_BUILD',
          attribute => 'repeat_interval',
          value     => 'freq=daily;byday=FRI,SAT;byhour=20;byminute=0;bysecond=0');
END;
/
```

Ändern des Oracle Scheduler-Wiederholungsintervalls für Jobs des Typs NAMED

Einige Oracle Scheduler-Jobs verwenden einen Zeitplannamen anstelle eines Intervalls. Für diese Art von Jobs müssen Sie im Master-Benutzerschema einen neuen benannten Zeitplan anlegen. Verwenden Sie dazu die standardmäßige Oracle `sys.dbms_scheduler.create_schedule`-Prozedur. Verwenden Sie außerdem die `rdsadmin.rdsadmin_dbms_scheduler.set_attribute procedure`, um dem Job den neuen benannten Zeitplan zuzuweisen.

Diese Vorgehensweise verwendet den folgenden allgemeinen Parameter für Oracle Scheduler-Tasks:

- `name`
- `attribute`
- `value`

Weitere Informationen finden Sie unter Geläufige Parameter für Oracle Scheduler-Prozeduren.

Das folgende Beispiel ändert das Wiederholungsintervall des Oracle Scheduler-Jobs `SYS.BSLN_MAINTAIN_STATS_JOB`.

```
BEGIN
     DBMS_SCHEDULER.CREATE_SCHEDULE (
          schedule_name   => 'rds_master_user.new_schedule',
          start_date      => SYSTIMESTAMP,
          repeat_interval =>
 'freq=daily;byday=MON,TUE,WED,THU,FRI;byhour=0;byminute=0;bysecond=0',
          end_date        => NULL,
          comments        => 'Repeats daily forever');
```

```
END;
/

BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute (
          name      => 'SYS.BSLN_MAINTAIN_STATS_JOB',
          attribute => 'schedule_name',
          value     => 'rds_master_user.new_schedule');
END;
/
```

Deaktivieren von Autocommit für die Erstellung von Oracle-Scheduler-Aufgaben

Wenn `DBMS_SCHEDULER.CREATE_JOB` Oracle-Scheduler-Aufgaben erstellt, werden die Aufgaben sofort erstellt und es wird ein Commit für die Änderungen ausgeführt. Möglicherweise müssen Sie die Erstellung von Oracle-Scheduler-Aufgaben in die Benutzertransaktion integrieren, um Folgendes zu tun:

- Setzen Sie die Oracle-Scheduler-Aufgabe zurück, wenn die Benutzertransaktion zurückgesetzt wird.
- Erstellen Sie die Oracle-Scheduler-Augabe, wenn ein Commit für die Hauptbenutzertransaktion ausgeführt wird.

Sie können die Prozedur `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` verwenden, um dieses Verhalten zu aktivieren. Diese Prozedur verwendet keine Parameter. Sie können dieses Verfahren im Folgenden RDS für Oracle-Releases verwenden:

- 21,0.0.0.ru-2022-07.rur-2022-07.r1 und höhere Versionen
- 19,0.0.0.ru-2022-07.rur-2022-07.r1 und höhere Versionen

Im folgenden Beispiel wird Autocommit für Oracle Scheduler deaktiviert, eine Oracle-Scheduler-Aufgabe erstellt und anschließend die Transaktion zurückgesetzt. Da Autocommit deaktiviert ist, setzt die Datenbank auch die Erstellung der Oracle-Scheduler-Aufgabe zurück.

```
BEGIN
  rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag;
  DBMS_SCHEDULER.CREATE_JOB(job_name   => 'EMPTY_JOB',
                            job_type   => 'PLSQL_BLOCK',
                            job_action => 'begin null; end;',
```

```
                              auto_drop  => false);
  ROLLBACK;
END;
/

PL/SQL procedure successfully completed.

SELECT * FROM DBA_SCHEDULER_JOBS WHERE JOB_NAME='EMPTY_JOB';

no rows selected
```

Diagnose von Problemen mit RDS für Oracle-DB-Instances

Oracle Database enthält eine Infrastruktur für die Fehlerdiagnose, mit der Sie Datenbankprobleme untersuchen können. In der Oracle-Terminologie ist ein Problem ein kritischer Fehler, z. B. ein Codefehler oder eine Datenbeschädigung. Ein Vorfall ist das Auftreten eines Problems. Wenn der gleiche Fehler dreimal auftritt, zeigt die Infrastruktur drei Vorfälle dieses Problems an. Weitere Informationen finden Sie unter Diagnosing and resolving problems in der Oracle Database-Dokumentation.

Das Automatic Diagnostic Repository Command Interpreter (ADRCI) -Hilfsprogramm ist ein Oracle-Befehlszeilentool, mit dem Sie Diagnosedaten verwalten. Sie können dieses Tool beispielsweise verwenden, um Probleme zu untersuchen und Diagnosedaten zu verpacken. Ein Vorfallpaket enthält Diagnosedaten für Vorfälle, die auf ein bestimmtes Problem hinweisen. Sie können ein Vorfallpaket, das als ZIP-Datei implementiert wird, zu Oracle Support hochladen.

Um ein Managed-Serviceerlebnis zu bieten, bietet Amazon RDS keinen Shell-Zugriff aufADRCI. Verwenden Sie das RDS Amazon-Paket, um Diagnoseaufgaben RDS für Ihre Oracle-DB-Instance durchzuführen`rdsadmin.rdsadmin_adrci_util`.

Mithilfe der Funktionen in `rdsadmin_adrci_util` können Sie Probleme und Vorfälle auflisten und verpacken sowie Ablaufverfolgungsdateien anzeigen. Alle Funktionen geben eine Aufgaben-ID zurück. Diese ID ist Teil des Namens der Protokolldatei, die die ADRCI Ausgabe enthält, wie in`dbtask-`*`task_id`*`.log`. Die Protokolldatei befindet sich im BDUMP Verzeichnis. Sie können die Protokolldatei herunterladen, indem Sie das unter beschriebene Verfahren befolgen. Herunterladen einer Datenbank-Protokolldatei

Allgemeine Parameter für Diagnoseverfahren

Verwenden Sie zur Durchführung von Diagnoseaufgaben die Funktionen im RDS Amazon-Paket`rdsadmin.rdsadmin_adrci_util`. Das Paket besitzt die folgenden geläufigen Parameter.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`incident_id`	Zahl	Eine gültige Vorfall-ID oder null	Null	Nein	Wenn der Wert null ist, zeigt die Funktion alle Vorfälle an. Wenn der Wert nicht null ist und eine gültige Vorfall-ID darstellt, zeigt die Funktion den angegebenen Vorfall an.
`problem_id`	Zahl	Eine gültige Problem-ID oder null	Null	Nein	Wenn der Wert null ist, zeigt die Funktion alle Probleme an. Wenn der Wert nicht null ist und eine gültige Problem-ID darstellt, zeigt die Funktion das angegebene Problem an.
`last`	Zahl	Eine gültige Ganzzahl größer als 0 oder null	Null	Nein	Wenn der Wert null ist, zeigt die Funktion maximal 50 Elemente an. Wenn der Wert nicht null ist, zeigt die Funktion die angegebene Anzahl an.

Auflistung von Vorfällen

Verwenden Sie die RDS Amazon-Funktion, um diagnostische Vorfälle für Oracle aufzulistenrdsadmin.rdsadmin_adrci_util.list_adrci_incidents. Sie können Vorfälle im Basis- oder im Detailmodus auflisten. Standardmäßig listet die Funktion die 50 letzten Vorfälle auf.

Diese Funktion verwendet die folgenden allgemeinen Parameter:

- `incident_id`
- `problem_id`
- `last`

Wenn Sie `incident_id` und `problem_id` angeben, wird `problem_id` von `incident_id` überschrieben. Weitere Informationen finden Sie unter Allgemeine Parameter für Diagnoseverfahren.

Diese Funktion verwendet den folgenden zusätzlichen Parameter.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`detail`	Boolean	`TRUE` oder `FALSE`	`FALSE`	Nein	Wenn `TRUE`, listet die Funktion Vorfälle im Detailmodus auf. Wenn `FALSE`, listet die Funktion Vorfälle im Basismodus auf.

Um alle Vorfälle aufzulisten, fragen Sie die Funktion `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` ohne Argumente ab. Die Abfrage gibt die Aufgaben-ID zurück.

```
SQL> SELECT rdsadmin.rdsadmin_adrci_util.list_adrci_incidents AS task_id FROM DUAL;

TASK_ID
------------------
1590786706158-3126
```

Oder rufen Sie die `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` Funktion ohne Argumente auf und speichern Sie die Ausgabe in einer SQL Client-Variablen. Sie können die Variable in anderen Anweisungen verwenden.

```
SQL> VAR task_id VARCHAR2(80);
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_incidents;

PL/SQL procedure successfully completed.
```

Rufen Sie das RDS Amazon-Verfahren auf, um die Protokolldatei zu lesen`rdsadmin.rds_file_util.read_text_file`. Geben Sie die Aufgaben-ID als Teil des Dateinamens an. Die folgende Ausgabe zeigt drei Vorfälle: 53523, 53522 und 53521.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:11:46.193 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:11:46.256 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID PROBLEM_KEY                                                 CREATE_TIME
----------- ------------------------------------------------------------
 ----------------------------------------
53523       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
 20:15:20.928000 +00:00
53522       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 2020-05-29
 20:15:15.247000 +00:00
53521       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 2020-05-29
 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:11:46.256 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:11:46.256 UTC [INFO ] The task finished successfully.

14 rows selected.
```

Um einen bestimmten Vorfall aufzulisten, geben Sie mithilfe des Parameters `incident_id` dessen ID an. Im folgenden Beispiel fragen Sie die Protokolldatei nur für Vorfall 53523 ab.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.list_adrci_incidents(incident_id=>53523);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:15:25.358 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:15:25.426 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID          PROBLEM_KEY
 CREATE_TIME
-------------------- ------------------------------------------------------------
 ---------------------------------
53523                ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003
 2020-05-29 20:15:20.928000 +00:00
1 rows fetched


2020-05-29 21:15:25.427 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:15:25.427 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Probleme mit der Auflistung

Verwenden Sie die RDS Amazon-Funktion, um Diagnoseprobleme für Oracle aufzulisten`rdsadmin.rdsadmin_adrci_util.list_adrci_problems`.

Standardmäßig listet die Funktion die 50 letzten Probleme auf.

Diese Funktion verwendet die allgemeinen Parameter `problem_id` und `last`. Weitere Informationen finden Sie unter Allgemeine Parameter für Diagnoseverfahren.

Um die Aufgaben-ID für alle Probleme abzurufen, rufen Sie die `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` Funktion ohne Argumente auf und speichern Sie die Ausgabe in einer SQL Client-Variablen.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems;

PL/SQL procedure successfully completed.
```

Um die Protokolldatei zu lesen, rufen Sie die Funktion `rdsadmin.rds_file_util.read_text_file` auf und geben Aufgaben-ID als Teil des Dateinamens an. In der folgenden Ausgabe zeigt die Protokolldatei drei Probleme an: 1, 2 und 3.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:18:50.764 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:18:50.829 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
PROBLEM_ID   PROBLEM_KEY                                                 LAST_INCIDENT
       LASTINC_TIME
---------- ---------------------------------------------------------- -------------
 ---------------------------------
2          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 53523
 2020-05-29 20:15:20.928000 +00:00
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 53521
 2020-05-29 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:18:50.829 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:18:50.829 UTC [INFO ] The task finished successfully.

14 rows selected.
```

Im folgenden Beispiel listen Sie nur Problem 3 auf.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems(problem_id=>3);

PL/SQL procedure successfully completed.
```

Um die Protokolldatei für Problem 3 zu lesen, rufen Sie au `rdsadmin.rds_file_util.read_text_file`. Geben Sie die Aufgaben-ID als Teil des Dateinamens an.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:19:42.533 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:19:42.599 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*********************************************************************
PROBLEM_ID PROBLEM_KEY                                                 LAST_INCIDENT
 LASTINC_TIME
---------- ---------------------------------------------------------- -------------
 -------------------------------
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1 rows fetched


2020-05-29 21:19:42.599 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:19:42.599 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Erstellen von Vorfallpaketen

Mit der RDS Amazon-Funktion können Sie Incident-Pakete erstellen`rdsadmin.rdsadmin_adrci_util.create_adrci_package`. Bei der Ausgabe handelt es sich um eine ZIP-Datei, die Sie Oracle Support bereitstellen können.

Diese Funktion verwendet die folgenden allgemeinen Parameter:

- `problem_id`
- `incident_id`

Sie müssen einen der vorhergehenden Parameter angeben. Wenn Sie beide Parameter angeben, überschreibt `incident_id` `problem_id`. Weitere Informationen finden Sie unter Allgemeine Parameter für Diagnoseverfahren.

Um ein Paket für einen bestimmten Vorfall zu erstellen, rufen Sie die RDS Amazon-Funktion `rdsadmin.rdsadmin_adrci_util.create_adrci_package` mit dem `incident_id` Parameter auf. Im folgenden Beispiel wird ein Paket für den Vorfall 53523 erstellt.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.create_adrci_package(incident_id=>53523);

PL/SQL procedure successfully completed.
```

Um die Protokolldatei zu lesen, rufen Sie au `rdsadmin.rds_file_util.read_text_file`. Sie können die Aufgaben-ID als Teil des Dateinamens bereitstellen. Die Ausgabe zeigt, dass Sie das Vorfallpaket generiert habe `ORA700EVE_20200529212043_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:20:43.031 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:20:47.641 UTC [INFO ] Generated package 1 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212043_COM_1.zip, mode complete
2020-05-29 21:20:47.642 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:20:47.642 UTC [INFO ] The task finished successfully.
```

Um Diagnosedaten für ein bestimmtes Problem zu verpacken, geben Sie mithilfe des Parameters `problem_id` dessen ID an. Im folgenden Beispiel verpacken Sie nur für Problem 3 Daten.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.create_adrci_package(problem_id=>3);

PL/SQL procedure successfully completed.
```

Um die Aufgabenausgabe zu lesen, rufen Sie `rdsadmin.rds_file_util.read_text_file` auf und die Aufgaben-ID als Teil des Dateinamens an. Die Ausgabe zeigt, dass Sie das Vorfallpaket generiert habe `ORA700EVE_20200529212111_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
```

```
2020-05-29 21:21:11.050 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:21:15.646 UTC [INFO ] Generated package 2 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212111_COM_1.zip, mode complete
2020-05-29 21:21:15.646 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:21:15.646 UTC [INFO ] The task finished successfully.
```

Sie können die Protokolldatei auch herunterladen. Weitere Informationen finden Sie unter Herunterladen einer Datenbank-Protokolldatei.

Anzeigen von Trace-Dateien

Sie können die RDS Amazon-Funktion verwenden`rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile`, um Trace-Dateien im Trace-Verzeichnis und alle Incident-Verzeichnisse unter dem aktuellen ADR Home aufzulisten. Außerdem können Sie die Inhalte von Trace-Dateien und Incident-Trace-Dateien anzeigen.

Diese Funktion verwendet den folgenden Parameter.

Parametername	Datentyp	Zulässige Werte	Standard	Erforderlich	Beschreibung
`filename`	varchar2	Ein gültiger Name für eine Trace-Datei	Null	Nein	Wenn der Wert null ist, zeigt die Funktion alle Trace-Dateien an. Wenn er nicht null ist, zeigt die Funktion die angegebene Datei an.

Rufen Sie die RDS Amazon-Funktion auf, um die Trace-Datei anzuzeigen`rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile`.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile;

PL/SQL procedure successfully completed.
```

Um die Namen der Trace-Dateien aufzulisten, rufen Sie das RDS Amazon-Verfahren `rdsadmin.rds_file_util.read_text_file` auf und geben Sie die Aufgaben-ID als Teil des Dateinamens an.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE TEXT LIKE '%/alert_%';

TEXT
--------------------------------------------------------------
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-28
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-27
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-26
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-25
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-24
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-23
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-22
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-21
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log

9 rows selected.
```

Im folgenden Beispiel generieren Sie die Ausgabe für`alert_ORCL.log`.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile('diag/rdbms/
orcl_a/ORCL/trace/alert_ORCL.log');

PL/SQL procedure successfully completed.
```

Um die Protokolldatei zu lesen, rufen Sie au `rdsadmin.rds_file_util.read_text_file`. Geben Sie die Aufgaben-ID als Teil des Dateinamens an. Die Ausgabe zeigt die ersten 10 Zeilen von alert_ ORCL .log.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE ROWNUM <= 10;

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:24:02.083 UTC [INFO ] The trace files are being displayed.
2020-05-29 21:24:02.128 UTC [INFO ] Thu May 28 23:59:10 2020
Thread 1 advanced to log sequence 2048 (LGWR switch)
  Current log# 3 seq# 2048 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_3_hbl2p8xs_.log
Thu May 28 23:59:10 2020
Archived Log entry 2037 added for thread 1 sequence 2047 ID 0x5d62ce43 dest 1:
Fri May 29 00:04:10 2020
Thread 1 advanced to log sequence 2049 (LGWR switch)
  Current log# 4 seq# 2049 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_4_hbl2qgmh_.log
```

```
Fri May 29 00:04:10 2020

10 rows selected.
```

Sie können die Protokolldatei auch herunterladen. Weitere Informationen finden Sie unter Herunterladen einer Datenbank-Protokolldatei.

Ausführen verschiedener Aufgaben für Oracle-DB-Instances

Im Folgenden erfahren Sie, wie Sie verschiedene DBA-Aufgaben für Ihre Amazon RDS-DB-Instances, auf denen Oracle ausgeführt wird, durchführen können. Um eine verwaltete Service-Erfahrung zu bieten, stellt Amazon RDS keinen Shell-Zugriff zu DB-Instances bereit und beschränkt den Zugriff auf bestimmte Systemprozeduren und -tabellen, die erweiterte Sonderrechte erfordern.

Themen

- Erstellen und Löschen von Verzeichnissen im Hauptdatenspeicherbereich
- Auflisten von Dateien in einem DB-Instance-Verzeichnis
- Lesen von Dateien in einem DB-Instance-Verzeichnis
- Zugreifen auf Opatch-Dateien
- Verwalten von Berateraufgaben
- Transport von Tabellenbereichen

Erstellen und Löschen von Verzeichnissen im Hauptdatenspeicherbereich

Um die Dateien in einem Verzeichnis aufzulisten, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.create_directory`. Sie können bis zu 10 000 Verzeichnisse erstellen, die sich alle in Ihrem Hauptdatenspeicherplatz befinden. Um Verzeichnisse zu entfernen, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_util.drop_directory`.

Die Prozeduren `create_directory` und `drop_directory` haben den folgenden erforderlichen Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_directory_name`	VARCHAR2	—	Ja	Der Name des Verzeichnisses

Im folgenden Beispiel wird ein neues Verzeichnis mit dem Namen erstell PRODUCT_DESCRIPTIONS.

```
EXEC rdsadmin.rdsadmin_util.create_directory(p_directory_name =>
 'product_descriptions');
```

Das Datenwörterbuch speichert den Verzeichnisnamen in Großbuchstaben. Sie können die Verzeichnisse durch das Abfragen von auflisten lasse DBA_DIRECTORIES. Das System wählt den tatsächlichen Host-Pfadnamen automatisch aus. Im folgenden Beispiel wird der Verzeichnispfad für das Verzeichnis mit dem Namen erhalte PRODUCT_DESCRIPTIONS:

```
SELECT DIRECTORY_PATH
  FROM DBA_DIRECTORIES
 WHERE DIRECTORY_NAME='PRODUCT_DESCRIPTIONS';

DIRECTORY_PATH
----------------------------------------
/rdsdbdata/userdirs/01
```

Der Hauptbenutzername für die DB-Instance hat Lese- und Schreibsonderrechte im neuen Verzeichnis und kann anderen Benutzern Zugriff gewähren. EXECUTE-Sonderrechte sind für die Verzeichnisse in einer DB-Instance nicht verfügbar. Verzeichnisse werden im Hautdatenspeicherplatz erstellt und verbrauchen Speicher und I/O-Bandbreite.

Im folgenden Beispiel wird das Verzeichnis mit dem Namen „“ entfern PRODUCT_DESCRIPTIONS.

```
EXEC rdsadmin.rdsadmin_util.drop_directory(p_directory_name => 'product_descriptions');
```

> **Note**
>
> Sie können ein Verzeichnis auch löschen, indem Sie den Oracle SQL-Befehl verwende DROP DIRECTORY.

Das Verwerfen eines Verzeichnisses entfernt nicht seine Inhalte. Da die Prozedur „rdsadmin.rdsadmin_util.create_directory“ Pfadnamen wiederverwenden kann, werden in Verzeichnissen verworfene Dateien in einem neu erstellten Verzeichnis wieder auftauchen. Bevor Sie ein Verzeichnis löschen, wird empfohlen, Dateien mit UTL_FILE.FREMOVE aus dem Verzeichnis zu entfernen. Weitere Informationen finden Sie unter FREMOVE Procedure in der Oracle-Dokumentation.

Auflisten von Dateien in einem DB-Instance-Verzeichnis

Um die Dateien in einem Verzeichnis aufzulisten, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rds_file_util.listdir`. Dieses Verfahren wird auf einem Oracle-Replikat nicht unterstützt. Die Prozedur `listdir` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_directory`	varchar2	—	Ja	Der Name des aufzulist enden Verzeichnisses

Im folgenden Beispiel werden Lese-/Schreibberechtigungen für das Verzeichnis `PRODUCT_DESCRIPTIONS` an den Benutzer `rdsadmin` gewährt. Anschließend werden die Dateien in diesem Verzeichnis aufgelistet.

```
GRANT READ,WRITE ON DIRECTORY PRODUCT_DESCRIPTIONS TO rdsadmin;
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
 'PRODUCT_DESCRIPTIONS'));
```

Lesen von Dateien in einem DB-Instance-Verzeichnis

Um eine Textdatei zu lesen, verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rds_file_util.read_text_file`. Die Prozedur `read_text_file` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_directory`	varchar2	—	Ja	Der Verzeichnisname der die Datei beinhaltet
`p_filename`	varchar2	—	Ja	Der Name der zu lesenden Datei

Im folgenden Beispiel wird die Datei `rice.txt` im Verzeichnis `PRODUCT_DESCRIPTIONS` erstellt.

```
declare
  fh sys.utl_file.file_type;
```

```
begin
  fh := utl_file.fopen(location=>'PRODUCT_DESCRIPTIONS', filename=>'rice.txt',
 open_mode=>'w');
  utl_file.put(file=>fh, buffer=>'AnyCompany brown rice, 15 lbs');
  utl_file.fclose(file=>fh);
end;
/
```

Im folgenden Beispiel wird die Datei `rice.txt` aus dem Verzeichnis `PRODUCT_DESCRIPTIONS` gelesen.

```
SELECT * FROM TABLE
    (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'PRODUCT_DESCRIPTIONS',
        p_filename  => 'rice.txt'));
```

Zugreifen auf Opatch-Dateien

Opatch ist ein Oracle-Dienstprogramm, das die Anwendung und das Rollback von Patches auf Oracle-Software ermöglicht. Die Oracle-Methode, mit der bestimmt wird, welche Patches auf eine Datenbank angewendet wurden, ist der Befehl `opatch lsinventory`. Um Serviceanfragen für Kunden mit Bring Your Own License (BYOL) zu öffnen, fordert Oracle Support die `lsinventory`-Datei und manchmal die von Opatch generierte `lsinventory_detail`-Datei an.

Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Shell-Zugriff auf Opatch. Stattdessen enthält die Datei `lsinventory-`*`dbv`*`.txt` im BDUMP-Verzeichnis die Patch-Informationen zu Ihrer aktuellen Engine-Version. Wenn Sie ein Upgrade der Neben- oder Hauptversion durchführen, aktualisiert Amazon RDS `lsinventory-`*`dbv`*`.txt` innerhalb einer Stunde nach Patch-Anwendung. Informationen zur Überprüfung der angewendeten Patches finden Sie in der `lsinventory-`*`dbv`*`.txt`. Diese Aktion entspricht der Ausführung des Befehls `opatch lsinventory`.

> **Note**
>
> In den Beispielen in diesem Abschnitt wird davon ausgegangen, dass das BDUMP-Verzeichnis als benannt is `BDUMP`. Bei einem Lesereplikat unterscheidet sich der Name des BDUMP-Verzeichnisses. Informationen zum Abrufen des BDUMP-Namens durch Abfragen von `V$DATABASE.DB_UNIQUE_NAME` auf einem Lesereplikat finden Sie unter Auflisten von Dateien.

Die Bestandsdateien verwenden die Amazon RDS-Namenskonvention `lsinventory-`*`dbv`*`.txt` und `lsinventory_detail-`*`dbv`*`.txt`, wobei *`dbv`* der vollständige Name Ihrer DB-Version ist. Die Datei `lsinventory-`*`dbv`*`.txt` ist auf allen DB-Versionen verfügbar. Das entsprechende `lsinventory_detail-`*`dbv`*`.txt` ist auf 19.0.0.0, ru-2020-01.rur-2020-01.r1 oder höher verfügbar.

Wenn Ihre DB-Version beispielsweise 19.0.0.0.ru-2021-07.rur-2021-07.r1 ist, haben die Bestandsdateien die folgenden Namen.

```
lsinventory-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
lsinventory_detail-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
```

Stellen Sie sicher, dass Sie die Dateien herunterladen, die mit der aktuellen Version Ihrer DB-Engine übereinstimmen.

Konsole

So laden Sie eine Bestandsdatei über die Konsole herunter

1. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie den Namen der DB-Instance, welche die anzuzeigende Protokolldatei enthält.
4. Wählen Sie die Registerkarte Logs & events (Protokolle und Ereignisse).
5. Scrollen Sie nach unten bis zum Abschnitt Protokolle.
6. Suchen Sie im Abschnitt Protokolle nach `lsinventory`.
7. Wählen Sie die Datei aus, auf die Sie zugreifen möchten, und klicken Sie dann auf Herunterladen.

SQL

Zum Lesen der `lsinventory-`*`dbv`*`.txt` auf einem SQL-Client können Sie eine `SELECT`-Anweisung verwenden. Für diese Methode verwenden Sie eine der folgenden `rdsadmin`-Funktionen: `rdsadmin.rds_file_util.read_text_file` oder `rdsadmin.tracefile_listing`.

Ersetzen Sie in der folgenden Beispielabfrage *`dbv`* durch Ihre Oracle DB-Version. Ihre DB-Version könnte beispielsweise 19.0.0.0.ru-2020-04.rur-2020-04.r1 sein.

```
SELECT text
```

```
FROM   TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'lsinventory-dbv.txt'));
```

PL/SQL

Um die `lsinventory-dbv.txt` auf einem SQL-Client zu lesen, können Sie ein PL/SQL-Programm schreiben. Dieses Programm verwendet `utl_file`, um die Datei zu lesen, und `dbms_output`, um sie zu drucken. Dies sind von Oracle bereitgestellte Pakete.

Ersetzen Sie im folgenden Beispielprogramm *dbv* durch Ihre Oracle DB-Version. Ihre DB-Version könnte beispielsweise 19.0.0.0.ru-2020-04.rur-2020-04.r1 sein.

```
SET SERVEROUTPUT ON
DECLARE
  v_file              SYS.UTL_FILE.FILE_TYPE;
  v_line              VARCHAR2(1000);
  v_oracle_home_type  VARCHAR2(1000);
  c_directory         VARCHAR2(30) := 'BDUMP';
  c_output_file       VARCHAR2(30) := 'lsinventory-dbv.txt';
BEGIN
  v_file := SYS.UTL_FILE.FOPEN(c_directory, c_output_file, 'r');
  LOOP
    BEGIN
      SYS.UTL_FILE.GET_LINE(v_file, v_line,1000);
      DBMS_OUTPUT.PUT_LINE(v_line);
    EXCEPTION
      WHEN no_data_found THEN
        EXIT;
    END;
  END LOOP;
END;
/
```

Oder fragen Sie `rdsadmin.tracefile_listing` ab und spoolen Sie die Ausgabe in eine Datei. Im folgenden Beispiel wird die Ausgabe in gespool `/tmp/tracefile.txt`.

```
SPOOL /tmp/tracefile.txt
SELECT *
FROM   rdsadmin.tracefile_listing
WHERE  FILENAME LIKE 'lsinventory%';
SPOOL OFF;
```

Verwalten von Berateraufgaben

Oracle Database enthält eine Reihe von Beratern. Jeder Berater unterstützt automatisierte und manuelle Aufgaben. Sie können Prozeduren im `rdsadmin.rdsadmin_util`-Paket zur Verwaltung einiger Berateraufgaben verwenden.

Die Prozesse der Berateraufgaben sind in den folgenden Engine-Versionen verfügbar:

- Oracle Database 21c (21.0.0)
- Version 19.0.0.0.ru-2021-01.rur-2021-01.r1 und höhere Versionen von Oracle Database 19c

  Weitere Informationen finden Sie unter Version 19.0.0.0.ru-2021-01.rur-2021-01.r1 in den Versionshinweisen zu Amazon RDS for Oracle.

Themen

- Festlegen von Parametern für Berateraufgaben
- Deaktivieren von AUTO_STATS_ADVISOR_TASK
- Erneutes Aktivieren von AUTO_STATS_ADVISOR_TASK

Festlegen von Parametern für Berateraufgaben

Um Parameter für einige Berateraufgaben festzulegen, verwenden Sie das Amazon RDS-Verfahren `rdsadmin.rdsadmin_util.advisor_task_set_parameter`. Die Prozedur `advisor_task_set_parameter` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_task_name`	varchar2	—	Ja	Der Name der Berateraufgabe, deren Parameter Sie ändern möchten. Die folgenden Werte sind gültig: • `AUTO_STATS_ADVISOR_TASK` • `INDIVIDUAL_STATS_ADVISOR_TASK` • `SYS_AUTO_SPM_EVOLVE_TASK` • `SYS_AUTO_SQL_TUNING_TASK`

Parametername	Datentyp	Standard	Erforderlich	Beschreibung

p_parameter	varchar2	—	Ja	Der Name des Aufgaben-Parameters. Um gültige Parameter für eine Beraterаufgabe zu finden, führen Sie die folgende Abfrage aus. Ersetzen Sie *p_task_name* mit einem gültigen Wert für p_task_name :
				`COL PARAMETER_NAME FORMAT a30`
				`COL PARAMETER_VALUE FORMAT a30`
				`SELECT PARAMETER_NAME, PARAMETER_VALUE`
				`FROM DBA_ADVISOR_PARAMETERS`
				`WHERE TASK_NAME='` *p_task_name* `'`
				`AND PARAMETER_VALUE != 'UNUSED'`
				`ORDER BY PARAMETER_NAME;`

p_value	varchar2	—	Ja	Der Wert für einen Aufgabenparameter. Um gültige Werte für Aufgabenparameter zu finden, führen Sie die folgende Abfrage aus. Ersetzen Sie *p_task_name* mit einem gültigen Wert für p_task_name :
				`COL PARAMETER_NAME FORMAT a30`
				`COL PARAMETER_VALUE FORMAT a30`
				`SELECT PARAMETER_NAME, PARAMETER_VALUE`
				`FROM DBA_ADVISOR_PARAMETERS`
				`WHERE TASK_NAME='` *p_task_name* `'`
				`AND PARAMETER_VALUE != 'UNUSED'`
				`ORDER BY PARAMETER_NAME;`

Das folgenden PL/SQL-Programm setzt ACCEPT_PLANS auf FALSE für SYS_AUTO_SPM_EVOLVE_TASK. Die automatisierte Aufgabe „SQL-Planverwaltung“ überprüft die Pläne und erstellt einen Bericht über ihre Ergebnisse, entwickelt die Pläne jedoch nicht automatisch weiter. Sie können einen Bericht verwenden, um neue SQL-Plan-Baselines zu identifizieren und sie manuell zu akzeptieren.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
    p_parameter => 'ACCEPT_PLANS',
    p_value     => 'FALSE');
END;
```

Das folgenden PL/SQL-Programm setzt `EXECUTION_DAYS_TO_EXPIRE` auf `10` für `AUTO_STATS_ADVISOR_TASK`. Die vordefinierte Aufgabe `AUTO_STATS_ADVISOR_TASK` läuft automatisch einmal täglich im Wartungsfenster. Im Beispiel wird der Aufbewahrungszeitraum für die Aufgabenausführung auf 10 Tage festgelegt.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'AUTO_STATS_ADVISOR_TASK',
    p_parameter => 'EXECUTION_DAYS_TO_EXPIRE',
    p_value     => '10');
END;
```

Deaktivieren von AUTO_STATS_ADVISOR_TASK

Um `AUTO_STATS_ADVISOR_TASK` zu deaktivieren, verwenden Sie das Amazon RDS-Verfahren `rdsadmin.rdsadmin_util.advisor_task_drop`. Das `advisor_task_drop`-Verfahren akzeptiert den folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_task_name`	varchar2	—	Ja	Der Name der Berateraufgabe, die deaktiviert werden soll. Der einzige gültige Wert ist `AUTO_STATS_ADVISOR_TASK` .

Der folgenden Befehl wird verworfen: `AUTO_STATS_ADVISOR_TASK`.

```
EXEC rdsadmin.rdsadmin_util.advisor_task_drop('AUTO_STATS_ADVISOR_TASK')
```

Sie können `AUTO_STATS_ADVISOR_TASK` unter Verwendung von `rdsadmin.rdsadmin_util.dbms_stats_init` erneut aktivieren.

Erneutes Aktivieren von AUTO_STATS_ADVISOR_TASK

Um `AUTO_STATS_ADVISOR_TASK` wieder zu aktivieren, verwenden Sie das Amazon RDS-Verfahren `rdsadmin.rdsadmin_util.dbms_stats_init`. Die `dbms_stats_init`-Prozedur verwendet keine Parameter.

Der folgende Befehl reaktiviert `AUTO_STATS_ADVISOR_TASK`.

```
EXEC rdsadmin.rdsadmin_util.dbms_stats_init()
```

Transport von Tabellenbereichen

Verwenden Sie das RDS Amazon-Paket`rdsadmin.rdsadmin_transport_util`, um eine Reihe von Tablespaces aus einer lokalen Oracle-Datenbank in eine Oracle-DB-Instance RDS zu kopieren. Auf physischer Ebene kopiert die Funktion für transportable Tabellenbereiche inkrementell Quelldatendateien und Metadatendateien auf Ihre Ziel-Instance. Sie können die Dateien entweder mit Amazon EFS oder Amazon S3 übertragen. Weitere Informationen finden Sie unter Migrieren mithilfe von Oracle Transportable Tablespaces.

Themen

- Importieren transportabler Tabellenbereiche in Ihre DB-Instance
- Importieren von Metadaten transportabler Tabellenbereiche in Ihre DB-Instance
- Auflisten verwaister Dateien nach einem Tabellenbereichimport
- Löschen verwaister Dateien nach einem Tabellenbereichimport

Importieren transportabler Tabellenbereiche in Ihre DB-Instance

Verwenden Sie das Verfahren `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, um Tabellenbereiche wiederherzustellen, die Sie zuvor aus einer Quell-DB-Instance exportiert haben. In der Transportphase sichern Sie Ihre schreibgeschützten Tabellenbereiche und exportieren Data-Pump-Metadaten, übertragen diese Dateien auf Ihre Ziel-DB-Instance und importieren dann die Tabellenbereiche. Weitere Informationen finden Sie unter Phase 4: Transportieren der Tabellenbereiche.

Syntax

```
FUNCTION import_xtts_tablespaces(
```

```
    p_tablespace_list IN CLOB,
    p_directory_name  IN VARCHAR2,
    p_platform_id     IN NUMBER DEFAULT 13,
    p_parallel        IN INTEGER DEFAULT 0) RETURN VARCHAR2;
```

Parameter

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_tablespace_list	CLOB	—	Ja	Die Liste der zu importier enden Tabellenbereiche.
p_directory_name	VARCHAR2	—	Ja	Das Verzeichnis, das die Backups der Tabellenb ereiche enthält.
p_platform_id	NUMBER	13	Nein	Geben Sie eine Plattform-ID an, die mit der in der Backup-Phase angegeben en ID übereinst immt. Fragen Sie V $TRANSPORTABLE_PL ATFORM ab, um eine Liste der Plattformen zu finden. Die Standardp lattform ist Linux x86 64-Bit, was das Little-Endian-Format ist.
p_parallel	INTEGER	0	Nein	Der Grad der Parallelität. In der Standardeinstellun g ist Parallelität deaktivie rt.

Beispiele

Das folgende Beispiel importiert die Tablespaces *TBS1*, *TBS2*, und *TBS3* aus dem Verzeichnis *DATA_PUMP_DIR*. Die Quellplattform ist AIX Based Systems (64-Bit) mit der Plattform-ID von6. Sie können die Plattform finden, IDs indem Sie eine Anfrage stellenV$TRANSPORTABLE_PLATFORM.

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
        'TBS1,TBS2,TBS3',
        'DATA_PUMP_DIR',
        p_platform_id => 6);
END;
/

PRINT task_id
```

Importieren von Metadaten transportabler Tabellenbereiche in Ihre DB-Instance

Verwenden Sie das Verfahren`rdsadmin.rdsadmin_transport_util.import_xtts_metadata`, um transportable Tablespace-Metadaten in Ihre Oracle-DB-Instance RDS zu importieren. Während des Vorgangs wird der Status des Metadatenimports in der Tabelle `rdsadmin.rds_xtts_operation_info` angezeigt. Weitere Informationen finden Sie unter Schritt 5: Importieren der Tabellenbereich-Metadaten in Ihre Ziel-DB-Instance.

Syntax

```
PROCEDURE import_xtts_metadata(
    p_datapump_metadata_file IN SYS.DBA_DATA_FILES.FILE_NAME%TYPE,
    p_directory_name         IN VARCHAR2,
    p_exclude_stats          IN BOOLEAN DEFAULT FALSE,
    p_remap_tablespace_list  IN CLOB DEFAULT NULL,
    p_remap_user_list        IN CLOB DEFAULT NULL);
```

Parameter

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_datapump_metadata_file`	`SYS.DBA_DATA_FILES.FILE_NAME%TYPE`	—	Ja	Der Name der Oracle-Data-Pump-Datei, die die Metadaten für Ihre transportablen Tabellenbereiche enthält.
`p_directory_name`	`VARCHAR2`	—	Ja	Das Verzeichnis, das die Data-Pump-Datei enthält.
`p_exclude_stats`	`BOOLEAN`	`FALSE`	Nein	Flag, das angibt, ob Statistiken ausgeschlossen werden sollen.
`p_remap_tablespace_list`	`CLOB`	NULL	Nein	Eine Liste von Tabellenbereichen, die beim Metadatenimport neu zugeordnet werden sollen. Verwenden Sie dabei das Format *`from_tbs`*`:`*`to_tbs`*. Geben Sie beispielsweise `users:user_data` an.
`p_remap_user_list`	`CLOB`	NULL	Nein	Eine Liste von Benutzerschemata, die beim Metadatenimport neu zugeordnet werden

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				sollen. Verwenden Sie dabei das Format *from_schema_name*:*to_schema_name*. Geben Sie beispielsweise `hr:human_resources` an.

Beispiele

Das Beispiel importiert die Tablespace-Metadaten aus der Datei *xttdump.dmp*, das sich im Verzeichnis befindet *DATA_PUMP_DIR*.

```
BEGIN
  rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
/
```

Auflisten verwaister Dateien nach einem Tabellenbereichimport

Verwenden Sie das Verfahren `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`, um Datendateien aufzulisten, die nach einem Tabellenbereichimport verwaist waren. Nachdem Sie die Datendateien identifiziert haben, können Sie sie löschen, indem Sie `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` aufrufen.

Syntax

```
FUNCTION list_xtts_orphan_files RETURN xtts_orphan_files_list_t PIPELINED;
```

Beispiele

Das folgende Beispiel führt die Prozedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` aus. Die Ausgabe zeigt zwei verwaiste Datendateien.

```
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

Löschen verwaister Dateien nach einem Tabellenbereichimport

Verwenden Sie das Verfahren `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`, um Datendateien zu löschen, die nach einem Tabellenbereichimport verwaist waren. Wenn Sie diesen Befehl ausführen, wird eine Protokolldatei generiert, die das Namensformat `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` im Verzeichnis BDUMP verwendet. Verwenden Sie das Verfahren `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`, um die verwaisten Dateien zu finden. Sie können die Protokolldatei lesen, indem Sie das Verfahren `rdsadmin.rds_file_util.read_text_file` aufrufen. Weitere Informationen finden Sie unter Phase 6: Bereinigen übrig gebliebener Dateien.

Syntax

```
PROCEDURE cleanup_incomplete_xtts_import(
    p_directory_name IN VARCHAR2);
```

Parameter

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_directory_name`	`VARCHAR2`	—	Ja	Das Verzeichnis, das die verwaisten Datendateien enthält.

Beispiele

Das folgende Beispiel löscht die verwaisten Datendateien in *`DATA_PUMP_DIR`*.

```
BEGIN
```

```
  rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

Das folgende Beispiel liest die durch den vorherigen Befehl generierte Protokolldatei.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

Konfiguration erweiterter Funktionen von RDS für Oracle

RDS für Oracle unterstützt verschiedene erweiterte Funktionen, darunter HugePages, einen Instance-Speicher und erweiterte Datentypen.

Themen

- Speichern temporärer Daten in einem Instance-Speicher von RDS für Oracle
- Aktivieren von HugePages für eine Instance von RDS für Oracle
- Erweiterte Datentypen RDS für Oracle aktivieren

Speichern temporärer Daten in einem Instance-Speicher von RDS für Oracle

Verwenden Sie einen Instance-Speicher für die temporären Tabellenräume und den Smart-Flash-Cache (Flash-Cache) der Datenbank in unterstützten DB-Instance-Klassen von RDS für Oracle.

Themen

- Übersicht über den Instance-Speicher von RDS für Oracle
- Aktivieren eines Instance-Speichers von RDS für Oracle
- Konfiguration eines Instance-Speichers RDS für Oracle
- Arbeiten mit einem Instance-Speicher auf einer Oracle Read Replica
- Konfiguration einer temporären Tablespace-Gruppe in einem Instance-Speicher und Amazon EBS
- Entfernen eines Instance-Speichers von RDS für Oracle

Übersicht über den Instance-Speicher von RDS für Oracle

Ein Instance-Speicher stellt temporären Speicher auf Blockebene für eine DB-Instance von RDS für Oracle bereit. Sie können einen Instance-Speicher für das temporäre Speichern von Informationen verwenden, die sich häufig ändern.

Ein Instance-Speicher basiert auf Non-Volatile Memory Express (NVMe) -Geräten, die physisch an den Host-Computer angeschlossen sind. Der Speicher ist für niedrige Latenz, zufällige I/O Leistung und sequentiellen Lesedurchsatz optimiert.

Die Größe des Instance-Speichers variiert je nach DB-Instance-Typ. Weitere Informationen zum Instance-Store finden Sie unter Amazon EC2 Instance Store im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Themen

- Datentypen im Instance-Speicher von RDS für Oracle
- Vorteile des Instance-Speichers von RDS für Oracle
- Unterstützte Instance-Klassen für den Instance-Speicher von RDS für Oracle
- Unterstützte Engine-Versionen für den Instance-Speicher von RDS für Oracle
- Wird AWS-Regionen für den RDS for Oracle-Instance-Speicher unterstützt
- Kosten des Instance-Speichers von RDS für Oracle

Datentypen im Instance-Speicher von RDS für Oracle

Sie können die folgenden Typen temporärer Daten von RDS für Oracle in einem Instance-Speicher ablegen:

Ein temporärer Tabellenraum

Oracle Database verwendet temporäre Tabellenräume, um Zwischenergebnisse von Abfragen zu speichern, die nicht in den Speicher passen. Größere Abfragen können große Mengen an Zwischendaten generieren, die vorübergehend zwischengespeichert werden müssen, aber nicht dauerhaft bestehen müssen. Ein temporärer Tabellenraum ist insbesondere nützlich für Sortierungen, Hash-Aggregationen und Joins. Wenn Ihre DB-Instance von RDS für Oracle die Enterprise Edition oder Standard Edition 2 verwendet, können Sie einen temporären Tabellenraum in einem Instance-Speicher ablegen.

Der Flash-Cache

Der Flash-Cache verbessert die Leistung von zufälligen Einzelblock-Lesevorgängen im konventionellen Pfad. Es empfiehlt sich, den Cache so zu dimensionieren, dass er den größten Teil Ihres aktiven Datasets aufnehmen kann. Wenn Ihre DB-Instance von RDS für Oracle die Enterprise Edition verwendet, können Sie den Flash-Cache in einem Instance-Speicher ablegen.

Standardmäßig ist ein Instance-Speicher für einen temporären Tabellenraum konfiguriert, jedoch nicht für den Flash-Cache. Sie können Oracle-Datendateien und Datenbankprotokolldateien nicht in einem Instance-Speicher ablegen.

Vorteile des Instance-Speichers von RDS für Oracle

Sie könnten erwägen, einen Instance-Speicher zu verwenden, um temporäre Dateien und Caches zu speichern, deren Verlust Sie hinnehmen können. Wenn Sie die DB-Leistung verbessern möchten oder wenn eine steigende Workload Leistungsprobleme für Ihren Amazon-EBS-Speicher verursacht, sollten Sie eine Skalierung auf eine Instance-Klasse in Betracht ziehen, die einen Instance-Speicher unterstützt.

Wenn Sie Ihren temporären Tabellenraum und Flash-Cache in einem Instance-Speicher ablegen, erhalten Sie folgende Vorteile:

- Niedrigere Lese-Latenzen
- Höherer Durchsatz
- Geringere Auslastung Ihrer Amazon-EBS-Volumes
- Geringere Speicher- und Snapshot-Kosten aufgrund geringerer Amazon-EBS-Last
- Weniger Druck, hohe IOPS bereitzustellen, was möglicherweise Ihre Gesamtkosten senkt

Indem Sie Ihren temporären Tabellenraum im Instance-Speicher ablegen, erzielen Sie eine sofortige Leistungssteigerung für Abfragen, die temporären Speicherplatz verwenden. Wenn Sie den Flash-Cache im Instance-Speicher ablegen, haben zwischengespeicherte Blocklesevorgänge in der Regel eine viel geringere Latenz als Amazon-EBS-Lesevorgänge. Der Flash-Cache muss „aufgewärmt“ werden, bevor er Leistungsvorteile bietet. Der Cache wärmt sich selbst auf, da die Datenbank Blöcke in den Flash-Cache schreibt, wenn sie für den Datenbankpuffer-Cache zu alt werden.

> **Note**
>
> In einigen Fällen verursacht der Flash-Cache aufgrund der Cache-Verwaltung einen Leistungs-Overhead. Bevor Sie den Flash-Cache in einer Produktionsumgebung aktivieren, sollten Sie Ihre Workload analysieren und den Cache in einer Testumgebung testen.

Unterstützte Instance-Klassen für den Instance-Speicher von RDS für Oracle

Amazon RDS unterstützt den Instance-Speicher für die folgenden DB-Instance-Klassen:

- db.m5d
- db.m6id

- db.r5d
- db.r6id
- db.x2idn
- db.x2iedn

RDS für Oracle unterstützt die vorgenannten DB-Instance-Klassen nur für das BYOL-Lizenzmodell. Weitere Informationen erhalten Sie unter Unterstützte RDS für Oracle-DB-Instance-Klassen und Bringen Sie Ihre eigene Lizenz (BYOL) für EE mit und SE2.

Um den gesamten Instance-Speicher für die unterstützten DB-Instance-Typen anzuzeigen, führen Sie den folgenden Befehl in der AWS CLI aus.

Example

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=*5d.*large*,*6id.*large*" \
  --query "InstanceTypes[?contains(InstanceType,'m5d')||contains(InstanceType,'r5d')||
contains(InstanceType,'m6id')||contains(InstanceType,'r6id')][InstanceType,
 InstanceStorageInfo.TotalSizeInGB]" \
  --output table
```

Der vorhergehende Befehl gibt die unformatierte Gerätegröße für den Instance-Speicher zurück. RDS für Oracle verwendet einen kleinen Teil dieses Speicherplatzes für die Konfiguration. Der Speicherplatz im Instance-Speicher, der für temporäre Tabellenräume oder den Flash-Cache zur Verfügung steht, ist etwas kleiner.

Unterstützte Engine-Versionen für den Instance-Speicher von RDS für Oracle

Der Instance-Speicher wird von folgenden Engine-Versionen von RDS für Oracle unterstützt:

- Version 21.0.0.0.ru-2022-01.rur-2022-01.r1 oder höhere Versionen von Oracle Database 21c
- 19.0.0.0.ru-2021-10.rur-2021-10.r1 oder höhere Versionen von Oracle Database 19c

Wird AWS-Regionen für den RDS for Oracle-Instance-Speicher unterstützt

Der Instance-Speicher ist überall verfügbar AWS-Regionen , wo einer oder mehrere dieser Instance-Typen unterstützt werden. Weitere Informationen zu den Instance-Klassen db.m5d und db.r5d finden Sie unter . Weitere Informationen zu den Instance-Klassen, die von Amazon RDS für Oracle unterstützt werden, finden Sie unter RDS für Oracle-DB-Instance-Klassen.

Kosten des Instance-Speichers von RDS für Oracle

Die Kosten des Instance-Speichers sind in die Kosten der Instances integriert, für die der Instance-Speicher aktiviert ist. Es entstehen keine zusätzlichen Kosten, wenn Sie einen Instance-Speicher auf einer DB-Instance von RDS für Oracle aktivieren. Weitere Informationen zu Instances, für die der Instance-Speicher aktiviert ist, finden Sie unter Unterstützte Instance-Klassen für den Instance-Speicher von RDS für Oracle.

Aktivieren eines Instance-Speichers von RDS für Oracle

Führen Sie einen der folgenden Schritte aus, um den Instance-Speicher für temporäre Daten von RDS für Oracle zu aktivieren:

- Erstellen Sie eine DB-Instance von RDS für Oracle mithilfe einer unterstützten Instance-Klasse. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine vorhandene DB-Instance von RDS für Oracle mithilfe einer unterstützten Instance-Klasse. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Konfiguration eines Instance-Speichers RDS für Oracle

Standardmäßig werden 100 % des Instance-Speicherplatzes dem temporären Tabellenraum zugewiesen. Wenn Sie den Instance-Speicher für die Zuweisung von Speicherplatz für den Flash-Cache und den temporären Tabellenraum konfigurieren möchten, legen Sie die folgenden Parameter in der Parametergruppe für Ihre Instance fest:

db_flash_cache_size= {* {0,2,4,6,8,10} /10} DBInstanceStore

Dieser Parameter gibt den Speicherplatz an, der für den Flash-Cache zugewiesen ist. Dieser Parameter ist nur für Oracle Database Enterprise Edition gültig. Der Standardwert ist `{DBInstanceStore*0/10}`. Wenn Sie einen Wert ungleich Null für `db_flash_cache_size` festlegen, aktiviert Ihre für Oracle-Instance den Flash-Cache, nachdem Sie die Instanz neu gestartet haben. RDS

rds.instance_store_temp_size= {* {0,2,4,6,8,10} /10} DBInstanceStore

Dieser Parameter gibt den Speicherplatz an, der für den temporären Tabellenraum zugewiesen ist. Der Standardwert ist `{DBInstanceStore*10/10}`. Dieser Parameter kann für Oracle Database Enterprise Edition geändert werden und ist für Standard Edition 2 schreibgeschützt. Wenn Sie einen Wert ungleich Null für festlegen`rds.instance_store_temp_size`, RDS weist Amazon Speicherplatz im Instance-Speicher für den temporären Tablespace zu.

Sie können die Parameter `db_flash_cache_size` und `rds.instance_store_temp_size` für DB-Instances festlegen, die keinen Instance-Speicher verwenden. In diesem Fall werden beide Einstellungen mit `0` ausgewertet, wodurch die Funktion deaktiviert wird. In diesem Fall können Sie dieselbe Parametergruppe für verschiedene Instance-Größen und für Instances verwenden, die keinen Instance-Speicher verwenden. Wenn Sie diese Parameter ändern, stellen Sie sicher, dass Sie die zugehörigen Instances neu starten, damit die Änderungen wirksam werden.

> **Important**
>
> Wenn Sie Speicherplatz für einen temporären Tablespace zuweisen, erstellt Amazon den temporären Tablespace RDS nicht automatisch. Informationen zum Erstellen des temporären Tabellenraums im Instance-Speicher finden Sie unter Erstellen eines temporären Tabellenraums im Instance-Speicher.

Der kombinierte Wert der vorhergehenden Parameter darf 10/10 oder 100 % nicht überschreiten. Die folgende Tabelle veranschaulicht gültige und ungültige Parametereinstellungen.

db_flash_cache_size	rds.instance_store_temp_size	Erklärung
db_flash_cache_size= {*0/10} DBInstanceStore	rds.instance_store_temp_size= {DBInstanceStore*10/10}	Dies ist eine gültige Konfiguration für alle Editionen von Oracle Database. Amazon RDS weist dem temporären Tablespace 100% des Instance-Speicherplatzes zu. Dies ist die Standardeinstellung.

db_flash_cache_size	rds.instance_store_temp_size	Erklärung
db_flash_cache_size= {*10/10} DBInstanceStore	rds.instance_store_temp_size= {DBInstanceStore*0/10}	Dieser Parameter ist nur für Oracle Database Enterprise Edition gültig. Amazon RDS weist dem Flash-Cache 100% des Instance-Speicherplatzes zu.
db_flash_cache_size= {*2/10} DBInstanceStore	rds.instance_store_temp_size= {DBInstanceStore*8/10}	Dieser Parameter ist nur für Oracle Database Enterprise Edition gültig. Amazon RDS weist 20% des Instance-Speicherplatzes dem Flash-Cache und 80% des Instance-Speicherplatzes dem temporären Tablespace zu.

db_flash_cache_size	rds.instance_store_temp_size	Erklärung
db_flash_cache_size= {*6/10} DBInstanceStore	rds.instance_store_temp_size= {DBInstanceStore*4/10}	Dieser Parameter ist nur für Oracle Database Enterprise Edition gültig. Amazon RDS weist 60% des Instance-Speicherplatzes dem Flash-Cache und 40% des Instance-Speicherplatzes dem temporären Tablespace zu.
db_flash_cache_size= {*2/10} DBInstanceStore	rds.instance_store_temp_size= {DBInstanceStore*4/10}	Dieser Parameter ist nur für Oracle Database Enterprise Edition gültig. Amazon RDS weist 20% des Instance-Speicherplatzes dem Flash-Cache und 40% des Instance-Speicherplatzes dem temporären Tablespace zu.

db_flash_cache_size	rds.instance_store_temp_size	Erklärung
db_flash_cache_size= {*8/10} DBInstanceStore	rds.instance_store_temp_size= {DBInstanceStore*8/10}	Dies ist eine ungültige Konfiguration, da der kombinierte Prozentsatz des Instance-Speicherplatzes 100 % übersteigt. In solchen Fällen RDS schlägt Amazon den Versuch fehl.

Überlegungen beim Ändern des DB-Instance-Typs

Wenn Sie den DB-Instance-Typ ändern, kann sich dies auf die Konfiguration des Flash-Caches oder des temporären Tabellenraums im Instance-Speicher auswirken. Berücksichtigen Sie die folgenden Änderungen und deren Auswirkungen:

Sie skalieren die DB-Instance, die den Instance-Speicher unterstützt, hoch oder herunter.

Die folgenden Werte erhöhen oder verringern sich proportional zur neuen Größe des Instance-Speichers:

- Die neue Größe des Flash-Caches.
- Der Speicherplatz, der den temporären Tabellenräumen zugewiesen ist, die sich im Instance-Speicher befinden.

Beispielsweise stellt die Einstellung `db_flash_cache_size={DBInstanceStore*6/10}` auf einer db.m5d.4xlarge-Instance etwa 340 GB Flash-Cache-Speicherplatz bereit. Wenn Sie den Instance-Typ auf db.m5d.8xlarge hochskalieren, erhöht sich der Flash-Cache-Speicherplatz auf etwa 680 GB.

Sie ändern eine DB-Instance, die keinen Instance-Speicher verwendet, in eine Instance mit Instance-Speicher.

Wenn `db_flash_cache_size` auf einen Wert größer als `0` festgelegt wird, ist der Flash-Cache konfiguriert. Wird `rds.instance_store_temp_size` auf einen Wert größer als `0` festgelegt, wird der Instance-Speicherplatz zur Verwendung durch einen temporären Tabellenraum zugewiesen. RDSdenn Oracle verschiebt temporäre Dateien nicht automatisch in den Instanzspeicher. Informationen zur Verwendung des zugewiesenen Speicherplatzes finden Sie unter Erstellen eines temporären Tabellenraums im Instance-Speicher oder Hinzufügen einer temporären Datei zum Instance-Speicher auf einer Read Replica.

Sie ändern eine DB-Instance, die einen Instance-Speicher verwendet, in eine Instance ohne Instance-Speicher.

In diesem Fall entfernt RDS for Oracle den Flash-Cache. RDSerstellt die temporäre Datei neu, die sich derzeit im Instance-Speicher auf einem EBS Amazon-Volume befindet. Die maximale Größe der neuen temporären Datei entspricht der früheren Größe des `rds.instance_store_temp_size`-Parameters.

Arbeiten mit einem Instance-Speicher auf einer Oracle Read Replica

Read Replicas unterstützen den Flash-Cache und temporäre Tabellenräume in einem Instance-Speicher. Während der Flash-Cache genauso funktioniert wie auf der primären DB-Instance, sind bei temporären Tabellenräume folgende Unterschiede zu beachten:

- Sie können keinen temporären Tabellenraum auf einer Read Replica erstellen. Wenn Sie einen neuen temporären Tablespace auf der primären Instance erstellen, repliziert Oracle RDS die Tablespace-Informationen ohne temporäre Dateien. Wenn Sie eine neue temporäre Datei hinzufügen möchten, verwenden Sie eine der folgenden Methoden:
  - Verwenden Sie das RDS Amazon-Verfahren`rdsadmin.rdsadmin_util.add_inst_store_tempfile`. RDSfür Oracle erstellt eine temporäre Datei im Instance-Speicher auf Ihrer Read Replica und fügt sie dem angegebenen temporären Tablespace hinzu.
  - Führen Sie den Befehl `ALTER TABLESPACE … ADD TEMPFILE` aus. RDSfür Oracle platziert die temporäre Datei im EBS Amazon-Speicher.

> **Note**
>
> Die Größen und Speichertypen der temporären Datei können auf der primären DB-Instance und der Read Replica unterschiedlich sein.

- Sie können die standardmäßige temporäre Tabellenraumeinstellung nur auf der primären DB-Instance verwalten. RDSfür Oracle repliziert die Einstellung auf alle Read Replicas.
- Sie können die temporären Tabellenraumgruppen nur auf der primären DB-Instance konfigurieren. RDSfür Oracle repliziert die Einstellung auf alle Read Replicas.

Konfiguration einer temporären Tablespace-Gruppe in einem Instance-Speicher und Amazon EBS

Sie können eine temporäre Tablespace-Gruppe so konfigurieren, dass sie temporäre Tablespaces sowohl in einem Instance-Speicher als auch in Amazon enthält. EBS Diese Methode ist nützlich, wenn Sie mehr temporären Speicher benötigen, als aufgrund der Maximaleinstellung von `rds.instance_store_temp_size` zulässig ist.

Wenn Sie eine temporäre Tablespace-Gruppe sowohl in einem Instance-Speicher als auch in Amazon konfigurierenEBS, weisen die beiden Tablespaces deutlich unterschiedliche Leistungsmerkmale auf. Oracle Database wählt den Tabellenraum für Abfragen basierend auf einem internen Algorithmus aus. Daher können ähnliche Abfragen in der Leistung variieren.

In der Regel erstellen Sie einen temporären Tabellenraum im Instance-Speicher wie folgt:

1. Erstellen Sie einen temporären Tabellenraum im Instance-Speicher.
2. Legen Sie den neuen Tabellenraum als temporären Standardtabellenraum der Datenbank fest.

Wenn die Größe des Tabellenraums im Instance-Speicher nicht ausreicht, können Sie wie folgt zusätzlichen temporären Speicher erstellen:

1. Weisen Sie den temporären Tabellenraum im Instance-Speicher einer temporären Tabellenraumgruppe zu.
2. Erstellen Sie einen neuen temporären Tablespace in AmazonEBS, falls noch keiner existiert.
3. Weisen Sie den temporären Tablespace in Amazon derselben Tablespace-Gruppe EBS zu, die den Instance-Speicher-Tablespace umfasst.

4. Legen Sie die Tabellenraumgruppe als temporären Standardtabellenraum fest.

Im folgenden Beispiel wird davon ausgegangen, dass die Größe des temporären Tabellenraums im Instance-Speicher nicht Ihren Anwendungsanforderungen entspricht. Das Beispiel erstellt den temporären Tablespace `temp_in_inst_store` im Instance-Speicher, weist ihn einer Tablespace-Gruppe zu`temp_group`, fügt den vorhandenen EBS Amazon-Tablespace mit dem Namen dieser Gruppe `temp_in_ebs` hinzu und legt diese Gruppe als temporären Standard-Tablespace fest.

```
SQL> EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace('temp_in_inst_store');

PL/SQL procedure successfully completed.

SQL> ALTER TABLESPACE temp_in_inst_store TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> ALTER TABLESPACE temp_in_ebs TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace('temp_group');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM DBA_TABLESPACE_GROUPS;

GROUP_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------
TEMP_GROUP                     TEMP_IN_EBS
TEMP_GROUP                     TEMP_IN_INST_STORE

SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE
 PROPERTY_NAME='DEFAULT_TEMP_TABLESPACE';

PROPERTY_VALUE
--------------
TEMP_GROUP
```

Entfernen eines Instance-Speichers von RDS für Oracle

Wenn Sie den Instance-Speicher entfernen möchten, ändern Sie Ihre DB-Instance von RDS für Oracle so, dass sie einen Instance-Typ verwendet, der keinen Instance-Speicher unterstützt, wie db.m5 oder db.r5.

Aktivieren von HugePages für eine Instance von RDS für Oracle

Amazon RDS for Oracle unterstützt Huge Pages mit Linux-Kernel für eine erhöhte Datenbank-Skalierbarkeit. HugePages erzeugt kleinere Seitentabellen und benötigt weniger CPU-Zeit für die Speicherverwaltung, so dass die Leistung von großen Datenbank-Instances erhöht wird. Weitere Informationen finden Sie unter Overview of HugePages in der Oracle-Dokumentation.

Sie können HugePages mit allen unterstützten Versionen und Editionen von RDS für Oracle verwenden.

Der `use_large_pages`-Parameter steuert, ob HugePages für eine DB-Instance aktiviert werden. Die möglichen Einstellungen für diesen Parameter sind `ONLY`, `FALSE`, und `{DBInstanceClassHugePagesDefault}`. Der Parameter `use_large_pages` ist in der Standard-DB-Parametergruppe von Oracle auf `{DBInstanceClassHugePagesDefault}` gesetzt.

Wenn Sie steuern möchten, ob HugePages für eine DB-Instance automatisch aktiviert werden, können Sie die Formelvariable `DBInstanceClassHugePagesDefault` in Parametergruppen verwenden. Der Wert ist wie folgt bestimmt:

- Für die in der folgenden Tabelle erwähnten DB-Instance-Klassen wird `DBInstanceClassHugePagesDefault` standardmäßig immer als `FALSE` und `use_large_pages` als `FALSE` ausgewertet. Sie können HugePages für diese DB-Instance-Klassen manuell aktivieren, wenn die DB-Instance-Klasse einen Speicherplatz von mindestens 14 GiB hat.
- Für nicht in der folgenden Tabelle erwähnte DB-Instance-Klassen wird `DBInstanceClassHugePagesDefault` immer als `FALSE` ausgewertet, wenn die Instance-Klasse weniger als 14 GiB Arbeitsspeicher hat. Außerdem wird `use_large_pages` als `FALSE` ausgewertet.
- Für nicht in der folgenden Tabelle erwähnte DB-Instance-Klassen wird `DBInstanceClassHugePagesDefault` standardmäßig als `TRUE` ausgewertet, wenn die Instance-Klasse mindestens 14 GiB und weniger als 100 GiB Arbeitsspeicher hat. Außerdem wird `use_large_pages` als `ONLY` ausgewertet. Sie können HugePages manuell deaktivieren, indem Sie `use_large_pages` auf `FALSE` festlegen.

- Für nicht in der folgenden Tabelle erwähnte DB-Instance-Klassen wird `DBInstanceClassHugePagesDefault` immer als `TRUE` ausgewertet, wenn die Instance-Klasse mindestens 100 GiB Arbeitsspeicher hat. Außerdem wird `use_large_pages` als `ONLY` ausgewertet und HugePages kann nicht deaktiviert werden.

Für die folgenden DB-Instance-Klassen werden HugePages standardmäßig nicht aktiviert.

DB-Instance-Klassenfamilie	DB-Instance-Klassen ohne standardmäßige Aktivierung von HugePages
db.m5	db.m5.large
db.m4	db.m4.large, db.m4.xlarge, db.m4.2xlarge, db.m4.4xlarge, db.m4.10xlarge
db.t3	db.t3.micro, db.t3.small, db.t3.medium, db.t3.large

Weitere Informationen zu DB-Instance-Klassen finden Sie unter Hardware-Spezifikationen für DB-Instance-Klassen .

Legen Sie den Parameter `use_large_pages` auf `ONLY` fest, um HugePages für neue oder vorhandene DB-Instances manuell zu aktivieren. Sie können HugePages nicht mit Oracle Automatic Memory Management (Automatische Arbeitsspeicher Verwaltung – AMM) verwenden. Wenn Sie den Parameter `use_large_pages` auf `ONLY` setzten, dann müssen Sie auch `memory_target` und `memory_max_target` auf `0` setzen. Weitere Informationen über die Einstellung von DB-Parametern für Ihre DB-Instance finden Sie unter Parametergruppen für Amazon RDS.

Sie können auch die Parameter `sga_target`, `sga_max_size` und `pga_aggregate_target` festlegen. Wenn Sie die Speicher-Parameter System Global Area (SGA) und Program Global Area (PGA) festlegen, addieren Sie beide Werte. Subtrahieren Sie die Summe von Ihrem verfügbaren Instance-Arbeitsspeicher (`DBInstanceClassMemory`), um den freien Arbeitsspeicher nach der HugePages-Zuteilung zu ermitteln. Sie müssen mindestens 2 GiB Speicher oder 10 Prozent des gesamten verfügbaren Instance-Speichers frei lassen, je nachdem, welcher Wert kleiner ist.

Nachdem Sie Ihre Parameter konfiguriert haben, müssen Sie Ihre DB-Instance neu starten, damit die Änderungen wirksam werden. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

> **Note**
>
> Die Oracle DB-Instance schiebt Änderungen an SGA-bezogenen Initialisierungsparametern auf, bis Sie die Instance ohne Failover neu starten. Wählen Sie in der Amazon RDS-Konsole Neustart, wählen Sie aber nicht Neustart mit Failover aus. Rufen Sie in der AWS CLI den `reboot-db-instance` Befehl mit dem Parameter `--no-force-failover` auf. Die DB-Instance verarbeitet die SGA-bezogenen Parameter nicht während eines Failovers oder bei anderen Wartungsvorgängen, die einen Neustart der Instance bewirken.

Nachfolgend finden Sie eine Beispielparameterkonfiguration für HugePages, bei der HugePages manuell aktiviert wird. Sie sollten die Werte so festlegen, dass sie Ihren Anforderungen entsprechen.

```
memory_target            = 0
memory_max_target        = 0
pga_aggregate_target     = {DBInstanceClassMemory*1/8}
sga_target               = {DBInstanceClassMemory*3/4}
sga_max_size             = {DBInstanceClassMemory*3/4}
use_large_pages          = ONLY
```

Nehmen wir an, dass in einer Parametergruppe die folgenden Parameterwerte eingestellt sind.

```
memory_target            = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target        = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target     = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target               = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size             = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages          = {DBInstanceClassHugePagesDefault}
```

Die Parametergruppe wird von einer db.r4-DB-Instance-Klasse mit weniger als 100 GiB Speicher verwendet. Mit diesen Parametereinstellungen und der Einstellung `{DBInstanceClassHugePagesDefault}` für `use_large_pages` sind HugePages auf der db.r4-Instance aktiviert.

Betrachten Sie ein weiteres Beispiel mit folgenden eingestellten Parameterwerten in einer Parametergruppe.

```
memory_target           = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target       = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target    = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target              = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size            = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages         = FALSE
```

Die Parametergruppe wird von einer db.r4 DB-Instance-Klasse und einer db.r5 DB-Instance-Klasse verwendet - beide mit weniger als 100 GiB Speicher. Bei diesen Parametereinstellungen sind HugePages auf der db.r4- und db.r5-Instance deaktiviert.

> **Note**
>
> Wenn diese Parametergruppe von einer db.r4 DB-Instance-Klasse oder db.r5 DB-Instance-Klasse mit mindestens 100 GiB Speicher verwendet wird, wird die `FALSE`-Einstellung für `use_large_pages` überschrieben und auf `ONLY` festgelegt. In diesem Fall wird eine Kundenbenachrichtigung über die erfolgte Überschreibung versendet.

Nachdem HugePages in Ihrer DB-Instance aktiv ist, können Sie HugePages-Informationen anzeigen, indem Sie „Enhanced Monitoring“ (Erweiterte Überwachung) aktivieren. Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Erweiterte Datentypen RDS für Oracle aktivieren

Amazon RDS for Oracle unterstützt erweiterte Datentypen. Bei erweiterten Datentypen beträgt die maximale Größe 32 767 Byte für die Datentypen `VARCHAR2`, `NVARCHAR2` und `RAW`. Wenn Sie erweiterte Datentypen verwenden möchten, setzen Sie den Parameter `MAX_STRING_SIZE` auf `EXTENDED`. Weitere Informationen finden Sie unter Extended Data Types in der Oracle-Dokumentation.

Wenn Sie keine erweiterten Datentypen verwenden möchten, belassen Sie für den Parameter `MAX_STRING_SIZE` die Standardeinstellung `STANDARD`. In diesem Fall liegen die Größenbeschränkungen bei 4.000 Byte für die `VARCHAR2` `NVARCHAR2` Datentypen und und 2.000 Byte für den RAW Datentyp.

Sie können erweiterte Datentypen auf einer neuen oder einer vorhandenen DB-Instance aktivieren. Bei neuen DB-Instances dauert es gewöhnlich länger, die DB-Instance zu erstellen, wenn Sie erweiterte Datentypen aktivieren. Bei vorhandenen DB-Instances ist die DB-Instance während des Konvertierungsvorgangs nicht verfügbar.

Überlegungen zu erweiterten Datentypen

Beachten Sie Folgendes, wenn Sie erweiterte Datentypen für Ihre DB-Instance aktivieren:

- Wenn Sie erweiterte Datentypen für eine neue oder bestehende DB-Instance aktivieren, müssen Sie die Instance neu starten, damit die Änderung wirksam wird.
- Nachdem Sie erweiterte Datentypen aktiviert haben, können Sie die DB-Instance nicht wieder so ändern, dass sie die Standardgröße für Datentypen verwendet. Wenn Sie den `MAX_STRING_SIZE` Parameter wieder auf diesen Wert setzen, `STANDARD` wird der `incompatible-parameters` Status angezeigt.
- Wenn Sie eine DB-Instance, die erweiterte Datentypen verwendet, wiederherstellen, müssen Sie eine Parametergruppe angeben, für die der Parameter `MAX_STRING_SIZE` auf `EXTENDED` gesetzt ist. Wenn Sie bei der Wiederherstellung die Standard-Parametergruppe oder eine andere Parametergruppe angeben, für die `MAX_STRING_SIZE` auf `STANDARD` gesetzt ist, wird der Status `incompatible-parameters` ausgegeben.
- Wenn der Status der DB-Instance aufgrund der Einstellung für `incompatible-parameters` `MAX_STRING_SIZE` lautet, ist die DB-Instance nicht verfügbar, bis Sie den Parameter `MAX_STRING_SIZE` auf `EXTENDED` setzen und die DB-Instance neu starten.

Aktivieren erweiterter Datentypen für eine neue DB-Instance

Wenn Sie eine DB-Instance mit der `MAX_STRING_SIZE` Einstellung auf erstellen`EXTENDED`, wird für die Instance die `MAX_STRING_SIZE` Standardeinstellung angezeigt`STANDARD`. Starten Sie die Instance neu, um die Änderung zu aktivieren.

So aktivieren Sie erweiterte Datentypen für eine neue DB-Instance

1. Setzen Sie den `MAX_STRING_SIZE` in einer Parametergruppe auf `EXTENDED`.

Zur Festlegung des Parameters können Sie entweder eine neue Parametergruppe erstellen oder eine vorhandene Parametergruppe ändern.

Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

2. Erstellen Sie eine neue RDS für Oracle DB-Instance.

   Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

3. Ordnen Sie die Parametergruppe, bei der `MAX_STRING_SIZE` auf `EXTENDED` gesetzt ist, der DB-Instance zu.

   Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

4. Starten Sie die DB-Instance neu, damit der Parameteränderung in Kraft tritt.

   Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Aktivieren erweiterter Datentypen für eine vorhandene DB-Instance

Wenn Sie eine DB-Instance ändern, um erweiterte Datentypen zu aktivieren, werden die Daten in der Datenbank so RDS konvertiert, dass sie die erweiterten Größen verwenden. Zu der Konvertierung und Ausfallzeit kommt es, wenn Sie die Datenbank nach der Parameteränderung das nächste Mal neu starten. Die DB-Instance ist während der Konvertierung nicht verfügbar.

Wie lange die Datenkonvertierung dauert, hängt von der DB-Instance-Klasse, der Datenbankgröße und dem Zeitpunkt des letzten DB-Snapshots ab. Um die Ausfallzeit zu verkürzen, können Sie unmittelbar vor dem Neustart einen Snapshot erstellen. Dadurch verkürzt sich die Zeit des Backups, das während des Konvertierungs-Workflows stattfindet.

> **Note**
>
> Nachdem Sie erweiterte Datentypen aktiviert haben, können Sie keine point-in-time Wiederherstellung zu einem Zeitpunkt während der Konvertierung durchführen. Eine Wiederherstellung auf den Zeitpunkt unmittelbar vor oder nach der Konvertierung ist möglich.

So aktivieren Sie erweiterte Datentypen für eine vorhandene DB-Instance

1. Erstellen Sie einen Snapshot der Datenbank.

Wenn die Datenbank ungültige Objekte enthält, RDS versucht Amazon, sie neu zu kompilieren. Die Konvertierung in erweiterte Datentypen kann fehlschlagen, wenn Amazon ein ungültiges Objekt nicht neu kompilieren RDS kann. Mithilfe des Snapshots können Sie die Datenbank wiederherstellen, wenn es ein Problem mit der Konvertierung gibt. Prüfen Sie immer vor der Konvertierung, ob ungültige Objekte vorhanden sind, und korrigieren oder löschen Sie diese ungültigen Objekte. Für Produktionsdatenbanken empfehlen wir, den Konvertierungsvorgang zunächst an einer Kopie Ihrer DB-Instance zu testen.

Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

2. Setzen Sie den `MAX_STRING_SIZE` in einer Parametergruppe auf `EXTENDED`.

   Zur Festlegung des Parameters können Sie entweder eine neue Parametergruppe erstellen oder eine vorhandene Parametergruppe ändern.

   Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

3. Ändern Sie die DB-Instance, um sie der Parametergruppe zuzuordnen, für die der Parameter `MAX_STRING_SIZE` auf `EXTENDED` gesetzt ist.

   Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

4. Starten Sie die DB-Instance neu, damit der Parameteränderung in Kraft tritt.

   Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Daten in Oracle auf Amazon importieren RDS

Wie Sie Daten in eine Amazon RDS for Oracle-DB-Instance importieren, hängt von folgenden Faktoren ab:

- Die Menge an Daten, die Sie haben
- Die Anzahl der Datenbankobjekte in Ihrer Datenbank
- Die Vielfalt der Datenbankobjekte in Ihrer Datenbank

Sie können beispielsweise das folgende Tool verwenden, je nach den Anforderungen:

- Oracle SQL Developer — Importieren Sie eine einfache 20-MB-Datenbank.
- Oracle Data Pump – Importieren Sie komplexe Datenbanken oder Datenbanken, die mehrere hundert Megabyte oder mehrere Terabyte groß sind. Sie können beispielsweise Tablespaces von einer lokalen Datenbank zu Ihrer RDS Oracle-DB-Instance transportieren. Sie können Amazon S3 oder Amazon verwendenEFS, um die Datendateien und Metadaten zu übertragen. Weitere Informationen finden Sie unter Migrieren mithilfe von Oracle Transportable Tablespaces, EFSAmazon-Integration und Amazon S3-Integration.
- AWS Database Migration Service (AWS DMS) — Migrieren Sie Datenbanken ohne Ausfallzeiten. Weitere Informationen zu AWS DMS finden Sie unter Was ist AWS Database Migration Service und im Blogbeitrag Migration von Oracle-Datenbanken mit nahezu keiner Ausfallzeit unter Verwendung von. AWS DMS

> **Important**
>
> Bevor Sie die vorherigen Migrationstechniken verwenden, empfehlen wir Ihnen, Ihre Datenbank zu sichern. Nachdem Sie die Daten importiert haben, können Sie Ihre RDS vier Oracle-DB-Instances sichern, indem Sie Snapshots erstellen. Später können Sie die Schnappschüsse wiederherstellen. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

Bei vielen Datenbank-Engines kann die laufende Replikation fortgesetzt werden, bis Sie bereit sind, zur Zieldatenbank zu wechseln. Sie können AWS DMS es verwenden, um von derselben Datenbank-Engine oder einer anderen Engine zu RDS for Oracle zu migrieren. Wenn Sie von einer

anderen Datenbank-Engine migrieren, können Sie die verwenden, AWS Schema Conversion Tool um Schemaobjekte zu migrieren, die AWS DMS nicht migriert werden.

Themen

- Importieren mit Oracle SQL Developer
- Migrieren mithilfe von Oracle Transportable Tablespaces
- Importieren mit Oracle Data Pump
- Import unter Verwendung von Oracle-Export/-Import
- Importieren mit Oracle SQL*Loader
- Migrieren mit materialisierten Oracle-Ansichten

Importieren mit Oracle SQL Developer

Oracle SQL Developer ist ein grafisches Java-Tool, das von Oracle kostenlos verteilt wird. SQL Developer bietet Optionen für das Migrieren von Daten zwischen Oracle-Datenbanken oder für Daten von anderen Datenbanken, wie zum Beispiel von MySQL zu einer Oracle-Datenbank. Dieses Tool eignet sich am besten für die Migration kleiner Datenbanken.

Sie können dieses Tool auf Ihrem Desktop-Computer (Windows, Linux oder Mac) oder auf einem Ihrer Server installieren. Nach der Installation von SQL Developer können Sie diesen verwenden, um eine Verbindung mit Ihrer Quell- und Ihrer Ziel-Datenbank herzustellen. Verwenden Sie den Befehl Database Copy im Menü Tools, um Ihre Daten in Ihre DB-Instance von RDS für Oracle zu kopieren.

Sie können SQL Developer unter http://www.oracle.com/technetwork/developer-tools/sql-developer herunterladen.

Wir empfehlen Ihnen, die Oracle SQL Developer Produkt-Dokumentation durchzulesen, bevor Sie mit dem Migrieren Ihrer Daten beginnen. Oracle biete auch eine Dokumentation über das Migrieren von anderen Datenbanken, einschließlich MySQL und SQL Server. Weitere Informationen finden Sie unter http://www.oracle.com/technetwork/database/migration in der Oracle-Dokumentation.

Migrieren mithilfe von Oracle Transportable Tablespaces

Sie können die Oracle-Funktion Transportable Tablespaces verwenden, um eine Reihe von Tabellenbereichen aus einer On-Premises-Oracle-Datenbank in eine DB-Instance von RDS für Oracle zu kopieren. Auf physischer Ebene übertragen Sie Quelldatendateien und Metadatendateien mit Amazon EFS oder Amazon S3 auf Ihre Ziel-DB-Instance. Die Funktion für transportable

Tablespaces verwendet das Paket. `rdsadmin.rdsadmin_transport_util` Informationen zur Syntax und Semantik dieses Pakets finden Sie unter. Transport von Tabellenbereichen

Blogbeiträge, in denen erklärt wird, wie Tablespaces transportiert werden, finden Sie unter Migrieren von Oracle-Datenbanken zur AWS Verwendung von transportierbarem Tablespace und Amazon RDS for Oracle Transportable Tablespaces mit RMAN.

Themen

- Überblick über Transportable Tablespaces von Oracle
- Phase 1: Einrichten Ihres Quell-Hosts
- Phase 2: Vorbereiten des vollständigen Tabellenbereich-Backups
- Phase 3: Erstellen und Übertragen inkrementeller Backups
- Phase 4: Transportieren der Tabellenbereiche
- Phase 5: Validieren der transportierten Tabellenbereiche
- Phase 6: Bereinigen übrig gebliebener Dateien

Überblick über Transportable Tablespaces von Oracle

Ein Transportable-Tablespace-Set besteht aus Datendateien für den Satz von Tabellenbereichen, der transportiert wird, und einer Export-Dump-Datei, die Metadaten zu den Tabellenbereichen enthält. In einer physischen Migrationslösung wie Transportable Tablespaces übertragen Sie physische Dateien: Datendateien, Konfigurationsdateien und Data-Pump-Dump-Dateien.

Themen

- Vor- und Nachteile von Transportable Tablespaces
- Einschränkungen für Transportable Tablespaces
- Voraussetzungen für Transportable Tablespaces

Vor- und Nachteile von Transportable Tablespaces

Wir empfehlen, Transportable Tablespaces zu verwenden, wenn Sie einen oder mehrere große Tabellenbereiche mit minimalen Ausfallzeiten zu RDS migrieren müssen. Transportable Tablespaces bieten gegenüber der logischen Migration die folgenden Vorteile:

- Die Ausfallzeiten sind geringer als bei den meisten anderen Oracle-Migrationslösungen.

- Da die Transportable-Tablespaces-Funktion nur physische Dateien kopiert, werden Datenintegritätsfehler und logische Beschädigungen vermieden, die bei der logischen Migration auftreten können.
- Es ist keine zusätzliche Lizenz erforderlich.
- Sie können einen Satz Tabellenbereiche zwischen verschiedenen Plattformen und Endianness-Typen migrieren, z. B. von einer Oracle-Solaris-Plattform nach Linux. Der Transport von Tabellenbereichen zu und von Windows-Servern wird jedoch nicht unterstützt.

> **Note**
>
> Linux wurde vollständig getestet und wird vollständig unterstützt. Nicht alle UNIX-Varianten wurden getestet.

Wenn Sie Transportable Tablespaces verwenden, können Sie Daten entweder mit Amazon S3 oder mit Amazon EFS transportieren:

- Wenn Sie EFS verwenden, verbleiben Ihre Backups für die Dauer des Imports im EFS-Dateisystem. Sie können die Dateien anschließend entfernen. Bei dieser Methode müssen Sie keinen EBS-Speicher für Ihre DB-Instance bereitstellen. Aus diesem Grund empfehlen wir, Amazon EFS anstelle von S3 zu verwenden. Weitere Informationen finden Sie unter EFSAmazon-Integration.
- Wenn Sie S3 verwenden, laden Sie RMAN-Backups auf den EBS-Speicher herunter, der an Ihre DB-Instance angehängt ist. Die Dateien verbleiben während des Imports in Ihrem EBS-Speicher. Nach dem Import können Sie diesen Speicherplatz freigeben, der Ihrer DB-Instance zugewiesen bleibt.

Der Hauptnachteil von Transportable Tablespaces besteht darin, dass Sie relativ fortgeschrittene Kenntnisse über Oracle Database benötigen. Weitere Informationen finden Sie unter Transporting Tablespaces Between Databases im Oracle-Database-Administratorhandbuch.

Einschränkungen für Transportable Tablespaces

Oracle-Database-Beschränkungen für Transportable Tablespaces gelten, wenn Sie diese Funktion in RDS für Oracle verwenden. Weitere Informationen finden Sie unter Limitations on Transportable Tablespaces und General Limitations on Transporting Data im Oracle-Database-

Administratorhandbuch. Beachten Sie die folgenden zusätzlichen Einschränkungen für Transportable Tablespaces in RDS für Oracle:

- Weder die Quell- noch die Zieldatenbank können Standard Edition 2 () verwenden. SE2 Es wird nur die Enterprise Edition unterstützt.
- Sie können keine Oracle Database 11g-Datenbank als Quelle verwenden. Die plattformübergreifende Funktion für transportable RMAN-Tablespaces basiert auf dem RMAN-Transportmechanismus, den Oracle Database 11g nicht unterstützt.
- Mithilfe von Transportable Tablespaces können Sie keine Daten aus einer DB-Instance von RDS für Oracle migrieren. Mit Transportable Tablespaces können Sie nur Daten zu einer DB-Instance von RDS für Oracle migrieren.
- Das Windows-Betriebssystem wird nicht unterstützt.
- Sie können Tabellenbereiche nicht in eine Datenbank auf einer niedrigeren Versionsebene transportieren. Die Zieldatenbank muss sich auf der gleichen oder einer höheren Versionsebene wie die Quelldatenbank befinden. Sie können beispielsweise keine Tabellenbereiche von Oracle Database 21c in Oracle Database 19c transportieren.
- Sie können keine administrativen Tabellenbereiche wie `SYSTEM` und `SYSAUX` transportieren.
- Sie können keine Objekte transportieren, die keine Daten sind, wie PL/SQL-Pakete, Java-Klassen, Views, Trigger, Sequenzen, Benutzer, Rollen und temporäre Tabellen. Um Objekte zu transportieren, die keine Daten sind, erstellen Sie sie manuell oder verwenden Sie den Export und Import von Data Pump-Metadaten. Weitere Informationen finden Sie in My Oracle Support Note 1454872.1.
- Sie können keine Tabellenbereiche transportieren, die verschlüsselt sind oder verschlüsselte Spalten verwenden.
- Wenn Sie Dateien mit Amazon S3 übertragen, beträgt die maximal unterstützte Dateigröße 5 TiB.
- Wenn die Quelldatenbank Oracle-Optionen wie „Spatial“ verwendet, können Sie keine Tabellenbereiche transportieren, es sei denn, in der Zieldatenbank sind dieselben Optionen konfiguriert.
- In einer Oracle-Replikatkonfiguration können Sie Tabellenbereiche nicht in eine DB-Instance von RDS für Oracle transportieren. Um dieses Problem zu umgehen, können Sie alle Replikate löschen, die Tabellenbereiche transportieren und die Replikate dann neu erstellen.

Voraussetzungen für Transportable Tablespaces

Führen Sie als Erstes die folgenden Schritte aus:

- Lesen Sie die Anforderungen für Transportable Tablespaces, die in den folgenden Support-Dokumenten von Oracle beschrieben werden:
  - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup (Doc ID 2471245.1)
  - Transportable Tablespace (TTS) Restrictions and Limitations: Details, Reference, and Version Where Applicable (Doc ID 1454872.1)
  - Primary Note for Transportable Tablespaces (TTS) - Common Questions and Issues (Doc ID 1166564.1)
- Planen Sie für eine Endianismuskonvertierung. Wenn Sie die Quellplattform-ID angeben, konvertiert RDS für Oracle den Endianismus automatisch. Informationen zur Suche nach einer Plattform IDs finden Sie unter Data Guard-Unterstützung für heterogene primäre und physische Standbys in derselben Data Guard-Konfiguration (Doc ID 413484.1).
- Stellen Sie sicher, dass die Transportable-Tablespace-Funktion auf Ihrer Ziel-DB-Instance aktiviert ist. Die Funktion ist nur aktiviert, wenn Sie beim Ausführen der folgenden Abfrage keine `ORA-20304`-Fehlermeldung erhalten:

  ```
  SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
  ```

  Wenn die Transportable-Tablespace-Funktion nicht aktiviert ist, starten Sie Ihre DB-Instance neu. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.
- Stellen Sie sicher, dass die Zeitzonendatei in der Quell- und Zieldatenbank identisch ist.
- Stellen Sie sicher, dass die Datenbankzeichensätze der Quell- und Zieldatenbanken eine der folgenden Anforderungen erfüllen:
  - Die Zeichensätze sind identisch.
  - Die Zeichensätze sind kompatibel. Eine Liste der Kompatibilitätsanforderungen finden Sie in der Dokumentation zu Oracle Database unter Allgemeine Einschränkungen beim Transport von Daten.
- Wenn Sie planen, Dateien mit Amazon S3 zu übertragen, gehen Sie wie folgt vor:
  - Stellen Sie sicher, dass ein Amazon S3 S3-Bucket für Dateiübertragungen verfügbar ist und dass sich der Amazon S3 S3-Bucket in derselben AWS Region wie Ihre DB-Instance befindet. Weitere Anleitungen finden Sie unter Erstellen eines Buckets im Amazon Simple Storage Service Handbuch "Erste Schritte".

  - Bereiten Sie den Amazon-S3-Bucket auf die Amazon RDS-Integration vor, indem Sie die Anleitungen unter Konfigurieren von IAM-Berechtigungen für die Integration von RDS for Oracle in Amazon S3 befolgen.
- Wenn Sie planen, Dateien mit Amazon EFS zu übertragen, stellen Sie sicher, dass Sie EFS gemäß den Anweisungen in EFSAmazon-Integration konfiguriert haben.
- Wir empfehlen dringend, automatische Backups in Ihrer Ziel-DB-Instance zu aktivieren. Da der Schritt zum Importieren von Metadaten potenziell fehlschlagen kann, ist es wichtig, dass Sie Ihre DB-Instance in den Zustand vor dem Import zurückversetzen können, sodass Sie Ihre Tabellenbereiche nicht erneut sichern, übertragen und importieren müssen.

Phase 1: Einrichten Ihres Quell-Hosts

In diesem Schritt kopieren Sie die von My Oracle Support bereitgestellten Transportable-Tablespaces-Skripts und richten die erforderlichen Konfigurationsdateien ein. In den folgenden Schritten führt der Quell-Host die Datenbank aus, die die Tabellenbereiche enthält, die zu Ihrer Ziel-Instance transportiert werden sollen.

So richten Sie Ihren Quell-Host ein

1. Melden Sie sich als Eigentümer Ihres Oracle-Basisverzeichnisses bei Ihrem Quell-Host an.
2. Stellen Sie sicher, dass Ihre Umgebungsvariablen `ORACLE_HOME` und `ORACLE_SID` auf Ihre Quelldatenbank verweisen.
3. Melden Sie sich als Administrator bei Ihrer Datenbank an und stellen Sie sicher, dass die Zeitzonenversion, der DB-Zeichensatz und der nationale Zeichensatz mit denen in Ihrer Zieldatenbank übereinstimmen.

   ```
   SELECT * FROM V$TIMEZONE_FILE;
   SELECT * FROM NLS_DATABASE_PARAMETERS
     WHERE PARAMETER IN ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
   ```

4. Richten Sie das Transportable-Tablespace-Dienstprogramm wie im Oracle-Support-Hinweis 2471245.1 beschrieben ein.

   Das Setup beinhaltet die Bearbeitung der `xtt.properties`-Datei auf Ihrem Quell-Host. Die folgende `xtt.properties`-Beispieldatei spezifiziert Backups von drei Tabellenbereichen im `/dsk1/backups`-Verzeichnis. Dies sind die Tabellenbereiche, die Sie zu Ihrer Ziel-DB-

Instance transportieren möchten. Es gibt auch die Quell-Plattform-ID an, um den Endianismus automatisch zu konvertieren.

> **Note**
>
> Informationen zu einer gültigen Plattform IDs finden Sie unter Data Guard-Unterstützung für heterogene primäre und physische Standbys in derselben Data Guard-Konfiguration (Doc ID 413484.1).

```
#linux system
platformid=13
#list of tablespaces to transport
tablespaces=TBS1,TBS2,TBS3
#location where backup will be generated
src_scratch_location=/dsk1/backups
#RMAN command for performing backup
usermantransport=1
```

Phase 2: Vorbereiten des vollständigen Tabellenbereich-Backups

In dieser Phase sichern Sie Ihre Tabellenbereiche zum ersten Mal, übertragen die Backups auf Ihren Ziel-Host und stellen sie dann mithilfe der Prozedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` wieder her. Wenn diese Phase abgeschlossen ist, befinden sich die ersten Tabellenbereich-Backups auf Ihrer Ziel-DB-Instance und können mit inkrementellen Backups aktualisiert werden.

Themen

- Schritt 1: Sichern der Tabellenbereiche auf Ihrem Quell-Host
- Schritt 2: Übertragen der Backup-Dateien auf Ihre Ziel-DB-Instance
- Schritt 3: Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance

Schritt 1: Sichern der Tabellenbereiche auf Ihrem Quell-Host

In diesem Schritt verwenden Sie das `xttdriver.pl`-Skript, um ein vollständiges Backup Ihrer Tabellenbereiche zu erstellen. Die Ausgabe von `xttdriver.pl` wird in der Umgebungsvariablen `TMPDIR` gespeichert.

So sichern Sie Ihre Tabellenbereiche

1. Wenn sich Ihre Tabellenbereiche im schreibgeschützten Modus befinden, melden Sie sich als Benutzer mit `ALTER TABLESPACE`-Berechtigung bei Ihrer Quelldatenbank an und versetzen Sie Ihre Tabellenbereiche in den Lese-/Schreibmodus. Andernfalls überspringen Sie diesen Schritt und gehen Sie direkt zum nächsten.

   Im folgenden Beispiel werden `tbs1`, `tbs2` und `tbs3` in den Lese-/Schreibmodus versetzt.

   ```
   ALTER TABLESPACE tbs1 READ WRITE;
   ALTER TABLESPACE tbs2 READ WRITE;
   ALTER TABLESPACE tbs3 READ WRITE;
   ```

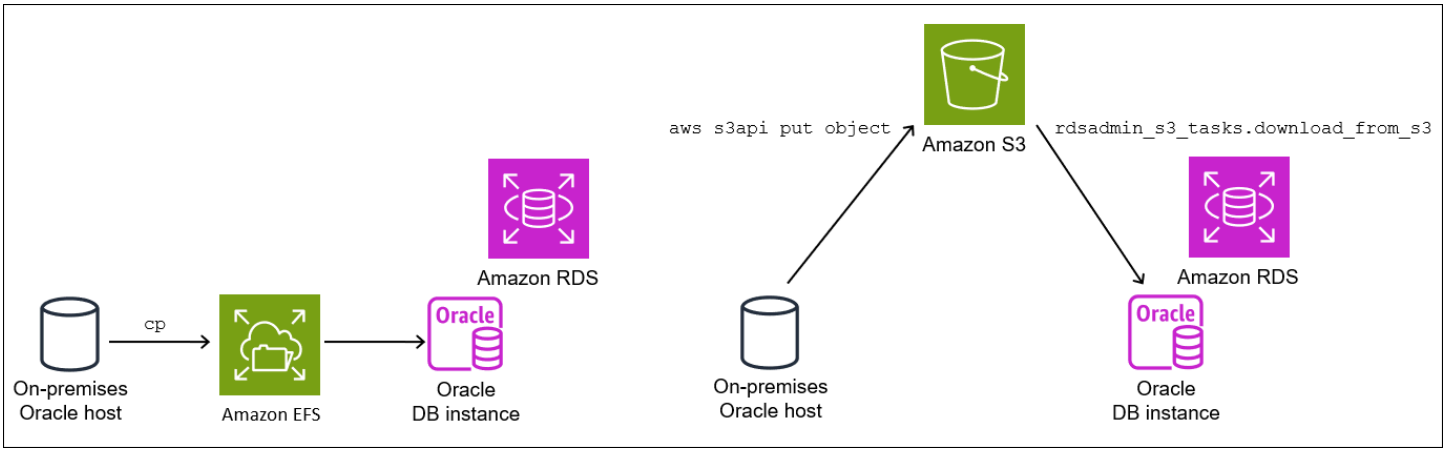
2. Sichern Sie Ihre Tabellenbereiche mithilfe des `xttdriver.pl`-Skripts. Optional können Sie `--debug` angeben, damit das Skript im Debug-Modus ausgeführt wird.

   ```
   export TMPDIR=location_of_log_files
   cd location_of_xttdriver.pl
   $ORACLE_HOME/perl/bin/perl xttdriver.pl --backup
   ```

Schritt 2: Übertragen der Backup-Dateien auf Ihre Ziel-DB-Instance

In diesem Schritt kopieren Sie die Sicherungs- und Konfigurationsdateien von Ihrem Scratch-Speicherort in Ihre Ziel-DB-Instance. Wählen Sie eine der folgenden Optionen:

- Wenn Quell- und Ziel-Host ein Amazon-EFS-Dateisystem gemeinsam nutzen, verwenden Sie ein Betriebssystem-Dienstprogramm wie `cp`, um Ihre Sicherungsdateien und die `res.txt`-Datei von Ihrem Scratch-Speicherort in ein freigegebenes Verzeichnis zu kopieren. Fahren Sie anschließend fort mit der unter Schritt 3: Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance beschriebenen Anleitung.
- Wenn Sie Ihre Backups in einem Amazon-S3-Bucket bereitstellen müssen, führen Sie die folgenden Schritte aus.



Schritt 2.2: Hochladen der Backups in Ihren Amazon-S3-Bucket

Laden Sie Ihre Backups und die `res.txt`-Datei aus Ihrem Scratch-Verzeichnis in Ihren Amazon-S3-Bucket hoch. Weitere Informationen finden Sie unter Hochladen von Objekten im Benutzerhandbuch von Amazon Simple Storage Service.

Schritt 2.3: Herunterladen der Backups aus Ihrem Amazon-S3-Bucket in Ihre Ziel-DB-Instance

In diesem Schritt verwenden Sie die Prozedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3`, um Ihre Backups auf Ihre DB-Instance von RDS für Oracle herunterzuladen.

So laden Sie Ihre Backups aus Ihrem Amazon-S3-Bucket herunter

1. Starten Sie SQL*Plus oder Oracle SQL Developer und melden Sie sich bei Ihrer DB-Instance von RDS für Oracle an.
2. Laden Sie die Backups aus dem Amazon-S3-Bucket auf Ihre Ziel-DB-Instance herunter, indem Sie die Amazon-RDS-Prozedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3` verwenden. Das folgende Beispiel lädt alle Dateien von einem Amazon S3-Bucket mit dem Namen *`amzn-s3-demo-bucket`* in das Verzeichnis *`DATA_PUMP_DIR`* herunter.

```
EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'res.txt');
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name    =>  'amzn-s3-demo-bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

Die Anweisung `SELECT` gibt die ID der Aufgabe in einem `VARCHAR2`-Datentyp zurück. Weitere Informationen finden Sie unter Hochladen von Dateien aus einem Amazon S3-Bucket zu einer Oracle-DB-Instance.

Schritt 3: Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance

Gehen Sie wie folgt vor, um Ihre Tablespaces auf Ihrer Ziel-DB-Instance wiederherzustellen. `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` Diese Prozedur konvertiert die Datendateien automatisch in das richtige Endian-Format.

Wenn Sie von einer anderen Plattform als Linux importieren, geben Sie `p_platform_id` beim Aufruf die Quellplattform mithilfe des Parameters an. `import_xtts_tablespaces` Stellen Sie sicher, dass die von Ihnen angegebene Plattform-ID mit der in der `xtt.properties` Datei unter angegebenen übereinstimmtSchritt 2: Exportieren der Tabellenbereich-Metadaten auf Ihren Quell-Host.

Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance

1. Starten Sie einen Oracle-SQL-Client und melden Sie sich als Hauptbenutzer bei Ihrer DB-Instance von RDS für Oracle an.
2. Führen Sie die Prozedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` aus und geben Sie dabei die zu importierenden Tabellenbereiche und das Verzeichnis mit den Backups an.

   Im folgenden Beispiel werden die Tablespaces *`TBS1TBS2`*, und *`TBS3`* aus dem Verzeichnis importiert. *`DATA_PUMP_DIR`* Die Quellplattform ist AIX-Based Systems (64-Bit) mit der Plattform-ID von. 6 Sie können die Plattform finden, IDs indem Sie eine Anfrage stellen. `V $TRANSPORTABLE_PLATFORM`

   ```
   VAR task_id CLOB

   BEGIN
     :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
           'TBS1,TBS2,TBS3',
           'DATA_PUMP_DIR',
           p_platform_id => 6);
   END;
   /
   ```

```
PRINT task_id
```

3. (Optional) Überwachen Sie den Fortschritt, indem Sie die Tabelle `rdsadmin.rds_xtts_operation_info` abfragen. Die Spalte `xtts_operation_state` enthält den Wert `EXECUTING`, `COMPLETED` oder `FAILED`.

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

> **Note**
>
> Für lang andauernde Operationen können Sie auch `V$SESSION_LONGOPS`, `V$RMAN_STATUS` und `V$RMAN_OUTPUT` abfragen.

4. Sehen Sie sich das Protokoll des abgeschlossenen Imports an, indem Sie die Task-ID aus dem vorherigen Schritt verwenden.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||'&task_id'||'.log'));
```

Stellen Sie sicher, dass der Import erfolgreich abgeschlossen wurde, bevor Sie mit dem nächsten Schritt fortfahren.

Phase 3: Erstellen und Übertragen inkrementeller Backups

In dieser Phase erstellen und übertragen Sie regelmäßig inkrementelle Backups, während die Quelldatenbank aktiv ist. Mit dieser Methode wird die Größe Ihres endgültigen Tabellenbereich-Backups reduziert. Wenn Sie mehrere inkrementelle Backups erstellen, müssen Sie die `res.txt`-Datei nach dem letzten inkrementellen Backup kopieren, bevor Sie es auf die Ziel-Instance anwenden können.

Die Schritte sind dieselben wie in Phase 2: Vorbereiten des vollständigen Tabellenbereich-Backups, außer dass der Importschritt optional ist.

Phase 4: Transportieren der Tabellenbereiche

In dieser Phase sichern Sie Ihre schreibgeschützten Tabellenbereiche und exportieren Data-Pump-Metadaten, übertragen diese Dateien auf Ihren Ziel-Host und importieren sowohl die Tabellenbereiche als auch die Metadaten.

Themen

- Schritt 1: Sichern Ihrer schreibgeschützten Tabellenbereiche
- Schritt 2: Exportieren der Tabellenbereich-Metadaten auf Ihren Quell-Host
- Schritt 3: (nur Amazon S3) Übertragen der Backup- und Exportdateien auf Ihre Ziel-DB-Instance
- Schritt 4: Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance
- Schritt 5: Importieren der Tabellenbereich-Metadaten in Ihre Ziel-DB-Instance

Schritt 1: Sichern Ihrer schreibgeschützten Tabellenbereiche

Dieser Schritt ist identisch mit Schritt 1: Sichern der Tabellenbereiche auf Ihrem Quell-Host, mit einem wesentlichen Unterschied: Sie versetzen Ihre Tabellenbereiche in den schreibgeschützten Modus, bevor Sie sie zum letzten Mal sichern.

Im folgenden Beispiel werden `tbs1`, `tbs2` und `tbs3` in den schreibgeschützten Modus versetzt.

```
ALTER TABLESPACE tbs1 READ ONLY;
ALTER TABLESPACE tbs2 READ ONLY;
ALTER TABLESPACE tbs3 READ ONLY;
```

Schritt 2: Exportieren der Tabellenbereich-Metadaten auf Ihren Quell-Host

Exportieren Sie Ihre Tabellenbereich-Metadaten, indem Sie das Dienstprogramm `expdb` auf Ihrem Quell-Host ausführen. Im folgenden Beispiel werden Tablespaces *TBS1*, und exportiert *TBS2*, um die Datei im *TBS3* Verzeichnis abzulegen. *xttdump.dmp DATA_PUMP_DIR*

```
expdp username/pwd \
dumpfile=xttdump.dmp \
directory=DATA_PUMP_DIR \
statistics=NONE \
transport_tablespaces=TBS1,TBS2,TBS3 \
transport_full_check=y \
logfile=tts_export.log
```

Wenn *DATA_PUMP_DIR* es sich um ein gemeinsam genutztes Verzeichnis in Amazon EFS handelt, fahren Sie mit fortSchritt 4: Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance.

Schritt 3: (nur Amazon S3) Übertragen der Backup- und Exportdateien auf Ihre Ziel-DB-Instance

Wenn Sie Amazon S3 verwenden, um Ihre Tabellenbereich-Backups und die Data-Pump-Exportdatei bereitzustellen, führen Sie die folgenden Schritte aus.

Schritt 3.1: Hochladen der Backups und der Dump-Datei von Ihrem Quell-Host in Ihren Amazon-S3-Bucket

Laden Sie Ihre Backup- und Dump-Dateien von Ihrem Quell-Host in Ihren Amazon-S3-Bucket hoch. Weitere Informationen finden Sie unter Hochladen von Objekten im Benutzerhandbuch von Amazon Simple Storage Service.

Schritt 3.2: Herunterladen der Backups und der Dump-Datei aus Ihrem Amazon-S3-Bucket in Ihre DB-Instance

In diesem Schritt verwenden Sie die Prozedur `rdsadmin.rdsadmin_s3_tasks.download_from_s3`, um Ihre Backups und die Dump-Datei auf Ihre DB-Instance von RDS für Oracle herunterzuladen. Führen Sie die Schritte unter Schritt 2.3: Herunterladen der Backups aus Ihrem Amazon-S3-Bucket in Ihre Ziel-DB-Instance aus.

Schritt 4: Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance

Verwenden Sie die Prozedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, um die Tabellenbereiche wiederherzustellen. Informationen zu Syntax und Semantik dieses Verfahrens finden Sie unter Importieren transportabler Tabellenbereiche in Ihre DB-Instance.

> **Important**
>
> Nachdem Sie Ihren endgültigen Tabellenbereichimport abgeschlossen haben, ist der nächste Schritt das Importieren der Metadaten von Oracle Data Pump. Wenn der Import fehlschlägt, ist es wichtig, dass Sie Ihre DB-Instance in den Zustand vor dem Fehler zurückversetzen. Daher empfehlen wir Ihnen, den Anweisungen unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS zu folgen und einen DB-Snapshot Ihrer DB-Instance zu erstellen. Der Snapshot enthält alle importierten Tabellenbereiche. Wenn der Import fehlschlägt, müssen Sie den Backup- und Importvorgang nicht wiederholen.
> Wenn für Ihre Ziel-DB-Instance automatische Backups aktiviert sind und Amazon RDS nicht erkennt, dass vor dem Import der Metadaten ein gültiger Snapshot initiiert wurde, versucht RDS, einen Snapshot zu erstellen. Abhängig von Ihrer Instance-Aktivität kann dieser Snapshot erfolgreich erstellt werden oder auch nicht. Wenn kein gültiger Snapshot erkannt

wird oder ein Snapshot nicht initiiert werden kann, wird der Metadatenimport mit Fehlern beendet.

Importieren der Tabellenbereiche in Ihre Ziel-DB-Instance

1. Starten Sie einen Oracle-SQL-Client und melden Sie sich als Hauptbenutzer bei Ihrer DB-Instance von RDS für Oracle an.
2. Führen Sie die Prozedur `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` aus und geben Sie dabei die zu importierenden Tabellenbereiche und das Verzeichnis mit den Backups an.

   Im folgenden Beispiel werden die Tablespaces *TBS1TBS2*, und *TBS3* aus dem Verzeichnis importiert. *DATA_PUMP_DIR*

   ```
   BEGIN

    :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces('TBS1,TBS2,TBS3','DATA_
   END;
   /
   PRINT task_id
   ```

3. (Optional) Überwachen Sie den Fortschritt, indem Sie die Tabelle `rdsadmin.rds_xtts_operation_info` abfragen. Die Spalte `xtts_operation_state` enthält den Wert `EXECUTING`, `COMPLETED` oder `FAILED`.

   ```
   SELECT * FROM rdsadmin.rds_xtts_operation_info;
   ```

   > **Note**
   >
   > Für lang andauernde Operationen können Sie auch `V$SESSION_LONGOPS`, `V$RMAN_STATUS` und `V$RMAN_OUTPUT` abfragen.

4. Sehen Sie sich das Protokoll des abgeschlossenen Imports an, indem Sie die Task-ID aus dem vorherigen Schritt verwenden.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
    'dbtask-'||'&task_id'||'.log'));
   ```

Stellen Sie sicher, dass der Import erfolgreich abgeschlossen wurde, bevor Sie mit dem nächsten Schritt fortfahren.

5. Folgen Sie den Anweisungen unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS und erstellen Sie einen manuellen DB-Snapshot.

Schritt 5: Importieren der Tabellenbereich-Metadaten in Ihre Ziel-DB-Instance

In diesem Schritt importieren Sie die Transportable-Tablespace-Metadaten mithilfe der Prozedur `rdsadmin.rdsadmin_transport_util.import_xtts_metadata` in Ihre DB-Instance von RDS für Oracle. Informationen zu Syntax und Semantik dieses Verfahrens finden Sie unter Importieren von Metadaten transportabler Tabellenbereiche in Ihre DB-Instance. Während des Vorgangs wird der Status des Imports in der Tabelle `rdsadmin.rds_xtts_operation_info` angezeigt.

> **Important**
>
> Bevor Sie Metadaten importieren, empfehlen wir Ihnen dringend, zu überprüfen, ob nach dem Import Ihrer Tabellenbereiche ein DB-Snapshot erfolgreich erstellt wurde. Wenn der Importschritt fehlschlägt, stellen Sie Ihre DB-Instance wieder her, beheben Sie die Importfehler und versuchen Sie dann erneut, den Import durchzuführen.

Importieren der Data-Pump-Metadaten in Ihre DB-Instance von RDS für Oracle

1. Starten Sie Ihren Oracle-SQL-Client und melden Sie sich als Hauptbenutzer bei Ihrer Ziel-DB-Instance an.
2. Erstellen Sie die Benutzer, denen Schemas in Ihren transportierten Tabellenbereichen gehören, falls diese Benutzer noch nicht existieren.

```
CREATE USER tbs_owner IDENTIFIED BY password;
```

3. Importieren Sie die Metadaten und geben Sie dabei den Namen der Dump-Datei und deren Verzeichnispfad an.

```
BEGIN

 rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
```

```
/
```

4. (Optional) Fragen Sie die Transportable-Tablespace-Verlaufstabelle ab, um den Status des Metadatenimports zu sehen.

   ```
   SELECT * FROM rdsadmin.rds_xtts_operation_info;
   ```

   Wenn der Vorgang abgeschlossen ist, befinden sich Ihre Tabellenbereiche im schreibgeschützten Modus.

5. (Optional) Sehen Sie sich die Protokolldatei an.

   Das folgende Beispiel listet den Inhalt des BDUMP-Verzeichnisses auf und fragt dann das Importprotokoll ab.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'BDUMP'));

   SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file(
     p_directory => 'BDUMP',
     p_filename => 'rds-xtts-
   import_xtts_metadata-2023-05-22.01-52-35.560858000.log'));
   ```

Phase 5: Validieren der transportierten Tabellenbereiche

In diesem optionalen Schritt validieren Sie Ihre transportierten Tabellenbereiche mithilfe der Prozedur `rdsadmin.rdsadmin_rman_util.validate_tablespace` und versetzen sie dann in den Lese-/Schreibmodus.

So validieren Sie die transportierten Daten

1. Starten Sie SQL*Plus oder SQL Developer und melden Sie sich als Hauptbenutzer bei Ihrer Ziel-DB-Instance an.
2. Validieren Sie die Tabellenbereiche mithilfe der Prozedur `rdsadmin.rdsadmin_rman_util.validate_tablespace`.

   ```
   SET SERVEROUTPUT ON
   BEGIN
       rdsadmin.rdsadmin_rman_util.validate_tablespace(
           p_tablespace_name     => 'TBS1',
           p_validation_type     => 'PHYSICAL+LOGICAL',
   ```

```
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS2',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS3',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
END;
/
```

3. Versetzen Sie Ihre Tabellenbereiche in den Lese-/Schreibmodus.

```
ALTER TABLESPACE TBS1 READ WRITE;
ALTER TABLESPACE TBS2 READ WRITE;
ALTER TABLESPACE TBS3 READ WRITE;
```

Phase 6: Bereinigen übrig gebliebener Dateien

In diesem optionalen Schritt entfernen Sie alle nicht benötigten Dateien. Verwenden Sie das Verfahren `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`, um Datendateien aufzulisten, die nach einem Tablespace-Import verwaist waren, und dann das Verfahren `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`, um sie zu löschen. Informationen zu Syntax und Semantik dieser Verfahren finden Sie unter Auflisten verwaister Dateien nach einem Tabellenbereichimport und Löschen verwaister Dateien nach einem Tabellenbereichimport.

So bereinigen Sie übrig gebliebene Dateien

1. Entfernen Sie alte Backups *DATA_PUMP_DIR* wie folgt:

   a. Listen Sie die Backup-Dateien auf, indem Sie den Befehl `rdsadmin.rdsadmin_file_util.listdir` ausführen.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
    'DATA_PUMP_DIR'));
   ```

   b. Entfernen Sie die Backups nacheinander, indem Sie `UTL_FILE.FREMOVE` aufrufen.

```
EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'backup_filename');
```

2. Wenn Sie Tabellenbereiche, aber keine Metadaten für diese Tabellenbereiche importiert haben, können Sie die verwaisten Datendateien wie folgt löschen:

   a. Listen Sie die verwaisten Datendateien auf, die Sie löschen müssen. Das folgende Beispiel führt die Prozedur `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` aus.

```
SQL> SELECT * FROM
 TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

   b. Löschen Sie die verwaisten Dateien, indem Sie die Prozedur `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` aus

```
BEGIN

 rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

   Der Bereinigungsvorgang generiert eine Protokolldatei, die das Namensformat `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` im BDUMP-Verzeichnis verwendet.

   c. Lesen Sie die im vorherigen Schritt generierte Protokolldatei. Das folgende Beispiel zeigt das Protokoll `rds-xtts-delete_xtts_orphaned_files-`*`2023-06-01.09-33-11.868894000`*`.log`.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));
```

```
TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

3. Wenn Sie Tabellenbereiche sowie Metadaten für diese Tabellenbereiche importiert haben, aber auf Kompatibilitätsfehler oder andere Probleme mit Oracle Data Pump gestoßen sind, bereinigen Sie die teilweise transportierten Datendateien wie folgt:

   a. Listen Sie die Tabellenbereiche auf, die teilweise transportierte Datendateien enthalten, indem Sie `DBA_TABLESPACES` abfragen.

   ```
   SQL> SELECT TABLESPACE_NAME FROM DBA_TABLESPACES WHERE PLUGGED_IN='YES';

   TABLESPACE_NAME
   --------------------------------------------------------------------------------
   TBS_3
   ```

   b. Löschen Sie die Tabellenbereiche und die teilweise transportierten Datendateien.

   ```
   DROP TABLESPACE TBS_3 INCLUDING CONTENTS AND DATAFILES;
   ```

Importieren mit Oracle Data Pump

Oracle Data Pump ist ein Dienstprogramm, mit dem Sie Oracle-Daten in eine Dump-Datei exportieren und in eine andere Oracle-Datenbank importieren können. Das ist ein langfristiger Ersatz für die Oracle-Export/-Import-Dienstprogramme. Oracle Data Pump ist die empfohlene Methode zum Verschieben großer Datenmengen von einer Oracle-Datenbank in eine Amazon-RDS-DB-Instance.

Die Beispiele in diesem Abschnitt zeigen eine Möglichkeit, Daten in eine Oracle-Datenbank zu importieren. Oracle Data Pump unterstützt jedoch weitere Methoden. Weitere Informationen finden Sie in der Oracle Database-Dokumentation.

Für die Beispiele in diesem Abschnitt wird das `DBMS_DATAPUMP`-Paket verwendet. Dieselben Aufgaben können mithilfe der Befehlszeilendienstprogramme `impdp` und `expdp` von Oracle Data Pump ausgeführt werden. Sie können diese Dienstprogramme, einschließlich Oracle Instant Client, auf einem Remote-Host als Teil einer Oracle Client-Installation installieren. Weitere Informationen finden Sie unter So wird Oracle Instant Client verwendet, um den Import oder Export von Data Pump für Amazon RDS für Oracle-DB-Instances auszuführen

Themen

- Übersicht über Oracle Data Pump
- Importieren von Daten mit Oracle Data Pump und einem Amazon S3-Bucket
- Importieren von Daten mit Oracle Data Pump und einer Datenbankverbindung

Übersicht über Oracle Data Pump

Oracle Data Pump besteht aus den folgenden Komponenten:

- Befehlszeilenclients `expdp` und `impdp`
- Dem `DBMS_DATAPUMP`-PL/SQL-Paket
- Dem `DBMS_METADATA`-PL/SQL-Paket

Sie können Oracle Data Pump für die folgenden Szenarien verwenden:

- Importieren Sie Daten aus einer Oracle-Datenbank, entweder lokal oder auf einer EC2 Amazon-Instance, in eine RDS for Oracle DB-Instance.
- Importieren Sie Daten aus einer RDS for Oracle DB-Instance in eine Oracle-Datenbank, entweder lokal oder auf einer EC2 Amazon-Instance.
- Importieren Sie Daten zwischen RDS for Oracle DB-Instances, um beispielsweise Daten von EC2 - Classic zu VPC zu migrieren.

Sie können Oracle Data Pump-Dienstprogramme unter Oracle Database Software Downloads auf der Oracle Technology Network-Website herunterladen. Überlegungen zur Kompatibilität bei der Migration zwischen Versionen der Oracle-Datenbank finden Sie in der Oracle-Database-Dokumentation.

Oracle Data-Pump-Workflow

In der Regel verwenden Sie Oracle Data Pump in den folgenden Phasen:

1. Exportieren Sie Ihre Daten in eine Dump-Datei in der Quelldatenbank.
2. Laden Sie Ihre Dump-Datei in Ihre Ziel-DB-Instance hoch. Sie können die Übertragung mithilfe eines Amazon-S3-Buckets oder über eine Datenbankverbindung zwischen zwei Datenbanken vornehmen.
3. Importieren Sie die Daten aus Ihrer Dump-Datei in Ihre Instance von RDS für Oracle DB.

Bewährte Methoden für Oracle Data Pump

Wenn Sie Oracle Data Pump zum Importieren von Daten in eine Oracle-Instance verwenden, empfehlen wir die folgenden bewährten Methoden:

- Führen Sie Importe im Modus `schema` oder `table` durch, damit bestimmte Schemata und Objekte importiert werden.
- Beschränken Sie die Schemata, die Sie importieren, auf solche, die von Ihrer Anwendung benötigt werden.
- Importieren Sie nicht im `full`-Modus oder keine Importschemas für vom System verwaltete Komponenten.

  Weil RDS für Oracle keinen Zugriff auf `SYS` oder `SYSDBA` für administrative Benutzer gewährt, können diese Aktionen das Oracle-Datenwörterbuch beschädigen und die Stabilität Ihrer Datenbank gefährden.
- Beim Laden großer Datenmengen gehen Sie folgendermaßen vor:
  1. Übertragen Sie die Dump-Datei an die Ziel-DB-Instance von RDS für Oracle.
  2. Erstellen Sie einen DB-Snapshot Ihrer Instance.
  3. Testen Sie den Importvorgang, um zu prüfen, ob er erfolgreich ist.

  Wenn Datenbankkomponenten unwirksam sind, können Sie die DB-Instance löschen und sie über den DB-Snapshot neu erstellen. Die wiederhergestellte DB-Instance enthält alle Dumpdateien, die auf der DB-Instance bereitgestellt wurden, als Sie den DB-Snapshot erstellt haben.
- Importieren Sie keine Dump-Dateien, die mit den Exportparametern von Oracle Data Pump `TRANSPORT_TABLESPACES`,`TRANSPORTABLE`, oder `TRANSPORT_FULL_CHECK` erstellt wurden. DB-Instances von RDS für Oracle unterstützen das Importieren dieser Dump-Dateien nicht.
- Importieren Sie keine Speicherabbilddateien, die Oracle Scheduler-Objekte in `SYS`, `SYSTEM`, `RDSADMIN`, `RDSSEC` und `RDS_DATAGUARD`enthalten und zu den folgenden Kategorien gehören:
  - Aufträge
  - Programme
  - Schedules
  - Ketten
  - Regeln
  - Auswertungskontexte
  - Regelsätze

RDS für Oracle -DB-Instances unterstützen das Importieren dieser Dump-Dateien nicht.

- Um nicht unterstützte Oracle-Scheduler-Objekte auszuschließen, verwenden Sie zusätzliche Anweisungen während des Data Pump-Exports. Wenn Sie `DBMS_DATAPUMP` verwenden, fügen Sie einen zusätzlichen `METADATA_FILTER` vor dem `DBMS_METADATA.START_JOB` hinzu:

```
DBMS_DATAPUMP.METADATA_FILTER(
  v_hdnl,
  'EXCLUDE_NAME_EXPR',
  q'[IN (SELECT NAME FROM SYS.OBJ$
         WHERE TYPE# IN (66,67,74,79,59,62,46)
         AND OWNER# IN
           (SELECT USER# FROM SYS.USER$
            WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
            )
        )
  ]',
  'PROCOBJ'
);
```

Wenn Sie `expdp` verwenden, erstellen Sie eine Parameterdatei, die die im folgenden Beispiel dargestellte `exclude`-Anweisung enthält. Dann benutze es `PARFILE=`*`parameter_file`* mit deinem `expdp` Kommando.

```
exclude=procobj:"IN
  (SELECT NAME FROM sys.OBJ$
   WHERE TYPE# IN (66,67,74,79,59,62,46)
   AND OWNER# IN
     (SELECT USER# FROM SYS.USER$
      WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
     )
  )"
```

Importieren von Daten mit Oracle Data Pump und einem Amazon S3-Bucket

Der folgende Importvorgang verwendet Oracle Data Pump und einen Amazon S3-Bucket. Die Schritte sind wie folgt:

1. Exportieren Sie Daten aus der Quelldatenbank mit dem Oracle DBMS_DATAPUMP-Paket.
2. Platzieren Sie die Dump-Datei in einem Amazon-S3-Bucket.

3. Laden Sie die Dump-Datei aus dem Amazon-S3-Bucket in das Verzeichnis `DATA_PUMP_DIR` auf der Ziel-DB-Instance von RDS für Oracle herunter.
4. Importieren Sie die Daten aus der kopierten Dump-Datei in die DB-Instance von RDS für Oracle mithilfe des `DBMS_DATAPUMP`-Pakets.

Themen

- Anforderungen zum Importieren von Daten mit Oracle Data Pump und einem Amazon-S3-Bucket
- Schritt 1: Erteilen von Berechtigungen für den Datenbankbenutzer auf der Ziel-DB-Instance von RDS für Oracle
- Schritt 2: Exportieren von Daten in eine Dump-Datei mit DBMS_DATAPUMP
- Schritt 3: Hochladen der Dumpdatei in Ihren Amazon S3-Bucket
- Schritt 4: Herunterladen der Dump-Datei aus Ihrem Amazon-S3-Bucket in Ihre DB-Instance.
- Schritt 5: Importieren Sie Ihre Dump-Datei mit DBMS_DATAPUMP in Ihre Ziel-DB-Instance
- Schritt 6: Bereinigen

Anforderungen zum Importieren von Daten mit Oracle Data Pump und einem Amazon-S3-Bucket

Der Vorgang hat folgende Anforderungen:

- Stellen Sie sicher, dass ein Amazon S3 S3-Bucket für Dateiübertragungen verfügbar ist und dass sich der Amazon S3 S3-Bucket in derselben Datenbank AWS-Region wie die DB-Instance befindet. Weitere Anleitungen finden Sie unter Erstellen eines Buckets im Amazon Simple Storage Service Handbuch "Erste Schritte".
- Die Größe des Objekts, das Sie zum Amazon S3-Bucket hochladen, darf höchstens 5 TB betragen. Weitere Informationen zur Arbeit mit Objekten in Amazon S3 finden Sie im Amazon Simple Storage Service User Guide.

> **Note**
>
> Wenn die Größe der Speicherabbilddatei 5 TB überschreitet, können Sie den Oracle Data Pump-Export mit Paralleloption ausführen. Diese Operation verteilt die Daten auf mehrere Speicherabbilddateien, sodass die Grenze von 5 TB für einzelne Dateien nicht überschritten wird.

- Sie müssen den Amazon S3-Bucket auf die Amazon RDS-Integration vorbereiten, indem Sie die Anleitungen unter befolge Konfigurieren von IAM-Berechtigungen für die Integration von RDS for Oracle in Amazon S3.
- Sie müssen sicherstellen, dass Sie über ausreichend Speicherplatz verfügen, um die Dump-Datei in der Quell-Instance und der Ziel-DB-Instance zu speichern.

> **Note**
>
> Dieser Vorgang importiert Dump-Dateien in das Verzeichnis `DATA_PUMP_DIR`, ein vorkonfiguriertes Verzeichnis auf allen Oracle-DB-Instances. Das Verzeichnis befindet sich im selben Speicher-Volume wie Ihre Datendateien. When Sie die Dump-Datei importieren, belegen die vorhandenen Oracle-Datendateien mehr Speicherplatz. Sie sollten daher sicherstellen, dass Ihre DB-Instance diesen zusätzlichen Platzbedarf erfüllen kann. Die importierte Dump-Datei wird nicht automatisch aus dem Verzeichnis `DATA_PUMP_DIR` gelöscht oder bereinigt. Zum Entfernen importierter Dump-Dateien verwenden Sie UTL_FILE.FREMOVE, das auf der Oracle-Website zu finden ist.

Schritt 1: Erteilen von Berechtigungen für den Datenbankbenutzer auf der Ziel-DB-Instance von RDS für Oracle

In diesem Schritt erstellen Sie Schemas, in die Sie Daten importieren möchten, und erteilen den Benutzern die erforderlichen Berechtigungen.

So erstellen Sie Benutzer und erteilen die erforderlichen Berechtigungen für die Ziel-Instance von RDS für Oracle

1. Verwenden Sie SQL*Plus oder Oracle SQL Developer, um sich als Hauptbenutzer bei der DB-Instance von RDS für Oracle anzumelden, in die die Daten importiert werden sollen. Weitere Information über das Verbinden mit der DB-Instance finden Sie unter Herstellen der Verbindung zu Ihrer Oracle-DB-Instance.
2. Erstellen Sie die erforderlichen Tabellenräume, bevor Sie Daten importieren. Weitere Informationen finden Sie unter Erstellen und Größenfestlegung von Tabellenräumen.
3. Wenn das Benutzerkonto, in das die Daten importiert werden sollen, nicht vorhanden ist, erstellen Sie das Benutzerkonto und erteilen Sie die erforderlichen Berechtigungen und Rollen. Wenn Sie Daten in mehrere Benutzerkonten importieren möchten, erstellen Sie alle Benutzerkonten und Rollen und erteilen Sie die erforderlichen Berechtigungen.

Die folgenden SQL-Anweisungen erstellen beispielsweise einen neuen Benutzer und gewähren die erforderlichen Berechtigungen und Rollen, um die Daten in das Schema zu importieren, das diesem Benutzer gehört. Ersetzen Sie *schema_1* durch den Namen Ihres Schemas in diesem Schritt und in den folgenden Schritten.

```
CREATE USER schema_1 IDENTIFIED BY my_password;
GRANT CREATE SESSION, RESOURCE TO schema_1;
ALTER USER schema_1 QUOTA 100M ON users;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Die vorangegangenen Anweisungen gewähren dem neuen Benutzer die `CREATE SESSION`-Berechtigung und die `RESOURCE`-Rolle. Je nach den Datenbankobjekten, die Sie importieren, benötigen Sie möglicherweise zusätzliche Berechtigungen und Rollen.

Schritt 2: Exportieren von Daten in eine Dump-Datei mit DBMS_DATAPUMP

Um eine Dump-Datei zu erstellen, verwenden Sie das `DBMS_DATAPUMP`-Paket.

Exportieren von Oracle-Daten in eine Dump-Datei

1. Verwenden Sie SQL Plus oder Oracle SQL Developer mit einem Benutzer mit Administratorrechten, um sich mit der DB-Instance von RDS für Oracle zu verbinden. Wenn die Quelldatenbank eine DB-Instance von RDS für Oracle ist, stellen Sie eine Verbindung mit dem Amazon-RDS-Hauptbenutzer her.
2. Exportieren Sie die Daten durch einen Aufruf der `DBMS_DATAPUMP`-Verfahren.

   Das folgende Skript exportiert das *SCHEMA_1*-Schema in eine Dump-Datei mit dem Namen `sample.dmp` im `DATA_PUMP_DIR`-Verzeichnis. Ersetzen Sie *SCHEMA_1* durch den Namen des Schemas, das Sie exportieren möchten.

   ```
   DECLARE
     v_hdnl NUMBER;
   BEGIN
     v_hdnl := DBMS_DATAPUMP.OPEN(
   ```

```
    operation => 'EXPORT',
    job_mode  => 'SCHEMA',
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl         ,
    filename  => 'sample.dmp'   ,
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_exp.log',
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM SYS.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Data Pump startet Jobs asynchron. Weitere Informationen über die Überwachung einer Data Pump-Aufgabe finden Sie unter Überwachung des Aufgabenstatus in der Oracle-Dokumentation.

3. (Optional) Sie können den Inhalt des Exportprotokolls mithilfe des `rdsadmin.rds_file_util.read_text_file`-Verfahrens anzeigen. Weitere Informationen finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis.

Schritt 3: Hochladen der Dumpdatei in Ihren Amazon S3-Bucket

Verwenden Sie die Amazon RDS-Prozedur `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`, um die Dump-Datei in den Amazon S3-Bucket zu kopieren. Das folgende Beispiel lädt alle Dateien aus dem Verzeichnis `DATA_PUMP_DIR` in einen Amazon S3-Bucket namens *`amzn-s3-demo-bucket`* hoch.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
  p_bucket_name    =>  'amzn-s3-demo-bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

Die Anweisung `SELECT` gibt die ID der Aufgabe in einem `VARCHAR2`-Datentyp zurück. Weitere Informationen finden Sie unter Dateien von Ihrer RDS Oracle-DB-Instance in einen Amazon S3 S3-Bucket hochladen.

Schritt 4: Herunterladen der Dump-Datei aus Ihrem Amazon-S3-Bucket in Ihre DB-Instance.

Führen Sie diesen Schritt mit dem Amazon-RDS-Verfahren `rdsadmin.rdsadmin_s3_tasks.download_from_s3` aus. Wenn Sie eine Datei in ein Verzeichnis herunterladen, wird der Download durch `download_from_s3` übersprungen, falls bereits eine gleichnamige Datei in dem Verzeichnis vorhanden ist. Zum Entfernen einer Datei aus dem Download-Verzeichnis verwenden Sie UTL_FILE.FREMOVE, das auf der Oracle-Website zu finden ist.

So laden Sie die Dump-Datei herunter

1. Starten Sie SQL*Plus oder Oracle SQL Developer und melden Sie sich als Hauptbenutzer für Ihre Ziel-Oracle-DB-Instance von Amazon RDS an.
2. Laden Sie die Dump-Datei mit dem Amazon-RDS-Verfahren `rdsadmin.rdsadmin_s3_tasks.download_from_s3` herunter.

   Das folgende Beispiel lädt alle Dateien von einem Amazon-S3-Bucket mit dem Namen *`amzn-s3-demo-bucket`* in das Verzeichnis `DATA_PUMP_DIR` herunter.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name    =>  'amzn-s3-demo-bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

Die Anweisung `SELECT` gibt die ID der Aufgabe in einem `VARCHAR2`-Datentyp zurück. Weitere Informationen finden Sie unter Hochladen von Dateien aus einem Amazon S3-Bucket zu einer Oracle-DB-Instance.

Schritt 5: Importieren Sie Ihre Dump-Datei mit DBMS_DATAPUMP in Ihre Ziel-DB-Instance

Verwenden Sie `DBMS_DATAPUMP`, um das Schema in Ihre DB-Instance von RDS für Oracle zu importieren. Zusätzliche Optionen wie `METADATA_REMAP` könnten erforderlich sein.

So importieren Sie Daten in Ihre Ziel-DB-Instance

1. Starten Sie SQL*Plus oder SQL Developer und melden Sie sich als Hauptbenutzer bei Ihrer DB-Instance von RDS für Oracle an.
2. Importieren Sie die Daten, indem Sie `DBMS_DATAPUMP` Prozeduren aufrufen.

   Im folgenden Beispiel *`SCHEMA_1`* werden die Daten aus `sample_copied.dmp` Ihrer Ziel-DB-Instance importiert.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
```

```
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Data Pump-Aufträge werden asynchron gestartet. Weitere Informationen über die Überwachung einer Data Pump-Aufgabe finden Sie unter Überwachung des Aufgabenstatus in der Oracle-Dokumentation. Sie können den Inhalt des Importprotokolls mithilfe des Verfahrens `rdsadmin.rds_file_util.read_text_file` anzeigen. Weitere Informationen finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis.

3. Überprüfen Sie den Datenimport, indem Sie die Schematabellen Ihrer Ziel-DB-Instance auflisten.

   Beispiel: Die folgende Abfrage gibt die Anzahl der Tabellen für zurüc *SCHEMA_1*.

   ```
   SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
   ```

Schritt 6: Bereinigen

Nachdem die Daten importiert wurden, können Sie die Dateien, die Sie nicht länger benötigen, löschen.

So entfernen Sie nicht benötigte Dateien

1. Starten Sie SQL*Plus oder SQL Developer und melden Sie sich als Hauptbenutzer bei Ihrer DB-Instance von RDS für Oracle an.
2. Listen Sie die Dateien in `DATA_PUMP_DIR` mit dem folgenden Befehl auf.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY
    MTIME;
   ```

3. Um Dateien in `DATA_PUMP_DIR` zu löschen, die nicht länger benötigt werden, verwenden Sie den folgenden Befehl.

   ```
   EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','filename');
   ```

So wird mit dem folgenden Befehl beispielsweise die Datei gelösch `sample_copied.dmp`.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Importieren von Daten mit Oracle Data Pump und einer Datenbankverbindung

Der folgende Importvorgang verwendet Oracle Data Pump und das Oracle-Paket DBMS_FILE_TRANSFER. Die Schritte sind wie folgt:

1. Stellen Sie eine Connect zu einer Oracle-Quelldatenbank her, bei der es sich um eine lokale Datenbank, eine EC2 Amazon-Instance oder eine RDS for Oracle DB-Instance handeln kann.
2. Exportieren Sie Daten mit dem DBMS_DATAPUMP-Paket.
3. Kopieren Sie damit die Dump-Datei mit `DBMS_FILE_TRANSFER.PUT_FILE` aus der Oracle-Datenbank in das `DATA_PUMP_DIR`-Verzeichnis auf der Ziel-DB-Instance von RDS für Oracle, die über einen Datenbank-Link verbunden ist.
4. Importieren Sie die Daten aus der kopierten Dump-Datei mithilfe des `DBMS_DATAPUMP`-Pakets in die RDS für Oracle DB-Instance.

Der Importvorgang mittels Oracle Data Pump und des `DBMS_FILE_TRANSFER`-Pakets besteht aus den folgenden Schritten.

Themen

- Anforderungen zm Importieren von Daten mit Oracle Data Pump und einer Datenbankverbindung
- Schritt 1: Erteilen von Berechtigungen für den Benutzer auf der Ziel-DB-Instance von RDS für Oracle
- Schritt 2: Erteilen von Berechtigungen für Benutzer in der Quell-Datenbank
- Schritt 3: Erstellen Sie eine Dump-Datei mithilfe von DBMS_DATAPUMP
- Schritt 4: Einen Datenbank-Link zur Ziel-DB-Instance erstellen
- Schritt 5: Kopieren der exportierten Dump-Datei mit DBMS_FILE_TRANSFER auf die Ziel-DB-Instance
- Schritt 6: Importieren der Datendatei in die Ziel-DB-Instance mit DBMS_DATAPUMP
- Schritt 7: Bereinigen

Anforderungen zm Importieren von Daten mit Oracle Data Pump und einer Datenbankverbindung

Der Vorgang hat folgende Anforderungen:

- Sie müssen über Ausführungsberechtigungen für die Pakete `DBMS_FILE_TRANSFER` und `DBMS_DATAPUMP` verfügen.
- Sie müssen über Schreibrechte für das Verzeichnis `DATA_PUMP_DIR` auf der Quell-DB-Instance verfügen.
- Sie müssen sicherstellen, dass Sie über ausreichend Speicherplatz verfügen, um die Dump-Datei in der Quell-Instance und der Ziel-DB-Instance zu speichern.

> **Note**
>
> Dieser Vorgang importiert Dump-Dateien in das Verzeichnis `DATA_PUMP_DIR`, ein vorkonfiguriertes Verzeichnis auf allen Oracle-DB-Instances. Das Verzeichnis befindet sich im selben Speicher-Volume wie Ihre Datendateien. When Sie die Dump-Datei importieren, belegen die vorhandenen Oracle-Datendateien mehr Speicherplatz. Sie sollten daher sicherstellen, dass Ihre DB-Instance diesen zusätzlichen Platzbedarf erfüllen kann. Die importierte Dump-Datei wird nicht automatisch aus dem Verzeichnis `DATA_PUMP_DIR` gelöscht oder bereinigt. Zum Entfernen importierter Dump-Dateien verwenden Sie UTL_FILE.FREMOVE, das auf der Oracle-Website zu finden ist.

Schritt 1: Erteilen von Berechtigungen für den Benutzer auf der Ziel-DB-Instance von RDS für Oracle

Um dem Benutzer auf der Ziel-DB-Instance von RDS für Oracle Berechtigungen zu erteilen, führen Sie die folgenden Schritte aus:

1. Importieren Sie die Daten aus der kopierten Dump-Datei mit Hilfe des Pakets in die RDS für Oracle DB-Instance. Stellen Sie eine Verbindung als Amazon RDS-Master-Benutzer her. Weitere Information über das Verbinden mit der DB-Instance finden Sie unter Herstellen der Verbindung zu Ihrer Oracle-DB-Instance.
2. Erstellen Sie die erforderlichen Tabellenräume, bevor Sie Daten importieren. Weitere Informationen finden Sie unter Erstellen und Größenfestlegung von Tabellenräumen.
3. Wenn das Benutzerkonto, in das die Daten importiert werden sollen, nicht vorhanden ist, erstellen Sie das Benutzerkonto und die Rollen und erteilen Sie die erforderlichen Berechtigungen. Wenn

Sie Daten in mehrere Benutzerkonten importieren möchten, erstellen Sie alle Benutzerkonten und Rollen und erteilen Sie die erforderlichen Berechtigungen.

Mit den folgenden Befehlen wird beispielsweise ein neuer Benutzer mit dem Namen erstellt *schema_1* und die erforderlichen Berechtigungen und Rollen erteilt, um die Daten für diesen Benutzer in das Schema zu importieren.

```
CREATE USER schema_1 IDENTIFIED BY my-password;
GRANT CREATE SESSION, RESOURCE TO schema_1;
ALTER USER schema_1 QUOTA 100M ON users;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Im vorherigen Beispiel wird dem neuen Benutzer die Berechtigung `CREATE SESSION` und die Rolle `RESOURCE` erteilt. Möglicherweise sind je nach zu importierenden Datenbankobjekten zusätzliche Berechtigungen und Rollen erforderlich.

> **Note**
>
> Ersetzen Sie *schema_1* durch den Namen Ihres Schemas in diesem Schritt und in den folgenden Schritten.

Schritt 2: Erteilen von Berechtigungen für Benutzer in der Quell-Datenbank

Verwenden Sie SQL*Plus oder Oracle SQL Developer, um eine Verbindung mit einer DB-Instance von RDS für Oracle herzustellen, die die zu importierenden Daten enthält. Falls nötig, erstellen Sie ein Benutzerkonto und gewähren die notwendigen Berechtigungen.

> **Note**
>
> Wenn die Quell-Datenbank eine Amazon RDS-Instance ist, können Sie diesen Schritt übergehen. Sie verwenden Ihr Amazon RDS-Master-Benutzerkonto zum Exportieren.

Die folgenden Befehle erstellen einen neuen Benutzer und gewähren die notwendigen Berechtigungen.

```
CREATE USER export_user IDENTIFIED BY my-password;
GRANT CREATE SESSION, CREATE TABLE, CREATE DATABASE LINK TO export_user;
ALTER USER export_user QUOTA 100M ON users;
GRANT READ, WRITE ON DIRECTORY data_pump_dir TO export_user;
GRANT SELECT_CATALOG_ROLE TO export_user;
GRANT EXECUTE ON DBMS_DATAPUMP TO export_user;
GRANT EXECUTE ON DBMS_FILE_TRANSFER TO export_user;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Schritt 3: Erstellen Sie eine Dump-Datei mithilfe von DBMS_DATAPUMP

Gehen Sie folgendermaßen vor, um eine Dump-Datei zu erstellen:

1. Verwenden Sie SQL*Plus oder Oracle SQL Developer, um sich mit einem administrativen Benutzer oder mit dem in Schritt 2 erstellten Benutzer mit der Oracle-Quell-Instance zu verbinden. Wenn die Quelldatenbank eine DB-Instance von Amazon RDS für Oracle ist, stellen Sie eine Verbindung mit dem Amazon-RDS-Hauptbenutzer her.
2. Erstellen Sie eine Dump-Datei mithilfe des Oracle Data Pump-Dienstprogramms.

   Das folgende Skript erstellt im Verzeichnis `DATA_PUMP_DIR` eine Dump-Datei mit dem Namen sample.dmp.

   ```
   DECLARE
     v_hdnl NUMBER;
   BEGIN
     v_hdnl := DBMS_DATAPUMP.OPEN(
       operation => 'EXPORT' ,
       job_mode  => 'SCHEMA' ,
       job_name  => null
     );
     DBMS_DATAPUMP.ADD_FILE(
       handle    => v_hdnl,
       filename  => 'sample.dmp'    ,
       directory => 'DATA_PUMP_DIR' ,
   ```

```
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl           ,
    filename  => 'sample_exp.log' ,
    directory => 'DATA_PUMP_DIR'  ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl              ,
    'SCHEMA_EXPR'       ,
    'IN (''SCHEMA_1'')'
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM sys.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
         )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Data Pump-Aufträge werden asynchron gestartet. Weitere Informationen über die Überwachung einer Data Pump-Aufgabe finden Sie unter Überwachung des Aufgabenstatus in der Oracle-Dokumentation. Sie können den Inhalt des Exportprotokolls mithilfe des Verfahrens `rdsadmin.rds_file_util.read_text_file` anzeigen. Weitere Informationen finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis.

Schritt 4: Einen Datenbank-Link zur Ziel-DB-Instance erstellen

Erstellen Sie einen Datenbank-Link zwischen Ihrer Quell-DB-Instance und Ihrer Ziel-DB-Instance. Ihre lokale Oracle-Instance muss über eine Netzwerkverbindung mit der DB-Instance verfügen, damit eine Datenbankverbindung erstellt und Ihre Dumpdatei übermittelt werden kann.

Führen Sie diesen Schritt mit demselben Benutzerkonto wie im vorherigen Schritt aus.

Wenn Sie einen Datenbank-Link zwischen zwei DB-Instances innerhalb derselben VPC oder Peering erstellen VPCs, sollten die beiden DB-Instances über eine gültige Route zwischen ihnen verfügen. Die Sicherheitsgruppe jeder DB-Instance muss den Eintritt und den Austritt von einer zur anderen DB-Instance erlauben. Die eingehenden und die ausgehenden Regeln der Sicherheitsgruppe können sich auf Sicherheitsgruppen aus derselben VPC oder aus einer gleichrangigen VPC beziehen. Weitere Informationen finden Sie unter Anpassen von Datenbank-Links für die Verwendung mit DB-Instances in einem VPC.

Mit dem folgenden Befehl wird eine Datenbankverbindung mit dem Namen `to_rds` erstellt, die eine Verbindung mit dem Amazon RDS-Master-Benutzer auf der Ziel-DB-Instance herstellt.

```
CREATE DATABASE LINK to_rds
  CONNECT TO <master_user_account> IDENTIFIED BY <password>
  USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>)
         (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';
```

Schritt 5: Kopieren der exportierten Dump-Datei mit DBMS_FILE_TRANSFER auf die Ziel-DB-Instance

Verwenden Sie `DBMS_FILE_TRANSFER`, um die Dump-Datei aus der Datenbank der Quell-Instance zur Ziel-DB-Instance zu kopieren. Mit dem folgenden Skript wird eine Dumpdatei mit dem Namen sample.dmp aus der Quell-Instance zu einer Ziel-Datenbankverbindung mit dem Namen to_rds (im vorherigen Schritt erstellt) kopiert.

```
BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object       => 'DATA_PUMP_DIR',
    source_file_name              => 'sample.dmp',
    destination_directory_object  => 'DATA_PUMP_DIR',
    destination_file_name         => 'sample_copied.dmp',
    destination_database          => 'to_rds' );
END;
/
```

Schritt 6: Importieren der Datendatei in die Ziel-DB-Instance mit DBMS_DATAPUMP

Verwenden Sie Oracle Data Pump, um das Schema in die DB-Instance zu kopieren. Möglicherweise sind zusätzliche Optionen wie METADATA_REMAP erforderlich.

Stellen Sie zum Importieren unter Verwendung des Amazon RDS-Master-Benutzerkontos eine Verbindung zur DB-Instance her.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Data Pump-Aufträge werden asynchron gestartet. Weitere Informationen über die Überwachung einer Data Pump-Aufgabe finden Sie unter Überwachung des Aufgabenstatus in der Oracle-Dokumentation. Sie können den Inhalt des Importprotokolls mithilfe des Verfahrens `rdsadmin.rds_file_util.read_text_file` anzeigen. Weitere Informationen finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis.

Sie können den Datenimport überprüfen, indem Sie die Tabelle des Benutzers zur DB-Instance aufrufen. Beispiel: Die folgende Abfrage gibt die Anzahl der Tabellen für zurüc *schema_1*.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Schritt 7: Bereinigen

Nachdem die Daten importiert wurden, können Sie die Dateien, die Sie nicht länger benötigen, löschen. Sie können die Dateien in DATA_PUMP_DIR mit dem folgenden Befehl auflisten.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

Um Dateien in DATA_PUMP_DIR zu löschen, die nicht länger benötigt werden, verwenden Sie den folgenden Befehl:

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','<file name>');
```

So wird mit dem folgenden Befehl beispielsweise die Datei "sample_copied.dmp" gelöscht.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Import unter Verwendung von Oracle-Export/-Import

Unter den folgenden Bedingungen können Sie Oracle-Export/-Import-Dienstprogramme für Migrationen in Betracht ziehen:

- Ihre Datengröße ist klein.
- Datentypen wie „Binary Float“ und „Double“ sind nicht erforderlich.

Beim Importvorgang werden die erforderlichen Schemaobjekte erstellt. So müssen Sie nicht selbst ein Skript ausführen, um diese Objekte vorab zu erstellen.

Die einfachste Möglichkeit, die Export- und Importdienstprogramme von Oracle zu installieren, ist die Installation des Oracle Instant Client. Wenn Sie die Software herunterladen möchten, gehen Sie zu https://www.oracle.com/database/technologies/instant-client.html. Die Dokumentation finden Sie unter Instant Client for SQL*Loader, Export, and Import im Handbuch Oracle Database Utilities.

So exportieren Sie Tabellen und importieren sie dann

1. Exportieren Sie Tabellen aus der Quelldatenbank mit dem exp-Befehl.

Der folgende Befehl exportiert die Tabellen mit den Namen `tab1`, `tab2` und `tab3`. Die Dump-Datei ist `exp_file.dmp`.

```
exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
```

Der Export erstellt eine binäre Dump-Datei, die sowohl das Schema als auch die Daten der angegebenen Tabellen enthält.

2. Importieren Sie das Schema und die Daten mit dem `imp`-Befehl in eine Zieldatenbank.

   Der folgende Befehl importiert die Tabellen `tab1`, `tab2` und `tab3` aus der Dump-Datei `exp_file.dmp`.

```
imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \
TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
```

Export und Import haben andere Varianten, die Ihren Bedürfnissen eher entsprechen könnten. Alle Einzelheiten finden Sie in der Oracle-Datenbank-Dokumentation.

Importieren mit Oracle SQL*Loader

Sie können Oracle SQL*Loader für große Datenbanken mit einer beschränkten Anzahl an Objekten in Betracht ziehen. Da der Vorgang für den Export aus einer Quelldatenbank und das Laden in eine Zieldatenbank genau an das Schema angepasst ist, werden im folgenden Beispiel Schema-Objekte erstellt, aus einer Quelle exportiert und anschließend die Daten in eine Zieldatenbank geladen.

Die einfachste Möglichkeit, Oracle SQL*Loader zu installieren, ist die Installation des Oracle Instant Client. Wenn Sie die Software herunterladen möchten, gehen Sie zu https://www.oracle.com/database/technologies/instant-client.html. Die Dokumentation finden Sie unter Instant Client for SQL*Loader, Export, and Import im Handbuch Oracle Database Utilities.

So importieren Sie Daten mit Oracle SQL*Loader

1. Erstellen Sie mit der folgenden SQL-Anweisung eine Beispiel-Quelltabelle.

```
CREATE TABLE customer_0 TABLESPACE users
   AS (SELECT ROWNUM id, o.*
       FROM   ALL_OBJECTS o, ALL_OBJECTS x
```

```
       WHERE  ROWNUM <= 1000000);
```

2. Erstellen Sie für die Ziel-Instance von RDS für Oracle DB eine Zieltabelle zum Laden der Daten. Die Klausel `WHERE 1=2` stellt sicher, dass Sie die Struktur von `ALL_OBJECTS`, aber keine der Zeilen kopieren.

```
CREATE TABLE customer_1 TABLESPACE users
  AS (SELECT 0 AS ID, OWNER, OBJECT_NAME, CREATED
      FROM   ALL_OBJECTS
      WHERE  1=2);
```

3. Exportieren Sie die Daten aus der Quelldatenbank in eine Textdatei. Im folgenden Beispiel wird SQL*Plus verwendet. Für Ihre Daten werden Sie höchstwahrscheinlich ein Skript erstellen müssen, das den Export für alle Objekte in der Datenbank übernimmt.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS'

SET LINESIZE 800 HEADING OFF FEEDBACK OFF ARRAY 5000 PAGESIZE 0
SPOOL customer_0.out
SET MARKUP HTML PREFORMAT ON
SET COLSEP ','

SELECT id, owner, object_name, created
FROM   customer_0;

SPOOL OFF
```

4. Erstellen Sie eine Steuerungsdatei, um die Daten zu beschreiben. Sie müssen eventuell ein Skript schreiben, damit Sie diesen Schritt durchführen können.

```
cat << EOF > sqlldr_1.ctl
load data
infile customer_0.out
into table customer_1
APPEND
fields terminated by "," optionally enclosed by '"'
(
  id           POSITION(01:10)    INTEGER EXTERNAL,
  owner        POSITION(12:41)    CHAR,
  object_name  POSITION(43:72)    CHAR,
  created      POSITION(74:92)    date "YYYY/MM/DD HH24:MI:SS"
)
```

Falls notwendig, kopieren Sie die erstellten Dateien mit dem vorstehenden Code in einen Bereitstellungsbereich, wie zum Beispiel eine Amazon EC2-Instance.

5. Importieren Sie die Daten mit SQL*Loader mit dem entsprechenden Benutzernamen und Passwort für die Zieldatenbank.

```
sqlldr cust_dba@targetdb CONTROL=sqlldr_1.ctl BINDSIZE=10485760 READSIZE=10485760
 ROWS=1000
```

Migrieren mit materialisierten Oracle-Ansichten

Um große Datenmengen effizient zu migrieren, können Sie die Oracle-Replikation von materialisierten Ansichten verwenden. Mit der Replikation können Sie die Zieltabellen mit den Quelltabellen synchronisieren. So können Sie bei Bedarf RDS später zu Amazon wechseln.

Bevor Sie mit materialisierten Ansichten migrieren können, müssen Sie sicherstellen, dass Sie die folgenden Anforderungen erfüllen:

- Konfigurieren Sie den Zugriff von der Zieldatenbank auf die Quelldatenbank. Im folgenden Beispiel wurden Zugriffsregeln für die Quelldatenbank aktiviert, damit die Zieldatenbank RDS für Oracle SQL über*Net eine Verbindung zur Quelle herstellen kann.
- Erstellen Sie einen Datenbank-Link von der RDS for Oracle-DB-Instance zur Quelldatenbank.

So migrieren Sie Daten mithilfe materialisierter Ansichten

1. Erstellen Sie ein Benutzerkonto sowohl für Quell- als auch RDS für Oracle-Ziel-Instances, das sich mit demselben Passwort authentifizieren kann. Im folgenden Beispiel wird ein Benutzer mit dem Namen `dblink_user` erstellt.

```
CREATE USER dblink_user IDENTIFIED BY my-password
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION TO dblink_user;

GRANT SELECT ANY TABLE TO dblink_user;

GRANT SELECT ANY DICTIONARY TO dblink_user;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

2. Erstellen Sie mit Ihrem neu erstellten Benutzer einen Datenbank-Link von der Ziel-Instance RDS für Oracle zur Quell-Instance.

```
CREATE DATABASE LINK remote_site
  CONNECT TO dblink_user IDENTIFIED BY my-password
  USING '(description=(address=(protocol=tcp) (host=my-host)
    (port=my-listener-port)) (connect_data=(sid=my-source-db-sid)))';
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

3. Testen Sie die Verbindung:

```
SELECT * FROM V$INSTANCE@remote_site;
```

4. Erstellen Sie eine Beispiel-Tabelle mit einem Primärschlüssel und einem Protokoll für materialisierte Ansichten in der Quell-Instance.

```
CREATE TABLE customer_0 TABLESPACE users
  AS (SELECT ROWNUM id, o.*
      FROM   ALL_OBJECTS o, ALL_OBJECTS x
      WHERE  ROWNUM <= 1000000);

ALTER TABLE customer_0 ADD CONSTRAINT pk_customer_0 PRIMARY KEY (id) USING INDEX;

CREATE MATERIALIZED VIEW LOG ON customer_0;
```

5. Erstellen Sie auf der Ziel-DB-Instance RDS für die Oracle-DB-Instance eine materialisierte Ansicht.

```
CREATE MATERIALIZED VIEW customer_0
  BUILD IMMEDIATE REFRESH FAST
  AS (SELECT *
      FROM   cust_dba.customer_0@remote_site);
```

6. Aktualisieren Sie auf dem Ziel RDS für die Oracle-DB-Instance die materialisierte Ansicht.

```
EXEC DBMS_MVIEW.REFRESH('CUSTOMER_0', 'f');
```

7. Verwerfen Sie die materialisierte Ansicht und schließen Sie die PRESERVE TABLE-Klausel ein, um die Container-Tabelle der materialisierten Ansicht und deren Inhalt beizubehalten.

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

   Die beibehaltene Tabelle hat denselben Namen wie die entfernte materialisierte Ansicht.

Arbeiten mit Lese-Replikaten für Amazon RDS für Oracle

Um die Replikation zwischen Oracle DB-Instances zu konfigurieren, können Sie Replikat-Datenbanken erstellen. Eine Übersicht über Amazon-RDS-Lesereplikate finden Sie unter Übersicht über Amazon RDS-Lesereplikate. Eine Zusammenfassung der Unterschiede zwischen Oracle-Replikaten und anderen DB-Engines finden Sie unter Unterschiede zwischen Read Replicas für DB-Engines.

Themen

- Übersicht über Replikate von RDS für Oracle
- Anforderungen und Überlegungen zu Backup und Wiederherstellung für RDS-für-Oracle-Replikate
- Vorbereiten der Erstellung eines Oracle-Replikats
- Erstellen eines Replikats von RDS für Oracle im aufgespielten Modus
- Ändern des Replikatmodus von RDS für Oracle
- Arbeiten mit RDS-für-Oracle-Replikat-Backups
- So führen Sie eine Oracle Data Guard-Umschaltung aus
- Fehlerbehebung bei Replikaten von RDS für Oracle

Übersicht über Replikate von RDS für Oracle

Eine Oracle-Replikatdatenbank ist eine physische Kopie Ihrer primären Datenbank. Ein Oracle-Replikat im schreibgeschützten Modus wird als Read Replica bezeichnet. Ein Oracle-Replikat im aufgespielten Modus wird als aufgespieltes Replikat bezeichnet. Oracle Database lässt keine Schreibvorgänge in einem Replikat zu, aber Sie können ein Replikat heraufstufen, um es beschreibbar zu machen. Das hochgestufte Lesereplikat weist die replizierten Daten bis zu dem Punkt auf, an dem die Anforderung zum Hochstufen ausgegeben wurde.

Das folgende Video bietet eine hilfreiche Übersicht über die Notfallwiederherstellung von RDS für Oracle.

Weitere Informationen finden Sie im Blogbeitrag Verwaltete Notfallwiederherstellung mit Amazon RDS for Oracle, regionsübergreifende automatisierte Backups – Teil 1 und Verwaltete Notfallwiederherstellung mit Amazon RDS for Oracle, regionsübergreifende automatisierte Backups – Teil 2.

Themen

- Schreibgeschützte und aufgespielte Replikate
- Replikate von CDBs lesen
- Archivierte Redo-Protokollaufbewahrung
- Ausfälle während der Oracle-Replikation

Schreibgeschützte und aufgespielte Replikate

Wenn Sie ein Oracle-Replikat erstellen oder ändern, können Sie es in einen der folgenden Modi versetzen:

Read-only

Dies ist die Standardeinstellung. Active Data Guard überträgt und wendet Änderungen aus der Quelldatenbank auf alle Read-Replica-Datenbanken an.

Sie können bis zu fünf Lesereplikate aus einer Quell-DB-Instance erstellen. Allgemeine Hinweise zu Read Replicas, die für alle DB-Engines gelten, finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten. Informationen zu Oracle Data Guard finden Sie unter Oracle-Data-Guard-Konzepte und -Administration in der Oracle-Dokumentation.

Bindungsbereitstellung

In diesem Fall wird für die Replikation Oracle Data Guard verwendet, die Replikatdatenbank akzeptiert jedoch keine Benutzerverbindungen. Die primäre Verwendung für aufgespielte Replikate ist die überregionale Notfallwiederherstellung.

Ein aufgespieltes Replikat kann keine schreibgeschützte Workload bereitstellen. Das aufgespielte Replikat löscht archivierte Redo-Protokolldateien, nachdem sie angewendet wurden, unabhängig von der Aufbewahrungsrichtlinie für archivierte Protokolle.

Sie können eine Kombination aus aufgespielten und schreibgeschützten DB-Replikaten für dieselbe Quell-DB-Instance erstellen. Sie können ein schreibgeschütztes Replikat in den aufgespielten Modus oder ein aufgespieltes Replikat in den schreibgeschützten Modus versetzen. In beiden Fällen behält die Oracle-Datenbank die Einstellung für die Aufbewahrung archivierter Protokolle bei.

Replikate von CDBs lesen

RDS für Oracle unterstützt Data-Guard-Lesereplikate für CDBs von Oracle Database 19c und 21c nur in der Single-Tenant-Konfiguration. Sie können Lesereplikate in einer CDB genauso wie in einer Non-

CDB erstellen, verwalten und hochstufen. Bereitgestellte Replikate werden ebenfalls unterstützt. Für Sie ergeben sich folgende Vorteile:

- Verwaltete Notfallwiederherstellung, hohe Verfügbarkeit und schreibgeschützter Zugriff auf Ihre Replikate
- Die Möglichkeit, Read Replicas in einem anderen AWS-Region zu erstellen.
- Integration mit den vorhandenen RDS-Read Replica-APIs: CreateDB InstanceReadReplica, und PromoteReadReplicaSwitchoverReadReplica

Um diese Funktion nutzen zu können, benötigen Sie eine aktive Data-Guard-Lizenz und eine Lizenz für Oracle Database Enterprise Edition sowohl für das Replikat als auch für die primären DB-Instances. Für die Verwendung der CDB-Architektur fallen keine zusätzlichen Kosten an. Sie zahlen nur für Ihre DB-Instances.

Weitere Informationen zur Single- und zur Multi-Tenant-Konfiguration der CDB-Architektur finden Sie unter Überblick über RDS for Oracle CDBs.

Archivierte Redo-Protokollaufbewahrung

Wenn eine primäre DB-Instance keine regionsübergreifenden Lesereplikate hat, behält Amazon RDS für Oracle archivierte Redo-Protokolle im Umfang von mindestens zwei Stunden auf der Quell-DB-Instance bei. Dies gilt unabhängig von der Einstellung für `archivelog retention hours` in `rdsadmin.rdsadmin_util.set_configuration`.

RDS löscht Protokolle aus der Quell-DB-Instance nach zwei Stunden oder nach Ablauf der Einstellung für die Aufbewahrungsdauer des Archivprotokolls, je nachdem, welcher Zeitraum länger ist. RDS löscht Protokolle aus dem Lesereplikat, nachdem die Einstellung für die Aufbewahrungsdauer des Archivprotokolls abgelaufen ist, nur wenn sie erfolgreich auf die Datenbank angewendet wurden.

In manchen Fällen kann eine primäre DB-Instance ein oder mehrere regionsübergreifende Lesereplikate haben. Wenn dies der Fall ist, behält Amazon RDS for Oracle die Transaktionsprotokolle auf der Quell-DB-Instance bei, bis diese übertragen und auf alle regionsübergreifenden Lesereplikate angewendet wurden. Weitere Informationen zu `rdsadmin.rdsadmin_util.set_configuration` finden Sie unter Beibehaltung von archivierten Redo-Protokollen.

Ausfälle während der Oracle-Replikation

Wenn Sie ein Lesereplikat erstellen, fertigt Amazon RDS einen DB-Snapshot Ihrer Quell-DB-Instance an und beginnt mit der Replikation. Bei der Quell-DB-Instance kommt es zu einer sehr kurzen I/O-Unterbrechung, wenn der DB-Snapshot-Vorgang beginnt. Die I/O-Unterbrechung dauert in der Regel etwa eine Sekunde. Sie können die I/O-Aussetzung vermeiden, wenn es sich bei der DB-Instance um eine Multi-AZ-Bereitstellung handelt, weil in diesem Fall der Snapshot von der sekundären DB-Instance aufgenommen wird.

Der DB-Snapshot wird zum Oracle-Replikat. Amazon RDS legt die erforderlichen Parameter und Berechtigungen für die Quelldatenbank und das Replikat ohne Serviceunterbrechung fest. Wenn Sie ein Replikat löschen, tritt ebenfalls kein Ausfall auf.

Anforderungen und Überlegungen zu Backup und Wiederherstellung für RDS-für-Oracle-Replikate

Machen Sie sich vor dem Erstellen eines Oracle-Replikats mit den folgenden Anforderungen und Überlegungen vertraut.

Themen

- Versions- und Lizenzierungsanforderungen für RDS-für-Oracle-Replikate
- Einschränkungen von Optionsgruppen für RDS for Oracle Replicas
- Überlegungen zu Backup und Wiederherstellung für RDS-für-Oracle-Replikate
- Anforderungen und Einschränkungen von Oracle Data Guard für Replikate von RDS für Oracle
- Sonstige Überlegungen zu RDS-für-Oracle-Replikaten

Versions- und Lizenzierungsanforderungen für RDS-für-Oracle-Replikate

Beachten Sie vor dem Erstellen eines RDS-für-Oracle-Replikats die folgenden Punkte:

- Wenn sich das Replikat im schreibgeschützten Modus befindet, stellen Sie sicher, dass Sie über eine Active Data Guard-Lizenz verfügen. Wenn Sie das Replikat in den aufgespielten Modus versetzen, benötigen Sie keine Active Data Guard-Lizenz. Nur die Oracle DB-Engine unterstützt aufgespielte Replikate.
- Oracle-Replikate werden nur für Oracle Enterprise Edition (EE) unterstützt.
- Oracle-Repliken von Non- CDBs werden nur für DB-Instances unterstützt, die mit Nicht-CDB-Instances erstellt wurden, auf denen Oracle Database 19c ausgeführt wird.

- Oracle-Repliken sind für DB-Instances verfügbar, die nur auf DB-Instance-Klassen mit zwei oder mehr v ausgeführt werden. CPUs Eine Quell-DB-Instance kann die Instance-Klasse db.t3.small nicht verwenden.
- Die Oracle-DB-Engine-Version der Quell-DB-Instance und all ihrer Replikate müssen identisch sein. Amazon RDS aktualisiert die Replikate sofort nach dem Upgrade der Quell-DB-Instance, ungeachtet des Wartungsfensters des Replikats. Für größere Versions-Upgrades von regionsübergreifenden Replikaten führt Amazon RDS automatisch Folgendes durch:
  - Automatisches Generieren einer Optionsgruppe für die Zielversion
  - Kopieren aller Optionen und Optionseinstellungen aus der ursprünglichen Optionsgruppe in die neue Optionsgruppe
  - Verknüpfen des aktualisierten regionsübergreifenden Replikats mit der neuen Optionsgruppe

  Weitere Informationen zum Aktualisieren der DB-Engine-Version finden Sie unter Aktualisieren der DB-Engine von RDS für Oracle.

Einschränkungen von Optionsgruppen für RDS for Oracle Replicas

Beachten Sie bei der Arbeit mit Optionsgruppen für Ihr RDS for Oracle-Replikat Folgendes:

- Sie können keine andere Replikat-Optionsgruppe als die Quell-DB-Instance-Optionsgruppe verwenden, wenn sich Quelle und Replikat in derselben Region befinden. AWS

  Änderungen an der Quelloptionsgruppe oder der Mitgliedschaft in der Quelloptionsgruppe werden auf Oracle-Replikate übertragen. Diese Änderungen werden unmittelbar, nachdem sie auf die Quell-DB-Instance angewandt wurden, auf die Replikate angewandt, ungeachtet des Wartungsfensters des Replikats. Weitere Informationen über Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.
- Sie können ein regionsübergreifendes Replikat von RDS for Oracle nicht aus der zugehörigen Optionsgruppe entfernen, die automatisch für das Replikat erstellt wird.
- Sie können die dedizierte Optionsgruppe für ein regionsübergreifendes RDS for Oracle-Replikat nicht zu einer anderen DB-Instance hinzufügen.
- Sie können einer dedizierten Optionsgruppe für ein regionsübergreifendes RDS for Oracle-Replikat keine nicht replizierten Optionen hinzufügen oder daraus entfernen, mit Ausnahme der folgenden Optionen:
  - `NATIVE_NETWORK_ENCRYPTION`
  - `OEM`

- `OEM_AGENT`
- `SSL`

  Um einem regionsübergreifenden Replikat von RDS für Oracle weitere Optionen hinzuzufügen, fügen Sie diese der Optionsgruppe der Quell-DB-Instance hinzu. Die Option wird auch auf allen Replikaten der Quell-DB-Instance installiert. Stellen Sie bei lizenzierten Optionen sicher, dass genügend Lizenzen für die Replikate vorhanden sind.

  Wenn Sie ein regionsübergreifendes Replikat von RDS für Oracle hochstufen, verhält sich das hochgestufte Replikat genau so wie andere Oracle-DB-Instances, einschließlich der Verwaltung seiner Optionen. Sie können ein Replikat explizit oder implizit hochstufen, indem Sie seine Quell-DB-Instance löschen.

  Weitere Informationen über Optionsgruppen finden Sie unter Arbeiten mit Optionsgruppen.
- Sie können die `EFS_INTEGRATION` Option nicht zu RDS für regionsübergreifende Oracle-Replikate hinzufügen.

Überlegungen zu Backup und Wiederherstellung für RDS-für-Oracle-Replikate

Beachten Sie vor dem Erstellen eines RDS-für-Oracle-Replikats die folgenden Punkte:

- Um Snapshots von RDS-für-Oracle-Replikate zu erstellen oder automatische Backups zu aktivieren, stellen Sie sicher, dass Sie den Aufbewahrungszeitraum für Backups manuell festlegen. Automatische Backups sind standardmäßig nicht aktiviert.
- Wenn Sie ein Replikat-Backup wiederherstellen, erfolgt die Wiederherstellung auf die Datenbankzeit, nicht auf die Uhrzeit, zu der das Backup erstellt wurde. Die Datenbank-Zeit bezieht sich auf die letzte angewendete Transaktionszeit der Daten im Backup. Der Unterschied ist erheblich, da ein Replikat Minuten oder Stunden hinter dem primären Replikat zurückbleiben kann.

  Um den Unterschied zu finden, verwenden Sie den `describe-db-snapshots`-Befehl. Vergleichen Sie `snapshotDatabaseTime`, das ist die Datenbankzeit des Replikat-Backups, und `OriginalSnapshotCreateTime`-Feld, das die letzte angewendete Transaktion in der Primärdatenbank darstellt.

Anforderungen und Einschränkungen von Oracle Data Guard für Replikate von RDS für Oracle

Beachten Sie die folgenden Anforderungen und Einschränkungen, bevor Sie ein Replikat von RDS für Oracle erstellen:

- Wenn Ihre primäre DB-Instance die Single-Tenant-Konfiguration der Multi-Tenant-Architektur verwendet, sollten Sie Folgendes berücksichtigen:
  - Sie müssen Oracle Database 19c oder höher mit der Enterprise Edition verwenden.
  - Ihre primäre CDB-Instance muss sich in einem `ACTIVE`-Lebenszyklus befinden.
  - Sie können eine Nicht-CDB-Primär-Instance nicht in eine CDB-Instance und ihre Replikate nicht im gleichen Vorgang konvertieren. Löschen Sie stattdessen die Nicht-CDB-Replikate, konvertieren Sie die primäre DB-Instance in eine CDB und erstellen Sie dann neue Replikate
- Stellen Sie sicher, dassin Anmeldeauslöser auf einer primären DB-Instance dem `RDS_DATAGUARD`-Benutzer und jedem Benutzer, dessen `AUTHENTICATED_IDENTITY`-Wert `RDS_DATAGUARD` oder `rdsdb` ist, den Zugriff erlauben. Außerdem darf der Auslöser das aktuelle Schema für den `RDS_DATAGUARD`-Benutzer nicht festlegen.
- Um zu vermeiden, dass Verbindungen vom Data Guard-Brokerprozess blockiert werden, aktivieren Sie keine eingeschränkten Sitzungen. Weitere Informationen zu eingeschränkten Sitzungen finden Sie unter Aktivieren und Deaktivieren von beschränkten Sitzungen.

Sonstige Überlegungen zu RDS-für-Oracle-Replikaten

Beachten Sie vor dem Erstellen eines RDS-für-Oracle-Replikats die folgenden Punkte:

- Wenn Ihre DB-Instance eine Quelle für ein oder mehrere regionsübergreifende Replikate ist, behält die Quell-DB ihre archivierten Redo-Log-Dateien bei, bis sie auf alle regionsübergreifenden Replikate angewendet werden. Die archivierten Redo-Protokolle können zu erhöhter Speichernutzung führen.
- Um die RDS-Automatisierung nicht zu beeinträchtigen, müssen Systemauslöser bestimmten Benutzern die Anmeldung bei der primären und der Replikatdatenbank ermöglichen. Zu den Systemauslösern gehören Auslöser für DDL, Anmeldung und Datenbankrollen. Wir empfehlen Ihnen, Code zu Ihren Auslösern hinzuzufügen, um die im folgenden Beispielcode aufgeführten Benutzer auszuschließen:

```
-- Determine who the user is
```

```
SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') INTO CURRENT_USER FROM DUAL;
-- The following users should always be able to login to either the Primary or
 Replica
IF CURRENT_USER IN ('master_user', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'rdsdb') THEN
RETURN;
END IF;
```

- Die Nachverfolgung von Blockänderungen wird für schreibgeschützte Replikate unterstützt, nicht jedoch für bereitgestellte Replikate. Sie können ein aufgespieltes Replikat in ein schreibgeschütztes Replikat ändern und dann die Blockänderungsverfolgung aktivieren. Weitere Informationen finden Sie unter Aktivieren und Deaktivieren der Nachverfolgung von Blockänderungen.
- Sie können keine Oracle-Read Replica erstellen, wenn die Quelldatenbank die Anmeldeinformationen des Hauptbenutzers mit Secrets Manager verwaltet.

Vorbereiten der Erstellung eines Oracle-Replikats

Führen Sie die folgenden Schritte aus, bevor Sie mit der Verwendung des Replikats beginnen können.

Themen

- Aktivieren automatischer Backups
- Aktivieren des erzwungenen Protokollierungsmodus
- Ändern der Protokollierungskonfiguration
- Festlegen des Parameters MAX_STRING_SIZE
- Planen von Datenverarbeitungs- und Speicherressourcen

Aktivieren automatischer Backups

Bevor eine DB-Instance als eine Quell-DB-Instance eingesetzt werden kann, müssen Sie automatische Backups auf der Quell-DB-Instance aktivieren. Weitere Informationen zum Ausführen dieses Verfahrens finden Sie unter Aktivieren von automatisierten Backups.

Aktivieren des erzwungenen Protokollierungsmodus

Es wird empfohlen, den erzwungenen Protokollierungsmodus zu aktivieren. Im erzwungenen Protokollierungsmodus schreibt die Oracle-Datenbank Redo-Datensätze, auch wenn `NOLOGGING` mit DDL-Anweisungen (Data Definition Language) verwendet wird.

Aktivieren Sie den erzwungenen Protokollierungsmodus wie folgt:

1. Melden Sie sich mit einem Client-Tool wie SQL Developer bei Ihrer Oracle-Datenbank an.
2. Aktivieren Sie den erzwungenen Protokollmodus, indem Sie das folgende Verfahren ausführen.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Weitere Informationen zu diesem Verfahren finden Sie unter Einstellen der erzwungenen Protokollierung.

Ändern der Protokollierungskonfiguration

Für n Online-Redo-Protokolle der Größe m erstellt RDS automatisch n +1 Standby-Protokolle der Größe m auf der primären DB-Instance und allen Replikaten. Wenn Sie die Protokollierungskonfiguration auf der primären Instance ändern, werden die Änderungen automatisch an die Replikate weitergegeben.

Wenn Sie Ihre Protokollierungskonfiguration ändern, beachten Sie die folgenden Richtlinien:

- Wir empfehlen Ihnen, die Änderungen vorzunehmen, bevor Sie eine DB-Instance zur Quelle für Replikate machen. RDS für Oracle unterstützt auch das Aktualisieren der Instance, nachdem sie zu einer Quelle geworden ist.
- Bevor Sie die Protokollierungskonfiguration auf der primären DB-Instance ändern, überprüfen Sie, ob jedes Replikat über genügend Speicher verfügt, um die neue Konfiguration zu berücksichtigen.

Sie können die Protokollierungskonfiguration für eine DB-Instance mithilfe der Amazon RDS-Verfahren `rdsadmin.rdsadmin_util.add_logfile` und ändern`rdsadmin.rdsadmin_util.drop_logfile`. Weitere Informationen erhalten Sie unter Hinzufügen von Online-Redo-Log-Dateien und Löschen von Online-Redo-Log-Dateien.

Festlegen des Parameters MAX_STRING_SIZE

Stellen Sie vor dem Erstellen eines Oracle-Replikats sicher, dass die Einstellung des Parameters `MAX_STRING_SIZE` auf der Quell-DB-Instance und auf dem Replikat identisch ist. Hierzu können Sie sie der gleichen Parametergruppe zuordnen. Wenn Sie unterschiedliche Parametergruppen für die Quelle und das Replikat haben, können Sie `MAX_STRING_SIZE` auf denselben Wert festlegen. Weitere Informationen zu diesem Parameter finden Sie unter Aktivieren erweiterter Datentypen für eine neue DB-Instance.

Planen von Datenverarbeitungs- und Speicherressourcen

Stellen Sie sicher, dass die Quell-DB-Instance und ihre Replikate in Bezug auf die Datenverarbeitungs- und Speicherkapazität die für ihre Betriebslast angemessene Größe aufweisen. Wenn ein Replikat die Kapazität von Rechen-, Netzwerk- und Speicherressourcen erreicht hat, stellt das Replikat den Empfang und die Anwendung von Änderungen aus seiner Quelle ein. Amazon RDS for Oracle greift nicht ein, um eine hohe Replikationsverzögerung zwischen einer Quell-DB-Instance und ihren Lesereplikaten zu minimieren. Sie können die Speicher- und CPU-Ressourcen eines Replikats unabhängig von seiner Quelle und anderen Replikaten ändern.

Erstellen eines Replikats von RDS für Oracle im aufgespielten Modus

Standardmäßig sind Oracle-Replikate schreibgeschützt. Um ein Replikat im aufgespielten Modus zu erstellen, verwenden Sie die Konsole, die AWS CLI oder die RDS-API.

Konsole

Erstellen Sie ein aufgespieltes Replikat aus einer Oracle DB-Quell-Instance wie folgt:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die Oracle DB-Instance, die Sie als Quelle für ein aufgespieltes Replikat verwenden möchten.
4. Wählen Sie unter Actions (Aktionen) Create replica (Replikat erstellen) aus.
5. Wählen Sie für Replica mode (Replikatmodus) die Option Mounted (Aufgespielt).
6. Wählen Sie die Einstellungen, die Sie verwenden möchten. Geben Sie unter DB instance identifier (DB-Instance-Kennung) einen Namen für das Lesereplikat ein. Passen Sie weitere Einstellungen nach Ihrem Bedarf an.

7. Wählen Sie unter Regions (Regionen) die Region aus, in der das aufgespielte Replikat gestartet wird.
8. Wählen Sie die Instance-Größe und den Speichertyp aus. Wir empfehlen Ihnen, dieselbe DB-Instance-Klasse und denselben Speichertyp wie bei der Quell-DB-Instance für das Lesereplikat zu verwenden.
9. Wählen Sie für Multi-AZ deployment (Multi-AZ-Bereitstellung) die Option Create a standby instance (Standby-Instance erstellen), um eine Standby-Version des Replikats in einer anderen Availability Zone zu erstellen, und um einen Failover-Support für das aufgespielte Replikat bereitzustellen. Das Erstellen Ihres aufgespielten Replikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist.
10. Wählen Sie weitere Einstellungen aus, die Sie verwenden möchten.
11. Wählen Sie Create replica (Replikat erstellen).

Auf der Seite Databases (Datenbanken) hat das aufgespielte Replikat die Rolle „Replikat“.

AWS CLI

Um ein Oracle-Replikat im aufgespielten Modus `--replica-mode` zu erstellen, setzen Sie `mounted` im AWS CLI Befehl create-db-instance-read-replica auf .

Example

Für Linux, macOSoder Unix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --replica-mode mounted
```

Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --replica-mode mounted
```

Um ein schreibgeschütztes Replikat `mounted` in einen aufgespielten Zustand zu ändern, setzen Sie im AWS CLI Befehl `--replica-mode` auf modify-db-instance. Um ein aufgespieltes Replikat in den schreibgeschützten Modus zu versetzen, setzen Sie `--replica-mode` auf `open-read-only`.

RDS-API

Um ein Oracle-Replikat im aufgespielten Modus zu erstellen, geben Sie `ReplicaMode=mounted` in der RDS-API-Operation CreateDBInstanceReadReplica an.

Ändern des Replikatmodus von RDS für Oracle

Um den Replikatmodus eines vorhandenen Replikats zu ändern, verwenden Sie entweder die Konsole, die AWS CLI oder die RDS-API. Wenn Sie in den aufgespielten Modus wechseln, trennt das Replikat alle aktiven Verbindungen. Wenn Sie in den schreibgeschützten Modus wechseln, initialisiert Amazon RDS Active Data Guard.

Die Wechseloperation kann einige Minuten dauern. Während der Operation ändert sich der Status der DB-Instance in Modifying (Wird geändert). Weitere Hinweise zu Statusänderungen finden Sie unter Anzeigen von Amazon RDS DB-Instance-Status.

Konsole

Ändern Sie den Replikatmodus eines Oracle-Replikats von aufgespielt zu schreibgeschützt wie folgt:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die aufgespielte Replikatdatenbank aus.
4. Wählen Sie Ändern aus.
5. Wählen Sie für Replica mode (Replikatmodus) die Option Read-only (Schreibgeschützt).
6. Wählen Sie die anderen Einstellungen aus, die Sie ändern möchten:
7. Klicken Sie auf Weiter.
8. Wählen Sie für Scheduling of modifications (Einplanung von Änderungen) die Option Apply immediately (Sofort anwenden) aus.
9. Wählen Sie Modify DB Instance (DB-Instance ändern) aus.

AWS CLI

Um ein Lesereplikat `mounted` in den aufgespielten Modus zu versetzen, setzen Sie im AWS CLI Befehl `--replica-mode` auf modify-db-instance. Um ein aufgespieltes Replikat in den schreibgeschützten Modus zu versetzen, setzen Sie `--replica-mode` auf `open-read-only`.

Example

Für Linux, macOSoder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier myreadreplica \
    --replica-mode mode
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myreadreplica ^
    --replica-mode mode
```

RDS-API

Um ein schreibgeschütztes Replikat in den aufgespielten Modus zu versetzen, stellen Sie `ReplicaMode=mounted` in ModifyDBInstance ein. Um ein aufgespieltes Replikat in den schreibgeschützten Modus zu versetzen, stellen Sie ei `ReplicaMode=read-only`.

Arbeiten mit RDS-für-Oracle-Replikat-Backups

Sie können Backups eines RDS-für-Oracle-Replikats erstellen und wiederherstellen. Sowohl automatische Backups als auch manuelle Snapshots werden unterstützt. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren. In den folgenden Abschnitten werden die wichtigsten Unterschiede zwischen der Verwaltung von Backups eines primären und eines RDS-für-Oracle-Replikats beschrieben.

Aktivieren von RDS-für-Oracle-Replikat-Backups

In einem Oracle-Replikat ist die Funktion für automatische Backups standardmäßig nicht aktiviert. Sie aktivieren automatisierte Backups, indem Sie den Aufbewahrungszeitraum für Backups auf einen Wert größer als null festlegen.

Konsole

So aktivieren Sie automatisierte Backups direkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Datenbanken und dann die DB-Instance oder den Multi-AZ-DB-Cluster, die/den Sie ändern möchten.
3. Wählen Sie Ändern aus.
4. Wählen Sie für die Aufbewahrungsdauer von Backup einen positiven Wert ungleich Null, z. B. drei Tage.
5. Klicken Sie auf Continue.
6. Wählen Sie Apply immediately (Sofort anwenden) aus.
7. Wählen Sie DB-Instance ändern oder Cluster ändern aus, um Ihre Änderungen zu speichern und automatisierte Backups zu aktivieren.

AWS CLI

Verwenden Sie den modify-db-clusterBefehl AWS CLI modify-db-instanceoder, um automatische Backups zu aktivieren.

Verwenden Sie die folgenden Parameter:

- `--db-instance-identifier` (oder `--db-cluster-identifier` für einen Multi-AZ-DB-Cluster)
- `--backup-retention-period`
- `--apply-immediately` oder `--no-apply-immediately`

In diesem Beispiel aktivieren wir automatische Backups, indem wir den Aufbewahrungszeitraum für Backups auf drei Tage festlegen. Die Änderungen werden sofort übernommen.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my_db_instance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my_db_instance  ^
```

```
    --backup-retention-period 3 ^
    --apply-immediately
```

RDS-API

Um automatische Backups zu aktivieren, verwenden Sie den DBCluster RDS-API-Vorgang Modify DBInstance oder Modify mit den folgenden erforderlichen Parametern:

- `DBInstanceIdentifier` oder `DBClusterIdentifier`
- `BackupRetentionPeriod`

Wiederherstellen eines Replikat-Backups von RDS für Oracle

Sie können ein Oracle-Replikat-Backup genauso wiederherstellen wie ein Backup der primären Instance. Weitere Informationen finden Sie hier:

- Wiederherstellung auf einer DB-Instance
- Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS

Die wichtigste Überlegung beim Wiederherstellen eines Replikat-Backups ist die Bestimmung des Zeitpunkts, zu dem Sie die Wiederherstellung durchführen. Die Datenbank-Zeit bezieht sich auf die letzte angewendete Transaktionszeit der Daten im Backup. Wenn Sie ein Replikat-Backup wiederherstellen, erfolgt die Wiederherstellung auf die Datenbankzeit, nicht auf den Zeitpunkt, zu dem das Backup abgeschlossen wurde. Der Unterschied ist erheblich, da ein RDS-für-Oracle-Replikat um Minuten oder Stunden hinter dem primären Replikat zurückbleiben kann. Daher kann die Datenbankzeit eines Replikat-Backups und damit der Zeitpunkt, zu dem Sie sie wiederherstellen, viel früher sein als die Backup-Erstellungszeit.

Um den Unterschied zwischen Datenbankzeit und Erstellungszeit zu ermitteln, verwenden Sie den `describe-db-snapshots`-Befehl. Vergleichen Sie `SnapshotDatabaseTime`, das ist die Datenbankzeit des Replikat-Backups, und `OriginalSnapshotCreateTime`-Feld, das die letzte angewendete Transaktion in der Primärdatenbank darstellt. Im folgenden Beispiel wird die Zeitspanne zwischen zwei Datumsangaben dargestellt.

```
aws rds describe-db-snapshots \
    --db-instance-identifier my-oracle-replica
    --db-snapshot-identifier my-replica-snapshot
```

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "my-replica-snapshot",
            "DBInstanceIdentifier": "my-oracle-replica",
            "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
            ...
            "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
        }
    ]
}
```

So führen Sie eine Oracle Data Guard-Umschaltung aus

Eine Umschaltung ist ein Rollentausch zwischen einer Primärdatenbank und einer Standby-Datenbank. Während einer Umschaltung wechselt die ursprüngliche Primärdatenbank in eine Standby-Rolle, während die ursprüngliche Standby-Datenbank in die primäre Rolle übergeht.

In einer Oracle Data Guard-Umgebung unterstützt eine Primärdatenbank eine oder mehrere Standby-Datenbanken. Sie können einen verwalteten, Umschaltungs-basierten Rollenübergang von einer Primärdatenbank zu einer Standby-Datenbank durchführen. Eine Umschaltung ist ein Rollentausch zwischen einer Primärdatenbank und einer Standby-Datenbank. Während einer Umschaltung wechselt die ursprüngliche Primärdatenbank in eine Standby-Rolle, während die ursprüngliche Standby-Datenbank in die primäre Rolle übergeht.

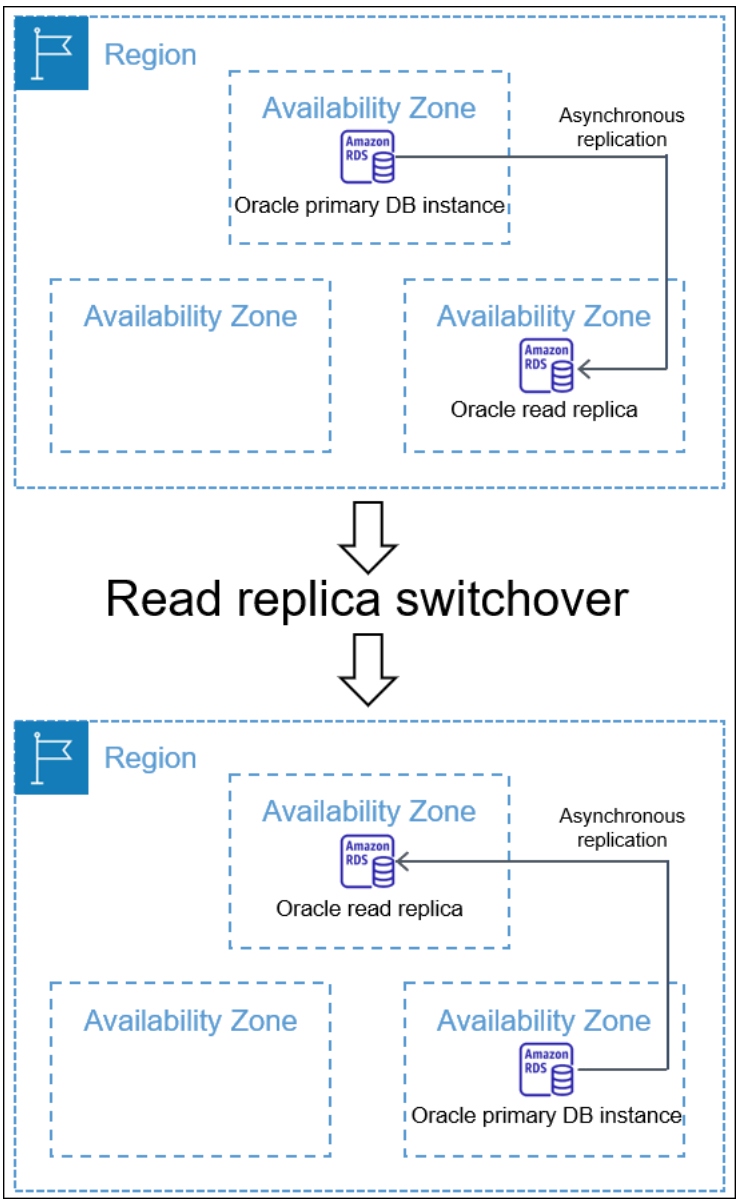
Themen

- Übersicht über Oracle Data Guard-Umschaltung
- Anforderungen für den Oracle Data Guard-Switchover
- Initiieren der Oracle Data Guard-Umschaltung
- So überwachen Sie Oracle Data Guard-Umschaltung

Übersicht über Oracle Data Guard-Umschaltung

Amazon RDS unterstützt einen vollständig verwalteten, Switchover-basierten Rollenübergang für Oracle-Datenbank-Replikate. Sie können nur eine Umschaltung zu einer Standby-Datenbank initiieren, die eingebunden oder schreibgeschützt geöffnet ist.

Die Replikate können sich in separaten AWS-Regionen oder in verschiedenen Availability Zones (AZs) einer einzigen Region befinden. Alle AWS-Regionen werden unterstützt.



Ein Switchover unterscheidet sich von einer Read Replica-Promotion. Bei einem Switchover wechseln die Rollen der Quell- und Replikat-DB-Instances. Bei einer Beförderung wird eine Read Replica zu einer Quell-DB-Instance, aber die Quell-DB-Instance wird nicht zu einer Replik. Weitere Informationen finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

Themen

- Vorteile von Oracle Data Guard-Umschaltung
- Unterstützte Oracle-Database-Versionen
- Kosten für Oracle Data Guard-Umschaltung
- So funktioniert die Oracle Data Guard-Umschaltung

Vorteile von Oracle Data Guard-Umschaltung

Genau wie RDS bei Oracle-Read Replicas stützt sich ein verwalteter Switchover auf Oracle Data Guard. Der Vorgang ist so ausgelegt, dass kein Datenverlust auftritt. Amazon RDS automatisiert die folgenden Aspekte der Umstellung:

- Kehrt die Rollen der Primärdatenbank und der angegebenen Standby-Datenbank um und versetzt die neue Standby-Datenbank in denselben Zustand (eingebunden oder schreibgeschützt) wie die ursprüngliche Standby-Datenbank
- Stellt Datenkonsistenz sicher
- Behält Ihre Replikationskonfiguration nach der Umstellung
- Unterstützt wiederholte Umkehrungen, sodass Ihre neue Standby-Datenbank zu ihrer ursprünglichen primären Rolle zurückkehren kann

Unterstützte Oracle-Database-Versionen

Der Oracle Data Guard-Switchover wird für Oracle Database 19c und höhere Versionen unterstützt.

Kosten für Oracle Data Guard-Umschaltung

Für die Oracle Data Guard-Umschaltungsfunktion fallen keine zusätzlichen Kosten an. Oracle Database Enterprise Edition unterstützt Standby-Datenbanken im aufgespielten Modus. Wenn Sie Standby-Datenbanken im schreibgeschützten Modus öffnen möchten, benötigen Sie die Option Oracle Active Data Guard.

So funktioniert die Oracle Data Guard-Umschaltung

Die Oracle Data Guard-Umschaltung ist ein vollständig verwalteter Vorgang. Sie initiieren den Switchover für eine Standby-Datenbank, indem Sie den Befehl ausführen. CLI `switchover-read-replica` Dann RDS ändert Amazon die Primär- und Standby-Rollen in Ihrer Replikationskonfiguration.

Ursprünglicher Standby und Ursprüngliche Primary sind die Rollen, die vor der Umschaltung existieren. Die neue Standby und neue Primary sind die Rollen, die nach der Umschaltung existieren. Ein Bystander-Replikat ist eine Replikatdatenbank, die als Standby-Datenbank in der Oracle Data Guard-Umgebung dient, aber nicht die Rollen wechselt.

Themen

- Phasen der Oracle Data Guard-Umschaltung
- Nach der Oracle Data Guard-Umschaltung

Phasen der Oracle Data Guard-Umschaltung

Um den Switchover durchzuführen, RDS muss Amazon die folgenden Schritte ausführen:

1. Blockieren Sie neue Transaktionen in der ursprünglichen Primärdatenbank. Während des Switchovers RDS unterbricht Amazon die Replikation für alle Datenbanken in Ihrer Oracle Data Guard-Konfiguration. Während der Umschaltung kann die ursprüngliche Primärdatenbank keine Schreibanforderungen verarbeiten.
2. Versenden Sie nicht angewendete Transaktionen an die ursprüngliche Standby-Datenbank und wenden Sie sie an.
3. Starten Sie die neue Standby-Datenbank im schreibgeschützten oder gemounteten Modus neu. Der Modus hängt vom offenen Zustand der ursprünglichen Standby-Datenbank vor der Umschaltung ab.
4. Öffnen Sie die neue Primärdatenbank im Lese-Schreibmodus.

Nach der Oracle Data Guard-Umschaltung

Amazon RDS wechselt die Rollen der Primär- und der Standby-Datenbank. Sie sind dafür verantwortlich, Ihre Anwendung erneut zu verbinden und jede andere gewünschte Konfiguration durchzuführen.

Themen

- Erfolgskriterien
- Verbindung zur neuen Primärdatenbank
- Konfiguration der neuen Primärdatenbank

Erfolgskriterien

Der Oracle Data Guard-Umschaltung ist erfolgreich, wenn die ursprüngliche Standby-Datenbank Folgendes ausführt:

- Übergang zu seiner Rolle als neue Primärdatenbank
- Schließt die Neukonfiguration ab

Um Ausfallzeiten zu begrenzen, wird Ihre neue Primärdatenbank so schnell wie möglich aktiv. Da Amazon Bystander-Replikate asynchron RDS konfiguriert, werden diese Replikate möglicherweise erst nach der ursprünglichen Primärdatenbank aktiv.

Verbindung zur neuen Primärdatenbank

Amazon RDS wird Ihre aktuellen Datenbankverbindungen nach dem Switchover nicht an die neue Primärdatenbank weitergeben. Nachdem der Oracle Data Guard-Umschaltung abgeschlossen ist, verbinden Sie Ihre Anwendung erneut mit der neuen Primärdatenbank.

Konfiguration der neuen Primärdatenbank

Um einen Wechsel zur neuen Primärdatenbank durchzuführen, RDS ändert Amazon den Modus der ursprünglichen Standby-Datenbank auf Öffnen. Die Änderung der Rolle ist die einzige Änderung an der Datenbank. Amazon RDS richtet keine Funktionen wie Multi-AZ-Replikation ein.

Wenn Sie eine Umschaltung auf ein regionsübergreifendes Replikat mit unterschiedlichen Optionen durchführen, behält die neue Primärdatenbank ihre eigenen Optionen bei. Amazon migriert die Optionen in der ursprünglichen Primärdatenbank RDS nicht. Wenn die ursprüngliche Primärdatenbank Optionen wieSSL,NNE, OEM und OEM _ hatteAGENT, propagiert Amazon sie RDS nicht an die neue Primärdatenbank.

Anforderungen für den Oracle Data Guard-Switchover

Stellen Sie vor Beginn der Oracle Data Guard-Umschaltung sicher, dass Ihre Replikationsumgebung die folgenden Anforderungen erfüllt:

- Die ursprüngliche Standby-Datenbank ist eingehängt oder schreibgeschützt geöffnet.
- Automatische Backups sind in der ursprünglichen Standby-Datenbank aktiviert.
- Die ursprüngliche Primärdatenbank und die ursprüngliche Standby-Datenbank befinden sich im `available` Status.
- Für die ursprüngliche Primärdatenbank und die ursprüngliche Standby-Datenbank stehen in keinem der folgenden Zustände Wartungsaktionen aus: `requirednext window`, oder`in progress`. Aktionen in diesen Zuständen blockieren den Switchover. Informationen zum Überprüfen des Status ausstehender Wartungsupdates finden Sie unterAusstehende Wartungsupdates anzeigen.

  Ausstehende Wartungsaktionen im `available` Status blockieren den Switchover nicht. RDS for Oracle veröffentlicht häufig Betriebssystemupdates (OS) im jeweiligen `available` Bundesstaat. Diese ausstehenden Betriebssystemupdates blockieren einen Switchover nicht, es sei denn, Sie

planen sie für das nächste Wartungsfenster, wodurch sie in den entsprechenden `next window` Status versetzt werden.

> **Note**
>
> Wenn Sie eine geplante Wartungsaktion verschieben möchten, sodass Sie einen Switchover ausführen können, wählen Sie in der RDS-Konsole Aktionen und dann Upgrade aufschieben aus. Sie können auch verhindern, dass ein Switchover blockiert wird, indem Sie eine ausstehende Wartungsaktion anwenden oder das Wartungsfenster auf ein Intervall vor Ihrem Switchover verschieben. Weitere Informationen finden Sie im re:POST-Artikel How to remove RDS Pending Maintenance Items.

- Die ursprüngliche Standby-Datenbank befindet sich im replizierenden Zustand.
- Sie versuchen nicht, eine Umschaltung zu initiieren, wenn sich entweder die Primärdatenbank oder die Standby-Datenbank derzeit in einem Umschaltungs-Lebenszyklus befindet. Wenn eine Replikatdatenbank nach einer Umschaltung neu konfiguriert wird, verhindert Amazon RDS, dass Sie eine weitere Umschaltung initiieren.

> **Note**
>
> Ein Bystander-Replikat ist ein Replikat in der Oracle Data Guard-Konfiguration, das nicht das Ziel der Umschaltung ist. Nachstehende Replikate können sich während der Umschaltung in einem beliebigen Status befinden.

- Die ursprüngliche Standby-Datenbank hat eine Konfiguration, die so nah wie gewünscht an der ursprünglichen Primärdatenbank liegt. Nehmen wir ein Szenario an, in dem die ursprüngliche primäre und die ursprüngliche Standby-Datenbank unterschiedliche Optionen haben. Nach Abschluss der Umschaltung konfiguriert Amazon RDS die neue Primärdatenbank nicht automatisch neu, sodass sie dieselben Optionen wie die ursprüngliche Primärdatenbank hat.
- Sie konfigurieren die gewünschte Multi-AZ-Bereitstellung, bevor Sie eine Umstellung einleiten. Amazon RDS verwaltet Multi-AZ im Rahmen der Umstellung nicht. Die Multi-AZ-Bereitstellung bleibt unverändert.

  Nehmen wir an, dass db_maz die primäre Datenbank in einer Multi-AZ-Bereitstellung ist und db_saz ein Single-AZ-Replikat ist. Sie initiieren eine Umstellung von db_maz auf db_saz. Danach ist db_maz eine Multi-AZ-Replikatdatenbank und db_saz ist eine Single-AZ-Primärdatenbank. Die neue Primärdatenbank ist jetzt nicht mehr durch eine Multi-AZ-Bereitstellung geschützt.

- In Vorbereitung auf eine regionsübergreifende Umstellung verwendet die Primärdatenbank außerhalb der Replikationskonfiguration nicht dieselbe Optionsgruppe wie eine DB-Instance. Damit eine regionsübergreifende Umstellung erfolgreich ist, müssen die aktuelle Primärdatenbank und ihre Lesereplikate die einzigen DB-Instances sein, die die Optionsgruppe der aktuellen Primärdatenbank verwenden. Andernfalls verhindert Amazon RDS die Umstellung.

Initiieren der Oracle Data Guard-Umschaltung

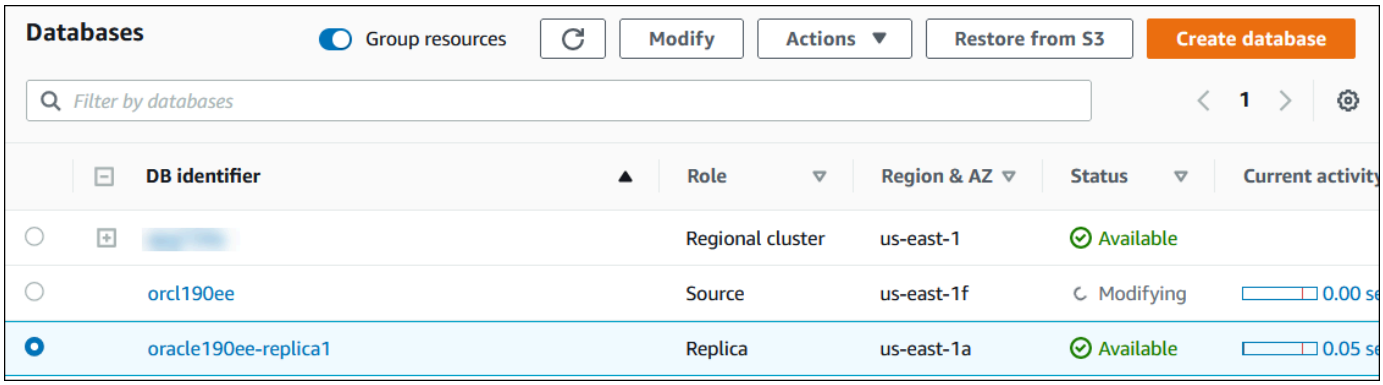
Sie können eine Read Replica RDS für Oracle auf die primäre Rolle und die ehemalige primäre DB-Instance auf eine Replikatrolle umstellen.

Konsole

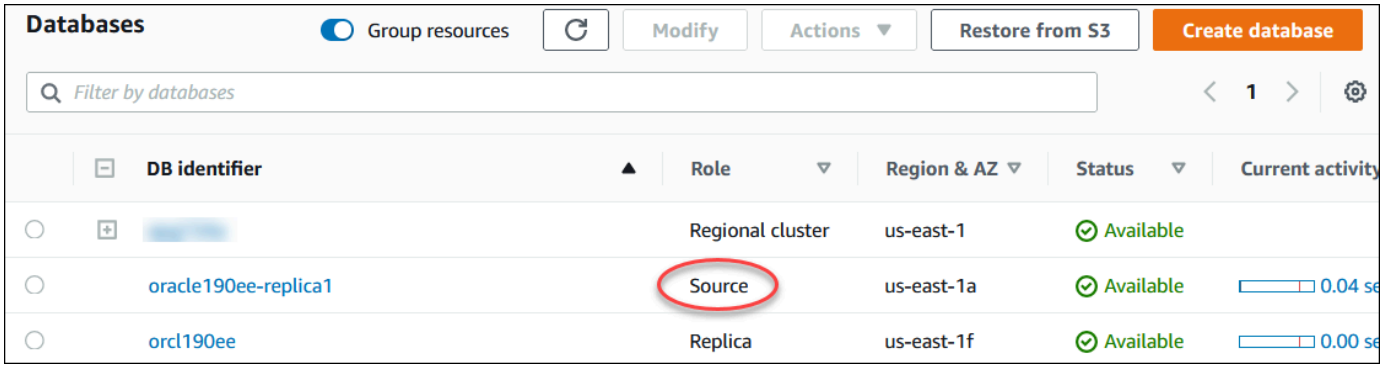
So stellen Sie ein Oracle-Lesereplikat auf die primäre DB-Rolle um

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie in der RDS Amazon-Konsole Datenbanken aus.

   Der Bereich Databases (Datenbanken) wird angezeigt. Jedes Lesereplikat zeigt Replica (Replikat) in der Spalte Role (Rolle) an.
3. Wählen Sie das Lesereplikat aus, das Sie zur primären Rolle wechseln möchten.
4. Für Aktionen wählen Sie So wechseln Sie Replikate.
5. Wählen Sie Ich bestätige. Dann wählen Sie Replikat umschalten.
6. Überwachen Sie auf derDatenbanken-Seite den Fortschritt der Umschaltung.



   Wenn die Umstellung abgeschlossen ist, ändert sich die Rolle des Umstellungsziels von Replica (Replikat) in Source (Quelle).



AWS CLI

Verwenden Sie den AWS CLI [switchover-read-replica](#)Befehl, um ein Oracle-Replikat auf die primäre DB-Rolle umzuschalten. In den folgenden Beispielen wird das Oracle-Replikat benannt *replica-to-be-made-primary* in die neue Primärdatenbank.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds switchover-read-replica \
    --db-instance-identifier replica-to-be-made-primary
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds switchover-read-replica ^
    --db-instance-identifier replica-to-be-made-primary
```

RDS API

Um ein Oracle-Replikat auf die primäre DB-Rolle umzuschalten, rufen Sie den RDS API [SwitchoverReadReplica](#)Amazon-Vorgang mit dem erforderlichen Parameter `DBInstanceIdentifier` auf. Dieser Parameter gibt den Namen des Oracle-Replikats an, das Sie als primäre DB-Rolle übernehmen möchten.

So überwachen Sie Oracle Data Guard-Umschaltung

Verwenden Sie den AWS CLI Befehl, um den Status Ihrer Instances zu überprüfen`describe-db-instances`. Der folgende Befehl überprüft den Status der DB-Instance *orcl2*. Diese

Datenbank war vor dem Switchover eine Standby-Datenbank, ist aber nach dem Switchover die neue Primärdatenbank.

```
aws rds describe-db-instances \
    --db-instance-identifier orcl2
```

Um zu bestätigen, dass die Umschaltung erfolgreich abgeschlossen wurde, fragen Sie `V$DATABASE.OPEN_MODE` ab. Stellen Sie sicher, dass der Wert für die neue Primärdatenbank `READ WRITE` ist.

```
SELECT OPEN_MODE FROM V$DATABASE;
```

Verwenden Sie den Befehl, um nach Ereignissen im Zusammenhang mit Switchover zu suchen. AWS CLI `describe-events` Das folgende Beispiel sucht nach Ereignissen auf *orcl2* sein.

```
aws rds describe-events \
    --source-identifier orcl2 \
    --source-type db-instance
```

Fehlerbehebung bei Replikaten von RDS für Oracle

In diesem Abschnitt werden mögliche Replikationsprobleme und -lösungen beschrieben.

Themen

- Überwachen einer Oracle-Replikationsverzögerung
- Fehlerbehebung bei der Oracle-Replikation nach dem Hinzufügen oder Ändern von Auslösern

Überwachen einer Oracle-Replikationsverzögerung

Wenn Sie die Replikationsverzögerung in Amazon CloudWatch überwachen möchten, zeigen Sie die `ReplicaLag`-Metrik von Amazon RDS an. Weitere Informationen zur zeitlichen Verzögerung bei der Replikation finden Sie unter Überwachen der Lesereplikation und CloudWatch Amazon-Metriken für Amazon RDS.

Wenn die Replikationsverzögerung für ein Lesereplikat zu lang ist, fragen Sie die folgenden Ansichten ab:

- `V$ARCHIVED_LOG` – Zeigt, welche Commits auf das Lesereplikat angewendet wurden.

- `V$DATAGUARD_STATS` – Zeigt eine detaillierte Aufschlüsselung der Komponenten, aus denen die Metrik `ReplicaLag` besteht.
- `V$DATAGUARD_STATUS` – Zeigt die Protokollausgabe der internen Replikationsvorgänge von Oracle.

Wenn die Verzögerungszeit für ein gemountetes Replikat zu lang ist, können Sie die `V$`-Ansichten nicht abfragen. Führen Sie stattdessen die folgenden Schritte aus:

- Überprüfen Sie die `ReplicaLag`-Metrik in CloudWatch.
- Überprüfen Sie die Warnungsprotokolldatei für das Replikat in der Konsole. Suchen Sie in den Wiederherstellungsmeldungen nach Fehlern. Die Meldungen enthalten die Log-Sequenznummer, die Sie mit der primären Sequenznummer vergleichen können. Weitere Informationen finden Sie unter Datenbankprotokolldateien von Amazon RDS für Oracle.

Fehlerbehebung bei der Oracle-Replikation nach dem Hinzufügen oder Ändern von Auslösern

Wenn Sie Auslöser hinzufügen oder ändern und die Replikation danach fehlschlägt, liegt das Problem möglicherweise bei den Auslösern. Stellen Sie sicher, dass der Auslöser die folgenden Benutzerkonten ausschließt, die von RDS für die Replikation benötigt werden:

- Benutzerkonten mit Administratorrechten
- `SYS`
- `SYSTEM`
- `RDS_DATAGUARD`
- `rdsdb`

Weitere Informationen finden Sie unter Sonstige Überlegungen zu RDS-für-Oracle-Replikaten.

Hinzufügen von Optionen zu Oracle DB-Instances

In Amazon RDS ist eine Option ein zusätzliches Feature. Nachfolgend finden Sie eine Beschreibung der Optionen, die Sie Amazon RDS-Instances hinzufügen können, auf denen die Oracle DB-Engine läuft

Themen

- Übersicht über Oracle-DB-Optionen
- Amazon S3-Integration
- Oracle Application Express (APEX)
- EFSAmazon-Integration
- Oracle Java Virtual Machine
- Oracle Enterprise Manager
- Oracle Label Security
- Oracle Locator
- Oracle Native Network Encryption
- Oracle OLAP
- Oracle Secure Sockets Layer
- Oracle Spatial
- Orakel SQLT
- Oracle Statspack
- Oracle-Zeitzone
- Automatische Aktualisierung der Oracle-Zeitzonendatei
- Oracle Transparent Data Encryption
- Orakel UTL _ MAIL
- Oracle XML DB

Übersicht über Oracle-DB-Optionen

Damit diese Optionen für Ihre Oracle-Datenbank aktiviert werden, fügen Sie diese einer Optionsgruppe hinzu und ordnen anschließend die Optionsgruppe Ihrer DB-Instance zu. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.

Themen

- Zusammenfassung der Optionen für Oracle Database
- Für die verschiedenen Editionen unterstützte Optionen
- Speicheranforderungen für spezifische Optionen

Zusammenfassung der Optionen für Oracle Database

Sie können die folgenden Optionen für Oracle-DB-Instances hinzufügen.

Option	Options-ID
Amazon S3-Integration	`S3_INTEGRATION`
Oracle Application Express (APEX)	`APEX` `APEX-DEV`
Oracle Enterprise Manager	`OEM` `OEM_AGENT`
Oracle Java Virtual Machine	`JVM`
Oracle Label Security	`OLS`
Oracle Locator	`LOCATOR`
Oracle Native Network Encryption	`NATIVE_NETWORK_ENCRYPTION`
Oracle OLAP	`OLAP`
Oracle Secure Sockets Layer	`SSL`
Oracle Spatial	`SPATIAL`
Orakel SQLT	`SQLT`
Oracle Statspack	`STATSPACK`

Option	Options-ID
Oracle-Zeitzone	`Timezone`
Automatische Aktualisierung der Oracle-Zeitzonendatei	`TIMEZONE_FILE_AUTOUPGRADE`
Oracle Transparent Data Encryption	`TDE`
Orakel UTL _ MAIL	`UTL_MAIL`
Oracle XML DB	`XMLDB`

Für die verschiedenen Editionen unterstützte Optionen

RDS for Oracle verhindert, dass Sie Optionen zu einer Edition hinzufügen, die nicht unterstützt werden. Mit dem Befehl `aws rds describe-option-group-options` können Sie herausfinden, welche RDS-Optionen in verschiedenen Oracle Database Editionen unterstützt werden. Im folgenden Beispiel werden die unterstützten Optionen für Oracle Database 19c Enterprise Edition aufgeführt.

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19
```

Weitere Informationen finden Sie unter describe-option-group-options in der AWS CLI-Befehlsreferenz.

Speicheranforderungen für spezifische Optionen

Einige Optionen erfordern zusätzlichen Arbeitsspeicher, um auf Ihrer DB-Instance ausgeführt zu werden. Oracle Enterprise Manager Database Control belegt beispielsweise etwa 300 MB RAM. Sollten Sie diese Option für eine kleine DB-Instance aktivieren, könnte es aufgrund von zu wenig Speicher zu Leistungsproblemen kommen. Sie können die Oracle-Parameter anpassen, sodass die Datenbank weniger RAM benötigt. Alternativ dazu können Sie auch auf eine größere DB-Instance skalieren.

Amazon S3-Integration

Sie können Dateien zwischen Ihrer RDS Oracle-DB-Instance und einem Amazon S3 S3-Bucket übertragen. Sie können die Amazon-S3-Integration mit Oracle-Database-Funktionen wie Oracle Data Pump nutzen. Sie können beispielsweise Data Pump-Dateien von Amazon S3 auf Ihre RDS Oracle-DB-Instance herunterladen. Weitere Informationen finden Sie unter Daten in Oracle auf Amazon importieren RDS.

> **Note**
>
> Die DB-Instance und der Amazon-S3-Bucket müssen sich in der gleichen AWS-Region befinden.

Themen

- Konfigurieren von IAM-Berechtigungen für die Integration von RDS for Oracle in Amazon S3
- Hinzufügen der Amazon S3-Integrationsoption
- Übertragung von Dateien zwischen Amazon RDS for Oracle und einem Amazon S3 S3-Bucket
- Fehlerbehebung für die Amazon-S3-Integration
- Entfernen der Amazon S3-Integrationsoption

Konfigurieren von IAM-Berechtigungen für die Integration von RDS for Oracle in Amazon S3

Damit RDS for Oracle in Amazon S3 integriert werden kann, benötigt Ihre DB-Instance Zugriff auf einen Amazon-S3-Bucket. Die von Ihrer DB-Instance verwendete Amazon VPC muss keinen Zugriff auf die Amazon S3-Endpunkte ermöglichen.

RDS for Oracle unterstützt die Übertragung von Dateien zwischen einer DB-Instance in einem Konto und einem Amazon S3 S3-Bucket in einem anderen Konto. Wenn zusätzliche Schritte erforderlich sind, werden diese in den folgenden Abschnitten vermerkt.

Themen

- Schritt 1: Erstellen einer IAM-Richtlinie für Ihre Amazon-RDS-Rolle
- Schritt 2: (Optional) Erstellen einer IAM-Richtlinie für Ihren Amazon-S3-Bucket

- Schritt 3: Erstellen einer IAM-Rolle für Ihre DB-Instance und Anfügen Ihrer Richtlinie
- Schritt 4: So ordnen Sie Ihre IAM-Rolle Ihrer DB-Instance von RDS für Oracle zu

Schritt 1: Erstellen einer IAM-Richtlinie für Ihre Amazon-RDS-Rolle

In diesem Schritt erstellen Sie eine AWS Identity and Access Management (IAM-) Richtlinie mit den erforderlichen Berechtigungen, um Dateien zwischen Ihrem Amazon S3 S3-Bucket und Ihrer RDS-DB-Instance zu übertragen. In diesem Schritt wird davon ausgegangen, dass Sie bereits einen S3-Bucket erstellt haben.

Notieren Sie sich vor dem Erstellen der Richtlinie die folgenden Informationen:

- Amazon-Ressourcenname (ARN) Ihres Buckets
- Der ARN für Ihren AWS KMS Schlüssel, wenn Ihr Bucket SSE-KMS- oder SSE-S3-Verschlüsselung verwendet

> **Note**
>
> Eine DB-Instance von RDS für Oracle kann nicht auf mit SSE-C verschlüsselte Amazon-S3-Buckets zugreifen.

Weitere Informationen finden Sie unter Schutz von Daten durch serverseitige Verschlüsselung im Amazon Simple Storage Service User Guide.

Konsole

So erstellen Sie eine IAM-Richtlinie, die Amazon RDS Zugriff auf Ihren Amazon-S3-Bucket gewährt

1. Öffnen Sie die IAM-Managementkonsole.
2. Wählen Sie unter Zugriffsverwaltung Richtlinien aus.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie auf der Registerkarte Visueller Editor die Option Service auswählen und dann S3 aus.
5. Wählen Sie unter Actions (Aktionen) die Option Expand all (Alle expandieren). Wählen Sie dann die Bucket-Berechtigungen und Objektberechtigungen aus, die benötigt werden, um Dateien aus einem Amazon S3-Bucket Amazon RDS zu übertragen. Führen Sie beispielsweise folgende Schritte aus:

- Erweitern Sie Liste und wählen Sie dann aus. ListBucket
- Erweitern Sie Lesen und wählen Sie dann aus GetObject.
- Erweitern Sie Schreiben und wählen Sie dann PutObject, DeleteObjectAbortMultipartUpload, und aus ListMultipartUploadParts. Die mehrteiligen Upload-Berechtigungen sind erforderlich, wenn große Dateien (100 MB oder größer) auf Amazon S3 hochgeladen werden.
- Erweitern Sie Berechtigungsverwaltung und wählen Sie dann. PutObjectAcl Diese Berechtigung ist erforderlich, wenn Sie planen, Dateien in einen Bucket hochzuladen, der einem anderen Konto gehört, und dieses Konto die volle Kontrolle über den Bucket-Inhalt benötigt.

Objektberechtigungen sind Berechtigungen für Objektoperationen in Amazon S3. Sie müssen sie für Objekte in einem Bucket und nicht für den Bucket selbst erteilen. Weitere Informationen finden Sie unter Berechtigungen für Objektoperationen.

6. Wählen Sie Ressourcen aus und gehen Sie wie folgt vor:

   a. Wählen Sie Spezifisch aus.

   b. Wählen Sie für Bucket die Option ARN hinzufügen aus. Geben Sie Ihren Bucket-ARN ein. Der Bucket-Name wird automatisch ausgefüllt. Wählen Sie dann Add (Hinzufügen).

   c. Wenn die Ressource Objekt angezeigt wird, wählen Sie entweder ARN hinzufügen aus, um Ressourcen manuell hinzuzufügen, oder klicken Sie auf Beliebige.

   > **Note**
   >
   > Sie können für Amazon-Ressourcenname (ARN) einen spezifischen ARN-Wert einstellen, um Amazon RDS nur den Zugriff auf spezifische Dateien oder Order in einem Amazon S3-Bucket zu gewähren. Weitere Informationen über das Definieren von Zugriffsrichtlinien für Amazon S3 finden Sie unter Verwaltung der Zugriffsberechtigungen zu Ihren Amazon S3-Ressourcen.

7. (Optional) Wählen Sie Zusätzliche Berechtigungen hinzufügen, um Ressourcen zur Richtlinie hinzuzufügen. Führen Sie beispielsweise folgende Schritte aus:

   a. Wenn Ihr Bucket mit einem benutzerdefinierten KMS-Schlüssel verschlüsselt ist, wählen Sie für den Service KMS aus.

   b. Wählen Sie für Manuelle Aktionen Folgendes aus:

- Encrypt
- ReEncrypt von und ReEncrypt nach
- Decrypt
- DescribeKey
- GenerateDataKey

c. Wählen Sie für Ressourcen die Option Spezifisch aus.

d. Wählen Sie unter Schlüssel die Option ARN hinzufügen aus. Geben Sie den ARN Ihres benutzerdefinierten Schlüssels als Ressource ein und wählen Sie Hinzufügen aus.

   Weitere Informationen finden Sie unter Schützen von Daten mithilfe serverseitiger Verschlüsselung mit in gespeicherten KMS-Schlüsseln AWS Key Management Service (SSE-KMS) im Amazon Simple Storage Service-Benutzerhandbuch.

e. Wenn Sie möchten, dass Amazon RDS auf andere Buckets zugreift, fügen Sie den ARNs für diese Buckets hinzu. Alternativ können Sie auch Zugriff auf alle Buckets und Objekte in Amazon S3 erlauben.

8. Wählen Sie Next: Tags (Weiter: Tags) und danach Next: Review (Weiter: Prüfen) aus.
9. Geben Sie unter Name einen Namen für Ihre IAM-Richtlinie ein, z. B. `rds-s3-integration-policy`. Sie verwenden diesen Namen, wenn Sie eine IAM-Rolle erstellen, um sie Ihrer DB-Instance zuzuweisen. Sie können auch einen optionalen Wert für Description (Beschreibung) hinzufügen.
10. Wählen Sie Richtlinie erstellen aus.

AWS CLI

Erstellen Sie eine AWS Identity and Access Management (IAM-) Richtlinie, die Amazon RDS Zugriff auf einen Amazon S3 S3-Bucket gewährt. Nachdem Sie die Richtlinie erstellt haben, notieren Sie den ARN der Richtlinie. Sie benötigen den ARN bei einem nachfolgenden Schritt.

Schließen Sie basierend auf dem erforderlichen Zugriffstyp die entsprechenden Aktionen ein:

- `GetObject` – Erforderlich für die Übertragung von Dateien aus einem Amazon S3-Bucket zu Amazon RDS.
- `ListBucket` – Erforderlich für die Übertragung von Dateien aus einem Amazon S3-Bucket zu Amazon RDS.

- `PutObject` – Erforderlich für die Übertragung von Dateien von Amazon RDS in einen Amazon S3-Bucket.
- `AbortMultipartUpload`— Erforderlich für mehrteilige Uploads bei der Übertragung großer Dateien (100 MB oder größer) von Amazon RDS in einen Amazon S3 S3-Bucket.
- `ListMultipartUploadParts`— Erforderlich für mehrteilige Uploads bei der Übertragung großer Dateien (100 MB oder größer) von Amazon RDS in einen Amazon S3 S3-Bucket.

Mit dem folgenden AWS CLI Befehl wird eine IAM-Richtlinie *`rds-s3-integration-policy`* mit diesen Optionen erstellt. Sie gewährt Zugriff auf einen Bucket namens *`amzn-s3-demo-bucket`*.

Example

Für LinuxmacOS, oderUnix:

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "s3:AbortMultipartUpload",
           "s3:ListMultipartUploadParts"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

Das folgende Beispiel enthält Berechtigungen für benutzerdefinierte KMS-Schlüssel.

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
```

```
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "kms:Decrypt",
           "kms:Encrypt",
           "kms:ReEncrypt*",
           "kms:GenerateDataKey",
           "kms:DescribeKey",
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*",
           "arn:aws:kms:::your-kms-arn"
         ]
       }
     ]
   }'
```

Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "s3:AbortMultipartUpload",
           "s3:ListMultipartUploadParts"
         ],
         "Effect": "Allow",
         "Resource": [
```

```
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
          ]
        }
      ]
    }'
```

Das folgende Beispiel enthält Berechtigungen für benutzerdefinierte KMS-Schlüssel.

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "kms:Decrypt",
           "kms:Encrypt",
           "kms:ReEncrypt",
           "kms:GenerateDataKey",
           "kms:DescribeKey",
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*",
           "arn:aws:kms:::your-kms-arn"
         ]
       }
     ]
   }'
```

Schritt 2: (Optional) Erstellen einer IAM-Richtlinie für Ihren Amazon-S3-Bucket

Dieser Schritt ist nur unter den folgenden Bedingungen erforderlich:

- Sie planen, Dateien von einem Konto (Konto A) in einen Amazon-S3-Bucket hochzuladen und von einem anderen Konto (Konto B) auf sie zuzugreifen.
- Konto B ist Eigentümer des Buckets.

- Konto B benötigt die volle Kontrolle über Objekte, die in den Bucket geladen wurden.

Wenn die vorhergehenden Bedingungen nicht auf Sie zutreffen, fahren Sie mit Schritt 3: Erstellen einer IAM-Rolle für Ihre DB-Instance und Anfügen Ihrer Richtlinie fort.

Zum Erstellen Ihrer Bucket-Richtlinie müssen Sie über Folgendes verfügen:

- Konto-ID für Konto A
- Benutzername für Konto A
- ARN-Wert für den Amazon-S3-Bucket in Konto B

Konsole

Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie erstellen wollen oder dessen Bucket-Richtlinie Sie bearbeiten wollen.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Dies öffnet die Seite Bucket-Richtlinie bearbeiten.
5. Untersuchen Sie auf der Seite Edit bucket policy (Bucket-Richtlinie bearbeiten) Policy examples (Richtlinienbeispiele) im Amazon-S3-Benutzerhandbuch, wählen Sie Policy generator (Richtliniengenerator) aus, um automatisch eine Richtlinie zu generieren, oder bearbeiten Sie die JSON im Abschnitt Policy (Richtlinie).

   Wenn Sie den Policy-Generator wählen, wird der AWS Policy-Generator in einem neuen Fenster geöffnet:

   a. Wählen Sie auf der Seite AWS -Richtliniengenerator unter Richtlinientyp auswählen die Option S3-Bucket-Richtlinie aus.
   b. Fügen Sie eine Anweisung hinzu, indem Sie die Informationen in die bereitgestellten Felder eingeben, und wählen Sie dann Anweisung hinzufügen. Wiederholen Sie diesen Vorgang für so viele Anweisungen, wie Sie hinzufügen möchten. Weitere Informationen zu diesen Feldern finden Sie in der Referenz zu den IAM-JSON-Richtlinienelementen im IAM-Benutzerhandbuch.

> **Note**
>
> Der Einfachheit halber zeigt die Seite Bucket-Richtlinie bearbeiten den Bucket-ARN (Amazon-Ressourcenname) des aktuellen Buckets über dem Richtlinientextfeld an. Sie können diesen ARN zur Verwendung in den Anweisungen auf der Seite AWS - Richtliniengenerator kopieren.

c. Wenn Sie mit dem Hinzufügen von Anweisungen fertig sind, wählen Sie Generieren von Richtlinien.

d. Kopieren Sie den generierten Richtlinientext, wählen Sie Schließen und kehren Sie zur Seite Bucket-Richtlinie bearbeiten in der Amazon-S3-Konsole zurück.

6. Bearbeiten Sie im Feld Richtlinie die vorhandene Richtlinie oder fügen Sie die Bucket-Richtlinie aus dem Richtliniengenerator ein. Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge bevor Sie Ihre Richtlinie speichern.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExamplePermissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/account-A-user"
      },
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket",
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      ]
    }
  ]
}
```

7. Wählen Sie Speichern Sie die Änderungen, wodurch Sie zu der Seite Bucket-Berechtigungen zurückkehren.

Schritt 3: Erstellen einer IAM-Rolle für Ihre DB-Instance und Anfügen Ihrer Richtlinie

In diesem Schritt wird davon ausgegangen, dass Sie die IAM-Richtlinie in Schritt 1: Erstellen einer IAM-Richtlinie für Ihre Amazon-RDS-Rolle erstellt haben. In diesem Schritt erstellen Sie eine Rolle für Ihre DB-Instance von RDS for Oracle und fügen dann Ihre Richtlinie an die Rolle an.

Konsole

So erstellen Sie eine IAM-Rolle, um Amazon RDS Zugriff auf einen Amazon-S3-Bucket zu gewähren

1. Öffnen Sie die IAM-Managementkonsole.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie einen AWS -Service aus.
5. Für Anwendungsfälle für andere AWS Dienste: Wählen Sie RDS und dann RDS — Rolle zur Datenbank hinzufügen. Wählen Sie anschließend Weiter.
6. Geben Sie für Suchen unter Berechtigungsrichtlinien anfügen den Namen der von Ihnen in Schritt 1: Erstellen einer IAM-Richtlinie für Ihre Amazon-RDS-Rolle erstellten IAM-Richtlinie ein. Wählen Sie die Richtlinie aus, wenn sie in der Liste erscheint. Wählen Sie anschließend Weiter.
7. Geben Sie unter Rollenname einen Namen für Ihre IAM-Rolle ein, z. B. `rds-s3-integration-role`. Sie können auch einen optionalen Wert für Description (Beschreibung) hinzufügen.
8. Wählen Sie Rolle erstellen aus.

AWS CLI

So erstellen Sie eine Rolle und fügen ihr eine Richtlinie an

1. Erstellen Sie eine IAM-Rolle, die Amazon RDS in Ihrem Auftrag annehmen kann, um auf Ihre Amazon S3-Buckets zuzugreifen.

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Vertrauensbeziehungen, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Sie können beide globalen Bedingungskontextschlüssel verwenden und der Wert `aws:SourceArn` enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der Wert `aws:SourceAccount` und das Konto im Wert `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Anweisung verwendet werden.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Stellen Sie in der Vertrauensbeziehung sicher, dass Sie den globalen Bedingungskontextschlüssel `aws:SourceArn` mit dem vollständigen Amazon-Ressourcennamen (ARN) der Ressourcen verwenden, die auf die Rolle zugreifen.

Der folgende AWS CLI Befehl erstellt die *`rds-s3-integration-role`* für diesen Zweck benannte Rolle.

Example

Für LinuxmacOS, oderUnix:

```
aws iam create-role \
   --role-name rds-s3-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
```

```
                "aws:SourceAccount": "my_account_ID",
                "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
            }
        }
      }
    ]
  }'
```

Windows:

```
aws iam create-role ^
   --role-name rds-s3-integration-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": "my_account_ID",
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.

2. Notieren Sie nach dem Erstellen der Rolle den ARN der Rolle. Sie benötigen den ARN bei einem nachfolgenden Schritt.
3. Fügen Sie die erstellte Richtlinie an die erstellte Rolle an.

   Mit dem folgenden AWS CLI Befehl wird die Richtlinie an die angegebene *rds-s3-integration-role* Rolle angehängt.

Example

Für LinuxmacOS, oderUnix:

```
aws iam attach-role-policy \
   --policy-arn your-policy-arn \
   --role-name rds-s3-integration-role
```

Windows:

```
aws iam attach-role-policy ^
   --policy-arn your-policy-arn ^
   --role-name rds-s3-integration-role
```

Ersetzen Sie *your-policy-arn* durch den Richtlinien-ARN, den Sie im vorherigen Schritt notiert haben.

Schritt 4: So ordnen Sie Ihre IAM-Rolle Ihrer DB-Instance von RDS für Oracle zu

Der letzte Schritt bei der Konfiguration der Berechtigungen für die Amazon S3-Integration ist die Zuordnung Ihrer IAM-Rolle zu Ihrer DB-Instance. Beachten Sie die folgenden Voraussetzungen:

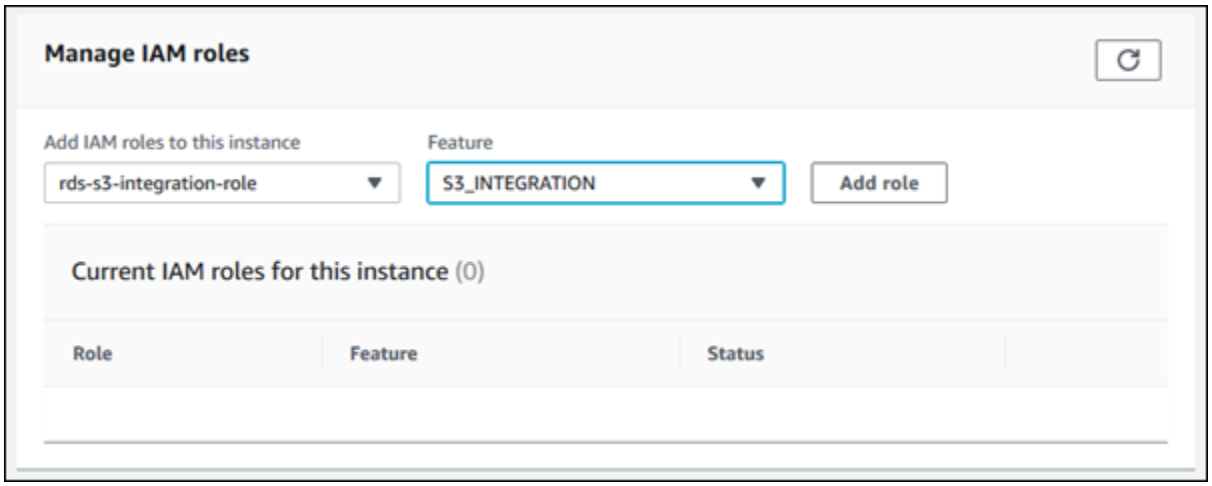
- Sie müssen Zugriff auf eine IAM-Rolle haben, der die Amazon-S3-Berechtigungsrichtlinie angefügt ist.
- Sie können jeweils nur eine IAM-Rolle Ihrer DB-Instance von RDS für Oracle hinzufügen.
- Ihre DB-Instance muss sich im Status Verfügbar befinden.

Konsole

So ordnen Sie Ihre IAM-Rolle Ihrer DB-Instance von RDS for Oracle zu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie aus dem Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance von RDS für Oracle aus, um deren Details anzuzeigen.
4. Auf derKonnektivität & SicherheitScrollen Sie nach unten zumIAM-Rollen verwaltenunten auf der Seite.

5. Wählen Sie unter IAM-Rollen zu dieser Instance hinzufügen die Rolle aus, die Sie in Schritt 3: Erstellen einer IAM-Rolle für Ihre DB-Instance und Anfügen Ihrer Richtlinie erstellt haben.
6. Wählen Sie unter Feature (Funktion) die Option S3_INTEGRATION aus.



7. Wählen Sie Rolle hinzufügen.

AWS CLI

Der folgende AWS CLI Befehl fügt die Rolle einer Oracle-DB-Instance mit dem Namen hinzu*mydbinstance*.

Example

Für LinuxmacOS, oderUnix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name S3_INTEGRATION \
   --role-arn your-role-arn
```

Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name S3_INTEGRATION ^
   --role-arn your-role-arn
```

Ersetzen Sie *your-role-arn* durch den Rollen-ARN, den Sie im vorherigen Schritt notiert haben. Für die Option `S3_INTEGRATION` muss `--feature-name` angegeben werden.

Hinzufügen der Amazon S3-Integrationsoption

Um Amazon RDS for Oracle mit Amazon S3 zu integrieren, muss Ihre DB-Instance einer Optionsgruppe zugeordnet sein, die die `S3_INTEGRATION` Option enthält.

Konsole

So konfigurieren Sie eine Optionsgruppe für die Amazon S3-Integration

1. Erstellen Sie eine neue Optionsgruppe oder identifizieren Sie eine vorhandene Optionsgruppe, der Sie die Option `S3_INTEGRATION` hinzufügen können.

   Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option `S3_INTEGRATION` zur Optionsgruppe hinzu.

   Weitere Informationen zum Hinzufügen einer Option zu einer Optionsgruppe finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Erstellen Sie eine neue RDS für Oracle DB-Instance und ordnen Sie ihr die Optionsgruppe zu, oder ändern Sie eine RDS für Oracle DB-Instance, um ihr die Optionsgruppe zuzuordnen.

   Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

   Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

So konfigurieren Sie eine Optionsgruppe für die Amazon S3-Integration

1. Erstellen Sie eine neue Optionsgruppe oder identifizieren Sie eine vorhandene Optionsgruppe, der Sie die Option `S3_INTEGRATION` hinzufügen können.

   Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option `S3_INTEGRATION` zur Optionsgruppe hinzu.

   Der folgende AWS CLI Befehl fügt die `S3_INTEGRATION` Option beispielsweise einer Optionsgruppe mit dem Namen hinzu**`myoptiongroup`**.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
   --option-group-name myoptiongroup \
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
   --option-group-name myoptiongroup ^
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. Erstellen Sie eine neue RDS für Oracle DB-Instance und ordnen Sie ihr die Optionsgruppe zu, oder ändern Sie eine RDS für Oracle DB-Instance, um ihr die Optionsgruppe zuzuordnen.

   Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

   Hinweise zum Ändern einer RDS For-Oracle-DB-Instance finden Sie unterÄndern einer Amazon RDS DB-Instance.

Übertragung von Dateien zwischen Amazon RDS for Oracle und einem Amazon S3 S3-Bucket

Um Dateien zwischen einer RDS Oracle-DB-Instance und einem Amazon S3 S3-Bucket zu übertragen, können Sie das RDS Amazon-Paket verwenden`rdsadmin_s3_tasks`. Sie können Dateien GZIP beim Hochladen komprimieren und beim Herunterladen dekomprimieren.

Themen

- Anforderungen und Einschränkungen für Dateiübertragungen
- Dateien von Ihrer RDS Oracle-DB-Instance in einen Amazon S3 S3-Bucket hochladen
- Hochladen von Dateien aus einem Amazon S3-Bucket zu einer Oracle-DB-Instance
- Überwachen des Status einer Dateiübertragung

Anforderungen und Einschränkungen für Dateiübertragungen

Bevor Sie Dateien zwischen Ihrer DB-Instance und einem Amazon S3 S3-Bucket übertragen, beachten Sie Folgendes:

- Das `rdsadmin_s3_tasks` Paket überträgt Dateien, die sich in einem einzigen Verzeichnis befinden. Sie können keine Unterverzeichnisse in eine Übertragung einbeziehen.
- Die maximale Objektgröße in einem Amazon S3-Bucket beträgt 5 TB.
- Aufgaben, die von `rdsadmin_s3_tasks` run asynchron erstellt wurden.
- Sie können Dateien aus dem Data Pump-Verzeichnis, z. B.`DATA_PUMP_DIR`, oder aus einem beliebigen vom Benutzer erstellten Verzeichnis hochladen. Sie können keine Dateien aus einem Verzeichnis hochladen, das von Oracle-Hintergrundprozessen verwendet wird, wie z. B. den `trace` Verzeichnissen `adump`bdump, oder.
- Das Download-Limit beträgt 2000 Dateien pro Prozeduraufruf für`download_from_s3`. Wenn Sie mehr als 2000 Dateien von Amazon S3 herunterladen müssen, teilen Sie den Download in separate Aktionen mit maximal 2000 Dateien pro Prozeduraufruf auf.
- Wenn in Ihrem Download-Ordner eine Datei vorhanden ist und Sie versuchen, eine Datei mit demselben Namen herunterzuladen überspringt `download_from_s3` den Download. Um eine Datei aus dem Download-Verzeichnis zu entfernen, verwenden Sie die SQL PL/-Prozedur UTL_FILE. FREMOVE.

Dateien von Ihrer RDS Oracle-DB-Instance in einen Amazon S3 S3-Bucket hochladen

Verwenden Sie zum Hochladen von Dateien aus Ihrer Oracle DB-Instance in einen Amazon-S3-Bucket das Verfahren `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`. Sie können beispielsweise Oracle Recovery Manager (RMAN) -Backup-Dateien oder Oracle Data Pump-Dateien hochladen. Weitere Informationen zur Arbeit mit Objekten finden Sie im Amazon Simple Storage Service User Guide. Weitere Hinweise zur Durchführung von RMAN Backups finden Sie unterAllgemeine RMAN Aufgaben für Oracle-DB-Instances ausführen.

Die Prozedur `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_bucket_name`	VARCHAR2	–	Erforderlich	Der Name des Amazon S3-Buckets, in den die

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
				Dateien hochgeladen werden sollen.
`p_directory_name`	VARCHAR2	–	Erforderlich	Der Name des Oracle-Verzeichnisobjekts, aus dem Dateien hochgeladen werden sollen. Das Verzeichnis kann jedes beliebige vom Benutzer erstellte Verzeichnisobjekt oder das Data Pump-Verzeichnis, z. B. , sei `DATA_PUMP_DIR` . Sie können keine Dateien aus einem Verzeichnis hochladen, das von Hintergrundprozessen wie `adump`bdump, und verwendet wird`trace`. **Note** Sie können nur Dateien aus dem angegebenen Verzeichnis hochladen . Sie können keine Dateien in Unterverzeichnisse des angegebenen Verzeichnisses hochladen.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_s3_prefix`	VARCHAR2	–	Erforderlich	Ein Amazon S3-Dateinamenspräfix, zu dem Dateien hochgeladen werden. Ein leeres Präfix lädt alle Dateien zur obersten Ebene im angegebenen Amazon S3-Bucket hoch und fügt kein Präfix an die Dateinamen an. Bei dem Präfix `folder_1/oradb` werden Dateien z. B. zu `folder_1` hochgeladen. In diesem Fall wird das Präfix `oradb` zu jeder Datei hinzugefügt.
`p_prefix`	VARCHAR2	–	Erforderlich	Ein Dateinamenspräfix, dem Dateinamen entsprechen müssen, damit die Dateien hochgeladen werden. Ein leeres Präfix lädt alle Dateien im angegebenen Verzeichnis hoch.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
`p_compression_level`	NUMBER	`0`	optional	Der Grad der GZIP Komprimierung. Der Bereich gültiger Werte lautet `0` bis 9: • `0` – Keine Komprimierung • 1 – Schnellste Komprimierung • 9 – Höchste Komprimierung
`p_bucket_owner_full_control`	VARCHAR2	–	optional	Die Einstellung der Zugriffssteuerung für den Bucket. Die einzigen gültigen Werte sind null und `FULL_CONTROL` . Diese Einstellung ist nur erforderlich, wenn Sie Dateien von einem Konto (Konto A) in einen Bucket hochladen, der einem anderen Konto (Konto B) gehört, und Konto B die vollständige Kontrolle über die Dateien benötigt.

Der Rückgabewert für die `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`-Prozedur ist eine Aufgaben-ID.

Im folgenden Beispiel werden alle Dateien im *`DATA_PUMP_DIR`* Verzeichnis in den Amazon S3 S3-Bucket mit dem Namen hochgeladen *`amzn-s3-demo-bucket`*. Die Dateien sind nicht komprimiert.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
```

```
      p_prefix         =>  '',
      p_s3_prefix      =>  '',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Das folgende Beispiel lädt alle Dateien mit dem Präfix *db* im Verzeichnis *DATA_PUMP_DIR* in den Amazon S3-Bucket *amzn-s3-demo-bucket* hoch. Amazon RDS wendet die höchste GZIP Komprimierungsstufe auf die Dateien an.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'amzn-s3-demo-bucket',
      p_prefix            =>  'db',
      p_s3_prefix         =>  '',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  9)
   AS TASK_ID FROM DUAL;
```

Das folgende Beispiel lädt alle Dateien im Verzeichnis *DATA_PUMP_DIR* in den Amazon S3-Bucket *amzn-s3-demo-bucket* hoch. Die Dateien werden in einen `dbfiles`-Ordner hochgeladen. In diesem Beispiel ist die GZIP Komprimierungsstufe *1*, das ist die schnellste Komprimierungsstufe.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'amzn-s3-demo-bucket',
      p_prefix            =>  '',
      p_s3_prefix         =>  'dbfiles/',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  1)
   AS TASK_ID FROM DUAL;
```

Das folgende Beispiel lädt alle Dateien im Verzeichnis *DATA_PUMP_DIR* in den Amazon S3-Bucket *amzn-s3-demo-bucket* hoch. Die Dateien werden in einen `dbfiles`-Order hochgeladen und `ora` wird an den Anfang eines jeden Dateinamens hinzugefügt. Es erfolgt keine Komprimierung.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
      p_prefix         =>  '',
      p_s3_prefix      =>  'dbfiles/ora',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Im folgenden Beispiel wird davon ausgegangen, dass der Befehl in Konto A ausgeführt wird, aber Konto B die vollständige Kontrolle über den Bucket-Inhalt benötigt. Der Befehl `rdsadmin_s3_tasks.upload_to_s3` überträgt alle Dateien im *`DATA_PUMP_DIR`*-Verzeichnis auf den Bucket namens *`s3bucket0wnedByAccountB`*. Die Zugriffskontrolle ist auf `FULL_CONTROL` eingestellt, damit Konto B auf die Dateien im Bucket zugreifen kann. Die GZIP Komprimierungsstufe ist *`6`*, wodurch Geschwindigkeit und Dateigröße ausgeglichen werden.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name               =>  's3bucket0wnedByAccountB',
      p_prefix                    =>  '',
      p_s3_prefix                 =>  '',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_bucket_owner_full_control =>  'FULL_CONTROL',
      p_compression_level         =>  6)
   AS TASK_ID FROM DUAL;
```

In jedem Beispiel gibt die Anweisung `SELECT` die ID der Aufgabe in einem `VARCHAR2`-Datentyp zurück.

Sie können das Ergebnis anzeigen, indem Sie die Ausgabedatei der Aufgabe anzeigen.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ersetzen *`Aufgaben-ID`* mit der Aufgaben-ID, die von der Prozedur zurückgegeben wurde.

> **Note**
>
> Die Aufgaben werden asynchron ausgeführt.

Hochladen von Dateien aus einem Amazon S3-Bucket zu einer Oracle-DB-Instance

Verwenden Sie das RDS Amazon-Verfahren, um Dateien von einem RDS Amazon S3 S3-Bucket auf eine Oracle-Instance herunterzuladen`rdsadmin.rdsadmin_s3_tasks.download_from_s3`.

Die Prozedur `download_from_s3` hat die folgenden Parameter.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_bucket_name	VARCHAI	–	Erforderlich	Der Name des Amazon S3-Buckets, aus dem die Dateien heruntergeladen werden sollen.
p_directory_name	VARCHAI	–	Erforderlich	Der Name des Oracle-Verzeichnisobjekts, in das die Dateien heruntergeladen werden sollen. Das Verzeichnis kann jedes beliebige vom Benutzer erstellte Verzeichnisobjekt oder das Data Pump-Verzeichnis, z. B. , sei DATA_PUMP_DIR .
p_error_on_zero_downloads	VARCHAI	FALSE	Optional	Ein Flag, das bestimmt, ob die Aufgabe einen Fehler auslöst, wenn keine Objekte im Amazon-S3-Bucket dem Präfix entsprechen. Wenn dieser Parameter nicht gesetzt oder auf FALSE (Standard) gesetzt ist, gibt die Aufgabe eine Meldung aus, dass keine Objekte gefunden wurden, löst aber keine Ausnahme aus oder schlägt fehl. Wenn dieser Parameter den Wert hatTRUE, löst die Aufgabe eine Ausnahme aus und schlägt fehl. Beispiele für Präfixspezifikationen, die bei Übereinstimmungsprüfungen problematisch sein können, sind Leerzeichen in Präfixen wie in ' import/test9.log' und Groß- und Kleinschreibung wie in test9.log und test9.LOG .

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_s3_prefix	VARCHAI	–	Erforderlich	Ein Dateinamenspräfix, dem Dateinamen entsprechen müssen, damit die Dateien heruntergeladen werden. Wenn das Präfix leer ist, werden alle Dateien auf der höchsten Verzeichnisebene des angegebenen Amazon S3-Buckets heruntergeladen, nicht jedoch Dateien in Unterordnern in diesem Bucket. Das Verfahren lädt Amazon S3-Objekte nur aus dem Ordner der ersten Ebene mit dem entsprechenden Präfix herunter. Verschachtelte Verzeichnisstrukturen, die dem angegebenen Präfix entsprechen, werden nicht heruntergeladen. Angenommen, ein Amazon S3-Bucket hat die Ordnerstruktur `folder_1/folder_2/folder_3` . Geben Sie das `'folder_1/folder_2/'` -Präfix an. In diesem Fall werden nur die Dateien in `folder_2` und nicht die Dateien in `folder_1` oder `folder_3` heruntergeladen. Wenn Sie stattdessen das Präfix `'folder_1/folder_2'` angeben, werden alle Dateien in `folder_1`, die mit dem Präfix `'folder_2'` übereinstimmen, heruntergeladen, und es werden keine Dateien in `folder_2` heruntergeladen.

Parametername	Datentyp	Standard	Erforderlich	Beschreibung
p_decompression_format	VARCHAR	–	Optional	Das Dekomprimierungsformat. Gültige Werte sind NONE für keine Dekomprimierung und GZIP für die Dekomprimierung.

Der Rückgabewert für die rdsadmin.rdsadmin_s3_tasks.download_from_s3-Prozedur ist eine Aufgaben-ID.

Das folgende Beispiel lädt alle Dateien namens *amzn-s3-demo-bucket* im Amazon-S3-Bucket in das *DATA_PUMP_DIR*-Verzeichnis herunter. Die Dateien werden nicht komprimiert, daher wird keine Dekomprimierung angewendet.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

Das folgende Beispiel lädt alle Dateien mit dem Präfix *db* im Amazon S3-Bucket *amzn-s3-demo-bucket* in das Verzeichnis *DATA_PUMP_DIR* herunter. Die Dateien werden mit komprimiertGZIP, sodass die Dekomprimierung angewendet wird. Der Parameter p_error_on_zero_downloads aktiviert die Präfixfehlerüberprüfung. Wenn das Präfix also mit keiner Datei im Bucket übereinstimmt, löst die Aufgabe eine Ausnahme und einen Fehler aus.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name               =>  'amzn-s3-demo-bucket',
      p_s3_prefix                 =>  'db',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_decompression_format      =>  'GZIP',
      p_error_on_zero_downloads   =>  'TRUE')
   AS TASK_ID FROM DUAL;
```

Das folgende Beispiel lädt alle Dateien im Ordner *myfolder/* im Amazon S3-Bucket *amzn-s3-demo-bucket* in das Verzeichnis *DATA_PUMP_DIR* herunter. Verwenden Sie zur Angabe des Amazon-S3-Ordners den Parameter p_s3_prefix. Die hochgeladenen Dateien werden mit komprimiertGZIP, aber beim Herunterladen nicht dekomprimiert.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name          =>  'amzn-s3-demo-bucket',
      p_s3_prefix            =>  'myfolder/',
      p_directory_name       =>  'DATA_PUMP_DIR',
      p_decompression_format =>  'NONE')
   AS TASK_ID FROM DUAL;
```

Das folgende Beispiel lädt die Datei *mydumpfile.dmp* im Amazon-S3-Bucket *amzn-s3-demo-bucket* ins Verzeichnis *DATA_PUMP_DIR* herunter. Es erfolgt keine Dekomprimierung.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'amzn-s3-demo-bucket',
      p_s3_prefix      =>  'mydumpfile.dmp',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

In jedem Beispiel gibt die Anweisung SELECT die ID der Aufgabe in einem VARCHAR2-Datentyp zurück.

Sie können das Ergebnis anzeigen, indem Sie die Ausgabedatei der Aufgabe anzeigen.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ersetzen *Aufgaben-ID* mit der Aufgaben-ID, die von der Prozedur zurückgegeben wurde.

> **Note**
>
> Die Aufgaben werden asynchron ausgeführt.
> Sie können mithilfe des Oracle-Verfahrens UTL_FILE.FREMOVE Dateien aus einem Verzeichnis entfernen. Weitere Informationen finden Sie unter FREMOVEVerfahren in der Oracle-Dokumentation.

Überwachen des Status einer Dateiübertragung

Dateiübertragungsaufgaben veröffentlichen RDS Amazon-Ereignisse, wenn sie beginnen und abgeschlossen sind. Die Ereignisnachricht enthält die Aufgaben-ID für die Dateiübertragung. Informationen zum Anzeigen dieser Grenze finden Sie unter RDSAmazon-Ereignisse anzeigen.

Sie können den Status einer laufenden Aufgabe in einer bdump-Datei einsehen. Die bdump-Dateien befinden sich im Verzeichnis `/rdsdbdata/log/trace`. Jeder bdump-Dateiname weist das folgende Format auf.

```
dbtask-task-id.log
```

Ersetzen Sie *`task-id`* durch die ID der Aufgabe, die Sie überwachen möchten.

> **Note**
>
> Die Aufgaben werden asynchron ausgeführt.

Sie können das gespeicherte Verfahren `rdsadmin.rds_file_util.read_text_file` zur Ansicht des Inhalts der bdump-Dateien verwenden. Beispiel: Die folgende Abfrage gibt den Inhalt der bdump-Datei *`dbtask-1234567890123-1234.log`* zurück.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-1234.log'));
```

Das folgende Beispiel zeigt die Protokolldatei für eine fehlgeschlagene Übertragung.

```
TASK_ID




                                                
```

```
------------------------------------------------------------------------------------------------
2023-04-17 18:21:33.993 UTC [INFO ] File #1: Uploading the file /rdsdbdata/datapump/
A123B4CDEF567890G1234567890H1234/sample.dmp to Amazon S3 with bucket name amzn-s3-demo-
bucket and key sample.dmp.
2023-04-17 18:21:34.188 UTC [ERROR] RDS doesn't have permission to write to Amazon S3
 bucket name amzn-s3-demo-bucket and key sample.dmp.
2023-04-17 18:21:34.189 UTC [INFO ] The task failed.
```

Fehlerbehebung für die Amazon-S3-Integration

Tipps zur Fehlerbehebung finden Sie im AWS re:POST-Artikel Wie behebe ich Probleme, wenn ich Amazon RDS for Oracle mit Amazon S3 integriere? .

Entfernen der Amazon S3-Integrationsoption

Sie können die Amazon S3-Integrationsoption aus einer DB-Instance entfernen.

Um die Amazon S3-Integrationsoption aus einer DB-Instance zu entfernen, führen Sie einen der folgenden Schritte durch:

- Um die Amazon S3-Integrationsoption aus mehreren DB-Instances zu entfernen, entfernen Sie die Option `S3_INTEGRATION` aus der Optionsgruppe, der die DB-Instances angehören. Diese Änderung wirkt sich auf alle DB-Instances aus, die die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.

- Um die Amazon S3-Integrationsoption aus einer einzelnen DB-Instance zu entfernen, ändern Sie die Instance und geben Sie eine andere Optionsgruppe an, in der die Option `S3_INTEGRATION` nicht enthalten ist. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle Application Express (APEX)

Amazon RDS unterstützt Oracle Application Express (APEX) mittels Verwendung von `APEX`- und `APEX-DEV`-Optionen. Sie können Oracle APEX als Laufzeitumgebung oder als vollständige Entwicklungsumgebung für webbasierte Anwendungen einsetzen. Mit Oracle APEX können Sie Anwendungen vollständig im Webbrowser erstellen. Weitere Informationen finden Sie unter Oracle Application Express in der Oracle-Dokumentation.

Themen

- Oracle APEX-Komponenten
- Anforderungen und Einschränkungen
- Oracle APEX und Oracle Rest Data Services (ORDS) einrichten
- Konfiguration von Oracle Rest Data Services (ORDS)
- Oracle APEX aktualisieren und entfernen

Oracle APEX-Komponenten

Oracle APEX besteht aus den folgenden Hauptkomponenten:

- Ein Repository, das die Metadaten für Oracle APEX-Anwendungen und -Komponenten speichert. Das Repository besteht aus Tabellen, Indizes und anderen Objekten, die in Ihrer Amazon RDS-DB-Instance installiert sind.
- Listener, der die HTTP-Kommunikation mit Oracle APEX-Clients verwaltet. Der Listener befindet sich auf einem separaten Host, z. B. einer EC2 Amazon-Instance, einem lokalen Server in Ihrem Unternehmen oder Ihrem Desktop-Computer. Der Listener akzeptiert eingehende Verbindungen von Webbrowsern, leitet sie zur Verarbeitung an die Amazon-RDS-DB-Instance weiter und sendet die Ergebnisse anschließend aus dem Repository an die Browser zurück.

  Amazon RDS für Oracle unterstützt die folgenden Listener:

  - Verwenden Sie für Oracle APEX Version 5.0 und höher Oracle REST Data Services (ORDS) Version 19.1 und höher. Es wird empfohlen, die neueste unterstützte Version von Oracle APEX und ORDS zu verwenden. Diese Dokumentation beschreibt ältere Versionen nur aus Gründen der Abwärtskompatibilität.
  - Für Oracle APEX Version 4.1.1 können Sie Oracle APEX Listener Version 1.1.4 verwenden.
  - Sie können Oracle HTTP Server und `mod_plsql` Listener verwenden.

> **Note**
>
> Amazon RDS unterstützt den Oracle XML DB-HTTP-Server mit dem eingebetteten PL/SQL gateway as a listener for Oracle APEX. In general, Oracle recommends against using the embedded PL/SQL Gateway für Anwendungen, die im Internet ausgeführt werden, nicht.

Weitere Informationen zu diesen Listener-Typen finden Sie unter Auswahl eines Web-Listeners in der Oracle-Dokumentation.

Wenn Sie die `APEX-DEV` Optionen `APEX` und zu Ihrer RDS for Oracle DB-Instance hinzufügen, installiert Amazon RDS nur das Oracle APEX-Repository. Installieren Sie Ihren Listener auf einem separaten Host.

Anforderungen und Einschränkungen

Im folgenden Thema werden die Anforderungen und Einschränkungen für Oracle APEX und ORDS aufgeführt.

Anforderungen an die Oracle APEX-Version

Die `APEX` Option verwendet Speicher in der DB-Instance-Klasse für Ihre DB-Instance. Nachfolgend finden Sie die unterstützten Versionen und die ungefähren Speicheranforderungen für Oracle APEX.

Oracle APEX-Version	Speicheranforderungen	Unterstützte Oracle-Datenbank-Versionen	Hinweise
Oracle APEX-Version 24.2.v1	114 MiB	Alle	Diese Version enthält Patch 37885097: PSE-BUNDLE FÜR APEX 24.2 (PSES ZUSÄTZLICH ZU 24.2.0), PATCH_VERSION 4.
Oracle APEX Version 24.1.v1	112 MiB	Alle	Diese Version enthält Patch 36695709: PSE-BUNDLE FÜR APEX 24.1 (PSES ZUSÄTZLICH ZU 24.1.0), PATCH_VERSION 3. Wenn Sie genau dieselbe APEX-Image-Version für die

Oracle APEX-Version	Speicheranforderungen	Unterstützte Oracle-Datenbank-Versionen	Hinweise
			Installation auf Ihrer EC2 Instance benötigen, laden Sie Patch 37544819:24.1.3 PSE BUNDLE FOR APEX 24.1 (PSES ZUSÄTZLICH ZU 24.1.0) herunter.
Oracle APEX Version 23.2.v1	110 MiB	Alle	Diese Version enthält Patch 35895964: PSE-BUNDLE FÜR APEX 23.2 (PSES ZUSÄTZLICH ZU 23.2.0), PATCH_VERSION 6. Wenn Sie genau dieselbe APEX-Image-Version für die Installation auf Ihrer EC2 Instance benötigen, laden Sie Patch 37593125:23.2.6 PSE BUNDLE FOR APEX 23.2 (PSES ZUSÄTZLICH ZU 23.2.0) herunter.
Oracle APEX-Version 23.1.v1	106 MiB	Alle	Diese Version enthält Patch 35283657: PSE BUNDLE FOR APEX 23.1 (PSES ON TOP OF 23.1.0), PATCH_VERSION 2.
Oracle APEX Version 22.2.v1	106 MiB	Alle	Diese Version enthält Patch 34628174: PSE BUNDLE FOR APEX 22.2 (PSES ON TOP OF 22.2.0), PATCH_VERSION 4.
Oracle APEX-Version 22.1.v1	124 MiB	Alle	Diese Version enthält Patch 34020981: PSE BUNDLE FOR APEX 22.1 (PSES ON TOP OF 22.1.0), PATCH_VERSION 6.
Oracle-APEX-Version 21.2.v1	125 MiB	Alle	Diese Version enthält Patch 33420059: PSE BUNDLE FOR APEX 21.2 (PSES ON TOP OF 21.2.0), PATCH_VERSION 8.
Oracle APEX-Version 21.1.v1	125 MiB	Alle	Diese Version enthält Patch 32598392: PSE BUNDLE FOR APEX 21.1, PATCH_VERSION 3.

Oracle APEX-Version	Speicheranforderungen	Unterstützte Oracle-Datenbank-Versionen	Hinweise
Oracle APEX-Version 20.2.v1	148 MiB	Alle außer Oracle Database 2.1.c	Diese Version enthält Patch 32006852: PSE BUNDLE FOR APEX 20.2, PATCH_VERSION 2020.11.12. Sie können die Patch-Nummer und das Datum sehen, indem Sie die folgende Abfrage ausführen: `SELECT PATCH_VERSION, PATCH_NUMBER FROM APEX_PATCHES;`
Oracle APEX-Version 20.1.v1	173 MiB	Alle außer Oracle Database 21c	Diese Version enthält Patch 30990551: PSE BUNDLE FOR APEX 20.1, PATCH_VERSION 2020.07.15.
Oracle APEX Version 19.2.v1	149 MiB	Alle außer Oracle Database 21c	
Oracle APEX Version 19.1.v1	148 MiB	Alle außer Oracle Database 21c	

Oracle APEX-ZIP-Dateien zum Herunterladen finden Sie in den Oracle APEX Prior Release-Archiven auf der Oracle-Website.

Voraussetzungen für Oracle APEX und ORDS

Beachten Sie die folgenden Voraussetzungen für die Verwendung von Oracle APEX und ORDS:

- Ihr System muss das Java Runtime Environment (JRE) verwenden.
- Ihre Oracle-Client-Installation muss Folgendes beinhalten:

- SQL*Plus oder SQL Developer für Administrationsaufgaben
- Oracle Net Services zum Konfigurieren von Verbindungen zu Ihrer DB-Instance von RDS für Oracle

Einschränkungen von Oracle APEX

Sie können das `APEX_version` Benutzerkonto, das von Amazon RDS verwaltet wird, nicht ändern. Daher können Sie für diesen Benutzer keine Datenbankprofile anwenden oder Kennwortregeln durchsetzen. Die Profile und Passworteinstellungen für `APEX_version` sind von Oracle vordefiniert AWS und so konzipiert, dass sie die Sicherheitsanforderungen für Amazon RDS erfüllen.

Oracle APEX und Oracle Rest Data Services (ORDS) einrichten

Im folgenden Thema sind die Schritte aufgeführt, die für die Einrichtung von Oracle APEX und ORDS erforderlich sind

Themen

- Hinzufügen der APEX- und APEX-DEV-Optionen zu Ihrer DB-Instance
- Entsperren des öffentlichen Benutzerkontos auf Ihrer DB-Instance
- RESTful Dienste für Oracle APEX konfigurieren
- Vorbereitung der Installation von ORDS auf einem separaten Host
- Einrichten von Oracle APEX Listener

Hinzufügen der APEX- und APEX-DEV-Optionen zu Ihrer DB-Instance

Gehen Sie wie folgt vor, um die `APEX-DEV` Optionen `APEX` und zu Ihrer RDS for Oracle DB-Instance hinzuzufügen:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie der Optionsgruppe die Optionen `APEX` und `APEX-DEV` hinzu.
3. Ordnen Sie die Optionsgruppe Ihrer DB-Instance zu.

Wenn Sie die `APEX-DEV` Optionen `APEX` und und hinzufügen, kommt es zu einem kurzen Ausfall, während Ihre DB-Instance automatisch neu gestartet wird.

> **Note**
>
> `APEX_MAIL` ist verfügbar, wenn die Option `APEX` installiert ist. Das Ausführungsrecht für das `APEX_MAIL`-Paket wird `PUBLIC` gewährt. Sie benötigen also kein APEX-Administratorkonto, um es verwenden zu können.

Um die APEX- und APEX-DEV-Optionen zu einer DB-Instance hinzuzufügen

1. Bestimmen Sie die Optionsgruppe, die Sie verwenden möchten. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie im Feld Engine die Oracle-Edition aus, die Sie verwenden möchten. Die `APEX-DEV` Optionen `APEX` und werden in allen Editionen unterstützt.

   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.

2. Fügen Sie der Optionsgruppe die Optionen hinzu. Wenn Sie nur die Oracle APEX-Laufzeitumgebung bereitstellen möchten, fügen Sie nur die `APEX` Option hinzu. Um die vollständige Entwicklungsumgebung bereitzustellen, fügen Sie die `APEX-DEV` Optionen `APEX` und hinzu.

   Wählen Sie unter Version die Version von Oracle APEX aus, die Sie verwenden möchten.

   > **Important**
   >
   > Wenn Sie die `APEX-DEV` Optionen `APEX` oder zu einer vorhandenen Optionsgruppe hinzufügen, die bereits mit einer oder mehreren DB-Instances verknüpft ist, kommt es zu einem kurzen Ausfall. Während dieses Ausfalls werden alle DB-Instances automatisch neu gestartet.

   Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Wenn Sie die `APEX-DEV` Optionen `APEX` oder zu einer vorhandenen DB-Instance hinzufügen, kommt es zu einem kurzen Ausfall, während Ihre DB-Instance automatisch neu gestartet wird. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Entsperren des öffentlichen Benutzerkontos auf Ihrer DB-Instance

Nachdem Sie die `APEX-DEV` Optionen `APEX` oder die Optionen Ihrer DB-Instance installiert haben, stellen Sie sicher, dass Sie wie folgt vorgehen:

1. Ändern Sie das Passwort für das `APEX_PUBLIC_USER` Konto.
2. Entsperren Sie das Konto.

Zu diesem Zweck können Sie das Befehlszeilen-Dienstprogramm Oracle SQL*Plus verwenden. Stellen Sie als Master-Benutzer eine Verbindung zur DB-Instance her und rufen Sie die folgenden Befehle auf. Ersetzen Sie `new_password` durch ein Passwort Ihrer Wahl.

```
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password;
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
```

RESTful Dienste für Oracle APEX konfigurieren

Um RESTful Dienste in Oracle APEX zu konfigurieren (für Oracle APEX 4.1.1.V1 nicht erforderlich), verwenden Sie SQL*Plus, um als Master-Benutzer eine Verbindung zu Ihrer DB-Instance herzustellen. Führen Sie anschließend die gespeicherte Prozedur `rdsadmin.rdsadmin_run_apex_rest_config` aus. Beim Ausführen der gespeicherten Prozedur geben Sie Passwörter für die folgenden Benutzer an:

- `APEX_LISTENER`
- `APEX_REST_PUBLIC_USER`

Die gespeicherte Prozedur führt das `apex_rest_config.sql`-Skript aus, mit dem neue Datenbankkonten für diese Benutzer erstellt werden.

> **Note**
>
> Für Oracle APEX Version 4.1.1.v1 ist keine Konfiguration erforderlich. Für diese Oracle APEX-Version müssen Sie die gespeicherte Prozedur nicht ausführen.

Der folgende Befehl führt die gespeicherte Prozedur aus.

```
EXEC rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',
 'apex_rest_public_user_password');
```

Vorbereitung der Installation von ORDS auf einem separaten Host

Installieren Sie ORDS auf einem separaten Host, z. B. einer EC2 Amazon-Instance, einem lokalen Server in Ihrem Unternehmen oder Ihrem Desktop-Computer. Bei den Beispielen in diesem Abschnitt wird davon ausgegangen, dass Ihr Host Linux ausführt und benannt `myapexhost.example.com` ist.

Bevor Sie ORDS installieren können, müssen Sie einen Betriebssystembenutzer ohne Zugriffsrechte erstellen und dann die Oracle APEX-Installationsdatei herunterladen und entpacken.

So bereiten Sie die ORDS-Installation vor:

1. Melden Sie sich bei `myapexhost.example.com` als `root` an.
2. Erstellen Sie einen Betriebssystembenutzer ohne administrative Rechte für die Listener-Installation. Mit dem folgenden Befehl wird ein neuer Benutzer mit dem Namen apexuser erstellt:

   ```
   useradd -d /home/apexuser apexuser
   ```

   Der folgende Befehl weist dem neuen Benutzer ein Passwort zu.

   ```
   passwd apexuser;
   ```

3. Melden Sie sich an `myapexhost.example.com` als `apexuser` und laden Sie die Oracle APEX-Installationsdatei von Oracle in Ihr Verzeichnis herunter: `/home/apexuser`

   - http://www.oracle. com/technetwork/developer-tools/apex/downloads/index.html

- Frühere Veröffentlichungsarchive von Oracle Application Express

4. Entpacken Sie die Datei im `/home/apexuser`-Verzeichnis.

```
unzip apex_version.zip
```

Nachdem Sie die Datei entpackt haben, ist ein `apex`-Verzeichnis im `/home/apexuser`-Verzeichnis vorhanden.

5. Während Sie noch angemeldet `myapexhost.example.com` sind`apexuser`, laden Sie die Oracle REST Data Services-Datei von Oracle in Ihr `/home/apexuser` Verzeichnis herunter: http://www.oracle. com/technetwork/developer-tools/apex-listener/downloads/index.html.

Einrichten von Oracle APEX Listener

> **Note**
>
> Oracle APEX Listener ist veraltet.

Amazon RDS for Oracle unterstützt weiterhin Oracle APEX Version 4.1.1 und Oracle APEX Listener Version 1.1.4. Es wird empfohlen, die neuesten unterstützten Versionen von Oracle APEX und ORDS zu verwenden.

Installieren Sie Oracle APEX Listener auf einem separaten Host, z. B. einer EC2 Amazon-Instance, einem lokalen Server in Ihrem Unternehmen oder Ihrem Desktop-Computer. Wir unterstellen `myapexhost.example.com` als Namen des Hosts, auf dem außerdem Linux ausgeführt wird.

Vorbereiten der Installation von Oracle APEX Listener

Bevor Sie Oracle APEX Listener installieren können, müssen Sie einen nicht privilegierten Betriebssystembenutzer erstellen und dann die Oracle APEX-Installationsdatei herunterladen und entpacken.

So bereiten Sie die Installation von Oracle APEX Listener vor:

1. Melden Sie sich bei `myapexhost.example.com` als `root` an.
2. Erstellen Sie einen Betriebssystembenutzer ohne administrative Rechte für die Listener-Installation. Mit dem folgenden Befehl wird ein neuer Benutzer mit dem Namen apexuser erstellt:

```
useradd -d /home/apexuser apexuser
```

Der folgende Befehl weist dem neuen Benutzer ein Passwort zu.

```
passwd apexuser;
```

3. Melden Sie sich an `myapexhost.example.com` als `apexuser` und laden Sie die Oracle APEX-Installationsdatei von Oracle in Ihr Verzeichnis herunter: `/home/apexuser`
   - http://www.oracle. com/technetwork/developer-tools/apex/downloads/index.html
   - Frühere Veröffentlichungsarchive von Oracle Application Express
4. Entpacken Sie die Datei im `/home/apexuser`-Verzeichnis.

   ```
   unzip apex_<version>.zip
   ```

   Nachdem Sie die Datei entpackt haben, ist ein `apex`-Verzeichnis im `/home/apexuser`-Verzeichnis vorhanden.
5. Wenn Sie bei `myapexhost.example.com` noch als `apexuser` angemeldet sind, laden Sie die APEX Listener-Datei von Oracle in Ihr `/home/apexuser`-Verzeichnis herunter:

Installieren und Konfigurieren von Oracle APEX Listener

Bevor Sie Oracle APEX verwenden können, müssen Sie die `apex.war` Datei herunterladen, Oracle APEX Listener mit Java installieren und dann den Listener starten.

So installieren und konfigurieren Sie Oracle APEX Listener:

1. Erstellen Sie ein neues Verzeichnis basierend auf Oracle APEX Listener und öffnen Sie die Listener-Datei.

   Führen Sie folgenden Code aus:

   ```
   mkdir /home/apexuser/apexlistener
   cd /home/apexuser/apexlistener
   unzip ../apex_listener.version.zip
   ```
2. Führen Sie folgenden Code aus.

```
java -Dapex.home=./apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -
jar ./apex.war
```

3. Geben Sie folgende Informationen in der Eingabeaufforderung des Programms ein:

   - Benutzername des APEX-Listener-Administrators. Der Standardwert ist adminlistener.
   - Passwort für den APEX-Listener-Administrator.
   - Benutzername des APEX-Listener-Managers. Der Standardwert ist managerlistener.
   - Passwort für den APEX-Listener-Administrator.

   Das Programm gibt eine URL aus, die Sie benötigen, um die Konfiguration abzuschließen.

   ```
   INFO: Please complete configuration at: http://localhost:8080/apex/
   listenerConfigure
   Database is not yet configured
   ```

4. Lassen Sie Oracle APEX Listener laufen, damit Sie Oracle Application Express verwenden können. Nachdem Sie dieses Konfigurationsverfahren abgeschlossen haben, können Sie den Listener im Hintergrund ausführen lassen.
5. Rufen Sie in Ihrem Webbrowser die URL auf, die vom Oracle APEX Listener-Programm bereitgestellt wird. Das Oracle Application Express Listener-Administrationsfenster wird angezeigt. Geben Sie die folgenden Informationen ein:

   - Benutzername – `APEX_PUBLIC_USER`
   - Password: Passwort für APEX_PUBLIC_USER. Dieses Passwort haben Sie zuvor bei der Konfiguration des Oracle APEX-Repositorys angegeben. Weitere Informationen finden Sie unter Entsperren des öffentlichen Benutzerkontos auf Ihrer DB-Instance.
   - Connection type (Verbindungstyp) – Basic (Einfach)
   - Hostname: Endpunkt der Amazon RDS-DB-Instance, z. B. `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`.
   - Port – 1521
   - SID: Name der Datenbank in der Amazon RDS-DB-Instance, z. B. `mydb`.
6. Wählen Sie Apply (Anwenden) aus. Das Oracle APEX-Administrationsfenster wird angezeigt.

7. Legen Sie ein Passwort für den Oracle APEX-Benutzer `admin` fest. Verwenden Sie dazu SQL*Plus, um sich mit Ihrer DB-Instance als Master-Benutzer zu verbinden, und führen Sie dann die folgenden Befehle aus.

   ```
   EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
   grant APEX_ADMINISTRATOR_ROLE to master;
   @/home/apexuser/apex/apxchpwd.sql
   ```

   Ersetzen Sie *`master`* durch den Masterbenutzernamen. Wenn Sie vom Skript `apxchpwd.sql` aufgefordert werden, geben Sie ein neues `admin`-Passwort ein.

8. Kehren Sie in Ihrem Browser zum Oracle APEX-Administrationsfenster zurück und wählen Sie Administration. Wählen Sie anschließend Application Express Internal Administration. Wenn Sie zur Eingabe von Anmeldeinformationen aufgefordert werden, geben Sie die folgenden Informationen ein:

   - Benutzername – `admin`
   - Passwort: Das mit dem Skript `apxchpwd.sql` festgelegte Passwort

   Wählen Sie Anmeldung und legen Sie dann ein neues Passwort für den Benutzer `admin` fest.

Ihr Listener ist nun einsatzbereit.

Konfiguration von Oracle Rest Data Services (ORDS)

Das folgende Thema listet die Konfigurationsoptionen für ORDS 21 und 22 auf:

Themen

- Installation und Konfiguration von ORDS 21 und niedriger
- Installation und Konfiguration von ORDS 2.2 und höher

Installation und Konfiguration von ORDS 21 und niedriger

Sie können jetzt Oracle Rest Data Services (ORDS) für die Verwendung mit Oracle APEX installieren und konfigurieren. Verwenden Sie für Oracle APEX Version 5.0 und höher die ORDS-Versionen 19.1 bis 21. Informationen zur Installation von ORDS 22 und höher finden Sie unter. Installation und Konfiguration von ORDS 2.2 und höher

Installieren Sie den Listener auf einem separaten Host, z. B. einer EC2 Amazon-Instance, einem lokalen Server in Ihrem Unternehmen oder Ihrem Desktop-Computer. Für die Beispiele in diesem Abschnitt gehen wir davon aus, dass der Name Ihres Hosts `myapexhost.example.com` lautet und dass auf Ihrem Host Linux ausgeführt wird.

Um ORDS 2.1 und niedriger für die Verwendung mit Oracle APEX zu installieren und zu konfigurieren

1. Gehen Sie zu Oracle REST Data Services und lesen Sie die Readme-Datei. Stellen Sie sicher, dass Sie die erforderliche Version von Java installiert haben.
2. Erstellen Sie ein neues Verzeichnis für Ihre ORDS-Installation.

   ```
   mkdir /home/apexuser/ORDS
   cd /home/apexuser/ORDS
   ```

3. Laden Sie die Datei `ords.version.number.zip` von Oracle REST Data Services herunter.
4. Entpacken Sie die Datei im Verzeichnis `/home/apexuser/ORDS`.
5. Wenn Sie ORDS in einer Mehrmandantendatenbank installieren, fügen Sie der Datei `/home/apexuser/ORDS/params/ords_params.properties` die folgende Zeile hinzu:

   ```
   pdb.disable.lockdown=false
   ```

6. Gewährt dem Master-Benutzer die erforderlichen Rechte für die Installation von ORDS.

   Nachdem die Optionen für Oracle APEX installiert sind, geben Sie dem Masterbenutzer die erforderlichen Rechte, um das ORDS-Schema zu installieren. Sie können dies tun, indem Sie sich mit der Datenbank verbinden und die folgenden Befehle ausführen. Ersetzen Sie *MASTER_USER* durch den Großbuchstaben Ihres Hauptbenutzers.

   > **Important**
   >
   > Verwenden Sie bei der Eingabe des Benutzernamens Großbuchstaben, es sei denn, Sie haben den Benutzer mit einer Kennung mit Groß- und Kleinschreibung erstellt. Wenn Sie z. B. `CREATE USER myuser` oder `CREATE USER MYUSER` ausführen , wird im Datenwörterbuch `MYUSER` gespeichert. Wenn Sie jedoch doppelte Anführungszeichen in `CREATE USER "MyUser"` verwenden, speichert das Datenwörterbuch `MyUser`. Weitere Informationen finden Sie unter Gewährung SELECT von EXECUTE Rechten für SYS Objekte.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);
```

> **Note**
>
> Diese Befehle gelten für ORDS Version 19.1 und später.

7. Installieren Sie das ORDS-Schema mit Hilfe der heruntergeladenen ords.war-Datei.

```
java -jar ords.war install advanced
```

Das Programm fordert Sie zur Eingabe der folgenden Informationen auf. Die Standardwerte stehen in eckigen Klammern. Weitere Informationen finden Sie unter Introduction to Oracle REST Data Services in der Oracle-Dokumentation.

- Geben Sie den Speicherort für die Konfigurationsdaten ein:

  Geben Sie ei */home/apexuser/ORDS*. Dies ist der Speicherort der ORDS-Konfigurationsdateien.

- Geben Sie den zu verwendenden Datenbankverbindungstyp an. Geben Sie die Nummer für [1] Basic [2] TNS [3] Benutzerdefinierte URL [1] ein:

  Wählen Sie den gewünschten Verbindungstyp.

- Geben Sie den Namen des Datenbankservers [localhost] ein: *DB_instance_endpoint*

  Wählen Sie den Standardwert aus oder geben Sie einen passenden Wert ein.

- Geben Sie den Datenbank-Listener-Port [1521] ein: *DB_instance_port*

  Wählen Sie den Standardwert aus oder geben Sie einen passenden Wert ein.

- Geben Sie 1 zum Angeben des Datenbank-Service-Namens und 2 zum Angeben der Datenbank-SID [1] ein.

  Wählen Sie 2, um die Datenbank-SID anzugeben.

- Datenbank-SID [xe]

  Wählen Sie den Standardwert aus oder geben Sie einen passenden Wert ein.

- Geben Sie 1 ein, wenn Sie das Oracle REST Data Services-Schema verifizieren/installieren möchten, oder 2, um diesen Schritt [1] zu überspringen:

  Wählen Sie 1. In diesem Schritt wird der Oracle REST Data Services-Proxy-Benutzer mit dem Namen ORDS_PUBLIC_USER erstellt.

- Geben Sie das Datenbankpasswort für ORDS_PUBLIC_USER ein:

  Geben Sie das Passwort ein und bestätigen Sie es.

- Erfordert für die Anmeldung mit Administratorrechten, um das Oracle REST Data Services-Schema zu überprüfen.

  Geben Sie den Administrator-Benutzernamen ein: *master_user*

  Geben Sie das Datenbankkennwort ein für*master_user*: *master_user_password*

  Bestätigen Sie das Passwort: *master_user_password*

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

- Geben Sie den Standard-Tablespace für ORDS_METADATA [SYSAUX] ein.

  Geben Sie den temporären Tablespace für ORDS_METADATA [TEMP] ein.

  Geben Sie den Standard-Tablespace für ORDS_PUBLIC_USER [USERS] ein.

  Geben Sie den temporären Tablespace für ORDS_PUBLIC_USER [TEMP] ein.
- Geben Sie 1 ein, wenn Sie das PL/SQL-Gateway verwenden möchten, oder 2, um diesen Schritt zu überspringen. Wenn Sie Oracle Application Express verwenden oder von mod_plsql migrieren, müssen Sie 1 [1] eingeben.

  Wählen Sie den Standardwert aus.
- Geben Sie den Benutzernamen [APEX_PUBLIC_USER] der PL/SQL-Gateway-Datenbank ein.

  Wählen Sie den Standardwert aus.
- Geben Sie das Datenbankpasswort für APEX_PUBLIC_USER ein:

  Geben Sie das Passwort ein und bestätigen Sie es.
- Geben Sie 1 ein, um Passwörter für Application Express RESTful Services-Datenbankbenutzer (APEX_LISTENER, APEX_REST_PUBLIC_USER) anzugeben, oder 2, um diesen Schritt zu überspringen [1]:

  Wählen Sie 2 für APEX 4.1.1.V1 und 1 für alle anderen APEX-Versionen.
- [Nicht erforderlich für APEX 4.1.1.v1] Datenbankpasswort für APEX_LISTENER

  Geben Sie das Passwort ein (falls erforderlich) und bestätigen Sie es.
- [Nicht erforderlich für APEX 4.1.1.v1] Datenbankpasswort für APEX_REST_PUBLIC_USER

  Geben Sie das Passwort ein (falls erforderlich) und bestätigen Sie es.
- Geben Sie eine Zahl ein, um eine Funktion auszuwählen, die aktiviert werden soll:

  Geben Sie 1 ein, um alle Funktionen zu aktivieren: SQL Developer Web, REST Enabled SQL und Datenbank-API.
- Geben Sie 1 ein, wenn Sie im Standalone-Modus starten möchten, oder 2, um [1] zu beenden:

Geben Sie ei 1.

- Geben Sie den Speicherort für statische APEX-Ressourcen ein:

  Wenn Sie APEX-Installationsdateien in `/home/apexuser` entpackt haben, geben Sie `/home/apexuser/apex/images` ein. Geben Sie andernfalls ein, wo sich das Verzeichnis befindet, in dem Sie die Datei entpackt *`unzip_path`*`/apex/images` haben. *`unzip_path`*

- Geben Sie 1 ein, wenn HTTP verwendet wird, oder 2 bei Verwendung von HTTPS [1]:

  Geben Sie bei Eingabe von 1 den HTTP-Port an. Geben Sie bei Eingabe von 2 den HTTPS-Port und den SSL-Hostnamen an. Die HTTPS-Option fordert Sie dazu auf, anzugeben, wie Sie das Zertifikat bereitstellen möchten:

  - Geben Sie 1 ein, um das selbstsignierte Zertifikat zu verwenden.
  - Geben Sie 2 ein, um Ihr eigenes Zertifikat vorzulegen. Geben Sie bei der Eingabe von 2 den Pfad für das SSL-Zertifikat und den Pfad für den privaten Schlüssel des SSL-Zertifikats an.

8. Legen Sie ein Passwort für den APEX `admin`-Benutzer fest. Verwenden Sie dazu SQL*Plus, um sich mit Ihrer DB-Instance als Master-Benutzer zu verbinden, und führen Sie dann die folgenden Befehle aus.

   ```
   EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
   grant APEX_ADMINISTRATOR_ROLE to master;
   @/home/apexuser/apex/apxchpwd.sql
   ```

   Ersetzen Sie *`master`* durch den Masterbenutzernamen. Wenn Sie vom Skript `apxchpwd.sql` aufgefordert werden, geben Sie ein neues `admin`-Passwort ein.

9. Starten Sie den ORDS-Listener. Führen Sie folgenden Code aus.

   ```
   java -jar ords.war
   ```

   Wenn Sie ORDS erstmals starten, werden Sie aufgefordert, die Position der APEX Static-Ressourcen anzugeben. Dieser Ordner für Abbilder befindet sich im Verzeichnis `/apex/images` im Installationsverzeichnis von APEX.

10. Kehren Sie in Ihrem Browser zum Oracle APEX-Administrationsfenster zurück und wählen Sie Administration. Wählen Sie anschließend Application Express Internal Administration. Wenn Sie zur Eingabe von Anmeldeinformationen aufgefordert werden, geben Sie die folgenden Informationen ein:

- Benutzername – `admin`
- Passwort: Das mit dem Skript `apxchpwd.sql` festgelegte Passwort

Wählen Sie Anmeldung und legen Sie dann ein neues Passwort für den Benutzer `admin` fest.

Ihr Listener ist nun einsatzbereit.

Installation und Konfiguration von ORDS 2.2 und höher

Sie können jetzt Oracle Rest Data Services (ORDS) für die Verwendung mit Oracle APEX installieren und konfigurieren. Für die Beispiele in diesem Abschnitt gehen wir davon aus, dass der Name Ihres separaten Hosts lautet `myapexhost.example.com` und dass auf Ihrem Host Linux ausgeführt wird. Die Anweisungen für ORDS 22 unterscheiden sich von den Anweisungen für frühere Versionen.

Um ORDS 22 und höher für die Verwendung mit Oracle APEX zu installieren und zu konfigurieren

1. Gehen Sie zu Oracle REST Data Services und lesen Sie die Readme-Datei für die ORDS-Version, die Sie herunterladen möchten. Stellen Sie sicher, dass Sie die erforderliche Version von Java installiert haben.
2. Erstellen Sie ein neues Verzeichnis für Ihre ORDS-Installation.

   ```
   mkdir /home/apexuser/ORDS
   cd /home/apexuser/ORDS
   ```

3. Laden Sie die Datei `ords.`*`version.number`*`.zip` oder `ords-latest.zip` von Oracle REST Data Services herunter.
4. Entpacken Sie die Datei im Verzeichnis `/home/apexuser/ORDS`.
5. Gewährt dem Master-Benutzer die erforderlichen Rechte für die Installation von ORDS.

   Nachdem die APEX Option installiert ist, geben Sie dem Master-Benutzer die erforderlichen Rechte, um das ORDS-Schema zu installieren. Sie können dies tun, indem Sie sich bei der Datenbank anmelden und die folgenden Befehle ausführen. Ersetzen Sie *`MASTER_USER`* durch den Großbuchstaben Ihres Hauptbenutzers.

> **Important**
>
> Verwenden Sie bei der Eingabe des Benutzernamens Großbuchstaben, es sei denn, Sie haben den Benutzer mit einer Kennung mit Groß- und Kleinschreibung erstellt. Wenn Sie z. B. `CREATE USER myuser` oder `CREATE USER MYUSER` ausführen , wird im Datenwörterbuch `MYUSER` gespeichert. Wenn Sie jedoch doppelte Anführungszeichen in `CREATE USER "MyUser"` verwenden, speichert das Datenwörterbuch `MyUser`. Weitere Informationen finden Sie unter Gewährung SELECT von EXECUTE Rechten für SYS Objekte.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);

exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOB', 'MASTER_USER', 'EXECUTE',
 true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_ASSERT', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_OUTPUT', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SCHEDULER', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('HTP', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('OWA', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPG_DOCLOAD', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_CRYPTO', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_METADATA', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SQL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('UTL_SMTP', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_NETWORK_ACL_ADMIN',
 'MASTER_USER', 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('SESSION_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_USERS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACL_PRIVILEGES',
 'MASTER_USER', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACLS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY', 'MASTER_USER',
 'SELECT', true);
```

> **Note**
>
> Die obigen Befehle gelten für ORDS 22 und höher.

6. Installieren Sie das ORDS-Schema mithilfe des heruntergeladenen `ords` Skripts. Geben Sie Verzeichnisse an, die Konfigurationsdateien und Protokolldateien enthalten sollen. Die Oracle Corporation empfiehlt, diese Verzeichnisse nicht in dem Verzeichnis zu platzieren, das die ORDS-Produktsoftware enthält.

```
mkdir -p /home/apexuser/ords_config /home/apexuser/ords_logs

/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs
```

Verwenden Sie für DB-Instances, auf denen die Container-Datenbank-Architektur (CDB) ausgeführt wird, ORDS 23.2 und höher und übergeben Sie das `--pdb-skip-disable-lockdown` Argument bei der Installation von ORDS.

```
/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs --pdb-skip-disable-
lockdown
```

Das Programm fordert Sie zur Eingabe der folgenden Informationen auf. Die Standardwerte stehen in eckigen Klammern. Weitere Informationen finden Sie unter Introduction to Oracle REST Data Services in der Oracle-Dokumentation.

- `Choose the type of installation:`

  Wählen Sie**2**, ob Sie ORDS-Schemas in der Datenbank installieren und einen Datenbankverbindungspool in den lokalen ORDS-Konfigurationsdateien erstellen möchten.

- `Specify the database connection type to use. Enter number for [1] Basic [2] TNS [3] Custom URL:`

  Wählen Sie den gewünschten Verbindungstyp. In diesem Beispiel wird davon ausgegangen, dass Sie wählen**1**.

- `Enter the name of the database server [localhost]:` ***DB_instance_endpoint***

  Wählen Sie den Standardwert aus oder geben Sie einen passenden Wert ein.

- `Enter the database listener port [1521]:` ***DB_instance_port***

  Wählen Sie die Standardeinstellung **1521** oder geben Sie den richtigen Wert ein.

- `Enter the database service name [orcl]:`

Geben Sie den Datenbanknamen ein, der von Ihrer RDS for Oracle DB-Instance verwendet wird.

- `Provide database user name with administrator privileges`

  Geben Sie den Master-Benutzernamen für Ihre RDS for Oracle DB-Instance ein.

- `Enter the database password for [username]:`

  Geben Sie das Master-Benutzerkennwort für Ihre RDS for Oracle DB-Instance ein.

- `Enter the default tablespace for ORDS_METADATA and ORDS_PUBLIC_USER [SYSAUX]:`
- `Enter the temporary tablespace for ORDS_METADATA [TEMP]. Enter the default tablespace for ORDS_PUBLIC_USER [USERS]. Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP].`
- `Enter a number to select additional feature(s) to enable [1]:`
- `Enter a number to configure and start ORDS in standalone mode [1]:`

  2Entscheiden Sie sich dafür, ORDS sofort im Standalone-Modus zu starten.

- `Enter a number to select the protocol [1] HTTP`
- `Enter the HTTP port [8080]:`
- `Enter the APEX static resources location:`

  Geben Sie den Pfad zu den Oracle APEX-Installationsdateien ein (`/home/apexuser/apex/images`).

7. Legen Sie ein Passwort für den Oracle APEX-Benutzer `admin` fest. Verwenden Sie dazu SQL*Plus, um sich mit Ihrer DB-Instance als Master-Benutzer zu verbinden, und führen Sie dann die folgenden Befehle aus.

   ```
   EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
   grant APEX_ADMINISTRATOR_ROLE to master;
   @/home/apexuser/apex/apxchpwd.sql
   ```

   Ersetzen Sie *`master`* durch den Masterbenutzernamen. Wenn Sie vom Skript `apxchpwd.sql` aufgefordert werden, geben Sie ein neues `admin`-Passwort ein.

8. Führen Sie ORDS im Standalone-Modus aus, indem Sie das `ords` Skript mit dem `serve` Befehl verwenden. Für Produktionsbereitstellungen sollten Sie die Verwendung unterstützter Java

EE-Anwendungsserver wie Apache Tomcat oder Oracle WebLogic Server in Betracht ziehen. Weitere Informationen finden Sie unter Deployment and Monitoring von Oracle REST Data Services in der Oracle Database-Dokumentation.

```
/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config serve \
  --port 8193 \
  --apex-images /home/apexuser/apex/images
```

Wenn ORDS ausgeführt wird, aber nicht auf die Oracle APEX-Installation zugreifen kann, wird möglicherweise der folgende Fehler angezeigt, insbesondere bei Nicht-CDB-Instances.

```
The procedure named apex_admin could not be accessed, it may not be declared,
 or the user executing this request may not have been granted execute privilege
 on the procedure, or a function specified by security.requestValidationFunction
 configuration property has prevented access.
```

Um diesen Fehler zu beheben, ändern Sie die von ORDS verwendete Anforderungsvalidierungsfunktion, indem Sie das `ords` Skript mit dem Befehl ausführen. `config` Standardmäßig verwendet ORDS das `ords_util.authorize_plsql_gateway` Verfahren, das nur auf CDB-Instances unterstützt wird. Für Nicht-CDB-Instances können Sie dieses Verfahren im Paket ändern. `wwv_flow_epg_include_modules.authorize` Bewährte Methoden zur Konfiguration der richtigen Anforderungsvalidierungsfunktion für Ihren Anwendungsfall finden Sie in der Oracle Database-Dokumentation und im Oracle-Support.

9. Kehren Sie in Ihrem Browser zum Oracle APEX-Administrationsfenster zurück und wählen Sie Administration. Wählen Sie anschließend Application Express Internal Administration. Wenn Sie zur Eingabe von Anmeldeinformationen aufgefordert werden, geben Sie die folgenden Informationen ein:

   - Benutzername – `admin`
   - Passwort: Das mit dem Skript `apxchpwd.sql` festgelegte Passwort

   Wählen Sie Anmeldung und legen Sie dann ein neues Passwort für den Benutzer `admin` fest.

Ihr Listener ist nun einsatzbereit.

Oracle APEX aktualisieren und entfernen

Folgen Sie den Anweisungen in diesem Thema, um Oracle APEX zu aktualisieren oder zu entfernen:

Themen

- Die Oracle APEX-Version aktualisieren
- Die APEX- und APEX-DEV-Optionen werden entfernt

Die Oracle APEX-Version aktualisieren

> **Important**
>
> Erstellen Sie eine Sicherungskopie Ihrer DB-Instance, bevor Sie ein Upgrade von Oracle APEX durchführen. Weitere Informationen erhalten Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS und Testen eines Oracle DB-Upgrades.

Gehen Sie wie folgt vor, um Oracle APEX mit Ihrer DB-Instance zu aktualisieren:

- Erstellen Sie eine neue Optionsgruppe für die aktualisierte Version Ihrer DB-Instance.
- Fügen Sie die aktualisierten Versionen der `APEX-DEV` Optionen `APEX` und der neuen Optionsgruppe hinzu. Achten Sie darauf, alle anderen Optionen aufzunehmen, die Ihre DB-Instance verwendet. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.
- Geben Sie beim Aktualisieren der DB-Instance die neue Optionsgruppe für Ihre aktualisierte DB-Instance an.

Nach dem Upgrade Ihrer Version von Oracle APEX ist das Oracle APEX-Schema für die vorherige Version möglicherweise noch in Ihrer Datenbank vorhanden. Wenn Sie es nicht mehr benötigen, können Sie das alte Oracle APEX-Schema nach dem Upgrade aus Ihrer Datenbank löschen.

Wenn Sie die Oracle APEX-Version aktualisieren und RESTful Dienste in der vorherigen Oracle APEX-Version nicht konfiguriert wurden, empfehlen wir Ihnen, Dienste zu konfigurieren. RESTful Weitere Informationen finden Sie unter RESTful Dienste für Oracle APEX konfigurieren.

In einigen Fällen, wenn Sie ein Upgrade der Hauptversion Ihrer DB-Instance planen, stellen Sie möglicherweise fest, dass Sie eine Oracle APEX-Version verwenden, die nicht mit Ihrer Zieldatenbankversion kompatibel ist. In diesen Fällen können Sie Ihre Version von Oracle APEX

aktualisieren, bevor Sie Ihre DB-Instance aktualisieren. Wenn Sie zuerst Oracle APEX aktualisieren, kann dies die Zeit reduzieren, die für das Upgrade Ihrer DB-Instance benötigt wird.

> **Note**
>
> Installieren und konfigurieren Sie nach dem Upgrade von Oracle APEX einen Listener für die Verwendung mit der aktualisierten Version. Detaillierte Anweisungen finden Sie unter Einrichten von Oracle APEX Listener.

Die APEX- und APEX-DEV-Optionen werden entfernt

Sie können die `APEX-DEV` Optionen `APEX` und aus einer DB-Instance entfernen. Gehen Sie wie folgt vor, um diese Optionen aus Ihrer DB-Instance zu entfernen:

- Um die `APEX-DEV` Optionen `APEX` und aus mehreren DB-Instances zu entfernen, entfernen Sie die Optionen aus der Optionsgruppe, zu der sie gehören. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Wenn Sie die Optionen aus einer Optionsgruppe entfernen, die an mehrere DB-Instances angehängt ist, kommt es beim Neustart der DB-Instances zu einem kurzen Ausfall.

  Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Um die `APEX-DEV` Optionen `APEX` und aus einer einzelnen DB-Instance zu entfernen, ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, die diese Optionen nicht enthält. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Wenn Sie die Optionen entfernen, kommt es zu einem kurzen Ausfall, während Ihre DB-Instance automatisch neu gestartet wird.

  Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Wenn Sie die `APEX-DEV` Optionen `APEX` und aus einer DB-Instance entfernen, wird das APEX-Schema aus Ihrer Datenbank entfernt.

EFSAmazon-Integration

Amazon Elastic File System (AmazonEFS) bietet serverlosen, vollständig elastischen Dateispeicher, sodass Sie Dateidaten gemeinsam nutzen können, ohne Speicherkapazität und Leistung bereitstellen oder verwalten zu müssen. Mit Amazon EFS können Sie ein Dateisystem erstellen und es dann VPC über die NFS Versionen 4.0 und 4.1 (NFSv4) in Ihr Protokoll einbinden. Dann können Sie das EFS Dateisystem wie jedes andere POSIX -konforme Dateisystem verwenden. Allgemeine Informationen finden Sie unter Was ist Amazon Elastic File System? und der AWS Blog Integrate Amazon RDS for Oracle with Amazon EFS.

Themen

- Überblick über die EFS Amazon-Integration
- Konfiguration von Netzwerkberechtigungen RDS für die Oracle-Integration mit Amazon EFS
- Konfigurieren von IAM-Berechtigungen für die Integration von RDS für Oracle in Amazon EFS
- Die INTEGRATION Option EFS _ hinzufügen
- Konfiguration von EFS Amazon-Dateisystemberechtigungen
- Übertragung von Dateien zwischen RDS Oracle und einem EFS Amazon-Dateisystem
- Die INTEGRATION Option EFS _ entfernen
- Fehlerbehebung bei der EFS Amazon-Integration

Überblick über die EFS Amazon-Integration

Mit Amazon EFS können Sie Dateien zwischen Ihrer RDS Oracle-DB-Instance und einem EFS Dateisystem übertragen. Sie können es beispielsweise verwenden, EFS um die folgenden Anwendungsfälle zu unterstützen:

- Gemeinsame Nutzung eines Dateisystems durch mehrere Anwendungen und Datenbankserver.
- Erstellen eines freigegebenen Verzeichnisses für migrationsbezogene Dateien, einschließlich transportierbarer Tabellenbereichsdatendateien. Weitere Informationen finden Sie unter Migrieren mithilfe von Oracle Transportable Tablespaces.
- Speichern und Teilen archivierter Redo-Protokolldateien, ohne zusätzlichen Speicherplatz auf dem Server zuzuweisen.
- Verwenden von Oracle-Database-Dienstprogrammen wie `UTL_FILE` zum Lesen und Schreiben von Dateien.

Vorteile der EFS Amazon-Integration

Wenn Sie sich für ein EFS Dateisystem gegenüber alternativen Datenübertragungslösungen entscheiden, profitieren Sie von den folgenden Vorteilen:

- Sie können Oracle Data Pump-Dateien zwischen Amazon EFS und Ihrer RDS Oracle-DB-Instance übertragen. Sie müssen diese Dateien nicht lokal kopieren, da Data Pump direkt aus dem EFS Dateisystem importiert. Weitere Informationen finden Sie unter Daten in Oracle auf Amazon importieren RDS.
- Die Datenmigration ist schneller als die Verwendung eines Datenbanklinks.
- Sie vermeiden es, Speicherplatz auf Ihrer RDS Oracle-DB-Instance zuzuweisen, um die Dateien zu speichern.
- Ein EFS Dateisystem kann Speicher automatisch skalieren, ohne dass Sie ihn bereitstellen müssen.
- Für die EFS Amazon-Integration fallen keine Mindestgebühren oder Einrichtungskosten an. Sie zahlen nur das, was Sie nutzen.
- EFSDie Amazon-Integration unterstützt zwei Formen der Verschlüsselung: Verschlüsselung von Daten bei der Übertragung und Verschlüsselung im Ruhezustand. Die Verschlüsselung von Daten während der Übertragung ist in TLS Version 1.2 standardmäßig aktiviert. Sie können die Verschlüsselung von Daten im Ruhezustand aktivieren, wenn Sie ein EFS Amazon-Dateisystem erstellen. Weitere Informationen finden Sie unter Verschlüsselung von Daten im Ruhezustand im Benutzerhandbuch zu Amazon Elastic File System.

Anforderungen für die EFS Amazon-Integration

Stellen Sie sicher, dass Sie die folgenden Anforderungen erfüllen:

- In Ihrer Datenbank muss die Datenbankversion 19.0.0.0.ru-2022-07.rur-2022-07.r1 oder höher ausgeführt werden.
- Ihre DB-Instance und Ihr Dateisystem müssen sich im selben,, und befindenEFS. AWS-Region VPC AWS-Konto RDSfür Oracle unterstützt keinen konto- und regionsübergreifenden Zugriff für. EFS
- Sie VPC müssen sowohl DNSAuflösung als auch DNSHostnamen aktiviert haben. Weitere Informationen finden Sie unter den DNSAttributen VPC in Ihrem Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.
- Ihr EFS Dateisystem muss die Speicherklasse Standard oder Standard-IA verwenden.

- Wenn Sie im `mount` Befehl einen DNS Namen verwenden, stellen Sie sicher, dass Sie für die Verwendung des von Amazon bereitgestellten DNS Servers VPC konfiguriert sind. Benutzerdefinierte DNS Server werden nicht unterstützt.
- Sie müssen für die Sicherung Ihres EFS Dateisystems andere RDS Lösungen verwenden. RDSfür Oracle unterstützt keine automatisierten Backups oder manuellen DB-Snapshots eines EFS Dateisystems. Weitere Informationen finden Sie unter Sichern Ihrer EFS Amazon-Dateisysteme.

Konfiguration von Netzwerkberechtigungen RDS für die Oracle-Integration mit Amazon EFS

RDSDamit Oracle in Amazon integriert werden kannEFS, stellen Sie sicher, dass Ihre DB-Instance Netzwerkzugriff auf ein EFS Dateisystem hat. Weitere Informationen finden Sie unter Steuern des Netzwerkzugriffs auf EFS Amazon-Dateisysteme für NFS Clients im Amazon Elastic File System-Benutzerhandbuch.

Themen

- Netzwerkzugriffskontrolle mit Sicherheitsgruppen
- Steuern des Netzwerkzugriffs mit Dateisystemrichtlinien

Netzwerkzugriffskontrolle mit Sicherheitsgruppen

Sie können den Zugriff Ihrer DB-Instance auf EFS Dateisysteme mithilfe von Sicherheitsmechanismen auf Netzwerkebene wie VPC Sicherheitsgruppen steuern. Um Ihrer DB-Instance den Zugriff auf ein EFS Dateisystem zu ermöglichen, stellen Sie sicher, dass Ihr EFS Dateisystem die folgenden Anforderungen erfüllt:

- In jeder Availability Zone, die von einer Oracle-DB-Instance verwendet wird, ist ein EFS RDS Mount-Ziel vorhanden.

  Ein EFSMount-Ziel stellt eine IP-Adresse für einen NFSv4 Endpunkt bereit, an dem Sie ein EFS Dateisystem mounten können. Sie mounten Ihr Dateisystem anhand seines DNS Namens, der in die IP-Adresse des EFS Mount-Ziels in der von der Availability Zone Ihrer DB-Instance verwendeten Datenbank aufgelöst wird.

  Sie können DB-Instances in verschiedenen Konfigurationen konfigurierenAZs, um dasselbe EFS Dateisystem zu verwenden. Für Multi-AZ benötigen Sie für jede AZ in Ihrer Bereitstellung einen Mountingpunkt. Möglicherweise müssen Sie eine DB-Instance in eine andere AZ

verschieben. Aus diesen Gründen empfehlen wir, dass Sie in jeder AZ Ihrer Umgebung einen EFS Bereitstellungspunkt erstellenVPC. Wenn Sie mit der Konsole ein neues EFS Dateisystem erstellen, werden standardmäßig Mount-Ziele für alle RDS erstelltAZs.

- Eine Sicherheitsgruppe ist an das Mountingziel angehängt.
- Die Sicherheitsgruppe hat eine Regel RDS für eingehenden Datenverkehr, die das Netzwerk-Subnetz oder die Sicherheitsgruppe der Oracle-DB-Instance auf TCP /2049 (Typ) zulässt. NFS

Weitere Informationen finden Sie unter Erstellen von EFS Amazon-Dateisystemen und Erstellen und Verwalten von EFS Mount-Zielen und Sicherheitsgruppen im Amazon Elastic File System-Benutzerhandbuch.

Steuern des Netzwerkzugriffs mit Dateisystemrichtlinien

Die EFS Amazon-Integration mit RDS for Oracle funktioniert mit der standardmäßigen (leeren) EFS Dateisystemrichtlinie. Die Standardrichtlinie wird nicht IAM zur Authentifizierung verwendet. Stattdessen gewährt sie jedem anonymen Client, der über ein Mountingziel eine Verbindung mit dem Dateisystem herstellen kann, Vollzugriff. Die Standardrichtlinie gilt immer dann, wenn keine vom Benutzer konfigurierte Dateisystemrichtlinie wirksam ist, auch bei der Erstellung des Dateisystems. Weitere Informationen finden Sie unter EFSStandard-Dateisystemrichtlinie im Amazon Elastic File System-Benutzerhandbuch.

Um den Zugriff auf Ihr EFS Dateisystem für alle Clients, auch RDS für Oracle, zu verbessern, können Sie IAM Berechtigungen konfigurieren. Bei diesem Ansatz erstellen Sie eine Dateisystemrichtlinie. Weitere Informationen finden Sie unter Erstellen von Dateisystemrichtlinien im Benutzerhandbuch für Amazon Elastic File System.

Konfigurieren von IAM-Berechtigungen für die Integration von RDS für Oracle in Amazon EFS

Standardmäßig verwendet die Amazon EFS-Integrationsfunktion keine IAM-Rolle: Die `USE_IAM_ROLE` Optionseinstellung ist`FALSE`. Um RDS for Oracle mit Amazon EFS und einer IAM-Rolle zu integrieren, muss Ihre DB-Instance über IAM-Berechtigungen für den Zugriff auf ein Amazon EFS-Dateisystem verfügen.

Themen

- Schritt 1: Erstellen einer IAM-Rolle für Ihre DB-Instance und Anfügen Ihrer Richtlinie
- Schritt 2: Erstellen einer Dateisystemrichtlinie für Ihr Amazon-EFS-Dateisystem

- Schritt 3: Zuordnen Ihrer IAM-Rolle zu Ihrer DB-Instance von RDS für Oracle

Schritt 1: Erstellen einer IAM-Rolle für Ihre DB-Instance und Anfügen Ihrer Richtlinie

Bei diesem Schritt erstellen Sie eine Rolle für Ihre DB-Instance von RDS für Oracle, um Amazon RDS den Zugriff auf Ihr EFS-Dateisystem zu ermöglichen.

Konsole

So erstellen Sie eine IAM-Rolle, um Amazon RDS; den Zugriff auf ein EFS-Dateisystem zu ermöglichen

1. Öffnen Sie die IAM-Managementkonsole.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie für denAWS Service RDS aus.
5. Wählen Sie unter Select your use case (Anwendungsfall auswählen) die Option RDS – Add Role to Database (RDS – Rolle zur Datenbank hinzufügen) aus.
6. Wählen Sie Weiter aus.
7. Fügen Sie keine Berechtigungsrichtlinien hinzu. Wählen Sie Weiter aus.
8. Legen Sie unter Role Name (Rollenname) einen Namen für Ihre IAM-Rolle fest, zum Beispiel `rds-efs-integration-role`. Sie können auch einen optionalen Wert für Description (Beschreibung) hinzufügen.
9. Wählen Sie Rolle erstellen aus.

AWS CLI

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Vertrauensbeziehungen, um die Berechtigungen des Services auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Sie können beide globalen Bedingungskontextschlüssel verwenden und der Wert `aws:SourceArn` enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der Wert `aws:SourceAccount` und das Konto im Wert `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Anweisung verwendet werden.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Stellen Sie in der Vertrauensbeziehung sicher, dass Sie den globalen Bedingungskontextschlüssel `aws:SourceArn` mit dem vollständigen Amazon-Ressourcennamen (ARN) der Ressourcen verwenden, die auf die Rolle zugreifen.

Der folgende AWS CLI Befehl erstellt die *`rds-efs-integration-role`* für diesen Zweck benannte Rolle.

Example

Für LinuxmacOS, oderUnix:

```
aws iam create-role \
   --role-name rds-efs-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": my_account_ID,
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Windows:

```
aws iam create-role ^
```

```
    --role-name rds-efs-integration-role ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole",
          "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": my_account_ID,
                  "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
              }
          }
        }
      ]
    }'
```

Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.

Schritt 2: Erstellen einer Dateisystemrichtlinie für Ihr Amazon-EFS-Dateisystem

Bei diesem Schritt erstellen Sie eine Dateisystemrichtlinie für Ihr Amazon-EFS-Dateisystem.

So erstellen oder bearbeiten Sie eine Dateisystemrichtlinie

1. Öffnen Sie die EFS-Managementkonsole.
2. Wählen Sie File Systems (Dateisysteme) aus.
3. Wählen Sie auf der Seite File systems (Dateisysteme) das Dateisystem aus, für das Sie eine Dateisystemrichtlinie bearbeiten oder erstellen möchten. Die Detailseite für dieses Dateisystem wird angezeigt.
4. Wählen Sie die Registerkarte File system policy (Dateisystemrichtlinie) aus.

   Wenn die Richtlinie leer ist, wird die standardmäßige EFS-Dateisystemrichtlinie verwendet. Weitere Informationen finden Sie unter Standardrichtlinie für das EFS-Dateisystem im Benutzerhandbuch für Amazon Elastic File System.
5. Wählen Sie Bearbeiten aus. Die Seite File system policy (Dateisystemrichtlinie) wird angezeigt.

6. Geben Sie im Policy editor (Richtlinien-Editor) eine Richtlinie wie die folgende ein, und wählen Sie dann Save (Speichern) aus.

   JSON

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/rds-efs-integration-
role"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-
east-1:123456789012:file-system/fs-1234567890abcdef0"
        }
    ]
}
```

Schritt 3: Zuordnen Ihrer IAM-Rolle zu Ihrer DB-Instance von RDS für Oracle

Bei diesem Schritt ordnen Sie Ihre IAM-Rolle Ihrer DB-Instance zu. Beachten Sie die folgenden Anforderungen:

- Sie müssen Zugriff auf eine IAM-Rolle haben, der die erforderliche Amazon-EFS-Berechtigungsrichtlinie angefügt ist.
- Sie können jeweils nur eine IAM-Rolle Ihrer DB-Instance von RDS für Oracle hinzufügen.
- Der Status Ihrer Instance muss Available lauten.

Weitere Informationen finden Sie unter Identity and Access Management für Amazon EFS im Benutzerhandbuch für Amazon Elastic File System.

Konsole

So ordnen Sie Ihre IAM-Rolle Ihrer DB-Instance von RDS for Oracle zu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbanken aus.
3. Wenn Ihre Datenbank-Instance nicht verfügbar ist, wählen Sie die Option Aktionen und anschließend Starten. Sobald der Instance-Status Gestartet lautet, fahren Sie mit dem nächsten Schritt fort.
4. Wählen Sie den Namen der Oracle DB-Instance aus, um deren Details anzuzeigen.
5. Auf derKonnektivität & SicherheitScrollen Sie nach unten zumIAM-Rollen verwaltenunten auf der Seite.
6. Wählen Sie die Rolle aus, die Sie hinzufügen möchten in Fügen Sie dieser Instanz IAM-Rollen hinzu.
7. Wählen Sie unter Feature (Funktion) die Option EFS_INTEGRATION aus.
8. Wählen Sie Rolle hinzufügen aus.

AWS CLI

Der folgende AWS CLI Befehl fügt die Rolle einer Oracle-DB-Instance mit dem Namen hinzu*mydbinstance*.

Example

Für LinuxmacOS, oderUnix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name EFS_INTEGRATION \
   --role-arn your-role-arn
```

Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name EFS_INTEGRATION ^
```

```
    --role-arn your-role-arn
```

Ersetzen Sie *your-role-arn* durch den Rollen-ARN, den Sie im vorherigen Schritt notiert haben. Für die Option `EFS_INTEGRATION` muss `--feature-name` angegeben werden.

Die INTEGRATION Option EFS _ hinzufügen

Um Amazon RDS for Oracle mit Amazon zu integrierenEFS, muss Ihre DB-Instance einer Optionsgruppe zugeordnet sein, die die `EFS_INTEGRATION` Option enthält.

Mehrere Oracle-DB-Instances, die derselben Optionsgruppe angehören, verwenden dasselbe EFS Dateisystem. Verschiedene DB-Instances können auf dieselben Daten zugreifen, der Zugriff kann jedoch durch die Verwendung verschiedener Oracle-Verzeichnisse aufgeteilt werden. Weitere Informationen finden Sie unter Übertragung von Dateien zwischen RDS Oracle und einem EFS Amazon-Dateisystem.

Konsole

Um eine Optionsgruppe für die EFS Amazon-Integration zu konfigurieren

1. Erstellen Sie eine neue Optionsgruppe oder identifizieren Sie eine vorhandene Optionsgruppe, der Sie die Option `EFS_INTEGRATION` hinzufügen können.

   Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option `EFS_INTEGRATION` zur Optionsgruppe hinzu. Sie müssen die `EFS_ID`-Dateisystem-ID angeben und das `USE_IAM_ROLE`-Flag festlegen.

   Weitere Informationen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe auf eine der folgenden Arten Ihrer DB-Instance zu:

   - Erstellen Sie eine neue Oracle-DB-Instance und ordnen sie sie der Optionsgruppe zu. Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Ändern Sie eine Oracle-DB-Instance, um ihr die Optionsgruppe zuzuordnen. Informationen über das Ändern einer Oracle-DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

Um eine Optionsgruppe für die EFS Integration zu konfigurieren

1. Erstellen Sie eine neue Optionsgruppe oder identifizieren Sie eine vorhandene Optionsgruppe, der Sie die Option `EFS_INTEGRATION` hinzufügen können.

   Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option `EFS_INTEGRATION` zur Optionsgruppe hinzu.

   Mit dem folgenden AWS CLI Befehl wird die `EFS_INTEGRATION` Option beispielsweise einer Optionsgruppe mit dem Namen hinzugefügt**`myoptiongroup`**.

   Example

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   aws rds add-option-to-option-group \
      --option-group-name myoptiongroup \
      --options "OptionName=EFS_INTEGRATION,OptionSettings=\
      [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   aws rds add-option-to-option-group ^
      --option-group-name myoptiongroup ^
      --options "OptionName=EFS_INTEGRATION,OptionSettings=^
      [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
   ```
3. Ordnen Sie die Optionsgruppe auf eine der folgenden Arten Ihrer DB-Instance zu:

   - Erstellen Sie eine neue Oracle-DB-Instance und ordnen sie sie der Optionsgruppe zu. Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Ändern Sie eine Oracle-DB-Instance, um ihr die Optionsgruppe zuzuordnen. Informationen über das Ändern einer Oracle-DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Konfiguration von EFS Amazon-Dateisystemberechtigungen

Standardmäßig hat nur der Root-Benutzer (UID0) Lese-, Schreib- und Ausführungsberechtigungen für ein neu erstelltes EFS Dateisystem. Damit auch andere Benutzer das Dateisystem ändern können, muss Ihnen der Root-Benutzer ausdrücklich Zugriff gewähren. Der Benutzer RDS für die Oracle-DB-Instance befindet sich in der `others` Kategorie. Weitere Informationen finden Sie unter Arbeiten mit Benutzern, Gruppen und Berechtigungen auf Netzwerk-Dateisystemebene (NFS) im Amazon Elastic File System-Benutzerhandbuch.

Gehen Sie wie folgt vor, damit Ihre RDS Oracle-DB-Instance Dateien in einem EFS Dateisystem lesen und schreiben kann:

- Mounten Sie ein EFS Dateisystem lokal auf Ihrer Amazon EC2 - oder lokalen Instance.
- Konfigurieren Sie differenzierte Berechtigungen.

Um `other` Benutzern beispielsweise Schreibberechtigungen für das EFS Dateisystem-Stammverzeichnis zu gewähren, führen Sie die Ausführung in `chmod 777` diesem Verzeichnis aus. Weitere Informationen finden Sie unter Beispiele für Anwendungsfälle und Berechtigungen für das EFS Amazon-Dateisystem im Amazon Elastic File System-Benutzerhandbuch.

Übertragung von Dateien zwischen RDS Oracle und einem EFS Amazon-Dateisystem

Um Dateien zwischen einer RDS for Oracle-Instance und einem EFS Amazon-Dateisystem zu übertragen, erstellen Sie mindestens ein Oracle-Verzeichnis und konfigurieren Sie EFS Dateisystemberechtigungen, um den DB-Instance-Zugriff zu kontrollieren.

Themen

- Erstellen eines Oracle-Verzeichnisses
- Übertragung von Daten zu und von einem EFS Dateisystem: Beispiele

Erstellen eines Oracle-Verzeichnisses

verwenden Sie die Prozedur `rdsadmin.rdsadmin_util.create_directory_efs`, um ein Oracle-Verzeichnis zu erstellen. Die Prozedur hat die folgenden Parameter.

Parameter name	Datentyp	Standa	Erforde ich	Beschreibung
`p_directory_name`	VARCHAR	–	Ja	Der Name des Oracle-Verzeichnisses.
`p_path_on_efs`	VARCHAR	–	Ja	Der Pfad im EFS Dateisystem. Das Präfix des Pfadnamens verwendet das Muster `/rdsefs-`*`fsid`*`/`, wobei *`fsid`* ist ein Platzhalter für Ihre EFS Dateisystem-ID. Wenn Ihr EFS Dateisystem beispielsweise benannt `fs-1234567890abcdef0` ist und Sie in diesem Dateisystem ein Unterverzeichnis mit dem Namen erstellen`mydir`, könnten Sie den folgenden Wert angeben: `/rdsefs-fs-1234567890abcdef0/mydir`

Gehen Sie davon aus, dass Sie ein Unterverzeichnis mit dem Namen `/datapump1` auf dem EFS Dateisystem erstellen. `fs-1234567890abcdef0` Im folgenden Beispiel wird ein Oracle-Verzeichnis erstelltDATA_PUMP_DIR_EFS, das auf das `/datapump1` Verzeichnis im EFS Dateisystem verweist. Dem Dateisystempfadwert für den `p_path_on_efs`-Parameter wird die Zeichenfolge `/rdsefs-` vorangestellt.

```
BEGIN
  rdsadmin.rdsadmin_util.create_directory_efs(
    p_directory_name => 'DATA_PUMP_DIR_EFS',
    p_path_on_efs    => '/rdsefs-fs-1234567890abcdef0/datapump1');
END;
/
```

Übertragung von Daten zu und von einem EFS Dateisystem: Beispiele

Im folgenden Beispiel wird Oracle Data Pump verwendet, um die Tabelle mit dem Namen `MY_TABLE` in die Datei `datapump.dmp` zu exportieren. Diese Datei befindet sich in einem EFS Dateisystem.

```
DECLARE
```

```
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'EXPORT', job_mode => 'TABLE',
 job_name=>null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-exp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Im folgenden Beispiel wird Oracle Data Pump verwendet, um die Tabelle mit dem Namen `MY_TABLE` aus der Datei `datapump.dmp` zu importieren. Diese Datei befindet sich in einem EFS Dateisystem.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'TABLE',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-imp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Weitere Informationen finden Sie unter Daten in Oracle auf Amazon importieren RDS.

Die INTEGRATION Option EFS _ entfernen

Die Schritte zum Entfernen der `EFS_INTEGRATION` Option hängen davon ab, ob Sie die Option aus mehreren DB-Instances oder einer einzelnen Instance entfernen.

Anzahl von DB-Instances	Aktion	Ähnliche Informationen
Mehrere	Entfernen Sie die `EFS_INTEGRATION` Option aus der Optionsgruppe, zu der die DB-Instances gehören. Diese Änderung wirkt sich auf alle Instances aus, die die Optionsgruppe verwenden.	Entfernen einer Option aus einer Optionsgruppe
Einzel	Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, in der die `EFS_INTEGRATION` -Option nicht enthalten ist. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben.	Ändern einer Amazon RDS DB-Instance

Nachdem Sie die `EFS_INTEGRATION` Option entfernt haben, können Sie optional das EFS Dateisystem löschen, das mit Ihren DB-Instances verbunden war.

Fehlerbehebung bei der EFS Amazon-Integration

Ihre RDS Oracle-DB-Instance überwacht die Konnektivität zu einem EFS Amazon-Dateisystem. Wenn die Überwachung ein Problem erkennt, versucht sie möglicherweise, das Problem zu beheben und ein Ereignis in der RDS Konsole zu veröffentlichen. Weitere Informationen finden Sie unter RDSAmazon-Ereignisse anzeigen.

Verwenden Sie die Informationen in diesem Abschnitt, um häufig auftretende Probleme bei der Arbeit mit der EFS Amazon-Integration zu diagnostizieren und zu beheben.

Benachrichtigung	Beschreibung	Aktion
`The EFS for RDS Oracle instance instance_name isn't available on the primary host. NFS port 2049 of your EFS isn't reachable.`	Die DB-Instance kann nicht mit dem EFS Dateisystem kommunizieren.	Stellen Sie Folgendes sicher: • Das EFS Dateisystem ist vorhanden. • Die Sicherheitsgruppe, die an das EFS Mount-Ziel angehängt ist, verfügt über eine Eingangsregel, die die Sicherheitsgruppe oder das Netzwerk-Subnetz der RDS Oracle-DB-Instance auf TCP /2049 (Typ) zulässt. NFS
`The EFS isn't reachable.`	Bei der Installation der `EFS_INTEGRATION` -Option ist ein Fehler aufgetreten.	Stellen Sie Folgendes sicher: • Das EFS Dateisystem ist vorhanden. • Die Sicherheitsgruppe, die an das EFS Mount-Ziel angehängt ist, verfügt über eine Eingangsregel, die die Sicherheitsgruppe oder das Netzwerk-Subnetz der RDS Oracle-DB-Instance auf TCP /2049 (Typ) zulässt. NFS • Das `enableDnsSupport` Attribut ist für Ihren aktiviert . VPC • Sie verwenden den von Amazon bereitgestellten DNS Server in IhremVPC. EFSDie Amazon-Integration

Benachrichtigung	Beschreibung	Aktion
		funktioniert nicht mit einer benutzerdefinierten Version DHCPDNS.
`The associated role with your DB instance wasn't found.`	Bei der Installation der `EFS_INTEGRATION` -Option ist ein Fehler aufgetreten.	Stellen Sie sicher, dass Sie Ihrer RDS Oracle-DB-Instance eine IAM Rolle zugeordnet haben.
`The associated role with your DB instance wasn't found.`	Bei der Installation der `EFS_INTEGRATION` -Option ist ein Fehler aufgetreten. RDSfür Oracle wurde aus einem DB-Snapshot mit der `USE_IAM_ROLE` Optionsei nstellung von wiederher gestelltTRUE.	Stellen Sie sicher, dass Sie Ihrer RDS Oracle-DB-Instance eine IAM Rolle zugeordnet haben.
`The associated role with your DB instance wasn't found.`	Bei der Installation der `EFS_INTEGRATION` -Option ist ein Fehler aufgetreten. RDSfor Oracle wurde aus einer all-in-one CloudForm ation Vorlage mit der `USE_IAM_ROLE` Optionsei nstellung von erstelltTRUE.	Um das Problem zu umgehen, führen Sie die folgenden Schritte aus: 1. Erstellen Sie eine DB-Instan ce mit der IAM Rolle und der Standardoptionsgruppe. 2. Fügen Sie bei einem nachfolgenden Stack-Upd ate die benutzerdefinierte Optionsgruppe mit der `EFS_INTEGRATION` Option hinzu.

Benachrichtigung	Beschreibung	Aktion
`PLS-00302: component 'CREATE_DIRECTORY_EFS' must be declared`	Dieser Fehler kann auftreten, wenn Sie eine Version von RDS for Oracle verwenden, die Amazon nicht unterstützt EFS.	Stellen Sie sicher, dass Sie RDS für die Oracle-DB-Instance Version 19.0.0.0.ru-2022-07.r1 oder höher verwenden.
`Read access of your EFS is denied. Check your file system policy.`	Ihre EFS DB-Instance kann das Dateisystem nicht lesen.	Stellen Sie sicher, dass Ihr EFS Dateisystem Lesezugriff über die IAM Rolle oder auf EFS Dateisystemebene zulässt.
N/A	Ihre DB-Instance kann nicht in das EFS Dateisystem schreiben.	Gehen Sie dazu wie folgt vor: 1. Stellen Sie sicher, dass Ihr EFS Dateisystem auf einer EC2 Amazon-Instance gemountet ist. 2. Erteilen Sie Ihrem RDS Benutzer Schreibzugriff für die `others` Gruppe. Die einfachste Methode besteht darin, den `chmod 777` Befehl im obersten Verzeichnis des EFS Dateisystems auszuführen.

Oracle Java Virtual Machine

Amazon RDS unterstützt Oracle Java Virtual Machine (JVM) durch die Verwendung der Option `JVM`. Oracle Java bietet ein SQL-Schema und Funktionen, die Oracle Java-Funktionen in einer Oracle-Datenbank ermöglichen. Weitere Informationen finden Sie unter Introduction to Java in Oracle Database in der Oracle-Dokumentation. Sie können Oracle JVM mit allen Versionen von Oracle Database 21c (21.0.0) und Oracle Database 19c (19.0.0) verwenden.

Überlegungen zu Oracle JVM

Die Java-Implementierung Amazon RDS verfügt über einen begrenzten Satz von Berechtigungen. Dem Masterbenutzer wird die Rolle `RDS_JAVA_ADMIN` erteilt, die eine Teilmenge der von der Rolle `JAVA_ADMIN` gewährten Berechtigungen gewährt. Um die der Rolle `RDS_JAVA_ADMIN` erteilten Berechtigungen aufzulisten, führen Sie die folgende Abfrage für Ihre DB-Instance durch:

```
SELECT * FROM dba_java_policy
   WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
   AND enabled = 'ENABLED'
   ORDER BY type_name, name, grantee;
```

Voraussetzungen für Oracle JVM

Für die Verwendung von Oracle Java gelten folgende Voraussetzungen:

- Ihre DB-Instance muss einer ausreichend großen Klasse angehören. Oracle Java wird für die DB-Instance-Klassen db.t3.small nicht unterstützt. Weitere Informationen finden Sie unter .
- Für Ihre DB-Instance muss Auto Minor Version Upgrade (Automatisches Unterversionsupgrade) aktiviert sein. Mit dieser Option erhält Ihre DB-Instance automatisch kleinere Upgrades der DB-Engine-Version, wenn diese verfügbar sind. Amazon RDS verwendet diese Option, um Ihre DB-Instance mit dem neuesten Oracle-Patch-Set-Update (PSU) oder Release-Update (RU) zu aktualisieren. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Bewährte Methoden für Oracle JVM

Für die Verwendung von Oracle Java gelten folgende bewährte Methoden:

- Für maximale Sicherheit sollten Sie die `JVM`-Option mit Secure Sockets Layer (SSL) verwenden. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.

- Konfigurieren Sie Ihre DB-Instance, um den Netzwerkzugriff einzuschränken. Weitere Informationen erhalten Sie unter Szenarien für den Zugriff auf einen in einem VPC und Arbeiten mit einer DB-Instance in einer VPC.
- Aktualisieren Sie die Konfiguration Ihrer HTTPS-Endpoints zur Unterstützung von TLSv1 .2, wenn Sie die folgenden Bedingungen erfüllen:
  - Sie verwenden Oracle Java Virtual Machine (JVM), um einen HTTPS-Endpunkt über TLSv1 1 TLSv1 oder .1 Protokolle zu verbinden.
  - Ihr Endpunkt unterstützt das TLSv1 2.2-Protokoll nicht.
  - Sie haben das Release-Update vom April 2021 nicht auf Ihre Oracle DB angewendet.

  Durch das Aktualisieren Ihrer Endpunktkonfiguration stellen Sie sicher, dass die Konnektivität der JVM mit dem HTTPS-Endpunkt weiterhin funktioniert. Weitere Informationen zu TLS-Änderungen in Oracle JRE und JDK finden Sie unter Oracle JRE und JDK Cryptographic Roadmap.

Hinzufügen der Oracle JVM-Option

Es folgt der allgemeine Vorgang für das Hinzufügen der `JVM`-Option zu einer DB-Instance:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Während die `JVM`-Option hinzugefügt wird, kommt es zu einem kurzen Ausfall. Nachdem Sie die Funktion hinzugefügt haben, müssen Sie Ihre DB-Instance neu starten. Sobald die Optionsgruppe aktiv ist, ist auch Oracle Java verfügbar.

> **Note**
>
> Während dieses Ausfalls werden die Funktionen zur Passwortverifizierung kurzzeitig deaktiviert. Sie können auch erwarten, dass während des Ausfalls Ereignisse im Zusammenhang mit der Passwortverifizierung auftreten. Die Passwortverifikationsfunktionen werden wieder aktiviert, bevor die Oracle DB-Instance verfügbar ist.

So fügen Sie die JVM-Option zu einer DB-Instance hinzu

1. Bestimmen Sie die Optionsgruppe, die Sie verwenden möchten. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   - Wählen Sie für Engine die von der DB-Instance verwendete DB-Engine (oracle-ee, oracle-se, oracle-se1 oder oracle-se2) aus.
   - Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie der Optionsgruppe die Option JVM hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Weisen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
4. Erteilen Sie die erforderlichen Berechtigungen für die Benutzer

   Der Amazon RDS-Hauptbenutzer verfügt standardmäßig über die Berechtigungen zur Verwendung der Option `JVM`. Falls andere Benutzer diese Berechtigungen benötigen, stellen Sie als Hauptbenutzer in einem SQL-Client eine Verbindung zu der DB-Instance her und erteilen Sie den Benutzern die Berechtigungen.

   Im folgenden Beispiel werden die Berechtigungen zur Verwendung der Option `JVM` dem Benutzer `test_proc` erteilt.

```
create user test_proc identified by password;
CALL dbms_java.grant_permission('TEST_PROC',
 'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', '');
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Nachdem dem Benutzer die Berechtigungen erteilt wurden, müsste die folgende Abfrage eine Ausgabe zurückgeben.

```
select * from dba_java_policy where grantee='TEST_PROC';
```

> **Note**
>
> Beim Oracle-Benutzernamen wird zwischen Groß- und Kleinschreibung unterschieden. Gewöhnlich besteht er nur aus Großbuchstaben.

Entfernen der Oracle JVM-Option

Sie können die `JVM`-Option aus einer DB-Instance entfernen. Während die Option entfernt wird, kommt es zu einem kurzen Ausfall. Nachdem Sie die `JVM`-Option entfernt haben, müssen Sie Ihre DB-Instance nicht neu starten.

> **Warning**
>
> Das Entfernen der `JVM`-Option kann zu Datenverlust führen, wenn die DB-Instance Datentypen verwendet, die als Teil der Option aktiviert waren. Sichern Sie Ihre Daten, bevor Sie fortfahren. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

Führen Sie eine der folgenden Maßnahmen durch, um die `JVM`-Option aus einer DB-Instance zu entfernen:

- Entfernen Sie die `JVM`-Option aus der zugehörigen Optionsgruppe. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.

- Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, in der die `JVM`-Option nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle Enterprise Manager

Amazon RDS unterstützt Oracle Enterprise Manager (OEM). OEMist die Oracle-Produktlinie für integriertes Management von Unternehmens-Informationstechnologie.

Amazon RDS unterstützt OEM auf Oracle Database 19c Non- CDBs oderCDBs. In der folgenden Tabelle werden die unterstützten OEM Optionen beschrieben.

Option	Options-ID	Unterstützte OEM Versionen
OEMDatenbank Express	`OEM`	OEMDatabase Express 19c
OEM Management Agent	`OEM_AGENT`	OEMCloud Control für 13c

> **Note**
>
> Sie können OEM Database oder OEM Management Agent verwenden, aber nicht beide.

Oracle Enterprise Manager Database Express

Amazon RDS unterstützt Oracle Enterprise Manager Database Express (EM Express) durch die Verwendung der OEM Option. Amazon RDS unterstützt EM Express für Oracle Database 19c unter Verwendung der CDB oder CDB Nicht-Architektur.

EM Express ist ein webbasiertes Datenbankverwaltungstool, das in Ihrer Datenbank enthalten ist und nur verfügbar ist, wenn es geöffnet ist. Es unterstützt wichtige Leistungsmanagement- und grundlegende Datenbankverwaltungsfunktionen. Weitere Informationen finden Sie unter Einführung in Oracle Enterprise Manager Database Express in der Oracle Database-Dokumentation.

> **Note**
>
> EM Express wird in der DB-Instance-Klasse db.t3.small nicht unterstützt. Weitere Informationen zu DB-Instance-Klassen finden Sie unter RDS für Oracle-DB-Instance-Klassen.

OEMOptionseinstellungen

Amazon RDS unterstützt die folgenden Einstellungen für die OEM Option.

Optionseinstellung	Zulässige Werte	Beschreibung
Port	Ein Ganzzahlwert	Der Port auf der RDS Oracle-DB-Instance, die auf EM Express wartet. Die Standarde instellung ist 5500.
Sicherheitsgruppen	—	Eine Sicherheitsgruppe, die Zugriff auf den Port hat.

Schritt 1: Hinzufügen der Option OEM

Das allgemeine Verfahren zum Hinzufügen der OEM Option zu einer DB-Instance ist wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie die Option zur Optionsgruppe hinzu.

3. Ordnen Sie die Optionsgruppe Ihrer DB-Instance zu.

Wenn Sie die OEM Option hinzufügen, kommt es zu einem kurzen Ausfall, während Ihre DB-Instance automatisch neu gestartet wird.

Um die OEM Option zu einer DB-Instance hinzuzufügen

1. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie für Engine die Oracle Edition für Ihre DB-Instance aus.

   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.

2. Fügen Sie die OEM Option der Optionsgruppe hinzu und konfigurieren Sie die Optionseinstellungen. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter OEMOptionseinstellungen.

   > **Note**
   >
   > Wenn Sie die OEM Option zu einer vorhandenen Optionsgruppe hinzufügen, die bereits an eine oder mehrere DB-Instances angehängt ist, kommt es zu einem kurzen Ausfall, während alle DB-Instances automatisch neu gestartet werden.

3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Weisen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Wenn Sie die OEM Option hinzufügen, kommt es zu einem kurzen Ausfall, während Ihre DB-Instance automatisch neu gestartet wird. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

> **Note**
>
> Sie können auch die verwenden AWS CLI um die OEM Option hinzuzufügen. Beispiele finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

Schritt 2: (CDBnur) Entsperren des DBSNMP Benutzerkontos

Wenn Ihre DB-Instance die CDB Architektur verwendet, müssen Sie sich bei EM Express als`DBSNMP`. In aCDB, `DBSNMP` ist ein gewöhnlicher Benutzer anmelden. Dieses Konto ist standardmäßig gesperrt. Wenn Ihre DB-Instance die CDB Architektur nicht verwendet, überspringen Sie diesen Schritt.

Um das DBSNMP Benutzerkonto in einer CDB Instance zu entsperren

1. Melden Sie sich SQL in*Plus oder einer anderen SQL Oracle-Anwendung als Masterbenutzer bei Ihrer DB-Instance an.
2. Führen Sie die folgende gespeicherte Prozedur aus, um das `DBSNMP` Konto zu entsperren:

   ```
   EXEC rdsadmin.rdsadmin_util.reset_oem_agent_password('new_password');
   ```

   Wenn Sie eine Fehlermeldung erhalten, die besagt, dass die Prozedur nicht existiert, starten Sie Ihre CDB Instance neu, um sie automatisch zu installieren. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Schritt 3: Zugriff auf EM Express über Ihren Browser

Wenn Sie über Ihren Webbrowser auf EM Express zugreifen, wird ein Anmeldefenster angezeigt, in dem Sie nach einem Benutzernamen und einem Passwort gefragt werden.

So greifen Sie über Ihren Browser auf EM Express zu

1. Identifizieren Sie den Endpunkt und den EM Express-Port für Ihre Amazon RDS DB-Instance. Informationen zum Ermitteln des Endpunkts für Ihre Amazon RDS DB-Instance finden Sie unterErmitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle.
2. Geben Sie URL in der Locator-Leiste Ihres Browsers a im folgenden Format ein.

   ```
   https://endpoint.rds.amazonaws.com:port/em
   ```

Wenn der Endpunkt für Ihre Amazon RDS DB-Instance beispielsweise und Ihr EM Express-Port ist1158, verwenden Sie Folgendes, URL um auf EM Express zuzugreifen. `mydb.a1bcde234fgh.us-east-1.rds.amazonaws.com`

```
https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em
```

3. Wenn Sie zur Eingabe Ihrer Anmeldedaten aufgefordert werden, führen Sie je nach Ihrer Datenbankarchitektur eine der folgenden Aktionen aus:

   Ihre Datenbank ist nicht-CDB.

   Geben Sie den Hauptbenutzernamen und das Hauptpasswort für die DB-Instance ein.

   Ihre Datenbank ist eineCDB.

   Geben Sie `DBSNMP` den Benutzer und das `DBSNMP` Passwort ein. Lassen Sie das `Container` Feld leer.

OEMDatenbankeinstellungen ändern

Nachdem Sie die OEM Datenbank aktiviert haben, können Sie die Sicherheitsgruppeneinstellung für diese Option ändern.

Sie können die OEM Portnummer nicht ändern, nachdem Sie die Optionsgruppe einer DB-Instance zugeordnet haben. Gehen Sie wie folgt vor, um die OEM Portnummer für eine DB-Instance zu ändern:

1. Erstellen Sie eine neue Optionsgruppe.
2. Fügen Sie die OEM Option mit der neuen Portnummer der neuen Optionsgruppe hinzu.
3. Entfernen Sie die bestehende Optionsgruppe aus der DB-Instance.
4. Fügen Sie die neue Optionsgruppe zur DB-Instance hinzu.

Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter OEMOptionseinstellungen.

OEMDatabase Express-Aufgaben ausführen

Sie können RDS Amazon-Verfahren verwenden, um bestimmte OEM Database Express-Aufgaben auszuführen. Durch die Ausführung dieser Verfahren können Sie die nachfolgend aufgeführten Aufgaben ausführen.

> **Note**
>
> OEMDatabase Express-Aufgaben werden asynchron ausgeführt.

Aufgaben

- Umstellung des Website-Frontends für OEM Database Express auf Adobe Flash
- Umstellung des Website-Frontends für OEM Database Express auf Oracle JET

Umstellung des Website-Frontends für OEM Database Express auf Adobe Flash

> **Note**
>
> Diese Aufgabe ist nur für Oracle Database 19c non- CDBs verfügbar.

Ab Oracle Database 19c hat Oracle die frühere OEM Database Express-Benutzeroberfläche, die auf Adobe Flash basierte, als veraltet eingestuft. Stattdessen verwendet OEM Database Express jetzt eine mit Oracle erstellte Oberfläche. JET Wenn Sie Schwierigkeiten mit der neuen Oberfläche haben, können Sie zurück zur veralteten Flash-basierten Oberfläche wechseln. Zu den Schwierigkeiten, auf die Sie mit der neuen Oberfläche stoßen könnten, gehört, dass Sie nach der Anmeldung bei OEM Database Express auf einem `Loading` Bildschirm hängen bleiben. Möglicherweise vermissen Sie auch bestimmte Funktionen, die in der Flash-basierten Version von OEM Database Express vorhanden waren.

Um das Frontend der OEM Database Express-Website auf Adobe Flash umzustellen, führen Sie das RDS Amazon-Verfahren aus`rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash`. Dieses Verfahren entspricht dem `execemx emx` SQL Befehl.

Bewährte Methoden für die Sicherheit raten von der Verwendung von Adobe Flash ab. Sie können zwar zur Flash-basierten Version von OEM Database Express zurückkehren, wir empfehlen jedoch,

wenn möglich, die JET auf OEM Database Express basierenden Websites zu verwenden. Wenn Sie wieder Adobe Flash verwenden und wieder zu Oracle wechseln möchtenJET, gehen Sie wie folgt vor. `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet` Nach einem Oracle-Datenbank-Upgrade JET kann eine neuere Version von Oracle die damit JET verbundenen Probleme in OEM Database Express beheben. Weitere Hinweise zum Umstieg auf Oracle JET finden Sie unterUmstellung des Website-Frontends für OEM Database Express auf Oracle JET.

> **Note**
>
> Wenn Sie diese Aufgabe von der Quell-DB-Instance für eine Read Replica aus ausführen, wechselt die Read Replica auch von ihren OEM Database Express-Website-Frontends zu Adobe Flash.

Der folgende Prozeduraufruf erstellt eine Aufgabe, um die OEM Database Express-Website auf Adobe Flash umzustellen, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash() as TASK_ID from DUAL;
```

Sie können das Ergebnis anzeigen, indem Sie die Ausgabedatei der Aufgabe anzeigen.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ersetzen *`Aufgaben-ID`* mit der Aufgaben-ID, die von der Prozedur zurückgegeben wurde. Weitere Informationen zum RDS Amazon-Verfahren `rdsadmin.rds_file_util.read_text_file` finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis

Sie können den Inhalt der Ausgabedatei der Aufgabe auch in der AWS Management Console indem Sie in den Protokolleinträgen im Abschnitt Protokolle und Ereignisse nach dem suchen`task-id`.

Umstellung des Website-Frontends für OEM Database Express auf Oracle JET

> **Note**
>
> Diese Aufgabe ist nur für Oracle Database 19c non- CDBs verfügbar.

Um das Frontend der OEM Database Express-Website auf Oracle umzustellenJET, führen Sie das RDS Amazon-Verfahren ausrdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet. Dieses Verfahren entspricht dem execemx omx SQL Befehl.

Standardmäßig verwenden die OEM Database Express-Websites für Oracle-DB-Instances, auf denen 19c oder höher ausgeführt wird, OracleJET. Wenn Sie das rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash Verfahren verwendet haben, um das Frontend der OEM Database Express-Website auf Adobe Flash umzustellen, können Sie wieder zu Oracle JET wechseln. Verwenden Sie dazu das Verfahren rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet. Weitere Hinweise zum Wechseln zu Adobe Flash finden Sie unter Umstellung des Website-Frontends für OEM Database Express auf Adobe Flash.

> **Note**
>
> Wenn Sie diese Aufgabe von der Quell-DB-Instance für eine Read Replica aus ausführen, wechselt die Read Replica auch von ihren OEM Database Express-Website-Frontends zu Oracle. JET

Der folgende Prozeduraufruf erstellt eine Aufgabe zur Umstellung der OEM Database Express-Website auf Oracle JET und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet() as TASK_ID from DUAL;
```

Sie können das Ergebnis anzeigen, indem Sie die Ausgabedatei der Aufgabe anzeigen.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Ersetzen *Aufgaben-ID* mit der Aufgaben-ID, die von der Prozedur zurückgegeben wurde. Weitere Informationen zum RDS Amazon-Verfahren rdsadmin.rds_file_util.read_text_file finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis

Sie können den Inhalt der Ausgabedatei der Aufgabe auch in der AWS Management Console indem Sie in den Protokolleinträgen im Abschnitt Protokolle und Ereignisse nach dem suchentask-id.

Die OEM Datenbankoption wird entfernt

Sie können die OEM Option aus einer DB-Instance entfernen. Wenn Sie die OEM Option entfernen, kommt es zu einem kurzen Ausfall, während Ihre Instance automatisch neu gestartet wird. Daher müssen Sie Ihre DB-Instance nicht neu starten, nachdem Sie die OEM Option entfernt haben.

Gehen Sie wie folgt vor, um die OEM Option aus einer DB-Instance zu entfernen:

- Entfernen Sie die OEM Option aus der Optionsgruppe, zu der sie gehört. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, die die OEM Option nicht enthält. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle Management Agent für Enterprise Cloud Control

Oracle Enterprise Manager (OEM) Management Agent ist eine Softwarekomponente, die Ziele in Hosts überwacht und diese Informationen an den Oracle Management Service (OMS) mittlerer Stufe übermittelt. Amazon RDS unterstützt Management Agent durch die Verwendung der `OEM_AGENT`-Option.

Weitere Informationen finden Sie unter Overview of Oracle Enterprise Manager Cloud Control 12c und Overview of Oracle Enterprise Manager Cloud Control 13c in der Oracle-Dokumentation.

Themen

- Anforderungen für den Management Agent
- Voraussetzungen für die OMS-Host-Kommunikation
- Einschränkungen für Management Agent
- Optionseinstellungen für den Management-Agenten
- Aktivierung der Management Agent-Option für Ihre DB-Instance
- Entfernen der Management-Agent-Option
- Ausführen von Datenbankaufgaben mit dem Management Agent

Anforderungen für den Management Agent

Im Folgenden sind die allgemeinen Anforderungen für die Verwendung des Management Agents aufgeführt:

- Auf Ihrer DB-Instance muss Oracle Database 19c (19.0.0.0) ausgeführt werden. Sie können entweder die CDB- oder die Nicht-CDB-Architektur verwenden.
- Sie müssen einen Oracle Management Service (OMS) verwenden, der so konfiguriert ist, dass er eine Verbindung zu Ihrer DB-Instance herstellt. Beachten Sie die folgenden OMS-Anforderungen:
  - Die Management Agent-Versionen 13.5.0.0.v2 und 13.5.0.0.v3 erfordern die OMS-Version 13.5.0.23 oder 24.1.
  - Für die Management Agent-Version 13.5.0.0.v1 ist die OMS-Version 13.5.0.0 oder 24.1 erforderlich.
  - Die Management Agent-Versionen 13.4.0.9.v1 und 13.4.0.9.v2 erfordern OMS-Version 13.4.0.9 oder höher und Patch 32198287.
- In den meisten Fällen muss die VPC konfiguriert werden, damit Verbindungen von OMS zur DB-Instance zulässig sind. Falls Sie mit Amazon Virtual Private Cloud (Amazon VPC) nicht vertraut

sind, sollten Sie die Schritte unter Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur) ausführen, bevor Sie fortfahren.

- Sie können Management Agent mit Oracle Enterprise Manager Cloud Control für 12c und 13c verwenden. Stellen Sie sicher, dass Sie über ausreichend Speicherplatz für Ihr OEM-Version verfügen:
  - Mindestens 8,5 GiB für OEM 13c Version 5
  - Mindestens 8,5 GiB für OEM 13c Version 4
  - Mindestens 8,5 GiB für OEM 13c Version 3
  - Mindestens 5,5 GiB für OEM 13c Version 2
  - Mindestens 4,5 GiB für OEM 13c Release 1
  - Mindestens 2,5 GiB für OEM 12c
- Wenn Sie Management Agent-Versionen verwenden `OEM_AGENT 13.2.0.0.v3` und TCPS-Konnektivität verwenden möchten`13.3.0.0.v2`, folgen Sie den Anweisungen unter Konfiguration von CA-Zertifikaten von Drittanbietern für die Kommunikation mit Zieldatenbanken in der Oracle-Dokumentation. Aktualisieren Sie auch das JDK auf Ihrem OMS anhand der Anleitung im Oracle-Dokument mit der Oracle-Dokument-ID 2241358.1. Dieser Schritt stellt sicher, dass Ihr OMS alle Cipher Suites unterstützt, die von der Datenbank unterstützt werden.

> **Note**
>
> Die TCPS-Konnektivität zwischen dem Management-Agenten und der DB-Instance wird nur für die Management-Agent-Versionen `OEM_AGENT 13.2.0.0.v3`, `13.3.0.0.v2`, `13.4.0.9.v1` und höher unterstützt.

Voraussetzungen für die OMS-Host-Kommunikation

Stellen Sie sicher, dass Ihr OMS-Host und Ihre Amazon RDS-DB-Instance kommunizieren können. Gehen Sie wie folgt vor:

- Um eine Verbindung vom Management Agent zu Ihrem OMS-Host herzustellen, wenn sich Ihr OMS-Host hinter einer Firewall befindet, fügen Sie der Firewall die IP-Adressen Ihrer DB-Instances hinzu. Stellen Sie sicher, dass die Firewall für das OMS den folgenden Netzwerkverkehr zulässt:

Vom OMS-Host zu Ihrer DB-Instance

Konfigurieren Sie eine unidirektionale Firewallregel, die den Datenverkehr vom OMS-Host zum Datenbank-Listener-Port (Standard 1521) und zum OEM-Agent-Port (Standard 3872) zulässt.

Von Ihrer DB-Instance zum OMS-Host

Konfigurieren Sie eine unidirektionale Firewallregel, die den Datenverkehr von der DB-Instance zum OMS-HTTP-Port zulässt (Standard 4903).

- Wenn OMS einen öffentlich auflösbaren Hostnamen besitzt, fügen Sie die OMS-Adresse zu einer Sicherheitsgruppe hinzu, um die Verbindung von OMS zu Management Agent herzustellen. Ihre Sicherheitsgruppe muss über Eingangsregeln verfügen, die den Zugriff auf den DB-Listener-Port und den Management Agent-Port erlauben. Ein Beispiel zum Erstellen einer Sicherheitsgruppe und zum Hinzufügen von Regeln für eingehenden Datenverkehr finden Sie unter Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur).
- Wenn OMS keinen öffentlich auflösbaren Hostnamen besitzt, stellen Sie die Verbindung von OMS zu Management Agent mit einer der folgenden Methoden her:
  - Wenn Ihr OMS auf einer Amazon Elastic Compute Cloud (Amazon EC2) -Instance in einer privaten VPC gehostet wird, können Sie VPC-Peering einrichten, um eine Verbindung von OMS zum Management Agent herzustellen. Weitere Informationen finden Sie unter Ein in einer, auf die von einer EC2 Instance in einer anderen VPC zugegriffen wird VPC.
  - Falls OMS lokal gehostet wird, können Sie eine VPN-Verbindung einrichten, über die OMS auf Management Agent zugreifen kann. Weitere Informationen finden Sie unter Ein in einer Anwendung, auf die ein Client über das Internet VPC zugreift oder VPN-Verbindungen.
- Um den OEM Management Agent Version 13.5.0.0 (v1-v3) mit einem 24.1 OMS-Host zu verbinden, stellen Sie in Ihren Konfigurationsoptionen die Verwendung des TLS `MINIMUM_TLS_VERSION` 1.2-Protokolls ein. `TLSv1.2`

Einschränkungen für Management Agent

Nachfolgend finden Sie einige Einschränkungen bei der Verwendung von Management Agent:

- Sie können keine benutzerdefinierten Oracle Management Agent-Images bereitstellen.
- Administrative Aufgaben, wie die Ausführung eines Jobs und Datenbank-Patching, die Anmeldeinformationen des Hosts benötigen, werden nicht unterstützt.
- Host-Metriken und die Liste mit den Prozessen gewährleisten nicht die akkurate Wiedergabe des aktuellen Systemzustands. Daher sollten Sie OEM nicht verwenden, um das Root-Dateisystem

oder Mount-Point-Dateisystem zu überwachen. Weitere Informationen zum Überwachen des Betriebssystems finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

- Autodiscovery wird nicht unterstützt. Sie müssen Datenbankziele manuell hinzufügen.
- Die Verfügbarkeit von OMS-Modulen ist von der Datenbank-Edition abhängig. Das Diagnose- und Optimierungsmodul für die Datenbankleistung ist beispielsweise nur für Oracle Database Enterprise Edition verfügbar.
- Management Agent benötigt zusätzlichen Speicher und Rechenressourcen. Sollten nach der Aktivierung der `OEM_AGENT`-Option Leistungsprobleme auftreten, wird empfohlen, auf eine größere DB-Instance-Klasse zu skalieren. Weitere Informationen erhalten Sie unter und Ändern einer Amazon RDS DB-Instance.
- Der Benutzer, der `OEM_AGENT` auf Host Amazon RDS ausführt, hat keinen Betriebssystemzugriff auf das Warnungsprotokoll. Daher können Sie keine Metriken für `DB Alert Log` und `DB Alert Log Error Status` im OEM sammeln.

Optionseinstellungen für den Management-Agenten

Amazon RDS unterstützt die folgenden Einstellungen für die Management Agent-Option.

Optionseinstellung	Erforderlich	Zulässige Werte	Beschreibung

Version (`AGENT_VERSION`)	Ja	`13.5.0.0.v3`
		`13.5.0.0.v2`
		`13.5.0.0.v1`
		`13.4.0.9.v2`
		`13.4.0.9.v1`
			Die Version der Management Agent-Software. Die unterstützte Mindestversion ist`13.1.0.0.v1` .
			Der AWS CLI Optionsname ist`OptionVersion` .
			Note
			In den AWS GovCloud (US) Regionen sind 13.1-Versionen nicht verfügbar.

Optionseinstellung	Erforderlich	Zulässige Werte	Beschreibung

		`13.3.0.0.v2`
		`13.3.0.0.v1`
		`13.2.0.0.v3`
		`13.2.0.0.v2`
		`13.2.0.0.v1`
		`13.1.0.0.v1`	

Port (`AGENT_PORT`)	Ja	Ein Ganzzahlwert	Der Port auf der DB-Instance, der den OMS-Host überwacht. Der Standardwert ist 3872. Ihr OMS-Host muss einer Sicherheitsgruppe angehören, die Zugriff auf diesen Port hat. Der AWS CLI Optionsname lautet`Port`.
Sicherheitsgruppen	Ja	Bestehende Sicherheitsgruppen	Eine Sicherheitsgruppe, die Zugriff auf den Port hat. Ihr OMS-Host muss dieser Sicherheitsgruppe angehören. Der AWS CLI Optionsname ist `VpcSecurityGroupMemberships` oder`DBSecurityGroupMemberships`.

Optionseinstellung	Erforderlich	Zulässige Werte	Beschreibung
OMS_HOST	Ja	Ein Zeichenkettenwert, zum Beispiel *my.example.oms*	Der öffentlich zugängliche Hostname oder die IP-Adresse von OMS. Der AWS CLI Optionsname lautet`OMS_HOST`.
OMS_PORT	Ja	Ein Ganzzahlwert	Der HTTPS-Upload-Port auf dem OMS-Host, der den Management Agent überwacht. Um den HTTPS-Upload-Port zu ermitteln, verbinden Sie sich mit dem OMS-Host und führen den folgenden Befehl aus (für den das `SYSMAN`-Passwort erforderlich ist): `emctl status oms -details` Der AWS CLI Optionsname ist`OMS_PORT`.
AGENT_REGISTRATION_PASSWORD	Ja	Ein Zeichenfolgenwert	Das von Management Agent verwendete Passwort zur Authentifizierung bei OMS. Es wird empfohlen, vor der Aktivierung der `OEM_AGENT` -Option ein persistentes Passwort in OMS festzulegen. Mit einem persistenten Passwort können Sie eine einzelne Management Agent-Optionsgruppe für mehrere Amazon RDS-Datenbanken freigeben. Der AWS CLI Optionsname ist`AGENT_REGISTRATION_PASSWORD` .

Optionseinstellung	Erforderlich	Zulässige Werte	Beschreibung
ALLOW_TLS_ONLY	Nein	`true`, `false` (Standard)	Ein Wert, der den OEM Agent so konfiguriert, dass er nur das TLSv1-Protokoll unterstützt, während der Agent als Server überwacht. Diese Einstellung wird nicht mehr unterstützt. Die Management Agent-Versionen 13.1.0.0.v1 und höher unterstützen standardmäßig Transport Layer Security (TLS).
MINIMUM_TLS_VERSION	Nein	`TLSv1` (Standard), `TLSv1.2`	Ein Wert, der die minimale TLS-Version angibt, die vom OEM Agent unterstützt wird, während der Agent als Server überwacht. Nicht unterstützte Agentenversionen unterstützen nur die Einstellung. `TLSv1` Um 13.5.0.0 (v1-v3) mit einem 24.1 OMS-Host zu verbinden, setzen Sie diese Einstellung auf. `TLSv1.2`
TLS_CIPHER_SUITE	Nein	Siehe Optionseinstellungen für den Management-Agenten.	Ein Wert, der die TLS-Verschlüsselungssammlung angibt, die vom OEM Agent verwendet wird, während der Agent als Server überwacht.

In der folgenden Tabelle sind die TLS-Verschlüsselungssuiten aufgeführt, die von der Management-Agent-Option unterstützt werden.

Verschlüsselungssuite	Unterstützte Agent-Version	FedRAMP-konform
TLS_RSA_WITH_AES_128_CBC_SHA	Alle	Nein

Verschlüsselungssuite	Unterstützte Agent-Version	FedRAMP-konform
TLS_RSA_WITH_AES_128_CBC_ SHA256	13.1.0.0.v1 und höher	Nein
TLS_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 und höher	Nein
TLS_RSA_WITH_AES_256_CBC_ SHA256	13.2.0.0.v3 und höher	Nein
TLS_ECDHE_RSA_WITH_AES_128_ CBC_SHA	13.2.0.0.v3 und höher	Ja
TLS_ECDHE_RSA_WITH_AES_256_ CBC_SHA	13.2.0.0.v3 und höher	Ja
TLS_ECDHE_RSA_WITH_AES_128_ CBC_ SHA256	13.2.0.0.v3 und höher	Ja
TLS_ECDHE_RSA_WITH_AES_256_ CBC_ SHA384	13.2.0.0.v3 und höher	Ja

Aktivierung der Management Agent-Option für Ihre DB-Instance

Gehen Sie wie folgt vor, um die Management Agent-Option zu aktivieren:

Themen

- Schritt 1: Hinzufügen der Management Agent-Option zu Ihrer DB-Instance
- Schritt 2: Entsperren des DBSNMP-Benutzerkontos
- Schritt 3: Hinzufügen Ihrer Ziele zur Management Agent-Konsole

Schritt 1: Hinzufügen der Management Agent-Option zu Ihrer DB-Instance

Gehen Sie wie folgt vor, um Ihrer DB-Instance die Option Management Agent hinzuzufügen:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.

3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Wenn Sie auf Fehler stoßen, durchsuchen Sie die My Oracle Support-Dokumente nach Informationen zur Lösung bestimmter Probleme.

Nachdem Sie die Management Agent-Option hinzugefügt haben, ist kein Neustart der DB-Instance erforderlich. Sobald die Optionsgruppe aktiv ist, ist auch der OEM Agent aktiv.

Wenn Ihr OMS-Host ein nicht vertrauenswürdiges Drittanbieterzertifikat verwendet, gibt Amazon RDS den folgenden Fehler zurück.

```
You successfully installed the OEM_AGENT option. Your OMS host is using an untrusted
 third party certificate.
Configure your OMS host with the trusted certificates from your third party.
```

Wenn dieser Fehler zurückgegeben wird, wird die Option „Management Agent (Verwaltungsagent)“ erst aktiviert, wenn das Problem behoben wurde. Informationen zum Beheben des Problems finden Sie im My Oracle-Supportdokument 2202569.1.

Konsole

Um die Management Agent-Option zu Ihrer DB-Instance hinzuzufügen

1. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie für Engine die Oracle Edition für Ihre DB-Instance aus.
   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option OEM_AGENT zur Optionsgruppe hinzu und konfigurieren Sie die Optionseinstellungen. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Optionseinstellungen für den Management-Agenten.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

- Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

Im folgenden Beispiel wird der Befehl AWS CLI add-option-to-option-group verwendet, um die `OEM_AGENT` Option zu einer Optionsgruppe namens `myoptiongroup` hinzuzufügen.

Für LinuxmacOS, oderUnix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] \
    --apply-immediately
```

Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] ^
    --apply-immediately
```

Schritt 2: Entsperren des DBSNMP-Benutzerkontos

Der Management Agent verwendet das `DBSNMP` Benutzerkonto, um eine Verbindung zur Datenbank herzustellen und Probleme an Oracle Enterprise Manager zu melden. In einer CDB `DBSNMP` ist dies ein üblicher Benutzer. Dieses Benutzerkonto ist sowohl für den Management Agent als auch für OEM Database Express erforderlich. Dieses Konto ist standardmäßig gesperrt. Das Verfahren zum Entsperren dieses Kontos unterscheidet sich je nachdem, ob Ihre Datenbank die Nicht-CDB- oder die CDB-Architektur verwendet.

So entsperren Sie das DBSNMP-Benutzerkonto

1. Melden Sie sich in SQL*Plus oder einer anderen Oracle SQL-Anwendung als Masterbenutzer bei Ihrer DB-Instance an.
2. Führen Sie je nach Datenbankarchitektur eine der folgenden Aktionen aus:

   Ihre Datenbank ist keine CDB.

   Führen Sie die folgende SQL-Anweisung aus:

   ```
   ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
   ```

   Ihre Datenbank ist eine CDB.

   Führen Sie die folgende gespeicherte Prozedur aus, um das DBSNMP Konto zu entsperren:

   ```
   EXEC rdsadmin.rdsadmin_util.reset_oem_agent_password('new_password');
   ```

   Wenn Sie eine Fehlermeldung erhalten, die besagt, dass die Prozedur nicht existiert, starten Sie Ihre CDB-Instance neu, um sie automatisch zu installieren. Weitere Informationen finden Sie unter Neustarten einer DB-Instance.

Schritt 3: Hinzufügen Ihrer Ziele zur Management Agent-Konsole

Um eine DB-Instance als Ziel hinzuzufügen, stellen Sie sicher, dass Sie den Endpunkt und den Port kennen. Informationen zum Ermitteln des Endpunkts der Amazon RDS-DB-Instance finden Sie unter Ermitteln des Endpunkts Ihrer DB-Instance von RDS für Oracle. Wenn Ihre Datenbank die CDB-Architektur verwendet, fügen Sie den CDB$ROOT Container separat als Ziel hinzu.

Um Ziele zur Management Agent-Konsole hinzuzufügen

1. Wählen Sie in Ihrer OMS-Konsole Setup (Einrichten), Add Target (Ziel hinzufügen) und anschließend Add Targets Manually (Ziele manuell hinzufügen) aus.
2. Wählen Sie Add Targets Declaratively by Specifying Target Monitoring Properties (Ziele deklarativ durch Angabe von Zielüberwachungseigenschaften hinzufügen) aus.
3. Wählen Sie für Zieltyp die Option Database Instance (DB-Instance) aus.
4. Wählen Sie für Monitoring-Agent den Agent mit der ID, die mit der Ihrer RDS DB-Instance identisch ist.

5. Wählen Sie Add Manually (Manuell hinzufügen) aus.
6. Geben Sie den Endpunkt für Ihre Amazon RDS-DB-Instance ein, oder wählen Sie ihn aus der Hostnamenliste aus. Stellen Sie sicher, dass der angegebene Hostname mit dem Endpunkt der Amazon RDS DB-Instance übereinstimmt.
7. Legen Sie die folgenden Datenbankeigenschaften fest:
   - Geben Sie für Target name (Zielname) einen Namen ein.
   - Geben Sie für Database system name (Datenbanksystemname) einen Namen ein.
   - Geben Sie für Monitor username (Überwachungs-Benutzername) **`dbsnmp`** ein.
   - Geben Sie für Monitor-Passwort das Passwort von einSchritt 2: Entsperren des DBSNMP-Benutzerkontos.
   - Geben Sie für Role (Rolle) normal ein.
   - Geben Sie für Oracle home path (Oracle-Startpfad) **`/oracle`** ein.
   - Für Listener Machine name (Listener-Maschinenname) wird die Agent-Kennung bereits angezeigt.
   - Geben Sie für Port den Datenbank-Port ein. Der RDS-Standard-Port ist 1521.
   - Geben Sie unter Database name (Datenbankname) den Namen Ihrer Datenbank ein. Wenn es sich bei Ihrer Datenbank um eine CDB handelt, lautet `RDSCDB` dieser Name.
8. Wählen Sie Test Connection (Verbindung testen) aus.
9. Wählen Sie Weiter aus. Die Zieldatenbank wird in der Liste überwachter Ressourcen angezeigt.

Entfernen der Management-Agent-Option

Sie können OEM Agent aus einer DB-Instance entfernen. Nachdem Sie OEM Agent entfernt haben, ist kein Neustart der DB-Instance erforderlich.

Führen Sie die folgenden Schritte aus, um OEM Agent aus einer DB-Instance zu entfernen:

- Entfernen Sie die OEM Agent-Option aus der zugehörigen Optionsgruppe. Diese Änderung wirkt sich auf alle DB-Instances aus, die die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, in der OEM Agent nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Ausführen von Datenbankaufgaben mit dem Management Agent

Sie können Amazon RDS-Prozeduren verwenden, um bestimmte EMCTL-Befehle auf dem Management-Agent auszuführen. Durch die Ausführung dieser Verfahren können Sie die nachfolgend aufgeführten Aufgaben ausführen.

> **Note**
>
> Die Aufgaben werden asynchron ausgeführt.

Aufgaben

- Den Management Agent sichern
- Abrufen des Status des Management Agents
- Neustart des Management-Agents
- Auflistung der vom Management Agent überwachten Ziele
- Auflistung der vom Management Agent überwachten Sammlungs-Threads
- Löschen des Status des Management Agents
- Der Management-Agent muss sein OMS hochladen.
- Ping des OMS
- Anzeigen des Status einer laufenden Aufgabe

Den Management Agent sichern

Um den Management Agent zu sichern, führen Sie das Amazon RDS-Verfahren aus`rdsadmin.rdsadmin_oem_agent_tasks.secure_oem_agent`. Diese Vorgehensweise entspricht der Ausführung des Befehls `emctl secure agent`.

Das folgende Verfahren erstellt eine Aufgabe zur Sicherung des Management Agents und gibt die ID der Aufgabe zurück:

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.secure_oem_agent as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Abrufen des Status des Management Agents

Führen Sie die Amazon RDS-Prozedur aus, um den Status des Management Agents abzurufe `rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent`. Diese Vorgehensweise entspricht dem `emctl status agent`-Befehl.

Die folgende Prozedur erstellt eine Aufgabe, um den Status des Verwaltungsagenten zu erhalten, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent() as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Neustart des Management-Agents

Um den Management-Agent neu zu starten, führen Sie die Amazon RDS-Prozedur au `rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent`. Diese Vorgehensweise entspricht der Ausführung der Befehle `emctl stop agent` und `emctl start agent`.

Die folgende Prozedur erstellt eine Aufgabe, um den Management-Agent neu zu starten, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Auflistung der vom Management Agent überwachten Ziele

Um die vom Management-Agent überwachten Ziele aufzulisten, führen Sie die Amazon RDS-Prozedur au `rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent`. Diese Vorgehensweise entspricht der Ausführung des Befehls `emctl config agent listtargets`.

Die folgende Prozedur erstellt eine Aufgabe, um die vom Management-Agent überwachten Ziele aufzulisten, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Auflistung der vom Management Agent überwachten Sammlungs-Threads

Führen Sie die Amazon RDS-Prozedur aus, um alle laufenden, fertigen und geplanten Sammlungs-Threads aufzulisten, die vom Management-Agent überwacht werde `rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent`. Diese Vorgehensweise entspricht dem `emctl status agent scheduler`-Befehl.

Die folgende Prozedur erstellt eine Aufgabe zum Auflisten der Sammlungs-Threads und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent() as TASK_ID from
 DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Löschen des Status des Management Agents

Um den Status des Management-Agents zu löschen, führen Sie die Amazon RDS-Prozedur au `rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent`. Diese Vorgehensweise entspricht der Ausführung des Befehls `emctl clearstate agent`.

Die folgende Prozedur erstellt eine Aufgabe, die den Status des Management-Agents löscht, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent() as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Der Management-Agent muss sein OMS hochladen.

Damit der Management-Agent den ihm zugeordneten Oracle Management Server (OMS) hochladen kann, führen Sie die Amazon RDS-Prozedur au `rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent`. Diese Vorgehensweise entspricht der Ausführung des Befehls `emclt upload agent`.

Die folgende Prozedur erstellt eine Aufgabe, bei der der Management-Agent seinen zugeordneten OMS hochlädt, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent() as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Ping des OMS

Um den OMS des Management-Agents anzurufen, führen Sie die Amazon RDS-Prozedur au `rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent`. Diese Vorgehensweise entspricht der Ausführung des Befehls `emctl pingOMS`.

Die folgende Prozedur erstellt eine Aufgabe, die den OMS des Management-Agents anpingt, und gibt die ID der Aufgabe zurück.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent() as TASK_ID from DUAL;
```

Informationen zum Anzeigen der Ausgabedatei für die Aufgabe und zum Ergebnis finden Sie unterAnzeigen des Status einer laufenden Aufgabe.

Anzeigen des Status einer laufenden Aufgabe

Sie können den Status einer laufenden Aufgabe in einer bdump-Datei einsehen. Die bdump-Dateien befinden sich im Verzeichnis `/rdsdbdata/log/trace`. Jeder bdump-Dateiname weist das folgende Format auf.

```
dbtask-task-id.log
```

Wenn Sie eine Aufgabe überwachen möchten, ersetzen Sie *`task-id`* durch die ID der Aufgabe, die Sie überwachen möchten.

Um den Inhalt von bdump-Dateien anzuzeigen, führen Sie die Amazon RDS-Prozedur au `rdsadmin.rds_file_util.read_text_file`. Die folgende Abfrage gibt den Inhalt der bdump-Datei `dbtask-1546988886389-2444.log` zurück.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

Weitere Informationen über die Amazon RDS-Prozedur `rdsadmin.rds_file_util.read_text_file` finden Sie unter Lesen von Dateien in einem DB-Instance-Verzeichnis.

Oracle Label Security

Amazon RDS unterstützt Oracle Label Security für die Enterprise Edition von Oracle Database mithilfe der OLS Option.

Die meisten Datenbanksicherheitskontrollen greifen auf die Ebene von Objekten zu. Oracle Label Security ermöglicht eine feinere Steuerung des Zugriffs auf einzelne Tabellenspalten. Sie können beispielsweise Label Security verwenden, um regulatorische Compliance mit einem richtlinienbasierten Verwaltungsmodell durchzusetzen. Sie können mit Label Security-Richtlinien den Zugriff auf sensible Daten steuern und den Zugriff auf Benutzer mit einer entsprechenden Berechtigungsstufe beschränken. Weitere Informationen finden Sie unter Introduction to Oracle Label Security in der Oracle-Dokumentation.

Themen

- Anforderungen für Oracle Label Security
- Überlegungen bei der Verwendung von Oracle Label Security
- Hinzufügen der Oracle Label Security-Option
- Fehlerbehebung

Anforderungen für Oracle Label Security

Machen Sie sich mit den folgenden Anforderungen für Oracle Label Security vertraut:

- Ihre DB-Instance muss das Modell "Bring Your Own License" verwenden. Weitere Informationen finden Sie unter RDSfür Oracle-Lizenzoptionen.
- Sie benötigen eine gültige Lizenz für Oracle Enterprise Edition mit Softwareupdate-Lizenz und Support.
- Ihre Oracle-Lizenz muss die Label Security-Option enthalten.

Überlegungen bei der Verwendung von Oracle Label Security

Oracle Label Security wird verwendet, indem Richtlinien für die Steuerung des Zugriffs auf einzelne Tabellenzeilen erstellt werden. Weitere Informationen finden Sie unter Creating an Oracle Label Security Policy in der Oracle-Dokumentation.

Berücksichtigen Sie dabei Folgendes:

- Oracle Label Security ist eine permanente und persistente Option. Da die Option permanent ist, können Sie sie nicht aus einer Optionsgruppe entfernen. Wenn Sie Oracle Label Security einer Optionsgruppe hinzufügen und sie Ihrer DB-Instance zuordnen, können Sie Ihrer DB-Instance später eine andere Optionsgruppe zuordnen, aber diese Gruppe muss ebenfalls die Oracle Label Security-Option enthalten.
- Wenn Sie mit Label Security arbeiten, führen Sie alle Aktionen als `LBAC_DBA` Rolle aus. Dem Masterbenutzer für Ihre DB-Instance wird die `LBAC_DBA` Rolle zugewiesen. Sie können die `LBAC_DBA` Rolle anderen Benutzern zuweisen, damit diese die Label Security-Richtlinien verwalten können.
- Stellen Sie sicher, dass Sie allen neuen Benutzern, die Zugriff auf Oracle Label Security benötigen, Zugriff auf das `OLS_ENFORCEMENT` Paket gewähren. Um Zugriff auf das `OLS_ENFORCEMENT` Paket zu gewähren, stellen Sie als Master-Benutzer eine Verbindung zur DB-Instance her und führen Sie die folgende SQL Anweisung aus:

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

- Sie können Label Security über Oracle Enterprise Manager (OEM) Cloud Control konfigurieren. Amazon RDS unterstützt OEM Cloud Control über die Management Agent-Option. Weitere Informationen finden Sie unter Oracle Management Agent für Enterprise Cloud Control.

Hinzufügen der Oracle Label Security-Option

Der allgemeine Vorgang für das Hinzufügen der Oracle Label Security-Option zu einer DB-Instance ist wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie die Option zur Optionsgruppe hinzu.

> **Important**
>
> Oracle Label Security ist eine permanente und persistente Option.

3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie die Label Security-Option hinzugefügt haben, ist Label Security aktiviert, sobald die Optionsgruppe aktiviert ist.

So fügen Sie einer DB-Instance die Label Security-Option hinzu:

1. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie für Engine die Option oracle-ee aus.

   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.

2. Fügen Sie die OLSOption der Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

   > **Important**
   >
   > Wenn Sie einer bestehenden Optionsgruppe, die bereits mit einer oder mehreren DB-Instances verknüpft ist, die Label Security-Option hinzufügen, werden alle DB-Instances neu gestartet.

3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Wenn Sie die Label Security-Option zu einer bestehenden DB-Instance hinzufügen, entsteht während des automatischen Neustarts Ihrer DB-Instance ein kurzzeitiger Nutzungsausfall. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Fehlerbehebung

Die folgenden Probleme können bei der Verwendung von Oracle Label Security auftreten.

Problem	Vorschläge für die Fehlerbehebung
Wenn Sie eine Richtlinie erstellen möchten, wird die Fehlermeldung ähnlich der folgenden angezeig: `insufficient authorization for the SYSDBA package`.	Es gibt bei dem Oracle Label Security-Feature ein bekanntes Problem, das verhindert, dass Benutzer, deren Benutzername genau 16 oder genau 24 Zeichen lang ist, die Label Security-Befehle ausführen können. Sie können einen neuen Benutzer mit einer anderen Anzahl von Zeichen erstellen, DBA dem neuen Benutzer LBAC _ gewähren, sich als neuer Benutzer anmelden und die OLS Befehle als neuer Benutzer ausführen. Für weitere Informationen wenden Sie sich an den Oracle Support.

Oracle Locator

Amazon RDS unterstützt Oracle Locator durch die Verwendung der Option `LOCATOR`. Oracle Locator stellt Funktionalitäten zur Verfügung, die üblicherweise für die Unterstützung von Anwendungen auf Internet- und Wireless-Servicebasis sowie von GIS-Lösungen auf Partnerbasis erforderlich sind. Oracle Locator ist ein begrenzter Teilbereich von Oracle Spatial. Weitere Informationen finden Sie unter Oracle Locator in der Oracle-Dokumentation.

> **Important**
>
> Bei der Verwendung von Oracle Locator aktualisiert Amazon RDS automatisch Ihre DB-Instance auf die neueste Oracle-PSU, falls Schwachstellen mit einem Common Vulnerability Scoring System (CVSS)-Schweregrad von mehr als 9 oder andere gemeldete Schwachstellen vorliegen.

Unterstützte Datenbankversionen für Oracle Locator

RDS for Oracle unterstützt Oracle Locator für Oracle Database 19c. Oracle Locator wird für Oracle Database 21c nicht unterstützt, aber seine Funktionalität ist in der Oracle-Spatial-Option verfügbar. Früher erforderte die Spatial-Option zusätzliche Lizenzen. Oracle Locator stellte eine Teilmenge der Funktionen von Oracle Spatial dar und verlangte keine zusätzlichen Lizenzen. Im Jahr 2019 gab Oracle bekannt, dass alle Funktionen von Oracle Spatial ohne zusätzliche Kosten in den Lizenzen Enterprise Edition und Standard Edition 2 enthalten sind. Folglich erforderte die Oracle-Spatial-Option keine zusätzliche Lizenzierung mehr. Weitere Informationen finden Sie unter Machine Learning, Spatial und Graph – keine Lizenz erforderlich! im Oracle Database-Insider-Blog.

Voraussetzungen für Oracle Locator

Es folgen die Voraussetzungen für den Einsatz von Oracle Locator:

- Ihre DB-Instance muss eine ausreichende Klasse besitzen. Oracle Locator wird für die DB-Instance-Klassen db.t3.small nicht unterstützt. Weitere Informationen finden Sie unter RDS für Oracle-DB-Instance-Klassen.
- Für Ihre DB-Instance muss Auto Minor Version Upgrade (Automatisches Unterversionsupgrade) aktiviert sein. Diese Option ermöglicht es Ihrer DB-Instance, DB-Engine-Unterversions-Upgrades automatisch zu erhalten, sobald diese verfügbar sind, und ist für alle Optionen erforderlich, die die Oracle Java Virtual Machine (JVM) installieren. Amazon RDS verwendet diese Option, um Ihre

DB-Instance mit dem neuesten Oracle-Patch-Set-Update (PSU) oder Release-Update (RU) zu aktualisieren. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Bewährte Methoden für Oracle Locator

Es folgen die bewährten Methoden für den Einsatz von Oracle Locator:

- Für maximale Sicherheit sollten Sie die LOCATOR-Option mit Secure Sockets Layer (SSL) verwenden. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.
- Konfigurieren Sie Ihre DB-Instance, um den Zugriff auf Ihre DB-Instance einzuschränken. Weitere Informationen erhalten Sie unter Szenarien für den Zugriff auf einen in einem VPC und Arbeiten mit einer DB-Instance in einer VPC.

Hinzufügen der Oracle Locator-Option

Es folgt der allgemeine Vorgang für das Hinzufügen der LOCATOR-Option zu einer DB-Instance:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Wenn Oracle Java Virtual Machine (JVM) nicht auf der DB-Instance installiert ist, kommt es zu einem kurzen Ausfall, während die Option LOCATOR hinzugefügt wird. Es gibt keinen Ausfall, wenn Oracle Java Virtual Machine (JVM) bereits auf der DB-Instance installiert ist. Nachdem Sie die Funktion hinzugefügt haben, müssen Sie Ihre DB-Instance neu starten. Sobald die Optionsgruppe aktiv ist, ist auch Oracle Locator verfügbar.

> **Note**
>
> Während dieses Ausfalls werden die Funktionen zur Passwortverifizierung kurzzeitig deaktiviert. Sie können auch erwarten, dass während des Ausfalls Ereignisse im Zusammenhang mit der Passwortverifizierung auftreten. Die Passwortverifikationsfunktionen werden wieder aktiviert, bevor die Oracle DB-Instance verfügbar ist.

So können Sie die **LOCATOR**-Option zu einer DB-Instance hinzufügen

1. Bestimmen Sie die Optionsgruppe, die Sie verwenden möchten. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie für Engine die Oracle Edition für die DB-Instance aus.
   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option LOCATOR zur Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Verwenden von Oracle Locator

Nachdem Sie die Option Oracle Locator aktiviert haben, können Sie mit der Nutzung beginnen. Sie sollten nur Oracle Locator-Funktionen verwenden. Verwenden Sie keine Oracle Spatial-Funktionen, es sei denn, Sie haben eine Lizenz für Oracle Spatial.

Eine Liste der Funktionen, die für Oracle Locator unterstützt werden, finden Sie unter In Locator vorhandene Funktionen in der Oracle-Dokumentation.

Eine Liste der Funktionen, die für Oracle Locator nicht unterstützt werden, finden Sie unter In Locator nicht verfügbare Funktionen in der Oracle-Dokumentation.

Entfernen der Oracle Locator-Option

Nachdem Sie alle Objekte gelöscht haben, die Datentypen verwenden, die von der LOCATOR-Option bereitgestellt werden, können Sie die Option aus einer DB-Instance entfernen. Wenn Oracle Java

Virtual Machine (JVM) nicht auf der DB-Instance installiert ist, kommt es zu einem kurzen Ausfall, während die Option LOCATOR entfernt wird. Es gibt keinen Ausfall, wenn Oracle Java Virtual Machine (JVM) bereits auf der DB-Instance installiert ist. Nachdem Sie die LOCATOR-Option entfernt haben, müssen Sie Ihre DB-Instance nicht neu starten.

So löschen Sie die **LOCATOR**-Option:

1. Sichern Sie Ihre Daten.

   > **Warning**
   >
   > Wenn die Instance Datentypen verwendet, die als Teil der Option aktiviert wurden, und wenn Sie die LOCATOR-Option entfernen, können Sie Daten verlieren. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

2. Überprüfen Sie, ob vorhandene Objekte auf Datentypen oder Funktionen der LOCATOR-Option verweisen.

   Wenn LOCATOR-Optionen vorhanden sind, kann die Instance hängen bleiben, wenn die neue Optionsgruppe angewendet wird, die nicht über die LOCATOR-Option verfügt. Sie können die Objekte mithilfe der folgenden Abfragen identifizieren:

   ```
   SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE '%TABLE%'
   AND    (OWNER, SEGMENT_NAME) IN
          (SELECT DISTINCT OWNER, TABLE_NAME
           FROM   DBA_TAB_COLUMNS
           WHERE  DATA_TYPE='SDO_GEOMETRY'
           AND    OWNER <> 'MDSYS')
   ORDER BY 1,2,3,4;

   SELECT OWNER, TABLE_NAME, COLUMN_NAME
   FROM   DBA_TAB_COLUMNS
   WHERE  DATA_TYPE = 'SDO_GEOMETRY'
   AND    OWNER <> 'MDSYS'
   ORDER BY 1,2,3;
   ```

3. Löschen Sie alle Objekte, die auf Datentypen oder Funktionen der LOCATOR-Option verweisen.
4. Führen Sie eine der folgenden Aufgaben aus:

- Entfernen Sie die LOCATOR-Option aus der zugehörigen Optionsgruppe. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, in der die LOCATOR-Option nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle Native Network Encryption

Amazon RDS unterstützt die native Oracle-Netzwerkverschlüsselung (NNE). Mit dieser `NATIVE_NETWORK_ENCRYPTION` Option können Sie Daten verschlüsseln, während sie zu und von einer DB-Instance übertragen werden. Amazon RDS unterstützt NNE alle Editionen von Oracle Database.

Eine detaillierte Beschreibung von Oracle Native Network Encryption geht über den Rahmen dieses Handbuchs hinaus, jedoch sollten Sie die Stärken und Schwächen jedes Algorithmus und Schlüssels kennen, bevor Sie sich für eine Bereitstellungslösung entscheiden. Weitere Informationen zu den Algorithmen und Schlüsseln, die über Oracle Native Network Encryption verfügbar sind, finden Sie unter Configuring Network Data Encryption in der Oracle-Dokumentation. Weitere Informationen zur AWS -Sicherheit finden Sie imAWS -Sicherheitszentrum.

> **Note**
>
> Sie können entweder Native Network Encryption oder Secure Sockets Layer verwenden, jedoch nicht beides zusammen. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.

Themen

- NATIVE_ NETWORK _ ENCRYPTION Optionseinstellungen
- Hinzufügen der ENCRYPTION Option NATIVE NETWORK _ _
- NNEWerte in der Datei sqlnet.ora setzen
- Ändern der Einstellungen für ENCRYPTION die Option NATIVE NETWORK _ _
- Die ENCRYPTION Option NATIVE _ NETWORK _ entfernen

NATIVE_ NETWORK _ ENCRYPTION Optionseinstellungen

Sie können Verschlüsselungsanforderungen sowohl auf dem Server als auch auf dem Client angeben. Die DB-Instanz kann als Client fungieren, wenn sie beispielsweise einen Datenbanklink verwendet, um eine Verbindung mit einer anderen Datenbank herzustellen. Möglicherweise möchten Sie vermeiden, dass die Verschlüsselung auf der Serverseite erzwungen wird. Beispielsweise möchten Sie möglicherweise nicht alle Clientkommunikationen dazu zwingen, die Verschlüsselung zu verwenden, da der Server dies erfordert. In diesem Fall können Sie die Verschlüsselung auf der Clientseite mit dem`SQLNET.*CLIENT`-Optionen.

Amazon RDS unterstützt die folgenden Einstellungen für die NATIVE_NETWORK_ENCRYPTION Option.

> **Note**
>
> Wenn Sie Werte für eine Optionseinstellung durch Kommas trennen, setzen Sie kein Leerzeichen nach dem Komma.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS	TRUE, FALSE	TRUE	Das Verhalten des Servers, wenn ein Client, der eine nicht sichere Chiffre verwendet, versucht, sich mit der Datenbank zu verbinden. WennTRUE, können Kunden eine Verbindung herstellen, auch wenn sie nicht mit dem Juli 2021 PSU gepatcht wurden. Wenn die Einstellung aktiviert istFALSE, können sich Clients nur dann mit der Datenbank verbinden, wenn sie mit dem Juli 2021 gepatcht wurden. PSU Bevor Sie SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS auf FALSE festlegen, stellen Sie sicher, dass die folgenden Bedingungen erfüllt sind: • SQLNET.ENCRYPTION_TYPES_SERVER und SQLNET.ENCRYPTION_TYPES_CLIENT haben eine übereinstimmende Verschlüsselungsmethode, die nicht DES, 3DES, oder RC4 (alle Schlüssellängen) ist.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
			• SQLNET.CHECKSUM_TYPES_SERVER und SQLNET.CHECKSUM_TYPES_CLIENT haben eine passende sichere Prüfsummierungs-Methode, die nicht MD5 ist. • Der Client ist mit dem Juli 2021 gepatcht. PSU Wenn der Client nicht gepatcht ist, verliert der Client die Verbindung und erhält den ORA-12269 -Fehler.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.ALLOW_WEAK_CRYPTO	TRUE, FALSE	TRUE	Das Verhalten des Servers, wenn ein Client, der eine nicht sichere Chiffre verwendet, versucht, sich mit der Datenbank zu verbinden. Die folgenden Chiffren gelten als nicht sicher: • DES-Verschlüsselungsmethode (alle Schlüssellängen) • 3DES-Verschlüsselungsmethode (alle Schlüssellängen) • RC4-Verschlüsselungsmethode (alle Schlüssellängen) • MD5-Prüfsummierungs-Methode Wenn die Einstellung TRUE lautet, können Clients eine Verbindung herstellen, wenn sie die vorhergehenden nicht sicheren Chiffren verwenden. Wenn die Einstellung FALSE lautet, verhindert die Datenbank, dass Clients eine Verbindung herstellen, wenn sie die vorhergehenden nicht sicheren Chiffren verwenden. Bevor Sie SQLNET.ALLOW_WEAK_CRYPTO auf FALSE festlegen, stellen Sie sicher, dass die folgenden Bedingungen erfüllt sind: • SQLNET.ENCRYPTION_TYPES_SERVER und SQLNET.ENCRYPTION_TYPES_CLIENT

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
			haben eine übereinstimmende Verschlüsselungsmethode, die nicht DES, 3DES, oder RC4 (alle Schlüssellängen) ist. • SQLNET.CHECKSUM_TYPES_SERVER und SQLNET.CHECKSUM_TYPES_CLIENT haben eine passende sichere Prüfsummierungs-Methode, die nicht MD5 ist. • Für den Client ist der Juli 2021 gepatcht. PSU Wenn der Client nicht gepatcht ist, verliert der Client die Verbindung und erhält den ORA-12269 -Fehler.
SQLNET.CRYPTO_CHECKSUM_CLIENT	Accepted Rejected Requested , Required	Requested	Das Datenintegritätsverhalten, wenn eine DB-Instance eine Verbindung zum Client oder zu einem als Client fungierenden Server herstellt. Wenn eine DB-Instance einen Datenbanklink verwendet, fungiert sie als Client. Requested bedeutet, dass der Client von der DB-Instanz keine Prüfsumme verlangt.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.CRYPTO_CHECKSUM_SERVER	Accepted, Rejected, Requested, Required	Requested	Das Datenintegritätsverhalten, wenn ein Client oder ein Server, der als Client agiert, sich mit der DB-Instance verbindet. Wenn eine DB-Instance einen Datenbanklink verwendet, fungiert sie als Client. Requested bedeutet, dass die DB-Instanz vom Client keine Prüfsumme verlangt.
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512	Eine Liste von Prüfsummenalgorithmen. Sie können entweder einen Wert oder eine durch Kommas getrennte Wertelist e angeben. Wenn Sie ein Komma verwenden, fügen Sie kein Leerzeich en nach dem Komma ein, andernfalls wird ein InvalidParameterValue - Fehler angezeigt. Dieser Parameter undSQLNET.CR YPTO_CHECKSUM_TYPES_SERVER muss eine gemeinsame Chiffre haben.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512, SHA1, MD5	Eine Liste von Prüfsummenalgorithmen. Sie können entweder einen Wert oder eine durch Kommas getrennte Wertelist e angeben. Wenn Sie ein Komma verwenden, fügen Sie kein Leerzeich en nach dem Komma ein, andernfalls wird ein InvalidParameterValue - Fehler angezeigt. Dieser Parameter und SQLNET.CR YPTO_CHECKSUM_TYPES_CLIENT müssen eine gemeinsame Chiffre haben.
SQLNET.ENCRYPTION_CLIENT	Accepted Rejected Requested , Required	Requested	Das Verschlüsselungsverhalten des Clients, wenn ein Client oder ein Server, der als Client fungiert, eine Verbindun g zur DB-Instanz herstellt. Wenn eine DB-Instance einen Datenbanklink verwendet, fungiert sie als Client. Requested bedeutet, dass der Client keine Verschlüsselung des Datenverk ehrs mit dem Server benötigt.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.ENCRYPTION_SERVER	Accepted Rejected Requested , Required	Requested	Das Verschlüsselungsverhalten des Servers, wenn ein Client oder ein Server, der als Client fungiert, eine Verbindung zur DB-Instanz herstellt . Wenn eine DB-Instance einen Datenbanklink verwendet, fungiert sie als Client. Requested gibt an, dass für die DB-Instance der Datenverkehr vom Client nicht verschlüsselt sein muss.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Eine Liste der vom Client verwendeten Verschlüsselungsalgorithmen. Der Client verwendet jeden Algorithmus der Reihe nach, um zu versuchen, die Servereingabe zu entschlüsseln, bis ein Algorithmus erfolgreich oder das Ende der Liste erreicht ist. Amazon RDS verwendet die folgende Standardliste von Oracle. RDSbeginnt mit der Liste RC4_256 und setzt die Liste der Reihe nach nach unten fort. Sie können die Reihenfolge ändern oder die Anzahl der Algorithmen verringern, die von DB-Instance verwendet werden sollen. 1. RC4_256: RSA RC4 (256-Bit-Schlüsselgröße) 2. AES256: AES (256-Bit-Schlüsselgröße) 3. AES192: AES (192-Bit-Schlüsselgröße) 4. 3DES168: 3-Tasten-Triple- DES (effektive Tastengröße von 112 Bit) 5. RC4_128: RSA RC4 (128-Bit-Schlüsselgröße) 6. AES128: AES (128-Bit-Schlüsselgröße) 7. 3DES112: Triple mit 2 Tasten DES (effektive Schlüsselgröße von 80 Bit) 8. RC4_56: RSA RC4 (56-Bit-Schlüsselgröße)

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
			9. `DES`: Standard DES (56-Bit-Schlüsselgröße) 10. `RC4_40`: RSA RC4 (40-Bit-Schlüsselgröße) 11. `DES40`: DES4 0 (40-Bit-Schlüsselgröße) Sie können entweder einen Wert oder eine durch Kommas getrennte Wertelist e angeben. Wenn Sie ein Komma verwenden, fügen Sie kein Leerzeich en nach dem Komma ein, andernfalls wird ein `InvalidParameterValue` - Fehler angezeigt. Dieser Parameter und`SQLNET.SQ LNET.ENCRYPTION_TY PES_SERVER` muss eine gemeinsame Chiffre haben.

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Eine Liste der von der DB-Instance verwendeten Verschlüsselungsalgorithmen. Die DB-Instance nutzt die einzelnen Algorithmen (der Reihe nach), um die vom Client stammenden Daten zu entschlüsseln, bis ein Algorithmus zum Erfolg führt oder das Ende der Liste erreicht ist. Amazon RDS verwendet die folgende Standardliste von Oracle. Sie können die Reihenfolge ändern oder die Algorithmen einschränken, die der Kunde akzeptieren wird. 1. RC4_256: RSA RC4 (256-Bit-Schlüsselgröße) 2. AES256: AES (256-Bit-Schlüsselgröße) 3. AES192: AES (192-Bit-Schlüsselgröße) 4. 3DES168: 3-Tasten-Triple- DES (effektive Tastengröße von 112 Bit) 5. RC4_128: RSA RC4 (128-Bit-Schlüsselgröße) 6. AES128: AES (128-Bit-Schlüsselgröße) 7. 3DES112: Triple mit 2 Tasten DES (effektive Schlüsselgröße von 80 Bit) 8. RC4_56: RSA RC4 (56-Bit-Schlüsselgröße)

Optionseinstellung	Zulässige Werte	Standardwerte	Beschreibung
			9. `DES`: Standard DES (56-Bit-Schlüsselgröße) 10. `RC4_40`: RSA RC4 (40-Bit-Schlüsselgröße) 11. `DES40`: DES4 0 (40-Bit-Schlüsselgröße) Sie können entweder einen Wert oder eine durch Kommas getrennte Wertelist e angeben. Wenn Sie ein Komma verwenden, fügen Sie kein Leerzeich en nach dem Komma ein, andernfalls wird ein `InvalidParameterValue` - Fehler angezeigt. Dieser Parameter und`SQLNET.SQ LNET.ENCRYPTION_TY PES_SERVER` muss eine gemeinsame Chiffre haben.

Hinzufügen der ENCRYPTION Option NATIVE NETWORK _ _

Das allgemeine Verfahren zum Hinzufügen der `NATIVE_NETWORK_ENCRYPTION` Option zu einer DB-Instance ist wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Wenn die Optionsgruppe aktiv ist, NNE ist sie aktiv.

Um die ENCRYPTION Option NATIVE _ NETWORK _ zu einer DB-Instance hinzuzufügen, verwenden Sie AWS Management Console

1. Wählen Sie im Feld Engine die Oracle-Edition aus, die Sie verwenden möchten. NNEwird in allen Editionen unterstützt.
2. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
3. Fügen Sie die ENCRYPTION Option NATIVENETWORK_ _ zur Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

   > **Note**
   >
   > Nachdem Sie die ENCRYPTION Option NATIVE_ NETWORK _ hinzugefügt haben, müssen Sie Ihre DB-Instances nicht neu starten. Sobald die Optionsgruppe aktiv ist, NNE ist sie aktiv.
4. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Nachdem Sie die ENCRYPTION Option NATIVE_ NETWORK _ hinzugefügt haben, müssen Sie Ihre DB-Instance nicht neu starten. Sobald die Optionsgruppe aktiv ist, NNE ist sie aktiv. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

NNEWerte in der Datei sqlnet.ora setzen

Mit der nativen Netzwerkverschlüsselung von Oracle können Sie die Netzwerkverschlüsselung sowohl auf der Server- als auch auf der Client-Seite einstellen. Der Client ist der Computer, mit dem eine Verbindung zur DB-Instance hergestellt wird. Sie können die folgenden Client-Einstellungen in slqnet.ora festlegen:

- `SQLNET.ALLOW_WEAK_CRYPTO`
- `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS`

- SQLNET.CRYPTO_CHECKSUM_CLIENT
- SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT
- SQLNET.ENCRYPTION_CLIENT
- SQLNET.ENCRYPTION_TYPES_CLIENT

Weitere Informationen finden Sie unter Configuring Network Data Encryption and Integrity for Oracle Servers and Clients in der Oracle-Dokumentation.

Manchmal lehnt die DB-Instance eine Verbindungsanfrage von einer Anwendung ab. Beispielsweise kann eine Ablehnung auftreten, wenn die Verschlüsselungsalgorithmen auf dem Client und auf dem Server nicht übereinstimmen. Um die Oracle-eigene Netzwerkverschlüsselung zu testen, fügen Sie die folgenden Zeilen in die Datei sqlnet.ora auf dem Client ein:

```
DIAG_ADR_ENABLED=off
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

Wenn eine Verbindung versucht wird, erzeugen die vorangehenden Zeilen eine Trace-Datei auf dem Client mit der Bezeichnung `/tmp/nettrace*`. Die Trace-Datei enthält Informationen zur Verbindung. Weitere Informationen zu verbindungsbezogenen Problemen bei der Verwendung von Oracle Native Network Encryption finden Sie unter About Negotiating Encryption and Integrity in der Oracle-Database-Dokumentation.

Ändern der Einstellungen für ENCRYPTION die Option NATIVE NETWORK _ _

Nachdem Sie die Option `NATIVE_NETWORK_ENCRYPTION` aktiviert haben, können Sie ihre Einstellungen ändern. Derzeit können Sie `NATIVE_NETWORK_ENCRYPTION` Optionseinstellungen nur mit dem AWS CLI oder ändern RDSAPI. Die Konsole können Sie nicht verwenden. Im folgenden Beispiel werden zwei Einstellungen in der Option geändert.

```
aws rds add-option-to-option-group \
    --option-group-name my-option-group \
    --options
 "OptionName=NATIVE_NETWORK_ENCRYPTION,OptionSettings=[{Name=SQLNET.CRYPTO_CHECKSUM_TYPES_SERVE
{Name=SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER,Value=SHA256}]" \
    --apply-immediately
```

Informationen zum Ändern von Optionseinstellungen mithilfe der finden Sie CLI unterAWS CLI. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter NATIVE_ NETWORK _ ENCRYPTION Optionseinstellungen.

Themen

- CRYPTO_ CHECKSUM _*-Werte ändern
- ALLOW_ WEAK _ CRYPTO *-Einstellungen ändern

CRYPTO_ CHECKSUM _*-Werte ändern

Wenn Sie die ENCRYPTION Optionseinstellungen NATIVENETWORK_ _ ändern, stellen Sie sicher, dass die folgenden Optionseinstellungen mindestens eine gemeinsame Chiffre haben:

- `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`

Das folgende Beispiel zeigt ein Szenario, in dem Sie `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER` ändern. Die Konfiguration ist gültig, da `CRYPTO_CHECKSUM_TYPES_CLIENT` und `CRYPTO_CHECKSUM_TYPES_SERVER` beide `SHA256` verwenden.

Optionseinstellung	Werte vor Änderung	Werte nach Änderung
`SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`	**`SHA256`**, `SHA384`, `SHA512`	Keine Änderung
`SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`	**`SHA256`**, `SHA384`, `SHA512`, `SHA1`, `MD5`	`SHA1,MD5,` **`SHA256`**

Nehmen Sie für ein anderes Beispiel an, dass Sie`SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`von der Standardeinstellung auf`SHA1,MD5`aus. Stellen Sie in diesem Fall sicher, dass Sie`SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`auf`SHA1`oder`MD5`aus. Diese Algorithmen sind nicht in den Standardwerten für`SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`aus.

ALLOW_ WEAK _ CRYPTO *-Einstellungen ändern

Um die `SQLNET.ALLOW_WEAK_CRYPTO*`-Optionen vom Standardwert auf `FALSE` festzulegen, stellen Sie sicher, dass die folgenden Bedingungen erfüllt sind:

- SQLNET.ENCRYPTION_TYPES_SERVER und SQLNET.ENCRYPTION_TYPES_CLIENT haben eine passende sichere Verschlüsselungsmethode. Eine Methode gilt als sicher, wenn sie nicht DES, 3DES, oder RC4 (alle Schlüssellängen) ist.
- SQLNET.CHECKSUM_TYPES_SERVER und SQLNET.CHECKSUM_TYPES_CLIENT haben eine passende sichere Prüfsummierungs-Methode. Eine Methode gilt als sicher, wenn sie nicht MD5 ist.
- Der Client ist mit dem Juli 2021 PSU gepatcht. Wenn der Client nicht gepatcht ist, verliert der Client die Verbindung und erhält den ORA-12269-Fehler.

Das folgende Beispiel zeigt NNE Beispieleinstellungen. Gehen Sie davon aus, dass Sie SQLNET.ENCRYPTION_TYPES_SERVER und SQLNET.ENCRYPTION_TYPES_CLIENT auf FALSE einrichten und dadurch unsichere Verbindungen blockieren möchten. Die Einstellungen der Prüfsummenoption erfüllen die Voraussetzungen, da beide SHA256 haben. Allerdings. benutzen SQLNET.ENCRYPTION_TYPES_CLIENT und SQLNET.ENCRYPTION_TYPES_SERVER die DES-, 3DES-, und RC4-Verschlüsselungsmethoden, die nicht sicher sind. Um die SQLNET.ALLOW_WEAK_CRYPTO*-Optionen auf FALSE festzulegen, setzen Sie daher zuerst SQLNET.ENCRYPTION_TYPES_SERVER und SQLNET.ENCRYPTION_TYPES_CLIENT auf eine sichere Verschlüsselungsmethode wie AES256.

Optionseinstellung	Werte
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA1,MD5,SHA256
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, 3DES168, DES40
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, 3DES168, DES40

Die ENCRYPTION Option NATIVE _ NETWORK _ entfernen

Sie können NNE aus einer DB-Instance entfernen.

Führen Sie eine der folgenden Maßnahmen durch, um die NATIVE_NETWORK_ENCRYPTION-Option aus einer DB-Instance zu entfernen:

- Um die Option aus mehreren DB-Instances zu entfernen, entfernen Sie die NATIVE_NETWORK_ENCRYPTION Option aus der Optionsgruppe, zu der sie gehören. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Nachdem Sie die NATIVE_NETWORK_ENCRYPTION Option entfernt haben, müssen Sie Ihre DB-Instances nicht neu starten. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Um die Option aus einer einzelnen DB-Instance zu entfernen, ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, die die NATIVE_NETWORK_ENCRYPTION Option nicht enthält. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Nachdem Sie die NATIVE_NETWORK_ENCRYPTION-Option entfernt haben, müssen Sie Ihre DB-Instance nicht neu starten. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle OLAP

Amazon RDS unterstützt Oracle OLAP durch die Verwendung der `OLAP`-Option. Diese Option bietet On-line Analytical Processing (OLAP) für Oracle-DB-Instances. Mit Oracle OLAP können Sie große Datenmengen analysieren, indem Sie Dimensionsobjekte und Cubes gemäß dem OLAP-Standard erstellen. Weitere Informationen finden Sie in der Oracle-Dokumentation.

> **Important**
>
> Bei der Verwendung von Oracle OLAP aktualisiert Amazon RDS automatisch Ihre DB-Instance auf die neueste Oracle-PSU, falls Schwachstellen mit einem Common Vulnerability Scoring System (CVSS)-Schweregrad von mehr als 9 oder andere gemeldete Schwachstellen vorliegen.

Amazon RDS unterstützt Oracle OLAP für die Enterprise Edition von Oracle Database 19c und höher.

Voraussetzungen für Oracle OLAP

Für die Verwendung von Oracle OLAP gelten folgende Voraussetzungen:

- Sie benötigen eine Oracle OLAP-Lizenz von Oracle. Weitere Informationen finden Sie unter Lizenzierungsinformationen in der Oracle-Dokumentation.
- Ihre DB-Instance muss eine ausreichende Instance-Klasse besitzen. Oracle OLAP wird für die DB-Instance-Klassen db.t3.small nicht unterstützt. Weitere Informationen finden Sie unter RDS für Oracle-DB-Instance-Klassen.
- Für Ihre DB-Instance muss Auto Minor Version Upgrade (Automatisches Unterversionsupgrade) aktiviert sein. Diese Option ermöglicht es Ihrer DB-Instance, DB-Engine-Unterversions-Upgrades automatisch zu erhalten, sobald diese verfügbar sind, und ist für alle Optionen erforderlich, die die Oracle Java Virtual Machine (JVM) installieren. Amazon RDS verwendet diese Option, um Ihre DB-Instance mit dem neuesten Oracle-Patch-Set-Update (PSU) oder Release-Update (RU) zu aktualisieren. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Ihre DB-Instance darf keinen Benutzer mit dem Namen habe `OLAPSYS`. Wenn dies der Fall ist, schlägt die Installation der OLAP-Option fehl.

Bewährte Methoden für Oracle OLAP

Für die Verwendung von Oracle OLAP gelten folgende bewährte Methoden:

- Für maximale Sicherheit sollten Sie die OLAP-Option mit Secure Sockets Layer (SSL) verwenden. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.
- Konfigurieren Sie Ihre DB-Instance, um den Zugriff auf Ihre DB-Instance einzuschränken. Weitere Informationen erhalten Sie unter Szenarien für den Zugriff auf einen in einem VPC und Arbeiten mit einer DB-Instance in einer VPC.

Hinzufügen der Oracle OLAP-Option

Es folgt der allgemeine Vorgang für das Hinzufügen der OLAP-Option zu einer DB-Instance:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Wenn Oracle Java Virtual Machine (JVM) nicht auf der DB-Instance installiert ist, kommt es zu einem kurzen Ausfall, während die Option OLAP hinzugefügt wird. Es gibt keinen Ausfall, wenn Oracle Java Virtual Machine (JVM) bereits auf der DB-Instance installiert ist. Nachdem Sie die Funktion hinzugefügt haben, müssen Sie Ihre DB-Instance neu starten. Sobald die Optionsgruppe aktiv ist, ist auch Oracle OLAP verfügbar.

So fügen Sie die OLAP-Option zu einer DB-Instance hinzu

1. Bestimmen Sie die Optionsgruppe, die Sie verwenden möchten. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   - Wählen Sie für Engine die Oracle-Edition für Ihre DB-Instance aus.
   - Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie der Optionsgruppe die Option OLAP hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

- Weisen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Verwenden von Oracle OLAP

Nachdem Sie die Option Oracle OLAP aktiviert haben, können Sie mit der Nutzung beginnen. Eine Liste der Funktionen, die für Oracle OLAP unterstützt werden, finden Sie in der Oracle-Dokumentation.

Entfernen der Oracle OLAP-Option

Nachdem Sie alle Objekte gelöscht haben, die Datentypen verwenden, die von der OLAP-Option bereitgestellt werden, können Sie die Option aus einer DB-Instance entfernen. Wenn Oracle Java Virtual Machine (JVM) nicht auf der DB-Instance installiert ist, kommt es zu einem kurzen Ausfall, während die Option OLAP entfernt wird. Es gibt keinen Ausfall, wenn Oracle Java Virtual Machine (JVM) bereits auf der DB-Instance installiert ist. Nachdem Sie die OLAP-Option entfernt haben, müssen Sie Ihre DB-Instance nicht neu starten.

So löschen Sie die **OLAP**-Option:

1. Sichern Sie Ihre Daten.

   > **Warning**
   >
   > Wenn die Instance Datentypen verwendet, die als Teil der Option aktiviert wurden, und wenn Sie die OLAP-Option entfernen, können Sie Daten verlieren. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

2. Überprüfen Sie, ob vorhandene Objekte auf Datentypen oder Funktionen der OLAP-Option verweisen.
3. Löschen Sie alle Objekte, die auf Datentypen oder Funktionen der OLAP-Option verweisen.
4. Führen Sie eine der folgenden Aufgaben aus:

- Entfernen Sie die 0LAP-Option aus der zugehörigen Optionsgruppe. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, in der die 0LAP-Option nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle Secure Sockets Layer

Um die SSL Verschlüsselung RDS für eine Oracle-DB-Instance zu aktivieren, fügen Sie die SSL Oracle-Option der Optionsgruppe hinzu, die der DB-Instance zugeordnet ist. Amazon RDS verwendet einen zweiten Port, wie von Oracle gefordert, für SSL Verbindungen. Dieser Ansatz ermöglicht die gleichzeitige Kommunikation zwischen einer DB-Instance und SQL *Plus sowohl im Klartext als auch in SSL verschlüsselter Form. Sie können beispielsweise den Port mit Klartext-Kommunikation verwenden, um mit anderen Ressourcen innerhalb von zu kommunizieren, VPC während Sie den Port mit SSL -verschlüsselter Kommunikation für die Kommunikation mit Ressourcen außerhalb von verwenden. VPC

> **Note**
>
> Sie können entweder SSL oder Native Network Encryption (NNE) auf derselben RDS Oracle-DB-Instance verwenden, aber nicht beide. Wenn Sie SSL Verschlüsselung verwenden, stellen Sie sicher, dass Sie alle anderen Verbindungsverschlüsselungen ausschalten. Weitere Informationen finden Sie unter Oracle Native Network Encryption.

SSL/TLSund NNE sind nicht mehr Teil von Oracle Advanced Security. In RDS Oracle können Sie SSL Verschlüsselung mit allen lizenzierten Editionen der folgenden Datenbankversionen verwenden:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Themen

- TLSVersionen für die SSL Oracle-Option
- Cipher Suites für die Oracle-Option SSL
- FIPSunterstützen
- SSLOption hinzufügen
- SQL*Plus RDS für die Verwendung SSL mit einer Oracle-DB-Instance konfigurieren
- Herstellen der Verbindung mit einer DB-Instance von RDS für Oracle mit SSL
- SSLVerbindung einrichten über JDBC
- Einen DN-Match mit einer SSL Verbindung erzwingen
- Problembehandlung bei SSL Verbindungen

TLSVersionen für die SSL Oracle-Option

Amazon RDS for Oracle unterstützt die Transport Layer Security (TLS) Versionen 1.0 und 1.2. Wenn Sie eine neue SSL Oracle-Option hinzufügen, legen Sie diese `SQLNET.SSL_VERSION` explizit auf einen gültigen Wert fest. Die folgenden Werte sind für diese Optionseinstellung zulässig:

- `"1.0"`— Clients können nur mit TLS Version 1.0 eine Verbindung zur DB-Instance herstellen. `SQLNET.SSL_VERSION`Ist für bestehende SSL Oracle-Optionen auf `"1.0"` automatisch gesetzt. Sie können die Einstellung bei Bedarf ändern.
- `"1.2"`— Clients können nur mit TLS 1.2 eine Verbindung zur DB-Instance herstellen.
- `"1.2 or 1.0"`— Clients können entweder mit TLS 1.2 oder 1.0 eine Verbindung zur DB-Instance herstellen.

Cipher Suites für die Oracle-Option SSL

Amazon RDS for Oracle unterstützt mehrere SSL Cipher Suites. Standardmäßig ist die SSL Oracle-Option für die Verwendung der `SSL_RSA_WITH_AES_256_CBC_SHA` Cipher Suite konfiguriert. Verwenden Sie die Optionseinstellung, um eine andere Verschlüsselungssuite für SSL Verbindungen anzugeben. `SQLNET.CIPHER_SUITE`

Sie können mehrere Werte für angeben. `SQLNET.CIPHER_SUITE` Diese Technik ist nützlich, wenn Sie Datenbanklinks zwischen Ihren DB-Instances haben und beschließen, Ihre Cipher Suites zu aktualisieren.

In der folgenden Tabelle wird die SSL Unterstützung RDS für Oracle in allen Editionen von Oracle Database 19c und 21c zusammengefasst.

Verschlüsselungssuite (. SQLNET CIPHER_) SUITE	TLSVersio nsunterstützung (SQLNET. SSL_VERSION)	FIPSunter stützen	RAMPFed-konform
SSL_ RSA _ WITH _ AES _256_ CBC _ SHA (Standard)	1.0 und 1.2	Ja	Nein
SSL_ _ _ RSA _256_ WITH _ AES CBC SHA256	1.2	Ja	Nein

Verschlüsselungssuite (. SQLNET CIPHER_) SUITE	TLSVersionsunterstützung (SQLNET. SSL_VERSION)	FIPSunterstützen	RAMPFed-konform
SSL_ _ _ RSA _256_ WITH _ AES GCM SHA384	1.2	Ja	Nein
TLS_ _ _ ECDHE _ RSA _256_ WITH _ AES GCM SHA384	1.2	Ja	Ja
TLS_ _ _ ECDHE _ RSA _128_ WITH _ AES GCM SHA256	1.2	Ja	Ja
TLS_ _ _ ECDHE _ RSA _256_ WITH _ AES CBC SHA384	1.2	Ja	Ja
TLS_ _ _ ECDHE _ RSA _128_ WITH _ AES CBC SHA256	1.2	Ja	Ja
TLS_ _ _ ECDHE _ RSA _256_ WITH _ AES CBC SHA	1.2	Ja	Ja
TLS_ _ _ ECDHE _ RSA _128_ WITH _ AES CBC SHA	1.2	Ja	Ja

FIPSunterstützen

RDSfür Oracle ermöglicht es Ihnen, den Federal Information Processing Standard (FIPS) -Standard für 140-2 zu verwenden. FIPS140-2 ist ein Regierungsstandard der Vereinigten Staaten, der Sicherheitsanforderungen für kryptografische Module definiert. Sie aktivieren den FIPS Standard, indem Sie `TRUE` für die `FIPS.SSLFIPS_140` Oracle-Option auf setzen. SSL Wenn FIPS 140-2 für konfiguriert istSSL, verschlüsseln die kryptografischen Bibliotheken Daten zwischen dem Client und der RDS Oracle-DB-Instance.

Clients müssen die Cipher Suite verwenden, die -konform ist. FIPS Beim Aufbau einer Verbindung handeln der Client und die Oracle-DB-Instance aus, welche Cipher Suite RDS für die Hin- und Herübertragung von Nachrichten verwendet werden soll. Die Tabelle in Cipher Suites für die

Oracle-Option SSL zeigt die FIPS -konformen SSL Cipher Suites für jede Version. TLS Weitere Informationen finden Sie unter Oracle Database FIPS 140-2-Einstellungen in der Oracle Database-Dokumentation.

SSLOption hinzufügen

Um sie verwenden zu könnenSSL, muss Ihre RDS Oracle-DB-Instance einer Optionsgruppe zugeordnet sein, die die SSL Option enthält.

Konsole

Um die SSL Option einer Optionsgruppe hinzuzufügen

1. Erstellen Sie eine neue Optionsgruppe oder identifizieren Sie eine vorhandene Optionsgruppe, der Sie die Option SSL hinzufügen können.

   Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option SSL zur Optionsgruppe hinzu.

   Wenn Sie nur FIPS -verifizierte Cipher Suites für SSL Verbindungen verwenden möchten, setzen Sie die Option auf. FIPS.SSLFIPS_140 TRUE Hinweise zum FIPS Standard finden Sie unter. FIPSunterstützen

   Weitere Informationen zum Hinzufügen einer Option zu einer Optionsgruppe finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Erstellen Sie eine neue RDS für Oracle DB-Instance und ordnen Sie ihr die Optionsgruppe zu, oder ändern Sie eine RDS für Oracle DB-Instance, um ihr die Optionsgruppe zuzuordnen.

   Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

   Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

Um die SSL Option einer Optionsgruppe hinzuzufügen

1. Erstellen Sie eine neue Optionsgruppe oder identifizieren Sie eine vorhandene Optionsgruppe, der Sie die Option SSL hinzufügen können.

   Weitere Informationen zum Erstellen einer Optionsgruppe finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option SSL zur Optionsgruppe hinzu.

   Geben Sie die folgenden Einstellungen für die Option an:

   - `Port`— Die SSL Portnummer
   - `VpcSecurityGroupMemberships`— Die VPC Sicherheitsgruppe, für die die Option aktiviert ist
   - `SQLNET.SSL_VERSION`— Die TLS Version, mit der der Client eine Verbindung zur DB-Instance herstellen kann

   Mit dem folgenden AWS CLI Befehl wird die SSL Option beispielsweise einer Optionsgruppe mit dem Namen hinzugefügt`ora-option-group`.

   Example

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group --option-group-name ora-option-group \
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group --option-group-name ora-option-group ^
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

3. Erstellen Sie eine neue RDS für Oracle DB-Instance und ordnen Sie ihr die Optionsgruppe zu, oder ändern Sie eine RDS für Oracle DB-Instance, um ihr die Optionsgruppe zuzuordnen.

   Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

   Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

SQL*Plus RDS für die Verwendung SSL mit einer Oracle-DB-Instance konfigurieren

Bevor Sie eine Verbindung zu einer RDS Oracle-DB-Instance herstellen können, die die SSL Oracle-Option verwendet, müssen SQL Sie*Plus konfigurieren, bevor Sie eine Verbindung herstellen.

> **Note**
>
> Stellen Sie sicher, dass die Sicherheitsgruppen korrekt konfiguriert sind, um Zugriff auf die DB-Instance durch entsprechende Clients zu erlauben. Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen. Außerdem gelten diese Anweisungen für SQL *Plus und andere Clients, die direkt ein Oracle-Standardverzeichnis verwenden. JDBCVerbindungen finden Sie unterSSLVerbindung einrichten über JDBC.

So konfigurieren Sie SQL *Plus RDS für die SSL Verbindung mit einer Oracle-DB-Instance

1. Legen Sie in der Umgebungsvariablen `ORACLE_HOME` den Speicherort des Oracle-Stammverzeichnisses fest.

   Der Pfad zu Ihrem Oracle-Stammverzeichnis hängt von Ihrer Installation ab. Im folgenden Beispiel wird die Umgebungsvariable `ORACLE_HOME` festgelegt.

   ```
   prompt>export ORACLE_HOME=/home/user/app/user/product/19.0.0/dbhome_1
   ```

   Informationen zur Einstellung von Oracle-Umgebungsvariablen finden Sie unter SQL*Plus-Umgebungsvariablen in der Oracle-Dokumentation sowie im Oracle-Installationsleitfaden für Ihr Betriebssystem.

2. Fügen Sie `$ORACLE_HOME/lib` an die Umgebungsvariable `LD_LIBRARY_PATH` an.

Im Folgenden finden Sie ein Beispiel, das die PATH Umgebungsvariable LD_ LIBRARY _ festlegt.

```
prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

3. Erstellen Sie ein Oracle Wallet unter `$ORACLE_HOME/ssl_wallet`.

   Im folgenden Beispiel wird das Oracle Wallet-Verzeichnis erstellt.

   ```
   prompt>mkdir $ORACLE_HOME/ssl_wallet
   ```

4. Laden Sie die Zertifikatspaket-Datei (.pem) herunter, die für alle funktioniert, AWS-Regionen und legen Sie die Datei im Verzeichnis ssl_wallet ab. Weitere Informationen finden Sie unter .

5. Ändern oder erstellen Sie im Verzeichnis `$ORACLE_HOME/network/admin` die Datei `tnsnames.ora`, die folgenden Eintrag enthalten muss.

   ```
   net_service_name =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS =
           (PROTOCOL = TCPS)
           (HOST = endpoint)
           (PORT = ssl_port_number)
         )
       )
       (CONNECT_DATA =
         (SID = database_name)
       )
       (SECURITY =
         (SSL_SERVER_CERT_DN =
    "C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=endpoint")
       )
     )
   ```

6. Ändern oder erstellen Sie im selben Verzeichnis die Datei sqlnet.ora und binden Sie folgende Parameter ein.

> **Note**
>
> Um mit Entitäten über eine TLS sichere Verbindung zu kommunizieren, benötigt Oracle ein Wallet mit den für die Authentifizierung erforderlichen Zertifikaten. Sie können das ORAPKI Oracle-Hilfsprogramm verwenden, um Oracle-Wallets zu erstellen und zu verwalten, wie in Schritt 7 gezeigt. Weitere Informationen finden Sie ORAPKI in der Oracle-Dokumentation unter Oracle Wallet einrichten.

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
 $ORACLE_HOME/ssl_wallet)))
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 1.0
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
SSL_SERVER_DN_MATCH = ON
```

> **Note**
>
> Sie können für `SSL_VERSION` einen höheren Wert einstellen, sofern er von Ihrer DB-Instance unterstützt wird.

7. Führen Sie den folgenden Befehl aus, um das Oracle-Wallet zu erstellen.

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
```

8. Extrahieren Sie jedes Zertifikat in der .pem-Bundle-Datei mithilfe eines Betriebssystem-Dienstprogramms in eine separate .pem-Datei.
9. Fügen Sie jedes Zertifikat mit separaten `orapki` Befehlen zu Ihrer Wallet hinzu und *`certificate-pem-file`* ersetzen Sie es durch den absoluten Dateinamen der PEM-Datei.

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
      certificate-pem-file -auto_login_only
```

Weitere Informationen finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate.

Herstellen der Verbindung mit einer DB-Instance von RDS für Oracle mit SSL

Nachdem Sie SQL*Plus wie zuvor beschrieben für die Verwendung von SSL konfiguriert haben, können Sie die Verbindung mit der DB-Instance von RDS für Oracle mit der SSL-Option herstellen. Optional können Sie zuerst den `TNS_ADMIN`-Wert importieren, der auf das Verzeichnis mit den tnsnames.ora- und sqlnet.ora-Dateien verweist. Dadurch wird sichergestellt, dass SQL*Plus diese Dateien konsistent finden kann. Das folgende Beispiel exportiert den `TNS_ADMIN`-Wert.

```
export TNS_ADMIN=${ORACLE_HOME}/network/admin
```

Stellen Sie eine Verbindung mit der DB-Instance her. Sie können beispielsweise mithilfe von SQL*Plus und a *<net_service_name>* in einer Datei tnsnames.ora eine Verbindung herstellen.

```
sqlplus mydbuser@net_service_name
```

Sie können die Verbindung mit einer DB-Instance auch unter Verwendung von SQL*Plus ohne eine tnsnames.ora-Datei einrichten, indem Sie den folgenden Befehl ausführen.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = endpoint) (PORT
 = ssl_port_number))(CONNECT_DATA = (SID = database_name)))'
```

Sie können auch ohne SSL eine Verbindung mit der DB-Instance von RDS für Oracle aufbauen. Mit diesem Befehl kann beispielsweise eine Verbindung zur DB-Instance über den Clear-Text-Port ohne SSL-Verschlüsselung hergestellt werden.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = endpoint) (PORT
 = port_number))(CONNECT_DATA = (SID = database_name)))'
```

Wenn der TCP (Transmission Control Protocol)-Portzugriff beendet werden soll, erstellen Sie eine Sicherheitsgruppe ohne IP-Adresszugänge und fügen sie zur Instance hinzu. Dadurch werden Verbindungen über den TCP-Port geschlossen, wohingegen Verbindungen über den SSL-Port von IP-Adressen, die aus dem von der SSL-Option der Sicherheitsgruppe definierten Adressbereich stammen, weiterhin möglich sind.

SSLVerbindung einrichten über JDBC

Um eine SSL Verbindung über zu verwendenJDBC, müssen Sie einen Keystore erstellen, dem RDS Amazon-Root-CA-Zertifikat vertrauen und den unten angegebenen Codeausschnitt verwenden.

Um den Keystore im JKS Format zu erstellen, können Sie den folgenden Befehl verwenden. Weitere Informationen zum Erstellen des Keystores finden Sie unter Keystore erstellen in der Oracle-Dokumentation. Referenzinformationen finden Sie unter keytool in der Java Platform, Standard Edition Tools Reference.

```
keytool -genkey -alias client -validity 365 -keyalg RSA -keystore clientkeystore
```

Gehen Sie wie folgt vor, um dem RDS Amazon-Root-CA-Zertifikat zu vertrauen.

Um dem RDS Amazon-Root-CA-Zertifikat zu vertrauen

1. Laden Sie die .pem-Datei für das Zertifikatspaket herunter, die für alle funktioniert, AWS-Regionen und legen Sie die Datei im Verzeichnis ssl_wallet ab.

   Informationen zum Herunterladen von Zertifikaten finden Sie unter .
2. Extrahieren Sie jedes Zertifikat in der PEM-Datei mithilfe eines Betriebssystemdienstprogramms in eine separate Datei.
3. Konvertieren Sie jedes Zertifikat mithilfe eines separaten `openssl` Befehls in das.der-Format und ersetzen Sie dabei *`certificate-pem-file`* mit dem Namen der Zertifikats-.pem-Datei (ohne die Erweiterung.pem).

   ```
   openssl x509 -outform der -in certificate-pem-file.pem -out certificate-pem-file.der
   ```
4. Importieren Sie jedes Zertifikat mit dem folgenden Befehl in den Keystore.

   ```
   keytool -import -alias rds-root -keystore clientkeystore.jks -file certificate-pem-file.der
   ```

   Weitere Informationen finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate.
5. Bestätigen Sie, das der Schlüsselspeicher erfolgreich erstellt wurde.

   ```
   keytool -list -v -keystore clientkeystore.jks
   ```

   Geben Sie das Passwort des Schlüsselspeichers an, wenn Sie dazu aufgefordert werden.

Das folgende Codebeispiel zeigt, wie Sie die SSL Verbindung mit einrichten. JDBC

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "dns-name-provided-by-amazon-rds";
    private static final Integer SSL_PORT = "ssl-option-port-configured-in-option-
group";
    private static final String DB_SID = "oracle-sid";
    private static final String DB_USER = "user-name";
    private static final String DB_PASSWORD = "password";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
    private static final String KEY_STORE_PASS = "keystore-password";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Einen DN-Match mit einer SSL Verbindung erzwingen

Sie können den Oracle-Parameter `SSL_SERVER_DN_MATCH` verwenden, um eine Übereinstimmung des eindeutigen Namens (DN) für den Datenbank-Server mit dem Service-Namen zu erzwingen. Wenn Sie die Übereinstimmungsüberprüfungen erzwingen, SSL stellen Sie sicher, dass das Zertifikat vom Server stammt. Wenn Sie die Übereinstimmungsüberprüfung nicht erzwingen, SSL führt dann die Prüfung durch, lässt aber die Verbindung zu, unabhängig davon, ob eine Übereinstimmung vorliegt. Wenn Sie die Übereinstimmung nicht erzwingen, erlauben Sie dem Server, seine Identität möglicherweise zu fälschen.

Fügen Sie die Eigenschaft für die DN-Übereinstimmung hinzu und verwenden Sie unten angegebene Verbindungszeichenfolge, um eine DN-Übereinstimmung zu erzwingen.

Fügen Sie der Clientverbindung die Eigenschaft zum Erzwingen der DN-Übereinstimmung hinzu.

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

Verwenden Sie die folgende Verbindungszeichenfolge, um den DN-Abgleich bei der Verwendung zu erzwingenSSL.

```
final String connectionString = String.format(
    "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
    "(CONNECT_DATA=(SID=%s))" +
    "(SECURITY = (SSL_SERVER_CERT_DN =
\"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=%s\")))",
    DB_SERVER_NAME, SSL_PORT, DB_SID, DB_SERVER_NAME);
```

Problembehandlung bei SSL Verbindungen

Möglicherweise erhalten Sie beim Abfragen Ihrer Datenbank den Fehler `ORA-28860`.

```
ORA-28860: Fatal SSL error
28860. 00000 - "Fatal SSL error"
*Cause: An error occurred during the SSL connection to the peer. It is likely that this
 side sent data which the peer rejected.
*Action: Enable tracing to determine the exact cause of this error.
```

Dieser Fehler tritt auf, wenn der Client versucht, eine Verbindung mit einer Version herzustellenTLS, die der Server nicht unterstützt. Um diesen Fehler zu vermeiden, bearbeiten Sie die Datei sqlnet.ora

und stellen Sie `SSL_VERSION` sie auf die richtige Version ein. TLS Weitere Informationen finden Sie im Oracle-Supportdokument 2748438.1 auf der My Oracle Support Site.

Oracle Spatial

Amazon RDS unterstützt Oracle Spatial durch die Verwendung der `SPATIAL`-Option. Oracle Spatial stellt ein SQL-Schema und Funktionen bereit, die Speicherung, Abruf, Update und Abfrage von Sammlungen großer Datensätze in einer Oracle-Datenbank ermöglichen. Weitere Informationen finden Sie unter Konzepte von Spatial in der Oracle-Dokumentation. Amazon RDS unterstützt Oracle Spatial in allen Editionen aller unterstützten Versionen.

So funktionieren Spatial Patch Bundles (SPBs)

Jedes Quartal veröffentlicht RDS for Oracle neue kleinere Engine-Versionen für jede unterstützte Haupt-Engine. Eine Version der Engine Release Update (RU) enthält Fehlerkorrekturen von Oracle, indem sie die RU-Patches für das angegebene Quartal enthält. Eine Engine-Version des Spatial Patch Bundle (SPB) enthält RU-Patches sowie spezielle Patches für Oracle Spatial. Beispielsweise ist 19.0.0.0.ru-2025-01.spb-1.r1 eine kleinere Engine-Version, die die RU-Patches in der Engine-Version 19.0.0.0.ru-2025-01.rur-2025-01.r1 sowie Spatial-Patches enthält. SPBs werden nur für Oracle Database 19c unterstützt.

SPBs funktionieren auf die gleiche Weise wie RUs, obwohl sie unterschiedlich benannt sind. Ein RU verwendet das Benennungsformat 19.0.0.0.ru-2025-01.rur-2025-01.r1. Ein SPB-Name enthält den Text „spb“, wie in 19.0.0.0.ru-2025-01.spb-1.r1. In der Regel wird ein SPB 2—3 Wochen nach der entsprechenden vierteljährlichen Veröffentlichung veröffentlicht. Beispielsweise wird 19.0.0.0.ru-2025-01.spb-1.r1 nach 19.0.0.0.ru-2025-01.rur-2025-01.r1 veröffentlicht.

RDS for Oracle verfügt über separate Pfade für automatische Upgrades kleinerer Versionen von und. RUs SPBs Wenn Ihre DB-Instance eine RU verwendet, aktualisiert RDS Ihre Instance automatisch auf eine RU. Wenn Ihre DB-Instance ein SPB verwendet, aktualisiert RDS Ihre Instance auf ein SPB. Wenn Sie Ihrer DB-Instance die Option Oracle Spatial hinzufügen und automatische Upgrades aktivieren, platziert RDS Ihre Instance in den Upgrade-Pfad für. SPBs

Weitere Informationen zu RUs und finden SPBs Sie unterOracle-Unterversion-Upgrades. Eine Liste der unterstützten Oracle Database 19c RUs und SPBs für Oracle Database 19c finden Sie unter Amazon RDS for Oracle Database 19c (19.0.0.0) in den Versionshinweisen zu Amazon RDS for Oracle.

Voraussetzungen für Oracle Spatial

Es folgen die Voraussetzungen für den Einsatz von Oracle Spatial:

- Stellen Sie sicher, dass Ihre DB-Instance zu einer ausreichenden Instance-Klasse gehört. Oracle Spatial wird für die DB-Instance-Klassen db.t3.small nicht unterstützt. Weitere Informationen finden Sie unter RDS für Oracle-DB-Instance-Klassen.
- Stellen Sie sicher, dass für Ihre DB-Instance Automatisches Unterversion-Upgrade aktiviert ist. Diese Option ermöglicht es Ihrer DB-Instance, DB-Engine-Unterversions-Upgrades automatisch zu erhalten, sobald diese verfügbar sind, und ist für alle Optionen erforderlich, die die Oracle Java Virtual Machine (JVM) installieren. Amazon RDS verwendet diese Option, um Ihre DB-Instance mit dem neuesten Oracle-Patch-Set-Update (PSU) oder Release-Update (RU) zu aktualisieren. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Bewährte Methoden für Oracle Spatial

Es folgen die bewährten Methoden für den Einsatz von Oracle Spatial:

- Für maximale Sicherheit sollten Sie die `SPATIAL`-Option mit Secure Sockets Layer (SSL) verwenden. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.
- Konfigurieren Sie Ihre DB-Instance, um den Zugriff auf Ihre DB-Instance einzuschränken. Weitere Informationen erhalten Sie unter Szenarien für den Zugriff auf einen in einem VPC und Arbeiten mit einer DB-Instance in einer VPC.

Hinzufügen der Oracle Spatial-Option

Es folgt der allgemeine Vorgang für das Hinzufügen der `SPATIAL`-Option zu einer DB-Instance:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Wenn Oracle Java Virtual Machine (JVM) nicht auf der DB-Instance installiert ist, kommt es zu einem kurzen Ausfall, während die Option `SPATIAL` hinzugefügt wird. Es gibt keinen Ausfall, wenn Oracle Java Virtual Machine (JVM) bereits auf der DB-Instance installiert ist. Nachdem Sie die Funktion hinzugefügt haben, müssen Sie Ihre DB-Instance neu starten. Sobald die Optionsgruppe aktiv ist, ist auch Oracle Spatial verfügbar.

> **Note**
>
> Während dieses Ausfalls werden die Funktionen zur Passwortverifizierung kurzzeitig deaktiviert. Sie können auch erwarten, dass während des Ausfalls Ereignisse im Zusammenhang mit der Passwortverifizierung auftreten. Die Passwortverifikationsfunktionen werden wieder aktiviert, bevor die Oracle DB-Instance verfügbar ist.

So können Sie die **`SPATIAL`**-Option zu einer DB-Instance hinzufügen

1. Bestimmen Sie die Optionsgruppe, die Sie verwenden möchten. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie für Engine die Oracle-Edition für Ihre DB-Instance aus.

   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die Option SPATIAL zur Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Entfernen der Oracle Spatial-Option

Nachdem Sie alle Objekte gelöscht haben, die Datentypen verwenden, die von der `SPATIAL`-Option bereitgestellt werden, können Sie die Option aus einer DB-Instance löschen. Wenn Oracle Java Virtual Machine (JVM) nicht auf der DB-Instance installiert ist, kommt es zu einem kurzen Ausfall, während die Option `SPATIAL` entfernt wird. Es gibt keinen Ausfall, wenn Oracle Java Virtual Machine

(JVM) bereits auf der DB-Instance installiert ist. Nachdem Sie die SPATIAL-Option entfernt haben, müssen Sie Ihre DB-Instance nicht neu starten.

So löschen Sie die **SPATIAL**-Option:

1. Sichern Sie Ihre Daten.

   > **Warning**
   >
   > Wenn die Instance Datentypen verwendet, die als Teil der Option aktiviert wurden, und wenn Sie die SPATIAL-Option entfernen, können Sie Daten verlieren. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

2. Überprüfen Sie, ob vorhandene Objekte auf Datentypen oder Funktionen der SPATIAL-Option verweisen.

   Wenn SPATIAL-Optionen vorhanden sind, kann die Instance hängen bleiben, wenn die neue Optionsgruppe angewendet wird, die nicht über die SPATIAL-Option verfügt. Sie können die Objekte mithilfe der folgenden Abfragen identifizieren:

   ```
   SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE '%TABLE%'
   AND    (OWNER, SEGMENT_NAME) IN
          (SELECT DISTINCT OWNER, TABLE_NAME
           FROM   DBA_TAB_COLUMNS
           WHERE  DATA_TYPE='SDO_GEOMETRY'
           AND    OWNER <> 'MDSYS')
   ORDER BY 1,2,3,4;

   SELECT OWNER, TABLE_NAME, COLUMN_NAME
   FROM   DBA_TAB_COLUMNS
   WHERE  DATA_TYPE = 'SDO_GEOMETRY'
   AND    OWNER <> 'MDSYS'
   ORDER BY 1,2,3;
   ```

3. Löschen Sie alle Objekte, die auf Datentypen oder Funktionen der SPATIAL-Option verweisen.
4. Führen Sie eine der folgenden Aufgaben aus:

- Entfernen Sie die `SPATIAL`-Option aus der zugehörigen Optionsgruppe. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.
- Ändern Sie die DB-Instance und legen sie eine andere Optionsgruppe fest, in der die `SPATIAL`-Option nicht enthalten ist. Diese Änderung betrifft eine einzelne DB-Instance. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Orakel SQLT

Amazon RDS unterstützt Oracle SQLTXPLAIN (SQLT) durch die Verwendung der SQLT Option. Sie können es SQLT mit jeder Edition von Oracle Database 19c und höher verwenden.

Die `EXPLAIN PLAN` Oracle-Anweisung kann den Ausführungsplan einer SQL Anweisung bestimmen. Sie kann überprüfen, ob der Oracle-Optimierer einen bestimmten Ausführungsplan auswählt, z. B. einen verschachtelten Schleifen-Join. Sie hilft Ihnen auch, die Entscheidungen des Optimierers zu verstehen, z. B. warum er einen verschachtelten Schleifen-Join einem Hash-Join den Vorzug gegeben hat. `EXPLAIN PLAN` hilft Ihnen also, die Performance der Anweisung zu verstehen.

SQLTist ein Oracle-Hilfsprogramm, das einen Bericht erstellt. Der Bericht enthält Objektstatistiken, Objektmetadaten, Optimizer-bezogene Initialisierungsparameter und andere Informationen, die ein Datenbankadministrator verwenden kann, um eine SQL Anweisung für eine optimale Leistung zu optimieren. SQLTerstellt einen HTML Bericht mit Hyperlinks zu allen Abschnitten des Berichts.

SQLTArbeitet im Gegensatz zu Berichten aus Automatic Workload Repository oder Statspack mit einzelnen Anweisungen. SQL SQLTist eine Sammlung von SQL PL/- und SQL *Plus-DateienSQL, die Leistungsdaten sammeln, speichern und anzeigen.

Im Folgenden sind die unterstützten Oracle-Versionen für jede SQLT Version aufgeführt.

SQLTVersion	Oracle Database 21c	Oracle Database 19c
2018-07-25.v1	Unterstützt	Unterstützt
2018-03-31.v1	Nicht unterstützt	Nicht unterstützt
2016-04-29.v1	Nicht unterstützt	Nicht unterstützt

Um Anweisungen zur Verwendung herunterzuladen SQLT und darauf zuzugreifen:

- Melden Sie sich bei Ihrem My Oracle Support-Konto an und öffnen Sie die folgenden Dokumente:
- Zum HerunterladenSQLT: Dokument 215187.1
- Anweisungen zur SQLT Verwendung finden Sie in Dokument 1614107.1
- Für häufig gestellte Fragen SQLT zu: Dokument 1454160.1
- Informationen zum Lesen der SQLT Ausgabe finden Sie in Dokument 1456176.1
- Für die Interpretation des Hauptberichts: Document 1922234.1

Amazon unterstützt die folgenden SQLT Methoden RDS nicht:

- `XPLORE`
- `XHUME`

Voraussetzungen für SQLT

Die folgenden Voraussetzungen gelten für die Verwendung vonSQLT:

- Sie müssen Benutzer und Rollen entfernen, die für erforderlich sindSQLT, falls sie existieren.

  Die SQLT Option erstellt die folgenden Benutzer und Rollen in einer DB-Instance:

  - `SQLTXPLAIN` Benutzer
  - `SQLTXADMIN` Benutzer
  - `SQLT_USER_ROLE` Rolle

  Wenn Ihre DB-Instance über einen dieser Benutzer oder Rollen verfügt, melden Sie sich mit einem SQL Client bei der DB-Instance an und löschen Sie sie mit den folgenden Anweisungen:

  ```
  DROP USER SQLTXPLAIN CASCADE;
  DROP USER SQLTXADMIN CASCADE;
  DROP ROLE SQLT_USER_ROLE CASCADE;
  ```

- Sie müssen Tablespaces entfernen, die von benötigt werdenSQLT, falls sie existieren.

  Die SQLT Option erstellt die folgenden Tablespaces auf einer DB-Instance:

  - `RDS_SQLT_TS`
  - `RDS_TEMP_SQLT_TS`

  Wenn Ihre DB-Instance über diese Tablespaces verfügt, melden Sie sich mit einem SQL Client bei der DB-Instance an und löschen Sie sie.

SQLTOptionseinstellungen

SQLTkann mit lizenzierten Funktionen arbeiten, die vom Oracle Tuning Pack und dem Oracle Diagnostics Pack bereitgestellt werden. Das Oracle Tuning Pack beinhaltet den SQL Tuning

Advisor, und das Oracle Diagnostics Pack beinhaltet das Automatic Workload Repository. Die SQLT Einstellungen aktivieren oder deaktivieren den Zugriff auf diese Funktionen vonSQLT.

Amazon RDS unterstützt die folgenden Einstellungen für die SQLT Option.

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
`LICENSE_PACK`	`T`, `D`, `N`	`N`	Die Oracle Management Packs, mit denen Sie zugreifen möchtenSQLT. Geben Sie einen der folgenden Werte ein: • `T`gibt an, dass Sie über eine Lizenz für das Oracle Tuning Pack und das Oracle Diagnostics Pack verfügen und von dort aus auf den SQL Tuning Advisor und das Automatic Workload Repository zugreifen möchtenSQLT. • `D`gibt an, dass Sie eine Lizenz für das Oracle Diagnostics Pack besitzen und von dort auf das Automatic Workload Repository zugreifen möchtenSQLT. • `N`gibt an, dass Sie keine Lizenz für das Oracle Tuning Pack und das Oracle Diagnostics Pack haben oder dass Sie eine Lizenz für eines oder beide haben, aber nicht SQLT darauf zugreifen möchten. **Note** Amazon RDS stellt keine Lizenzen für diese Oracle Management Packs zur

Optionseinstellung	Zulässige Werte	Standardwert	Beschreibung
			Verfügung. Wenn Sie angeben, dass Sie ein Paket verwenden möchten, das nicht in Ihrer DB-Instance enthalten ist, können Sie es SQLT mit der DB-Instance verwenden . SQLTSie können jedoch nicht auf das Paket zugreifen, und der SQLT Bericht enthält keine Daten für das Paket. Wenn Sie beispielsweise angebenT, dass die DB-Instance das Oracle Tuning Pack nicht enthält, SQLT funktioniert sie auf der DB-Instance, aber der von ihr generierte Bericht enthält keine Daten, die sich auf das Oracle Tuning Pack beziehen.

`VERSION`	`2016-04-29.v1`
	
	`2018-03-31.v1`
	
	`2018-07-25.v1`
		`2016-04-29.v1`	Die VersionSQLT, die Sie installieren möchten.
			
			Note
			Für Oracle Database 19c und 21c ist `2018-07-25.v1` die einzige unterstützte Version. Diese Version ist die Standardversion für diese Releases.

Die SQLT Option wird hinzugefügt

Im Folgenden wird das allgemeine Verfahren zum Hinzufügen der SQLT Option zu einer DB-Instance beschrieben:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Fügen Sie die SQLT Option der Optionsgruppe hinzu.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie die SQLT Option hinzugefügt haben, ist sie aktiv, sobald die Optionsgruppe aktiv SQLT ist.

Um die SQLT Option zu einer DB-Instance hinzuzufügen

1. Bestimmen Sie die Optionsgruppe, die Sie verwenden möchten. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie im Feld Engine die Oracle-Edition aus, die Sie verwenden möchten. Die SQLT Option wird in allen Editionen unterstützt.
   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die SQLTOption der Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

   - Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
4. (Optional) Überprüfen Sie die SQLT Installation auf jeder DB-Instance mit der SQLT Option.

a. Verwenden Sie einen SQL Client, um als Masterbenutzer eine Verbindung zur DB-Instance herzustellen.

   Hinweise zum Herstellen einer Verbindung mit einer Oracle-DB-Instance mithilfe eines SQL Clients finden Sie unterHerstellen der Verbindung zu Ihrer Oracle-DB-Instance.

b. Führen Sie die folgende Abfrage aus:

```
SELECT sqltxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
```

   Die Abfrage gibt die aktuelle Version der SQLT Option bei Amazon zurückRDS. `12.1.160429`ist ein Beispiel für eine Version davonSQLT, die bei Amazon erhältlich istRDS.

5. Ändern Sie die Passwörter der Benutzer, die mit der SQLT Option erstellt wurden.

a. Verwenden Sie einen SQL Client, um sich als Masterbenutzer mit der DB-Instance zu verbinden.

b. Führen Sie die folgende SQL Anweisung aus, um das Passwort für den `SQLTXADMIN` Benutzer zu ändern:

```
ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

c. Führen Sie die folgende SQL Anweisung aus, um das Passwort für den `SQLTXPLAIN` Benutzer zu ändern:

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

> **Note**
>
> Für ein Upgrade SQLT muss eine ältere Version von deinstalliert SQLT und anschließend die neue Version installiert werden. Daher können beim Upgrade SQLT alle SQLT Metadaten verloren gehen. Bei einem Upgrade einer Hauptversion einer Datenbank wird auch eine Deinstallation und SQLT Neuinstallation durchgeführt. Ein Beispiel für ein Upgrade einer Hauptversion ist ein Upgrade von Oracle Database 19c auf Oracle Database 21c.

Verwenden SQLT

SQLTfunktioniert mit dem SQL Oracle*Plus-Hilfsprogramm.

Zu verwenden SQLT

1. Laden Sie die SQLT ZIP-Datei aus Dokument 215187.1 auf der My Oracle Support-Website herunter.

   > **Note**
   >
   > Sie können SQLT 12.1.160429 nicht von der My Oracle Support-Website herunterladen. Oracle hat diese ältere Version eingestellt.

2. Entpacken Sie die ZIP-Datei. SQLT
3. Wechseln Sie von einer Eingabeaufforderung in das Verzeichnis `sqlt/run` auf Ihrem Dateisystem.
4. Öffnen Sie in der Befehlszeile SQL *Plus und stellen Sie als Masterbenutzer eine Verbindung zur DB-Instance her.

   Hinweise zum Herstellen einer Verbindung mit einer DB-Instance mithilfe SQL von*Plus finden Sie unter. Herstellen der Verbindung zu Ihrer Oracle-DB-Instance

5. Ermittelt die SQL ID einer SQL Anweisung:

```
SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
```

Ihre Ausgabe sieht ähnlich aus wie:

```
SQL_ID
-------------
chvsmttqjzjkn
```

6. Analysieren Sie eine SQL Aussage mitSQLT:

```
START sqltxtract.sql sql_id sqltxplain_user_password
```

Geben Sie beispielsweise für die SQL ID `chvsmttqjzjkn` Folgendes ein:

```
START sqltxtract.sql chvsmttqjzjkn sqltxplain_user_password
```

SQLTgeneriert den HTML Bericht und die zugehörigen Ressourcen als ZIP-Datei in dem Verzeichnis, von dem aus der SQLT Befehl ausgeführt wurde.

7. (Optional) Damit Anwendungsbenutzer SQL Anweisungen diagnostizieren könnenSQLT, erteilen `SQLT_USER_ROLE` Sie jedem Anwendungsbenutzer die folgende Anweisung:

```
GRANT SQLT_USER_ROLE TO application_user_name;
```

> **Note**
>
> Oracle rät davon ab, die Ausführung SQLT mit dem `SYS` Benutzer oder mit Benutzern auszuführen, denen diese `DBA` Rolle zugewiesen wurde. Es hat sich bewährt, SQLT Diagnosen mit dem Konto des Anwendungsbenutzers auszuführen, indem dem Anwendungsbenutzer `SQLT_USER_ROLE` die Genehmigung erteilt wird.

Die SQLT Option wird aktualisiert

Mit Amazon RDS for Oracle können Sie die SQLT Option von Ihrer vorhandenen Version auf eine höhere Version aktualisieren. Um die SQLT Option zu aktualisieren, führen Sie die Schritte 1—3 Verwenden SQLT für die neue Version von ausSQLT. Wenn Sie SQLT in Schritt 7 dieses Abschnitts Berechtigungen für die vorherige Version von gewährt haben, gewähren Sie die Rechte auch für die neue SQLT Version erneut.

Ein Upgrade der SQLT Option führt zum Verlust der Metadaten der älteren SQLT Version. Das Schema der älteren SQLT Version und die zugehörigen Objekte werden gelöscht, und die neuere Version von SQLT wird installiert. Weitere Informationen zu den Änderungen in der neuesten SQLT Version finden Sie in Dokument 1614201.1 auf der My Oracle Support-Website.

> **Note**
>
> Versions-Downgrades werden nicht unterstützt.

SQLTEinstellungen ändern

Nach der Aktivierung SQLT können Sie die `VERSION` Einstellungen `LICENSE_PACK` und für die Option ändern.

Weitere Informationen über das Ändern von Optionseinstellungen finden Sie unter Ändern einer Optionseinstellung. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter SQLTOptionseinstellungen.

Die SQLT Option wird entfernt

Sie können SQLT aus einer DB-Instance entfernen.

Gehen Sie wie folgt vor, um SQLT aus einer DB-Instance zu entfernen:

- Um SQLT aus mehreren DB-Instances zu entfernen, entfernen Sie die SQLT Option aus der Optionsgruppe, zu der die DB-Instances gehören. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.

- Um SQLT aus einer einzelnen DB-Instance zu entfernen, ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, die die SQLT Option nicht enthält. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Oracle Statspack

Mit der Oracle Statspack-Option wird die Funktion der Oracle Statspack-Leistungsstatistik installiert und aktiviert. Oracle Statspack ist eine Sammlung von SQL PL/- und SQL *Plus-SkriptenSQL, die Leistungsdaten sammeln, speichern und anzeigen. Weitere Informationen über die Verwendung von Oracle Statspack finden Sie unter Oracle Statspack in der Oracle-Dokumentation.

> **Note**
>
> Oracle Statspack wird von Oracle nicht mehr unterstützt und wurde durch das erweiterte Automatic Workload Repository () ersetzt. AWR AWRist nur für Kunden der Oracle Enterprise Edition verfügbar, die das Diagnostics Pack gekauft haben. Sie können Oracle Statspack mit jeder Oracle-DB-Engine bei Amazon verwenden. RDS Sie können Oracle Statspack nicht auf Amazon RDS Read Replicas ausführen.

Einrichten von Oracle Statspack

Um Statspack-Skripts auszuführen, müssen Sie die Statspack-Option hinzufügen.

So richten Sie Oracle Statspack ein:

1. Melden Sie sich in einem SQL Client mit einem Administratorkonto bei der Oracle-Datenbank an.
2. Führen Sie je nachdem, ob Statspack installiert ist, eine der folgenden Aktionen aus:

   - Wenn Statspack installiert ist und das `PERFSTAT`-Konto Statspack zugeordnet ist, fahren Sie mit Schritt 4 fort.
   - Wenn Statspack nicht installiert ist und das `PERFSTAT`-Konto vorhanden ist, löschen Sie das Konto wie folgt:

     ```
     DROP USER PERFSTAT CASCADE;
     ```

     Andernfalls erzeugt der Versuch, die Statspack-Option hinzuzufügen, einen Fehler und `RDS-Event-0058`.
3. Fügen Sie die Statspack-Option zu einer Optionsgruppe hinzu. Siehe Hinzufügen einer Option zu einer Optionsgruppe.

Amazon installiert die Statspack-Skripts RDS automatisch auf der DB-Instance und richtet dann das PERFSTAT Konto ein.

4. Setzen Sie das Passwort mit der folgenden SQL Anweisung zurück und ersetzen Sie pwd durch Ihr neues Passwort:

```
ALTER USER PERFSTAT IDENTIFIED BY pwd ACCOUNT UNLOCK;
```

Sie können sich mit dem PERFSTAT-Benutzerkonto anmelden und die Statspack-Skripts ausführen.

5. Erteilen Sie dem PERFSTAT Konto die CREATE JOB Berechtigung mit der folgenden Anweisung:

```
GRANT CREATE JOB TO PERFSTAT;
```

6. Stellen Sie sicher, dass Leerlaufwarteereignisse in der PERFSTAT.STATS$IDLE_EVENT-Tabelle eingegeben werden.

Aufgrund von Oracle-Fehler 28523746 werden die Leerlaufwarteereignisse in PERFSTAT.STATS $IDLE_EVENT möglicherweise nicht eingegeben. Um sicherzustellen, dass alle Leerlaufereignisse verfügbar sind, führen Sie die folgende Anweisung aus:

```
INSERT INTO PERFSTAT.STATS$IDLE_EVENT (EVENT)
SELECT NAME FROM V$EVENT_NAME WHERE WAIT_CLASS='Idle'
MINUS
SELECT EVENT FROM PERFSTAT.STATS$IDLE_EVENT;
COMMIT;
```

Generieren von Statspack-Berichten

Ein Statspack-Bericht vergleicht zwei Snapshots.

So generieren Sie Statspack-Berichte:

1. Melden Sie sich in einem SQL Client mit dem PERFSTAT Konto bei der Oracle-Datenbank an.
2. Erstellen Sie einen Snapshot mit einer der folgenden Techniken:

   - Erstellen Sie einen Statspack-Snapshot manuell.

- Erstellen Sie eine Aufgabe, die nach einem bestimmten Zeitintervall einen Statspack-Snapshot erstellt. Der folgende Job erstellt beispielsweise jede Stunde einen Statspack-Snapshot:

```
VARIABLE jn NUMBER;
exec dbms_job.submit(:jn, 'statspack.snap;',SYSDATE,'TRUNC(SYSDATE
+1/24,''HH24'')');
COMMIT;
```

3. Zeigen Sie die Snapshots mithilfe der folgenden Abfrage an:

```
SELECT SNAP_ID, SNAP_TIME FROM STATS$SNAPSHOT ORDER BY 1;
```

4. Führen Sie das RDS Amazon-Verfahren `rdsadmin.rds_run_spreport` aus und ersetzen Sie begin_snap und end_snap durch den Snapshot: IDs

```
exec rdsadmin.rds_run_spreport(begin_snap,end_snap);
```

   Beispielsweise wird mit dem folgenden Befehl ein Bericht erstellt, der auf dem Intervall zwischen den Statspack-Snapshots 1 und 2 basiert:

```
exec rdsadmin.rds_run_spreport(1,2);
```

   Der Dateiname des Statspack-Berichts enthält die Nummer der beiden Snapshots. So würde beispielsweise eine Berichtsdatei, in der die Statspack-Snapshots 1 und 2 verwendet werden, den Namen erhalte `ORCL_spreport_1_2.lst`.

5. Überwachen Sie die Ausgabe auf Fehler.

   Oracle Statspack führt Prüfungen durch, bevor der Bericht ausgeführt wird. Daher können Sie möglicherweise auch Fehlermeldungen in der Befehlsausgabe sehen. Sie können beispielsweise versuchen, einen Bericht basierend auf einem ungültigen Bereich zu generieren, wobei der anfängliche Statspack-Snapshot-Wert größer als der Endwert ist. In diesem Fall zeigt die Ausgabe die Fehlermeldung an, aber die DB-Engine generiert keine Fehlerdatei.

```
exec rdsadmin.rds_run_spreport(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

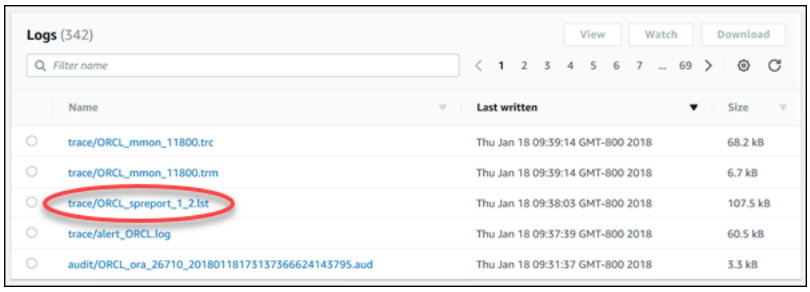
Wenn Sie eine ungültige Nummer für einen Statspack-Snapshot verwenden, zeigt die Ausgabe einen Fehler an. Wenn Sie beispielsweise versuchen, einen Bericht für Snapshots 1 und 50 zu erstellen, aber Snapshot 50 nicht vorhanden ist, zeigt die Ausgabe einen Fehler an.

```
exec rdsadmin.rds_run_spreport(1,50);
*
ERROR at line 1:
ORA-20000: Could not find both snapshot IDs
```

6. (Optional)

   Um den Bericht abzurufen, rufen Sie die Ablaufverfolgungsdatei-Prozeduren auf, wie unter erläuter Arbeiten mit Oracle-Trace-Dateien.

   Alternativ können Sie den Statspack-Bericht von der Konsole herunterladen. RDS Gehen Sie in den DB-Instance-Details zum Abschnitt Protokoll und wählen Sie Herunterladen aus. Das folgende Beispiel zeigt `trace/ORCL_spreport_1_2.lst`



   Wenn beim Generieren eines Berichts ein Fehler auftritt, verwendet die DB-Engine dieselben Namenskonventionen wie für einen Bericht, jedoch mit der Erweiterung von. `.err` Wenn beispielsweise ein Fehler beim Erstellen eines Berichts mit den Statspack-Snapshots 1 und 7 aufgetreten ist, hat die Berichtsdatei den Namen `ORCL_spreport_1_7.err`. Sie können den Fehlerbericht mit den gleichen Methoden herunterladen wie für einen standardmäßigen Snapshot-Bericht.

Entfernen von Statspack-Snapshots

Verwenden Sie den folgenden Befehl zum Entfernen eines Bereichs von Oracle-Statspack-Snapshots:

```
exec statspack.purge(begin snap, end snap);
```

Oracle-Zeitzone

Um die von Ihrer Oracle-DB-Instance verwendete Systemzeitzone zu ändern, können Sie die Zeitzonenoption verwenden. Gründe, die Zeitzone einer DB-Instance zu ändern, sind beispielsweise Kompatibilitätsanforderungen der Umgebung an einem Standort oder eine veraltete Anwendung. Mit der Zeitzonenoption wird die Zeitzone auf der Ebene des Hosts geändert. Wenn Sie die Zeitzone ändern, wirkt sich dies auf alle Datumsspalten und -werte aus, u. a. auf `SYSDATE` und `SYSTIMESTAMP`.

Die Zeitzonenoption unterscheidet sich von dem Befehl `rdsadmin_util.alter_db_time_zone`. Der Befehl `alter_db_time_zone` ändert die Zeitzone nur für bestimmte Datentypen. Mit der Zeitzonenoption wird die Zeitzone für alle Datumsspalten und -werte geändert. Mehr über `alter_db_time_zone` erfahren Sie unter Einstellen der Datenbank-Zeitzone. Weitere Informationen zu Upgrade-Überlegungen finden Sie unter Überlegungen zur Zeitzone.

Einschränkungen bei der Einstellung der Zeitzone

Die Zeitzonenoption ist eine permanente und persistente Option. Daher können Sie Folgendes nicht tun:

- Entfernen Sie die Option aus einer Optionsgruppe, nachdem Sie die Zeitzonenoption hinzugefügt haben.
- Entfernen Sie die Optionsgruppe aus einer DB-Instance, nachdem Sie die Gruppe hinzugefügt haben.
- Ändern Sie die Zeitzoneneinstellung für die Option zu einer anderen Zeitzone.

Empfehlungen für die Einstellung der Zeitzone

Bevor Sie Ihrer Produktionsdatenbank die Zeitzonenoption hinzufügen, raten wir dringend zu Folgendem:

- Erstellen Sie einen Snapshot Ihrer DB-Instance. Wenn Sie die Zeitzone versehentlich falsch eingestellt haben, müssen Sie die DB-Instance auf ihre vorherige Zeitzoneneinstellung zurücksetzen. Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
- Fügen Sie einer Test-DB-Instance die Zeitzonenoption hinzu. Bei dem Hinzufügen der Zeitzonenoption können Probleme in Zusammenhang mit Tabellen auftreten, die die Systemzeit verwenden, um Datums- bzw. Uhrzeitangaben einzufügen. Wir empfehlen Ihnen, Ihre Daten und

Anwendungen auf der Testinstanz zu analysieren. Auf diese Weise können Sie die Auswirkungen einer Änderung der Zeitzone auf Ihre Produktionsinstanz beurteilen.

Wenn Ihre DB-Instance die Standardoptionsgruppe verwendet, gehen Sie wie folgt vor:

1. Erstellen Sie einen Snapshot Ihrer DB-Instance.
2. Fügen Sie Ihrer DB-Instance die Zeitzonenoption hinzu.

Wenn Ihre DB-Instance derzeit eine nicht standardmäßige Optionsgruppe verwendet, gehen Sie wie folgt vor:

1. Erstellen Sie einen Snapshot Ihrer DB-Instance.
2. Erstellen Sie eine neue Optionsgruppe.
3. Fügen Sie ihr die Zeitzonenoption zusammen mit allen anderen Optionen hinzu, die derzeit der vorhandenen Optionsgruppe zugeordnet sind.

   Dadurch wird verhindert, dass die vorhandenen Optionen deinstalliert werden, während die Zeitzonenoption aktiviert wird.
4. Fügen Sie Ihrer DB-Instance die Optionsgruppe hinzu.

Einstellungen der Zeitzonenoption

Amazon RDS unterstützt die folgenden Einstellungen für die Zeitzonenoption.

Optionseinstellung	Zulässige Werte	Beschreibung
`TIME_ZONE`	Eine der verfügbaren Zeitzonen. Eine vollständige Liste finden Sie unter Verfügbare Zeitzonen.	Die neue Zeitzone für Ihre DB-Instance.

Hinzufügen der Zeitzonenoption

Gehen Sie wie folgt vor, um Ihrer DB-Instance die Zeitzonenoption hinzuzufügen:

1. (Empfohlen) Erstellen Sie einen Snapshot Ihrer DB-Instance.

2. Führen Sie eine der folgenden Aufgaben aus:
   - Erstellen Sie eine völlig neue Optionsgruppe. Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
   - Kopieren Sie eine bestehende Optionsgruppe mit dem AWS CLI oderAPI. Weitere Informationen finden Sie unter Kopieren einer Optionsgruppe.
   - Verwenden Sie eine vorhandene, nicht standardmäßige Optionsgruppe wieder. Es hat sich bewährt, eine Optionsgruppe zu verwenden, die derzeit keinen DB-Instances oder Snapshots zugeordnet ist.
3. Fügen Sie die neue Option der Optionsgruppe aus dem vorherigen Schritt hinzu.
4. Wenn für die Optionsgruppe, die derzeit mit Ihrer DB-Instance verknüpft ist, Optionen aktiviert sind, fügen Sie diese Optionen zu Ihrer neuen Optionsgruppe hinzu. Diese Strategie verhindert, dass die vorhandenen Optionen deinstalliert werden, während die neue Option aktiviert wird.
5. Fügen Sie die neue Optionsgruppe zu Ihrer DB-Instance hinzu.

Wenn Sie die Zeitzonenoption hinzufügen, entsteht während des automatischen Neustarts Ihrer DB-Instance ein kurzzeitiger Nutzungsausfall.

Konsole

Um die Zeitzonenoption zu einer Optionsgruppe hinzuzufügen und sie einer DB-Instance zuzuordnen

1. Wählen Sie in der RDS Konsole Optionsgruppen aus.
2. Wählen Sie den Namen der Optionsgruppe, zu der Sie die Option hinzufügen möchten.
3. Wählen Sie Add option (Option hinzufügen).
4. Wählen Sie als Optionsname die Option Timezone aus, und konfigurieren Sie dann die Optionseinstellungen.
5. Ordnen Sie die Optionsgruppe einer neuen oder vorhandenen DB-Instance zu:
   - Weisen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Wenn Sie die neue Option zu einer vorhandenen DB-Instance hinzufügen, kommt es zu einem kurzen Ausfall, während Ihre DB-Instance automatisch neu gestartet wird. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

Im folgenden Beispiel wird der Befehl AWS CLI add-option-to-option-group verwendet, um die `Timezone` Option und die Optionseinstellung zu einer `TIME_ZONE` Optionsgruppe namens hinzuzufügen. `myoptiongroup` Als Zeitzone wird festgeleg `Africa/Cairo`.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" ^
    --apply-immediately
```

Ändern der Zeitzoneneinstellungen

Die Zeitzonenoption ist eine permanente und persistente Option. Wenn Sie die Option einer Optionsgruppe hinzugefügt haben, kann sie nicht mehr aus der Gruppe entfernt werden. Wenn Sie die Optionsgruppe einer DB-Instance zugeordnet haben, kann die Zuordnung nicht mehr entfernt werden. Sie können auch die Zeitzoneneinstellung für die Option nicht mehr ändern, also keine andere Zeitzone angeben. Wenn Sie die Zeitzone falsch einstellen, müssen Sie Ihre DB-Instance anhand eines Snapshots wiederherstellen, den Sie vor dem Hinzufügen der Zeitzonenoption erstellt haben.

Entfernen der Zeitzonenoption

Die Zeitzonenoption ist eine permanente und persistente Option. Wenn Sie die Option einer Optionsgruppe hinzugefügt haben, kann sie nicht mehr aus der Gruppe entfernt werden. Wenn Sie die Optionsgruppe einer DB-Instance zugeordnet haben, kann die Zuordnung nicht mehr entfernt werden. Wenn Sie die Zeitzonenoption entfernen möchten, müssen Sie Ihre DB-Instance anhand eines Snapshots wiederherstellen, den Sie vor dem Hinzufügen der Zeitzonenoption erstellt haben.

Verfügbare Zeitzonen

Sie können die folgenden Werte für die Zeitzonenoption verwenden:

Bereich	Zeitzone
Afrika	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
Amerika	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York, America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asien	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damascus, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Jakarta, Asia/Jerusalem, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantik	Atlantic/Azores, Atlantic/Cape_Verde
Australien	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brasilien	Brazil/DeNoronha, Brazil/East
Kanada	Canada/Newfoundland, Canada/Saskatchewan
Etc	Etc/ -3 GMT

Bereich	Zeitzone
Europa	Europe/Amsterdam, Europe/Athens, Europe/Berlin, Europe/Dublin, Europe/Helsinki, Europe/Kaliningrad, Europe/London, Europe/Madrid, Europe/Moscow, Europe/Paris, Europe/Prague, Europe/Rome, Europe/Sarajevo
Pazifik	Pacific/Apia, Pacific/Auckland, Pacific/Chatham, Pacific/Fiji, Pacific/Guam, Pacific/Honolulu, Pacific/Kiritimati, Pacific/Marquesas, Pacific/Samoa, Pacific/Tongatapu, Pacific/Wake
US	US/Alaska, US/Central, US/East-Indiana, US/Eastern, US/Pacific
UTC	UTC

Automatische Aktualisierung der Oracle-Zeitzonendatei

Mit dieser `TIMEZONE_FILE_AUTOUPGRADE` Option können Sie die aktuelle Zeitzonendatei auf die neueste Version Ihrer RDS Oracle-DB-Instance aktualisieren.

Themen

- Übersicht über Oracle-Zeitzonendateien
- Strategien zum Aktualisieren Ihrer Zeitzonendatei
- Ausfallzeiten während der Aktualisierung der Zeitzonendatei
- Vorbereiten der Aktualisierung der Zeitzonendatei
- Hinzufügen der Option zur automatischen Aktualisierung der Zeitzonendatei
- Überprüfen Ihrer Daten nach der Aktualisierung der Zeitzonendatei

Übersicht über Oracle-Zeitzonendateien

Eine Zeitzonendatei von Oracle Database speichert die folgenden Informationen:

- Offset von der koordinierten Weltzeit (UTC)
- Übergangszeiten für die Sommerzeit (DST)
- Abkürzungen für Standardzeit und DST

Oracle Database stellt mehrere Versionen von Zeitzonendateien bereit. Wenn Sie eine Oracle-Datenbank in einer lokalen Umgebung erstellen, wählen Sie die Version der Zeitzonendatei aus. Weitere Informationen finden Sie unter Choosing a Time Zone-File im Oracle Database Globalization Support Guide.

Wenn sich die Regeln für ändernDST, veröffentlicht Oracle neue Zeitzonendateien. Oracle veröffentlicht diese neuen Zeitzonendateien unabhängig vom Zeitplan für vierteljährliche Release-Updates (RUs) und Release-Update-Revisionen (RURs). Die Zeitzonendateien befinden sich auf dem Datenbank-Host im Verzeichnis `$ORACLE_HOME/oracore/zoneinfo/`. Die Namen der Zeitzonendateien verwenden das Format DSTv*`version`*, wie inDSTv35.

Auswirkungen der Zeitzonendatei auf die Datenübertragung

In Oracle Database speichert der `TIMESTAMP WITH TIME ZONE`-Datentyp Zeitstempel- und Zeitzonendaten. Daten mit dem Datentyp `TIMESTAMP WITH TIME ZONE` verwenden die Regeln

in der zugeordneten Version der Zeitzonendatei. Daher sind vorhandene `TIMESTAMP WITH TIME ZONE` Daten betroffen, wenn Sie die Zeitzonendatei aktualisieren.

Probleme können auftreten, wenn Sie Daten zwischen Datenbanken übertragen, die unterschiedliche Versionen der Zeitzonendatei verwenden. Wenn Sie beispielsweise Daten aus einer Quelldatenbank mit einer höheren Zeitzonendateiversion als der Zieldatenbank importieren, gibt die Datenbank den `ORA-39405` Fehler aus. Zuvor mussten Sie diesen Fehler mit einer der folgenden Techniken umgehen:

- Erstellen Sie eine RDS Oracle-DB-Instance mit der gewünschten Zeitzonendatei, exportieren Sie Daten aus Ihrer Quelldatenbank und importieren Sie sie dann in die neue Datenbank.
- Verwenden Sie AWS DMS unsere logische Replikation, um Ihre Daten zu migrieren.

Automatische Updates mit der AUTOUPGRADE Option TIMEZONE FILE _ _

Wenn die an Ihre RDS Oracle-DB-Instance angehängte Optionsgruppe die `TIMEZONE_FILE_AUTOUPGRADE` Option enthält, werden Ihre Zeitzonendateien automatisch RDS aktualisiert. Indem Sie sicherstellen, dass Ihre Oracle-Datenbanken dieselbe Zeitzonen-Dateiversion verwenden, vermeiden Sie zeitaufwändige manuelle Verfahren, wenn Sie Daten zwischen verschiedenen Umgebungen verschieben. Die `TIMEZONE_FILE_AUTOUPGRADE` Option wird sowohl für Container-Datenbanken (CDBs) als auch für Nicht-Datenbanken unterstütztCDBs.

Wenn Sie die Option `TIMEZONE_FILE_AUTOUPGRADE` Ihrer Optionsgruppe hinzufügen, können Sie auswählen, ob die Option sofort oder während des Wartungsfensters hinzugefügt werden soll. Nachdem Ihre DB-Instance die neue Option angewendet hat, wird RDS geprüft, ob sie eine neuere installieren kann DSTv*`version`* file. Das Ziel DSTv*`version`* hängt von folgenden Faktoren ab:

- Die Engine-Nebenversion, die Ihre DB-Instance derzeit ausführt
- Die Engine-Nebenversion, auf die Sie Ihre DB-Instance aktualisieren möchten

Beispielsweise könnte Ihre aktuelle Zeitzonen-Dateiversion seinDSTv33. Wenn das Update auf Ihre Optionsgruppe RDS angewendet wird, DSTv34 wird möglicherweise festgestellt, dass es derzeit auf Ihrem DB-Instance-Dateisystem verfügbar ist. RDSaktualisiert dann Ihre Zeitzonendatei auf DSTv34 automatisch.

Die verfügbaren DST Versionen in den unterstützten RDS Versionsupdates finden Sie in den Patches in den Versionshinweisen für Amazon Relational Database Service (AmazonRDS) for Oracle. Zum

Beispiel listet Version 19.0.0.0.ru-2022-10.rur-2022-10.r1 Patch 34533061: - - 022C auf. RDBMS DSTV39 UPDATE TZDATA2

Strategien zum Aktualisieren Ihrer Zeitzonendatei

Das Aktualisieren Ihrer DB-Engine und das Hinzufügen der `TIMEZONE_FILE_AUTOUPGRADE` Option zu einer Optionsgruppe sind separate Vorgänge. Das Hinzufügen der `TIMEZONE_FILE_AUTOUPGRADE` Option initiiert die Aktualisierung Ihrer Zeitzonendatei, falls eine aktuellere verfügbar ist. Sie führen die folgenden Befehle (es werden nur relevante Optionen angezeigt) entweder sofort oder im nächsten Wartungsfenster aus:

- Aktualisieren Sie Ihre DB-Engine nur mit dem folgenden RDS CLI Befehl:

```
modify-db-instance --engine-version name ...
```

- Fügen Sie die `TIMEZONE_FILE_AUTOUPGRADE` Option nur mit dem folgenden CLI Befehl hinzu:

```
add-option-to-option-group --option-group-name name --options
 OptionName=TIMEZONE_FILE_AUTOUPGRADE ...
```

- Aktualisieren Sie Ihre DB-Engine und fügen Sie Ihrer Instance mit dem folgenden CLI Befehl eine neue Optionsgruppe hinzu:

```
modify-db-instance --engine-version name --option-group-name name ...
```

Ihre Aktualisierungsstrategie hängt davon ab, ob Sie Ihre Datenbank und Ihre Zeitzonendatei zusammen aktualisieren oder nur einen dieser Vorgänge ausführen möchten. Denken Sie daran, dass, wenn Sie Ihre Optionsgruppe aktualisieren und dann Ihre DB-Engine in separaten API Vorgängen aktualisieren, es möglich ist, dass beim Upgrade Ihrer DB-Engine gerade eine Aktualisierung der Zeitzonendatei ausgeführt wird.

Die Beispiele in diesem Abschnitt setzen Folgendes voraus:

- Sie haben noch nichts `TIMEZONE_FILE_AUTOUPGRADE` zu der Optionsgruppe hinzugefügt, die derzeit mit Ihrer DB-Instance verknüpft ist.
- Ihre DB-Instance verwendet die Datenbankversion 19.0.0.0.ru-2019-07.rur-2019-07.r1 und die Zeitzonendatei. DSTv33
- Ihr DB-Instance-Dateisystem enthält eine Datei. DSTv34

- Das Versionsupdate 19.0.0.0.ru-2022-10.rur-2022-10.r1 beinhaltet. DSTv35

Sie können für die Aktualisierung Ihrer Zeitzonendatei die folgenden Strategien verfolgen.

Themen

- Aktualisieren der Zeitzonendatei ohne Engine-Upgrade
- Durchführen eines Upgrades der Zeitzonendatei und der DB-Engine-Version
- Durchführen eines Upgrades Ihrer DB-Engine-Version, ohne die Zeitzonendatei zu aktualisieren

Aktualisieren der Zeitzonendatei ohne Engine-Upgrade

In diesem Szenario verwendet DSTv33 Ihre Datenbank das Dateisystem Ihrer DB-Instance, ist aber auf diesem verfügbar. DSTv34 Sie möchten die von Ihrer DB-Instance verwendete Zeitzonendatei von DSTv33 bis aktualisierenDSTv34, aber Sie möchten Ihre Engine nicht auf eine neue Nebenversion aktualisieren, die Folgendes beinhaltet:DSTv35.

Fügen Sie in einem `add-option-to-option-group` Befehl der Optionsgruppe `TIMEZONE_FILE_AUTOUPGRADE` hinzu, die von Ihrer DB-Instance verwendet wird. Geben Sie an, ob die Option sofort hinzugefügt oder in das Wartungsfenster verschoben werden soll. RDSFührt nach dem Anwenden der `TIMEZONE_FILE_AUTOUPGRADE` Option Folgendes aus:

1. Sucht nach einer neuen DST Version.
2. Stellt fest, DSTv34 dass diese im Dateisystem verfügbar ist.
3. Aktualisiert die Zeitzonendatei sofort.

Durchführen eines Upgrades der Zeitzonendatei und der DB-Engine-Version

In diesem Szenario verwendet Ihre Datenbank das Dateisystem Ihrer DB-InstanceDSTv33, DSTv34 ist aber in diesem verfügbar. Sie möchten Ihre DB-Engine auf die Nebenversion 19.0.0.0.ru-2022-10.rur-2022-10.r1 aktualisieren, die diese beinhaltetDSTv35, und Ihre Zeitzonendatei während des Engine-Upgrades auf diese aktualisieren. DSTv35 Ihr Ziel ist es also, Ihre Zeitzonendateien direkt zu überspringen und zu aktualisieren. DSTv34 DSTv35

Um die Engine und die Zeitzonendatei zusammen zu aktualisieren, führen Sie es `modify-db-instance` mit den `--engine-version` Optionen `--option-group-name` und aus. Sie können den Befehl sofort ausführen oder ihn auf das Wartungsfenster verschieben. `In --option-group-`

name, geben Sie eine Optionsgruppe an, die die TIMEZONE_FILE_AUTOUPGRADE Option enthält. Beispielsweise:

```
aws rds modify-db-instance
    --db-instance-identifier my-instance \
    --engine-version new-version \
    ----option-group-name og-with-timezone-file-autoupgrade \
    --apply-immediately
```

RDSbeginnt mit der Aktualisierung Ihrer Engine auf 19.0.0.0.ru-2022-10.rur-2022-10.r1. Sucht nach der Anwendung der TIMEZONE_FILE_AUTOUPGRADE Option nach einer neuen DST Version, stellt fest, dass diese in RDS 19.0.0.0.ru-2022-10.rur-2022-10.r1 verfügbar DSTv35 ist, und startet sofort das Update für. DSTv35

Um Ihre Engine sofort zu aktualisieren und anschließend Ihre Zeitzonendatei zu aktualisieren, führen Sie die folgenden Schritte nacheinander aus:

1. Aktualisieren Sie Ihre DB-Engine nur mit dem folgenden CLI Befehl:

   ```
   aws rds modify-db-instance \
       --db-instance-identifier my-instance \
       --engine-version new-version \
       --apply-immediately
   ```

2. Fügen Sie die TIMEZONE_FILE_AUTOUPGRADE Option mit dem folgenden CLI Befehl der Optionsgruppe hinzu, die an Ihre Instance angehängt ist:

   ```
   aws rds add-option-to-option-group \
       --option-group-name og-in-use-by-your-instance \
       --options OptionName=TIMEZONE_FILE_AUTOUPGRADE \
       --apply-immediately
   ```

Durchführen eines Upgrades Ihrer DB-Engine-Version, ohne die Zeitzonendatei zu aktualisieren

In diesem Szenario verwendet Ihre Datenbank das Dateisystem Ihrer DB-InstanceDSTv33, DSTv34 ist aber in diesem verfügbar. Sie möchten Ihre DB-Engine auf Version 19.0.0.0.ru-2022-10.rur-2022-10.r1 aktualisieren, die die Zeitzonendatei beinhaltet, aber beibehält. DSTv35 DSTv33 Sie können sich aus den folgenden Gründen für diese Strategie entscheiden:

- Ihre Daten verwenden nicht den Datentyp TIMESTAMP WITH TIME ZONE.

- Ihre Daten verwenden den Datentyp `TIMESTAMP WITH TIME ZONE`, aber Ihre Daten sind nicht von den Zeitzonenänderungen betroffen.
- Sie möchten die Aktualisierung der Zeitzonendatei verschieben, da Sie die zusätzliche Ausfallzeit nicht tolerieren können.

Ihre Strategie hängt davon ab, welche der folgenden Möglichkeiten zutrifft:

- Ihre DB-Instance ist keiner Optionsgruppe zugeordnet, die `TIMEZONE_FILE_AUTOUPGRADE` umfasst. Geben Sie in Ihrem `modify-db-instance` Befehl keine neue Optionsgruppe an, damit Ihre Zeitzonendatei nicht aktualisiert wird. RDS
- Ihre DB-Instance ist derzeit einer Optionsgruppe zugeordnet, die Folgendes umfasst`TIMEZONE_FILE_AUTOUPGRADE`. Ordnen Sie Ihre DB-Instance innerhalb eines einzigen `modify-db-instance` Befehls einer Optionsgruppe zu, die Ihre DB-Engine nicht enthält, `TIMEZONE_FILE_AUTOUPGRADE` und aktualisieren Sie sie auf 19.0.0.0.ru-2022-10.rur-2022-10.r1.

Ausfallzeiten während der Aktualisierung der Zeitzonendatei

Wenn Ihre Zeitzonendatei RDS aktualisiert wird, `TIMESTAMP WITH TIME ZONE` können sich vorhandene Daten, die verwendet werden, ändern. In diesem Fall geht es in erster Linie um Ausfallzeiten.

> **Warning**
>
> Wenn Sie die Option `TIMEZONE_FILE_AUTOUPGRADE` hinzufügen, können längere Ausfallzeiten bei Ihrem Engine-Upgrade auftreten. Das Aktualisieren von Zeitzonendaten für eine große Datenbank kann Stunden oder sogar Tage dauern.

Die Länge des Updates der Zeitzonendatei hängt von Faktoren wie den folgenden ab:

- Der Menge der Daten `TIMESTAMP WITH TIME ZONE` in Ihrer Datenbank
- Die DB-Instance-Konfiguration
- Der DB-Instance-Klasse
- Der Speicherkonfiguration
- Der Datenbankkonfiguration
- Den Einstellungen für Datenbankparameter

Wenn Sie die folgenden Schritte ausführen, können zusätzliche Ausfallzeiten die Folge sein:

- Hinzufügen der Option zur Optionsgruppe, wenn die DB-Instance eine veraltete Zeitzonendatei verwendet
- Aktualisieren des Oracle-Datenbankmoduls, wenn die neue Engine-Version eine neue Version der Zeitzonendatei enthält

> **Note**
>
> Während der Aktualisierung der Zeitzonendatei RDS für Oracle-Aufrufe`PURGE DBA_RECYCLEBIN`.

Vorbereiten der Aktualisierung der Zeitzonendatei

Ein Upgrade der Zeitzonendatei besteht aus zwei separaten Phasen: Vorbereiten und Aktualisieren. Er ist zwar nicht zwingend erforderlich, wir empfehlen Ihnen jedoch nachdrücklich, den Vorbereitungsschritt auszuführen. In diesem Schritt erfahren Sie, welche Daten von der Ausführung des PL/ SQL -Verfahrens `DBMS_DST.FIND_AFFECTED_TABLES` betroffen sind. Weitere Informationen zum Vorbereitungsfenster finden Sie unter Aktualisieren der Zeitzonendatei und des Zeitstempels mit Zeitzonendaten in der Oracle Database-Dokumentation.

So bereiten Sie die Aktualisierung der Zeitzonendatei vor

1. Stellen Sie über einen SQL Client eine Connect zu Ihrer Oracle-Datenbank her.
2. Bestimmen Sie die aktuelle Version der verwendeten Zeitzonendatei.

   ```
   SELECT * FROM V$TIMEZONE_FILE;
   ```

3. Bestimmen Sie die neueste Version der Zeitzonendatei, die auf Ihrer DB-Instance verfügbar ist.

   ```
   SELECT DBMS_DST.GET_LATEST_TIMEZONE_VERSION FROM DUAL;
   ```

4. Bestimmen Sie die Gesamtgröße von Tabellen, die Spalten vom Typ `TIMESTAMP WITH LOCAL TIME ZONE` oder `TIMESTAMP WITH TIME ZONE` enthalten.

   ```
   SELECT SUM(BYTES)/1024/1024/1024 "Total_size_w_TSTZ_columns_GB"
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE 'TABLE%'
   ```

```
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE');
```

5. Bestimmen Sie die Namen und Größen von Segmenten, die Spalten vom Typ `TIMESTAMP WITH LOCAL TIME ZONE` oder `TIMESTAMP WITH TIME ZONE` enthalten.

```
SELECT OWNER, SEGMENT_NAME, SUM(BYTES)/1024/1024/1024
 "SEGMENT_SIZE_W_TSTZ_COLUMNS_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE')
GROUP BY OWNER, SEGMENT_NAME;
```

6. Starten Sie den Vorbereitungsschritt.

   - Die Prozedur `DBMS_DST.CREATE_AFFECTED_TABLE` erstellt eine Tabelle zum Speichern aller betroffenen Daten. Sie übergeben den Namen dieser Tabelle an das Verfahren `DBMS_DST.FIND_AFFECTED_TABLES`. Weitere Informationen finden Sie unter CREATE_ AFFECTED _ TABLE Procedure in der Oracle Database-Dokumentation.
   - Mit dem Verfahren `CREATE_ERROR_TABLE` wird eine Tabelle zum Protokollieren von Fehlern erstellt. Weitere Informationen finden Sie unter CREATE_ ERROR _ TABLE Procedure in der Oracle Database-Dokumentation.

   Im folgenden Beispiel werden die betroffenen Daten und Fehlertabellen erstellt und alle betroffenen Tabellen gefunden.

```
EXEC DBMS_DST.CREATE_ERROR_TABLE('my_error_table')
EXEC DBMS_DST.CREATE_AFFECTED_TABLE('my_affected_table')

EXEC DBMS_DST.BEGIN_PREPARE(new_version);
EXEC DBMS_DST.FIND_AFFECTED_TABLES('my_affected_table', TRUE, 'my_error_table');
EXEC DBMS_DST.END_PREPARE;

SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

7. Fragen Sie die betroffenen und Fehlertabellen ab.

```
SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

Hinzufügen der Option zur automatischen Aktualisierung der Zeitzonendatei

Wenn Sie die Option einer Optionsgruppe hinzufügen, befindet sich die Optionsgruppe in einem der folgenden Zustände:

- Eine vorhandene Optionsgruppe ist derzeit mindestens an eine DB-Instance angefügt. Wenn Sie die Option hinzufügen, werden alle DB-Instances, die diese Optionsgruppe verwenden, automatisch neu gestartet. Dies führt zu einem kurzen Ausfall.
- Eine vorhandene Optionsgruppe ist an keine DB-Instance angefügt. Sie planen, die Option hinzuzufügen und die vorhandene Optionsgruppe dann vorhandenen DB-Instances oder einer neuen DB-Instance zuzuordnen.
- Sie erstellen eine neue Optionsgruppe und fügen die Option hinzu. Sie planen, die neue Optionsgruppe vorhandenen DB-Instances oder einer neuen DB-Instance zuzuordnen.

Konsole

So fügen Sie einer DB-Instance die Option zur automatischen Aktualisierung hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Option groups (Optionsgruppen) aus.
3. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie für Engine die Oracle Database Edition für Ihre DB-Instance aus.

   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.

4. Wählen Sie die Optionsgruppe aus, die Sie ändern möchten, und wählen Sie dann Add option (Option hinzufügen).
5. Führen Sie im Fenster Add option (Option hinzufügen) die folgenden Schritte aus:
   a. Wählen Sie TIMEZONE_ FILE _ AUTOUPGRADE.
   b. Um die Option in allen zugeordneten DB-Instanzen zu aktivieren, sobald Sie sie hinzufügen, wählen Sie für Apply Immediately (Direkt anwenden) Yes (Ja). Wenn Sie No (Nein) (Standard) wählen, wird die Option während des nächsten Wartungsfensters in jeder zugeordneten DB-Instanz aktiviert.
6. Wenn die Einstellungen Ihren Wünschen entsprechen, wählen Sie Add option (Option hinzufügen) aus.

AWS CLI

Im folgenden Beispiel wird der Befehl AWS CLI add-option-to-option-group verwendet, um die `TIMEZONE_FILE_AUTOUPGRADE` Option einer Optionsgruppe namens `myoptiongroup` hinzuzufügen.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" ^
    --apply-immediately
```

Überprüfen Ihrer Daten nach der Aktualisierung der Zeitzonendatei

Wir empfehlen, dass Sie Ihre Daten überprüfen, nachdem Sie die Zeitzonendatei aktualisiert haben. Während des Vorbereitungsschritts erstellt RDS for Oracle automatisch die folgenden Tabellen:

- `rdsadmin.rds_dst_affected_tables` – Listet die Tabellen auf, die von der Aktualisierung betroffene Daten enthalten

- `rdsadmin.rds_dst_error_table` – Listet die Fehler auf, die während der Aktualisierung generiert wurden

Diese Tabellen hängen nicht von den Tabellen ab, die Sie im Vorbereitungsfenster erstellen. Zum Aufrufen der Ergebnisse der Aktualisierung fragen Sie die Tabellen wie folgt ab.

```
SELECT * FROM rdsadmin.rds_dst_affected_tables;
SELECT * FROM rdsadmin.rds_dst_error_table;
```

Weitere Informationen zum Schema für die betroffenen Daten und Fehlertabellen finden Sie unter FIND_ AFFECTED _ TABLES Procedure in der Oracle-Dokumentation.

Oracle Transparent Data Encryption

Amazon RDS unterstützt Oracle Transparent Data Encryption (TDE), eine Funktion der Oracle Advanced Security-Option, die in der Oracle Enterprise Edition verfügbar ist. Mit dieser Funktion werden Daten vor dem Speichern automatisch verschlüsselt und beim Abruf aus dem Speicher automatisch entschlüsselt. Diese Option wird nur für das Modell Bring Your Own License (BYOL) unterstützt.

TDEist nützlich in Szenarien, in denen Sie vertrauliche Daten verschlüsseln müssen, falls Datendateien und Backups von Dritten abgerufen werden. TDEist auch nützlich, wenn Sie sicherheitsrelevante Vorschriften einhalten müssen.

Eine ausführliche Erklärung zu TDE Oracle Database würde den Rahmen dieses Handbuchs sprengen. Informationen finden Sie in den folgenden Oracle Database-Ressourcen:

- Einführung in die transparente Datenverschlüsselung in der Oracle-Datenbankdokumentation
- Die erweiterte Sicherheit von Oracle in der Dokumentation zu Oracle Database
- Oracle Advanced Security — Bewährte Methoden zur transparenten Datenverschlüsselung — ein Whitepaper von Oracle

Weitere Informationen zur Verwendung von TDE with RDS für Oracle finden Sie in den folgenden Blogs:

- Verschlüsselungsoptionen für Oracle-Datenbanken bei Amazon RDS
- Migrieren Sie eine kontoübergreifende TDE Amazon RDS for Oracle-DB-Instance mit reduzierter Ausfallzeit mithilfe von AWS DMS

TDEVerschlüsselungsmodi

Oracle Transparent Data Encryption unterstützt zwei Verschlüsselungsmodi: TDE Tablespace-Verschlüsselung und TDE Spaltenverschlüsselung. TDEDie Tablespace-Verschlüsselung wird verwendet, um ganze Anwendungstabellen zu verschlüsseln. TDEDie Spaltenverschlüsselung wird verwendet, um einzelne Datenelemente zu verschlüsseln, die sensible Daten enthalten. Sie können auch eine hybride Verschlüsselungslösung anwenden, die sowohl TDE Tablespace- als auch Spaltenverschlüsselung verwendet.

> **Note**
>
> Amazon RDS verwaltet das Oracle Wallet und den TDE Masterschlüssel für die DB-Instance. Sie müssen den Verschlüsselungsschlüssel nicht durch den Befehl festlege `ALTER SYSTEM set encryption key`.

Nachdem Sie die TDE Option aktiviert haben, können Sie den Status der Oracle Wallet mit dem folgenden Befehl überprüfen:

```
SELECT * FROM v$encryption_wallet;
```

Sie können mit folgendem Befehl einen verschlüsselten Tabellenraum erzeugen:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

Um den Verschlüsselungsalgorithmus anzugeben, verwenden Sie den folgenden Befehl:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

Die vorherigen Anweisungen zum Verschlüsseln eines Tablespaces entsprechen denen, die Sie in einer lokalen Oracle-Datenbank verwenden würden.

Einschränkungen für die Option TDE

Die TDE Option ist permanent und dauerhaft. Nachdem Sie Ihre DB-Instance einer Optionsgruppe zugeordnet haben, für die die TDE Option aktiviert ist, können Sie die folgenden Aktionen nicht mehr ausführen:

- Deaktivieren Sie die TDE Option in der aktuell verknüpften Optionsgruppe.
- Ordnen Sie Ihre DB-Instance einer anderen Optionsgruppe zu, die die TDE Option nicht enthält.
- Teilen Sie einen DB-Snapshot, der die TDE Option verwendet. Weitere Informationen zum Freigeben von DB-Snapshots finden Sie unter Einen DB-Snapshot für Amazon teilen RDS.

Weitere Hinweise zu persistenten und permanenten Optionen finden Sie unterPersistente und permanente Optionen.

Feststellen, ob Ihre DB-Instance TDE

Möglicherweise möchten Sie ermitteln, ob Ihre DB-Instance einer Optionsgruppe zugeordnet ist, für die die TDE Option aktiviert ist. Um die Optionsgruppe anzuzeigen, der eine DB-Instance zugeordnet ist, verwenden Sie die RDS Konsole, den describe-db-instance AWS CLI Befehl oder die API Operation escribeDBInstancesD.

Die TDE Option wird hinzugefügt

Gehen Sie wie folgt vor, um die TDE Option zu Ihrer DB-Instance hinzuzufügen:

1. (Empfohlen) Erstellen Sie einen Snapshot Ihrer DB-Instance.
2. Führen Sie eine der folgenden Aufgaben aus:
   - Erstellen Sie eine völlig neue Optionsgruppe. Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
   - Kopieren Sie eine bestehende Optionsgruppe mit dem AWS CLI oderAPI. Weitere Informationen finden Sie unter Kopieren einer Optionsgruppe.
   - Verwenden Sie eine vorhandene, nicht standardmäßige Optionsgruppe wieder. Es hat sich bewährt, eine Optionsgruppe zu verwenden, die derzeit keinen DB-Instances oder Snapshots zugeordnet ist.
3. Fügen Sie die neue Option der Optionsgruppe aus dem vorherigen Schritt hinzu.
4. Wenn für die Optionsgruppe, die derzeit mit Ihrer DB-Instance verknüpft ist, Optionen aktiviert sind, fügen Sie diese Optionen zu Ihrer neuen Optionsgruppe hinzu. Diese Strategie verhindert, dass die vorhandenen Optionen deinstalliert werden, während die neue Option aktiviert wird.
5. Fügen Sie die neue Optionsgruppe zu Ihrer DB-Instance hinzu.

Konsole

Um die TDE Option einer Optionsgruppe hinzuzufügen und sie Ihrer DB-Instance zuzuordnen

1. Wählen Sie in der RDS Konsole Optionsgruppen aus.
2. Wählen Sie den Namen der Optionsgruppe, zu der Sie die Option hinzufügen möchten.
3. Wählen Sie Add option (Option hinzufügen).
4. Wählen Sie unter Optionsname die Option aus TDE, und konfigurieren Sie dann die Optionseinstellungen.
5. Wählen Sie Add option (Option hinzufügen).

> **Important**
>
> Wenn Sie die TDEOption zu einer Optionsgruppe hinzufügen, die derzeit an eine oder mehrere DB-Instances angehängt ist, kommt es zu einem kurzen Ausfall, während alle DB-Instances automatisch neu gestartet werden.

Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.

6. Ordnen Sie die Optionsgruppe einer neuen oder vorhandenen DB-Instance zu:

   - Weisen Sie bei einer neuen DB-Instance die Optionsgruppe beim Starten der Instance zu. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
   - Weisen Sie bei einer bestehenden DB-Instance die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Die DB-Instance wird im Rahmen dieses Vorgangs nicht neu gestartet. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

AWS CLI

Im folgenden Beispiel verwenden Sie den Befehl AWS CLI add-option-to-option-group, um die TDE Option einer Optionsgruppe mit dem Namen `myoptiongroup` hinzuzufügen. Weitere Informationen finden Sie unter Erste Schritte: Flink 1.13.2.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=TDE" \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=TDE" ^
    --apply-immediately
```

Kopieren Sie Ihre Daten in eine DB-Instance, die die TDE Option nicht enthält

Sie können die TDE Option nicht aus einer DB-Instance entfernen oder sie einer Optionsgruppe zuordnen, die die TDE Option nicht enthält. Gehen Sie wie folgt vor, um Ihre Daten auf eine Instance zu migrieren, die die TDE Option nicht enthält:

1. Entschlüsseln Sie die Daten auf Ihrer DB-Instance.
2. Kopieren Sie die Daten in eine neue DB-Instance, die keiner `TDE` aktivierten Optionsgruppe zugeordnet ist.
3. Löschen Sie Ihre ursprüngliche DB-Instance.

Sie können für die neue Instance denselben Namen wie für die vorherige DB-Instance verwenden.

Überlegungen bei der Verwendung TDE mit Oracle Data Pump

Sie können Oracle Data Pump verwenden, um verschlüsselte Dump-Dateien zu importieren oder zu exportieren. Amazon RDS unterstützt den Passwortverschlüsselungsmodus (`ENCRYPTION_MODE=PASSWORD`) für Oracle Data Pump. Amazon unterstützt RDS keinen transparenten Verschlüsselungsmodus (`ENCRYPTION_MODE=TRANSPARENT`) für Oracle Data Pump. Weitere Informationen finden Sie unter Importieren mit Oracle Data Pump.

Orakel UTL _ MAIL

Amazon RDS unterstützt Oracle UTL _ MAIL durch die Verwendung der MAIL Option UTL _ und SMTP Server. Sie können E-Mails direkt aus Ihrer Datenbank senden, indem Sie das MAIL Paket UTL _ verwenden. Amazon RDS unterstützt UTL _ MAIL für die folgenden Versionen von Oracle:

- Oracle Database 21c (21.0.0.0), alle Versionen
- Oracle Database 19c (19.0.0.0), alle Versionen

Im Folgenden sind einige Einschränkungen bei der Verwendung von UTL _ aufgeführtMAIL:

- UTL_ MAIL unterstützt Transport Layer Security (TLS) nicht und daher werden E-Mails nicht verschlüsselt.

  Um eine sichere Verbindung zu SSL TLS Remote-/Ressourcen herzustellen, indem Sie benutzerdefinierte Oracle-Wallets erstellen und hochladen, folgen Sie den Anweisungen unter. Konfigurieren des UTL_HTTP-Zugriffs mit Zertifikaten und einer Oracle Wallet

  Die spezifischen Zertifikate, die für Ihr Wallet benötigt werden, variieren je nach Service. AWS Dienste finden Sie in der Regel im Amazon Trust Services-Repository.
- UTL_ MAIL unterstützt keine Authentifizierung mit SMTP Servern.
- Sie können nur einen Anhang pro E-Mail senden.
- Sie können nur Anhänge bis zu einer Größe von 32 K senden.
- Sie können nur Zeichenkodierungen ASCII und den Extended Binary Coded Decimal Interchange Code (EBCDIC) verwenden.
- SMTPPort (25) wird auf der Grundlage der Richtlinien des Besitzers der elastic network interface gedrosselt.

Wenn Sie UTL _ aktivierenMAIL, wird nur dem Master-Benutzer für Ihre DB-Instance die Ausführungsberechtigung gewährt. Falls erforderlich, kann der Master-Benutzer anderen Benutzern die Ausführungsberechtigung gewähren, sodass diese UTL _ verwenden könnenMAIL.

> **Important**
>
> Wir empfehlen, dass Sie die in Oracle integrierte Auditing-Funktion aktivieren, um die Verwendung von UTL MAIL _-Prozeduren nachzuverfolgen.

Voraussetzungen für Oracle UTL _ MAIL

Im Folgenden sind die Voraussetzungen für die Verwendung von Oracle UTL _ aufgeführtMAIL:

- Ein oder mehrere SMTP Server und die entsprechenden IP-Adressen oder öffentlichen oder privaten Domain Name Server (DNS) -Namen. Weitere Hinweise zu privaten DNS Namen, die über einen benutzerdefinierten DNS Server aufgelöst werden, finden Sie unterEinen benutzerdefinierten DNS Server einrichten.

Hinzufügen der MAIL Oracle-Option UTL _

Das allgemeine Verfahren zum Hinzufügen der MAIL Option Oracle UTL _ zu einer DB-Instance ist wie folgt:

1. Erstellen Sie eine neue Optionsgruppe oder kopieren oder ändern Sie eine bestehende Optionsgruppe.
2. Hinzufügen der Option zur Optionsgruppe.
3. Ordnen Sie die Optionsgruppe der DB-Instance zu.

Nachdem Sie die MAIL Option UTL _ hinzugefügt haben, ist UTL _ aktiv, sobald die Optionsgruppe aktiv MAIL ist.

Um die MAIL Option UTL _ zu einer DB-Instance hinzuzufügen

1. Bestimmen Sie die zu verwendende Optionsgruppe. Sie können eine Optionsgruppe erstellen oder eine bestehende Optionsgruppe verwenden. Wenn Sie eine bestehende Optionsgruppe verwenden möchten, fahren Sie mit dem nächsten Schritt fort. Erstellen Sie andernfalls eine benutzerdefinierte DB-Optionsgruppe mit folgenden Einstellungen:

   a. Wählen Sie im Feld Engine die Oracle-Edition aus, die Sie verwenden möchten.
   b. Wählen Sie für Major Engine Version (Engine-Hauptversion) die Version Ihrer DB-Instance aus.

   Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe.
2. Fügen Sie die MAIL Option UTL_ zur Optionsgruppe hinzu. Weitere Informationen über das Hinzufügen von Optionen finden Sie unter Hinzufügen einer Option zu einer Optionsgruppe.
3. Ordnen Sie die Optionsgruppe einer neuen oder bestehenden DB-Instance zu:

- Einer neuen DB-Instance wird die Optionsgruppe beim Starten der Instance zugewiesen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Bei einer bestehenden DB-Instance weisen Sie die Optionsgruppe zu, indem Sie die Instance ändern und die neue Optionsgruppe anhängen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Verwenden von Oracle UTL _ MAIL

Nachdem Sie die MAIL Option UTL _ aktiviert haben, müssen Sie den SMTP Server konfigurieren, bevor Sie ihn verwenden können.

Sie konfigurieren den SMTP Server, indem Sie den SERVER Parameter SMTP OUT _ _ auf eine gültige IP-Adresse oder einen öffentlichen DNS Namen setzen. Für den SERVER Parameter SMTP OUT _ _ können Sie eine durch Kommas getrennte Liste der Adressen mehrerer Server angeben. Wenn der erste Server nicht verfügbar ist, MAIL versucht UTL _ den nächsten Server usw.

Sie können den Standardwert SMTP _ OUT _ SERVER für eine DB-Instance festlegen, indem Sie eine DB-Parametergruppe verwenden. Sie können den SERVER Parameter SMTP _ OUT _ für eine Sitzung festlegen, indem Sie den folgenden Code in Ihrer Datenbank auf Ihrer DB-Instance ausführen.

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

Nachdem die MAIL Option UTL _ aktiviert und Ihr SMTP _ OUT _ konfiguriert SERVER ist, können Sie mithilfe der folgenden `SEND` Prozedur E-Mails senden. Weitere Informationen finden Sie unter UTL_ MAIL in der Oracle-Dokumentation.

Die MAIL Oracle-Option UTL _ wird entfernt

Sie können Oracle UTL _ MAIL aus einer DB-Instance entfernen.

Gehen Sie wie folgt vor, um UTL _ MAIL aus einer DB-Instance zu entfernen:

- Um UTL _ MAIL aus mehreren DB-Instances zu entfernen, entfernen Sie die MAIL Option UTL _ aus der Optionsgruppe, zu der sie gehören. Diese Änderung wirkt sich auf alle DB-Instances aus, welche die betreffende Optionsgruppe verwenden. Weitere Informationen finden Sie unter Entfernen einer Option aus einer Optionsgruppe.

- Um UTL _ MAIL aus einer einzelnen DB-Instance zu entfernen, ändern Sie die DB-Instance und geben Sie eine andere Optionsgruppe an, die die MAIL Option UTL _ nicht enthält. Sie können die (leere) Standardoptionsgruppe oder eine andere benutzerdefinierte Optionsgruppe angeben. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Fehlerbehebung

Die folgenden Probleme können auftreten, wenn Sie UTL _ MAIL mit Amazon verwendenRDS.

- Ablehnung. SMTPPort (25) wird auf der Grundlage der Richtlinien des Besitzers der elastic network interface gedrosselt. Wenn Sie mit UTL _ erfolgreich E-Mails senden können und der Fehler angezeigt `ORA-29278: SMTP transient error: 421 Service not available` wirdMAIL, werden Sie möglicherweise gedrosselt. Falls die E-Mail-Zustellung abgelehnt wird, empfehlen wir, einen Backoff-Algorithmus zu implementieren. Weitere Informationen zu Backoff-Algorithmen finden Sie unter Wiederholen bei Fehlern und Exponentielles Backoff in AWS und Umgang mit dem Fehler „Drosselung – Maximale Senderate überschritten“.

  Sie können eine Aufhebung der Drosselung anfordern. Weitere Informationen finden Sie unter Wie entferne ich die Drosselung an Port 25 aus meiner EC2 Instance? .

Oracle XML DB

Oracle XML DB unterstützt natives XML auf der DB-Instance. Mit XML DB können Sie strukturierte oder unstrukturierte XML- und relationale Daten speichern und abrufen. Der XML-DB-Protokollserver wird auf RDS for Oracle nicht unterstützt.

XML DB ist auf Oracle Database 12c und höher vorinstalliert. Daher müssen Sie keine Optionsgruppe verwenden, um XML DB explizit als zusätzliche Funktion zu installieren.

Informationen zur Konfiguration und Verwendung von XML DB finden Sie im Oracle XML DB Developer's Guide in der Oracle Database-Dokumentation.

Aktualisieren der DB-Engine von RDS für Oracle

Wenn Amazon RDS eine neue Version von Oracle Database unterstützt, können Sie Ihre DB-Instances auf die neue Version aktualisieren. Weitere Informationen zu unterstützten Oracle-Versionen in Amazon RDS finden Sie in den Versionshinweisen zu Amazon RDS for Oracle.

> **Important**
>
> RDS für Oracle Database 11g, 12c und 18c werden nicht mehr unterstützt. Wenn Sie Oracle Database 11g-, 12c oder 18c-Snapshots verwalten, können Sie diese auf eine spätere Version upgraden. Weitere Informationen finden Sie unter Aktualisieren eines Oracle-DB-Snapshots.

Themen

- Übersicht über RDS für Oracle DB Engine-Upgrades
- Upgrades der Oracle-Hauptversion
- Oracle-Unterversion-Upgrades
- Überlegungen zu Oracle-Datenbank-Upgrades
- Testen eines Oracle DB-Upgrades
- Aktualisierung der Version einer RDS für Oracle-DB-Instance
- Aktualisieren eines Oracle-DB-Snapshots

Übersicht über RDS für Oracle DB Engine-Upgrades

Machen Sie sich vor dem Upgrade Ihrer DB-Instance von RDS für Oracle mit den folgenden Konzepten vertraut.

Themen

- Aktualisierungen von Haupt- und Nebenversionen
- Support-Daten und obligatorische Upgrades für RDS for Oracle
- Oracle-Engine-Versionsverwaltung
- Automatische Snapshots während Engine-Upgrades

- Oracle-Upgrades in einer Multi-AZ-Bereitstellung
- Oracle-Upgrades von Read Replicas (Lesereplikaten)

Aktualisierungen von Haupt- und Nebenversionen

Hauptversionen von Oracle Database werden alle 1-2 Jahre veröffentlicht. Oracle Database 19c und Oracle Database 21c sind Hauptversionen.

Jedes Quartal veröffentlicht RDS for Oracle neue kleinere Engine-Versionen für jede unterstützte Haupt-Engine. Eine Version der Engine Release Update (RU) enthält Fehlerkorrekturen von Oracle, indem sie die RU-Patches für das angegebene Quartal enthält. Beispielsweise ist 21.0.0.0.ru-2024-10.rur-2024-10.r1 eine Nebenversion von Oracle Database 21c, die die RU vom Oktober 2024 enthält.

Eine Engine-Version des Spatial Patch Bundle (SPB) enthält RU-Patches und Patches, die für Oracle Spatial spezifisch sind. Beispielsweise ist 19.0.0.0.ru-2025-01.spb-1.r1 eine kleinere Engine-Version, die die RU-Patches in der Engine-Version 19.0.0.0.ru-2025-01.rur-2025-01.r1 sowie Spatial-Patches enthält. In der Regel wird RDS für Oracle SPBs 2—3 Wochen nach der entsprechenden RU veröffentlicht. Eine Erläuterung der Unterschiede zwischen RUs und finden Sie SPBs unterVersionsupdates (RUs) und Spatial Patch Bundles () SPBs. Informationen zu unterstützten RUs und SPBs finden Sie in den Versionshinweisen für Amazon Relational Database Service (Amazon RDS) for Oracle.

RDS für Oracle unterstützt die folgenden Aktualisierungen für eine DB-Instance:

Aktualisierungstyp	Anwendungskompatibilität	Aktualisierungsverfahren	Beispiel-Aktualisierungspfad
Hauptversion	In Hauptversions-Upgrades Änderungen können enthalten sein, die nicht mit vorhandenen Anwendungen kompatibel sind.	Nur Manuell	Von Oracle Database 19c zu Oracle Database 21c
Nebenversion	Ein Nebenversion-Upgrade enthalten nur Änderungen, die abwärtskompatibel mit bestehenden Anwendungen sind.	Automatisch oder manuell	Von 21.0.0.0.ru-2023-07.rur-2022-07.r1 bis 21.0.0.0.ru-2023-10.rur-2022-10.r1

> **Important**
>
> Wenn Sie Ihre DB-Engine aktualisieren, tritt ein Ausfall auf. Die Ausfallzeit hängt von Ihrer Engine-Version und der Größe der DB-Instance ab.
> Testen Sie alle Upgrades sorgfältig, um sicherzustellen, dass Ihre Anwendungen ordnungsgemäß funktionieren, bevor Sie das Upgrade auf Ihre Produktionsdatenbanken anwenden. Weitere Informationen finden Sie unter Testen eines Oracle DB-Upgrades.

Support-Daten und obligatorische Upgrades für RDS for Oracle

Für Datenbankversionen von RDS for Oracle gelten erwartete Support-Daten. Wenn sich eine Haupt- oder Nebenversion einer RDS for Oracle-DB-Engine ihrem end-of-support Datum nähert, beginnt RDS mit obligatorischen Upgrades, die auch als erzwungene Upgrades bezeichnet werden. RDS veröffentlicht die folgenden Informationen:

- Eine Empfehlung für Sie, mit dem manuellen Upgrade von Instances auf veraltete Versionen auf unterstützte Versionen zu beginnen
- Ein Datum, nach dem Sie keine Instanzen mehr für die nicht unterstützten Versionen erstellen können
- Ein Datum, an dem RDS beginnt, Ihre Instances während der Wartungsfenster automatisch auf unterstützte Versionen zu aktualisieren
- Ein Datum, an dem RDS außerhalb der Wartungsfenster automatisch mit dem Upgrade Ihrer Instances auf unterstützte Versionen beginnt

> **Important**
>
> Erzwungene Upgrades können unerwartete Folgen für AWS CloudFormation Stacks haben. Wenn Sie sich darauf verlassen, dass RDS Ihre DB-Instances automatisch aktualisiert, können Probleme mit AWS CloudFormation auftreten.

In diesem Abschnitt werden folgende Themen behandelt:

Themen

- Support-Daten für Hauptversionen von RDS for Oracle

- Support-Daten für Nebenversionen von RDS for Oracle

Support-Daten für Hauptversionen von RDS for Oracle

Die Hauptversionen von RDS für Oracle stehen mindestens bis zum Ende des Supportdatums für die entsprechende Oracle Database-Release-Version zur Verfügung. Sie können die folgenden Daten verwenden, um Ihre Test- und Upgrade-Zyklen zu planen. Diese Daten stellen das früheste Datum dar, an dem ein Upgrade auf eine neuere Version erforderlich sein könnte. Wenn Amazon die Unterstützung für eine RDS-für-Oracle-Version länger als ursprünglich geplant erweitert, planen wir, diese Tabelle zu aktualisieren, um das spätere Datum widerzuspiegeln.

> **Note**
>
> Sie können die Hauptversionen Ihrer Oracle-Datenbanken anzeigen, indem Sie den AWS CLI Befehl describe-db-major-engine-versions ausführen oder den API-Vorgang Describe DBMajor EngineVersions RDS verwenden.

Hauptversion von Oracle Database	Voraussichtliches Datum für das Upgrade auf eine neuere Version
Oracle Database 19c	31. Dezember 2029 mit BYOL Premier Support (keine Gebühren für Extended Support) 31. Dezember 2032 mit BYOL Extended Support (zusätzliche Kosten) oder einer unbegrenzten Lizenzvereinbarung 31. Dezember 2029 inklusive Lizenz (LI)
Oracle Database 21c	31. Juli 2027 (nicht verfügbar für Extended Support)

RDS benachrichtigt Sie mindestens 12 Monate, bevor Sie auf eine neuere Hauptversion aktualisieren müssen. Die Benachrichtigung beschreibt den Upgrade-Prozess, einschließlich des Zeitpunkts wichtiger Meilensteine, der Auswirkungen auf Ihre DB-Instances und der empfohlenen Maßnahmen. Wir empfehlen Ihnen, Ihre Anwendungen gründlich mit neuen Versionen von RDS for Oracle zu testen, bevor Sie Ihre Datenbank auf eine Hauptversion aktualisieren.

Nach dieser Vorankündigungsfrist kann ein automatisches Upgrade auf die nachfolgende Hauptversion auf alle DB-Instances von RDS für Oracle angewendet werden, auf denen noch die ältere Version ausgeführt wird. Falls zutreffend, wird das Upgrade während der geplanten Wartungsfenster gestartet.

Weitere Informationen finden Sie unter Release-Zeitplan der aktuellen Datenbank-Releases in My Oracle Support.

Support-Daten für Nebenversionen von RDS for Oracle

Die folgenden Nebenversionen von Oracle Database 19c nähern sich ihrem end-of-support Datum:

- 19.0.0.0.ru-2019-07.rur-2019-07.r1
- 19.0.0.0.ru-2019-10.rur-2019-10.r1
- 19.0.0.0.ru-2020-01.rur-2020-01.r1

Die folgenden Nebenversionen von Oracle Database 21c nähern sich ihrem Datum: end-of-support

- 21.0.0.0.ru-2022-01.rur-2022-01.r1
- 21.0.0.0.ru-2022-04.rur-2022-04.r1
- 21.0.0.0.ru-2022-07.rur-2022-07.r1
- 21.0.0.0.ru-2022-10.rur-2022-10.r1
- 21,0.0.0.ru-2023-01.rur-2023-01.r1
- 21.0.0.0.ru-2023-01.rur-2023-01.r2

Die folgende Tabelle zeigt einen Zeitplan der obligatorischen Upgrades für Nebenversionen der vorherigen, veralteten Versionen.

Datum	Aktion für veraltete Versionen
07/02/25	Sie können keine DB-Instances mehr erstellen, auf denen veraltete Versionen ausgeführt werden. Aktualisieren Sie Ihre DB-Instances entweder manuell auf das neueste Release-Update (RU) oder warten Sie, bis RDS sie automatisch aktualisiert.

Datum	Aktion für veraltete Versionen
08.01.25 — 15.09.25	RDS Force aktualisiert DB-Instances im geplanten Wartungsfenster auf die neueste RU. RDS aktualisiert auch alle Instances, die aus Snapshots der veralteten Versionen wiederhergestellt wurden.
15.09.25	RDS beginnt, unabhängig vom geplanten Wartungsfenster das Upgrade von DB-Instances auf die neueste RU zu erzwingen. Die erzwungenen Upgrades werden fortgesetzt, bis auf allen Instances unterstützte Nebenversionen ausgeführt werden.

Wenn Sie nicht möchten, dass RDS Ihre Instances, auf denen veraltete Versionen ausgeführt werden, automatisch aktualisiert, gehen Sie wie folgt vor:

- Aktualisieren Sie Ihre Oracle Database-Nebenversionen 19c und 21c manuell auf die neueste RU oder auf eine RU, für die keine obligatorischen Upgrades geplant sind. Sie können beispielsweise 21.0.0.0.ru-2023-01.rur-2023-01.r2 auf 21.0.0.0.ru-2023-04.rur-2023-04.r2 oder 19.0.0.0.ru-2019-07.rur-2019-07.r1 auf 19.0.0.0.ru-2025-01.rur-2025-01.rur-2025-01.r2 aktualisieren.
- Führen Sie ein Upgrade von DB-Snapshots durch, die veraltete Nebenversionen verwenden, und stellen Sie dann die Snapshots wieder her.

Weitere Informationen zu Nebenversionen finden Sie in den Versionshinweisen für Amazon Relational Database Service (Amazon RDS) for Oracle.

Oracle-Engine-Versionsverwaltung

Mit der DB-Engine-Versionsverwaltung steuern Sie, wann und wie die Datenbank-Engine gepatcht und upgegradet wird. Sie erhalten die Flexibilität, die Kompatibilität mit Datenbank-Engine-Patch-Versionen aufrechtzuerhalten. Sie können auch neue Patch-Versionen von RDS für Oracle testen, um sicherzustellen, dass sie effektiv mit Ihrer Anwendung funktionieren, bevor Sie sie in der Produktion bereitstellen. Darüber hinaus aktualisieren Sie die Versionen zu Ihren eigenen Bedingungen und Zeitplänen.

> **Note**
>
> Amazon RDS fasst periodische Oracle-Datenbank-Patches unter Verwendung einer Amazon RDS-spezifischen DB-Engine zusammen. Eine Liste mit den in einer Amazon RDS Oracle-spezifischen Engine-Version enthaltenen Oracle-Patches finden Sie unter Versionshinweise zu Amazon RDS for Oracle.

Automatische Snapshots während Engine-Upgrades

Während eines Upgrades einer Oracle-DB-Instance bieten Snapshots Schutz vor Upgrade-Problemen. Wenn der Aufbewahrungszeitraum für Backups für Ihre DB-Instance größer als 0 ist, erstellt Amazon RDS während des Upgrades die folgenden DB-Snapshots:

1. Einen Snapshot der DB-Instance, bevor Upgrade-Änderungen vorgenommen wurden. Wenn das Upgrade fehlschlägt, können Sie diesen Snapshot wiederherstellen, um eine DB-Instance zu erstellen, auf der die alte Version ausgeführt wird.
2. Einen Snapshot der DB-Instance nach Abschluss des Upgrades.

> **Note**
>
> Informationen über das Ändern Ihres Aufbewahrungszeitraums für Backups finden Sie unter Ändern einer Amazon RDS DB-Instance.

Nach einem Upgrade können Sie nicht zur die vorherigen Engine-Version zurückkehren. Sie können jedoch eine neue Oracle-DB-Instance erstellen, indem Sie den Snapshot vor dem Upgrade wiederherstellen.

Oracle-Upgrades in einer Multi-AZ-Bereitstellung

Wenn sich Ihre DB-Instance in einer Multi-AZ-Bereitstellung befindet, werden sowohl die Primär- als auch die Standby-Replikate von Amazon RDS aktualisiert. Wenn keine Betriebssystemaktualisierungen erforderlich sind, werden die primären und Standby-Upgrades gleichzeitig durchgeführt. Die Instances sind erst verfügbar, wenn das Upgrade abgeschlossen ist.

Wenn Betriebssystemupdates in einer Multi-AZ-Bereitstellung erforderlich sind, wendet Amazon RDS die Updates an, wenn Sie das Datenbank-Upgrade anfordern. Amazon RDS führt die folgenden Schritte durch:

1. Aktualisiert das Betriebssystem auf der aktuellen Standby-DB-Instance.
2. Führt ein Failover der primären DB-Instance zur Standby-DB-Instance durch.
3. Führt ein Upgrade der Datenbankversion auf der neuen primären DB-Instance durch, die früher die Standby-Instance war. Die Primärdatenbank ist während des Upgrades nicht verfügbar.
4. Aktualisiert das Betriebssystem auf der neuen Standby-DB-Instance, die früher die primäre DB-Instance war.
5. Führt ein Upgrade der Datenbankversion auf der neuen Standby-DB-Instance durch.
6. Failover der neuen primären DB-Instance zurück zur ursprünglichen primären DB-Instance und die neue Standby-DB-Instance zurück zur ursprünglichen Standby-DB-Instance. Somit setzt Amazon RDS die Replikationskonfiguration in ihren ursprünglichen Zustand zurück.

Oracle-Upgrades von Read Replicas (Lesereplikaten)

Die Oracle DB-Engine-Version der Quell-DB-Instance und alle Lese-Replikate müssen identisch sein. Amazon RDS führt das Upgrade in den folgenden Phasen durch:

1. Aktualisieren der Quell-DB-Instance. Die Read Replica sind in dieser Phase verfügbar.
2. Paralleles Aktualisieren der Read Replica unabhängig von den Replikatwartungsfenstern. Die Quell-DB ist in dieser Phase verfügbar.

Bei Upgrades von regionsübergreifenden Read Replica führt Amazon RDS zusätzliche Aktionen durch:

- Automatisches Generieren einer Optionsgruppe für die Zielversion
- Kopieren aller Optionen und Optionseinstellungen aus der ursprünglichen Optionsgruppe in die neue Optionsgruppe
- Verknüpfen des aktualisierten regionsübergreifenden Lese-Replikats mit der neuen Optionsgruppe

Upgrades der Oracle-Hauptversion

Sie müssen die DB-Instance manuell ändern, um ein Hauptversions-Upgrade durchführen zu können. Hauptversions-Upgrades werden nicht automatisch durchgeführt.

> **Important**
>
> Testen Sie alle Upgrades sorgfältig, um sicherzustellen, dass Ihre Anwendungen ordnungsgemäß funktionieren, bevor Sie das Upgrade auf Ihre Produktionsdatenbanken anwenden. Weitere Informationen finden Sie unter Testen eines Oracle DB-Upgrades.

Themen

- Unterstützte Versionen für Hauptversion-Upgrades
- Unterstützte Instance-Klassen für Hauptversion-Upgrades
- Erfassen von Statistiken vor Hauptversion-Upgrades
- Zulassen von Hauptversion-Upgrades

Unterstützte Versionen für Hauptversion-Upgrades

Amazon RDS unterstützt die folgenden Major-Versionsupgrades.

Aktuelle Version	Unterstütztes Upgrade
19.0.0.0 mit der CDB-Architektur	21.0.0.0

Ein Hauptversions-Upgrade von Oracle Database muss auf ein Release Update (RU) upgraden, das im selben Monat oder später veröffentlicht wurde. Herabstufungen der Hauptversion werden für Oracle-Database-Versionen nicht unterstützt.

Unterstützte Instance-Klassen für Hauptversion-Upgrades

In einigen Fällen wird Ihre aktuelle Oracle-DB-Instance möglicherweise auf einer DB-Instance-Klasse ausgeführt, die für die Version, auf die upgegradet werden soll, nicht unterstützt wird. In diesem Fall migrieren Sie die DB-Instance vor dem Upgrade auf eine unterstützte DB-Instance-Klasse. Weitere Informationen über die unterstützten DB-Instance-Klassen für alle Versionen und Editionen von Amazon RDS for Oracle finden Sie unter .

Erfassen von Statistiken vor Hauptversion-Upgrades

Vor dem Ausführen eines Hauptversions-Upgrades empfiehlt Oracle, dass Sie für die zu aktualisierende DB-Instance eine Optimierungsstatistik erheben. Diese Aktion kann die Ausfallzeiten der DB-Instance während des Upgrades reduzieren.

Zum Erheben von Optimierungsstatistiken stellen Sie als Hauptbenutzer eine Verbindung zur DB-Instance her und führen die `DBMS_STATS.GATHER_DICTIONARY_STATS`-Prozedur durch. Vgl. hierzu das folgende Beispiel.

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

Weitere Informationen finden Sie in der Oracle-Dokumentation unter GATHER_DICTIONARY_STATS Procedure.

Zulassen von Hauptversion-Upgrades

Ein Upgrade der Haupt-Engine-Version ist möglicherweise nicht mit Ihrer Anwendung kompatibel. Das Upgrade ist irreversibel. Wenn Sie für den EngineVersion Parameter eine Hauptversion angeben, die sich von der aktuellen Hauptversion unterscheidet, müssen Sie Hauptversions-Upgrades zulassen.

Wenn Sie eine Hauptversion mit dem CLI-Befehl modify-db-instance aktualisieren, müssen Sie `--allow-major-version-upgrade` angeben. Diese Einstellung ist nicht persistent, daher müssen Sie `--allow-major-version-upgrade` bei jeder Durchführung eines größeren Upgrades angeben. Dieser Parameter hat keine Auswirkungen auf Upgrades kleinerer Engine-Versionen. Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Wenn Sie eine Hauptversion mit der Konsole aktualisieren, müssen Sie keine Option zum Zulassen des Upgrades auswählen. Stattdessen zeigt die Konsole eine Warnung an, dass wichtige Upgrades irreversibel sind.

Oracle-Unterversion-Upgrades

In RDS for Oracle ist ein kleines Versionsupgrade ein Update auf eine größere DB-Engine-Version. In RDS ist eine kleinere Engine-Version entweder ein Release-Update (RU) oder ein Spatial Patch Bundle (SPB). Wenn auf Ihrer DB-Instance beispielsweise die Hauptversion Oracle Database 21c und die Nebenversion 21.0.0.0.ru-2024-10.rur-2024-10.r1 ausgeführt werden, können Sie Ihre DB-

Engine auf die Nebenversion 21.0.0.0.ru-2025-01.rur-2025-01.r1 aktualisieren. RDS für Oracle unterstützt keine Downgrades von Nebenversionen.

Sie können Ihre DB-Engine manuell oder automatisch auf eine Unterversion aktualisieren. Zur manuellen Aktualisierung vgl. Manuelles Upgraden der Engine-Version. Zur Konfiguration automatischer Aktualisierungen vgl. Automatisches Upgraden der Engine-Unterversion. Unabhängig davon, ob Sie die Aktualisierung manuell oder automatisch durchführen, führt ein Unterversionsupgrade zu Ausfallzeiten. Berücksichtigen Sie diese Ausfallzeit bei der Planung Ihrer Upgrades.

> **Important**
>
> Testen Sie alle Upgrades sorgfältig, um sicherzustellen, dass Ihre Anwendungen ordnungsgemäß funktionieren, bevor Sie das Upgrade auf Ihre Produktionsdatenbanken anwenden. Weitere Informationen finden Sie unter Testen eines Oracle DB-Upgrades.

Themen

- Versionsupdates (RUs) und Spatial Patch Bundles () SPBs
- Einschalten von automatischen Nebenversions-Upgrades
- Benachrichtigung über automatische Upgrades kleinerer Versionen in RDS for Oracle
- Wenn RDS automatische Upgrades für kleinere Versionen in RDS für Oracle plant
- Verwaltung eines automatischen Upgrades für Nebenversionen in RDS für Oracle

Versionsupdates (RUs) und Spatial Patch Bundles () SPBs

In RDS ist ein Release-Update (RU) eine vierteljährliche kleinere Engine-Version, die Sicherheitskorrekturen, Bugfixes und neue Funktionen für Oracle Database enthält. Ein Spatial Patch Bundle (SPB) ist eine RU-Engine-Version, die Patches enthält, die für die Oracle Spatial-Option entwickelt wurden. Das SPB mit dem Namen 19.0.0.0.ru-2025-01.spb-1.r1 enthält beispielsweise alle Patches in der entsprechenden RU 19.0.0.0.ru-2025-01.rur-2025-01.r1 sowie Patches, die für Spatial spezifisch sind. SPBs werden nur für Oracle Database 19c unterstützt.

Wenn Ihre Instanz für automatische Upgrades von Nebenversionen konfiguriert ist RUs und SPBs sich auf separaten Upgrade-Pfaden befindet. In der Regel wird ein SPB 2—3 Wochen nach der entsprechenden RU veröffentlicht. Die folgende Tabelle zeigt Beispiele für Nebenversionen für Oracle Database 19c.

Standard-RU-Upgrade-Pfad	SPB-Upgrade-Pfad
19.0.0.0.ru-2025-01.rur-2025-01.r1	19.0.0.0.ru-2025-01.spb-1.r1
19.0.0.0.ru-2025-04.rur-2025-04.r1	19.0.0.0.ru-2025-04.spb-1.r1
19.0.0.0.ru-2025-07.rur-2025-07.r1	19.0.0.0.ru-2025-07.spb-1.r1
19.0.0.0.ru-2025-10.rur-2025-10.r1	19.0.0.0.ru-2025-10.spb-1.r1

Wenn Ihre DB-Instance für automatische Upgrades konfiguriert ist, befindet sich Ihre Instance auf dem Upgrade-Pfad, der Ihrer aktuellen Version entspricht. Wenn auf Ihrer DB-Instance beispielsweise Version 19.0.0.0.ru-2025-01.rur-2025-01.r1 ausgeführt wird, wird Ihre Instance bei der Veröffentlichung von 19.0.0.0.ru-2025-04.rur-2025-04.r1 automatisch auf diese RU aktualisiert. Wenn Ihre DB-Instance 19.0.0.0.ru-2025-01.spb-1.r1 ausführt, wird Ihre Instance ebenfalls automatisch auf dieses SPB aktualisiert, wenn 19.0.0.0.ru-2025-04.spb-1.r1 veröffentlicht wird. Eine Instanz, auf der 19.0.0.0.ru-2025-01.rur-2025-01.r1 ausgeführt wird, bei der es sich um eine RU handelt, wird nicht automatisch auf 19.0.0.0.ru-2025-04.spb-1.r1 aktualisiert, was ein SPB auf einem separaten Upgrade-Pfad ist.

Sie können Ihre DB-Instance auch dann aktualisieren, wenn Ihre SPBs Instance Spatial nicht verwendet, aber die Spatial-Patches gelten nur für Oracle Spatial. Sie können manuell ein Upgrade von einem RU auf ein SPB mit derselben Engine-Version oder höher durchführen. Sie können Ihre Instance beispielsweise von 19.0.0.0.ru-2025-01.rur-2025-01.r1 auf eine der folgenden Engine-Versionen aktualisieren:

- 19.0.0.0.ru-2025-01.spb-1.r1
- 19.0.0.0.ru-2025-04.spb-1.r1

Sie können Ihre Instance nur dann von einem SPB auf eine RU aktualisieren, wenn es sich bei der RU um eine höhere Engine-Version handelt. Sie können beispielsweise von der SPB-Version 19.0.0.0.ru-2025-04.spb-1.r1 auf eine höhere RU-Version 19.0.0.0.ru-2025-07.rur-2025-07.r1 aktualisieren, aber nicht auf dieselbe RU-Version 19.0.0.0.ru-2025-04.rur-2025-04.r1.

Wenn Ihre DB-Instance für automatische Upgrades kleinerer Versionen konfiguriert ist und Sie manuell von einem RU auf ein SPB oder von einem SPB auf ein RU aktualisieren, ändert sich Ihr automatischer Upgrade-Pfad. Angenommen, Sie führen ein manuelles Upgrade von der RU-

Version 19.0.0.0.ru-2025-01.r1 auf die SPB-Version 19.0.0.0.ru-2025-01.spb-1.r1 durch. Ihr nächstes automatisches Upgrade der Nebenversion erfolgt auf SPB-Version 19.0.0.0.ru-2025-04.spb-1.r1.

Weil die SPBs Funktionen wie, die RDS für das Upgrade Ihrer Instance, identisch sind und identisch sind. RUs APIs RUs SPBs Die folgenden Befehle veranschaulichen das Upgrade auf eine RU und auf einen SPB.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version 19.0.0.0.ru-2025-01.rur-2025-01.r1

aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version 19.0.0.0.ru-2025-01.spb-1.r1
```

Weitere Hinweise zur Option Oracle Spatial finden Sie unterSo funktionieren Spatial Patch Bundles (SPBs). Informationen zu unterstützten RUs und SPBs für Oracle Database 19c finden Sie unter Amazon RDS for Oracle Database 19c (19.0.0.0).

Einschalten von automatischen Nebenversions-Upgrades

Bei einem automatischen Unterversion-Upgrade wendet RDS die neueste verfügbare Unterversion ohne manuelles Eingreifen auf Ihre Oracle-Datenbank an. Eine Instance von Amazon RDS für Oracle-plant die Aktualisierung unter den folgenden Umständen für das nächste Wartungszeitfenster:

- Für Ihre DB-Instance ist die Option Automatisches Unterversion-Upgrade eingeschaltet.
- Ihre DB-Instance führt die neueste DB-Engine-Unterversion nicht bereits aus.
- Für Ihre DB-Instance ist noch keine ausstehende Aktualisierung geplant.

Weitere Informationen zum Aktivieren von automatischen Aktualisierungen finden Sie unter Automatisches Upgraden der Engine-Unterversion.

Benachrichtigung über automatische Upgrades kleinerer Versionen in RDS for Oracle

RDS veröffentlicht eine Vorankündigung, bevor es mit der Planung automatischer Aktualisierungen beginnt. Sie finden die Benachrichtigung auf der Seite mit den Datenbankdetails auf der Registerkarte Wartung und Sicherungen. Die Nachricht hat das folgende Format:

```
An automatic minor version upgrade to engine version will become available
 on availability-date and will be applied during a subsequent maintenance window.
```

*availability-date*In der Vorankündigung steht das Datum, an dem RDS mit der Planung von Upgrades für DB-Instances in Ihrem beginnt AWS-Region. Dies ist nicht das Datum, für das die Aktualisierung Ihrer DB-Instance geplant ist. Wenn es beispielsweise der 1. März *availability-date* ist, plant RDS an diesem Tag Ihr Upgrade möglicherweise für den 14. April.

Sie können das Verfügbarkeitsdatum des Upgrades auch mithilfe des `describe-pending-maintenance-actions` Befehls im abrufen AWS CLI, wie im folgenden Beispiel gezeigt:

```
aws rds describe-pending-maintenance-actions

{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:orclinst1",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "db-upgrade",
                    "Description": "Automatic minor version upgrade to
 21.0.0.0.ru-2024-07.rur-2024-07.r1",
                    "CurrentApplyDate": "2024-12-02T08:10:00Z",
                    "OptInStatus": "next-maintenance"
                }
            ]
        }, ...
```

In der folgenden Tabelle sind Ihre Optionen für jeden Typ von Meldung zu einer ausstehenden Wartungsaktion aufgelistet.

Meldung zu ausstehender Wartungsaktion	Wenn die Meldung angezeigt wird	Soll beim nächsten Wartungsfenster angewendet werden?	Soll sofort angewendet werden?	Opt-In soll rückgängig gemacht werden?
Ein automatisches Upgrade der Nebenvers	4-6 Wochen, bevor automatische	Ja	Ja	Ja

Meldung zu ausstehender Wartungsaktion	Wenn die Meldung angezeigt wird	Soll beim nächsten Wartungsfenster angewendet werden?	Soll sofort angewendet werden?	Opt-In soll rückgängig gemacht werden?
ion auf *engine-version* wird am verfügbar sein *availability-date* und sollte während eines nachfolgenden Wartungsfensters installiert werden.	Aktualisierungen geplant sind.			
Automatisches Upgrade der Nebenversion auf *engine-version*	Am oder danach*availability-date* . RDS wendet diese Aktualisierung automatisch im nächsten Wartungsfenster der DB-Instance an.	Ja	Ja	Nein

Weitere Informationen zu describe-pending-maintenance-actions finden Sie in der AWS CLI - Befehlsreferenz.

Wenn RDS automatische Upgrades für kleinere Versionen in RDS für Oracle plant

Wenn das Verfügbarkeitsdatum für automatische Aktualisierungen erreicht ist, beginnt RDS mit der Planung der Aktualisierungen. In den meisten AWS-Regionen Fällen plant RDS Ihr Upgrade auf die neueste vierteljährliche RU etwa vier bis sechs Wochen nach dem Verfügbarkeitsdatum. Das geplante Datum hängt von den AWS-Region und anderen Faktoren ab. Weitere Informationen zu RUs und RURs finden Sie in den Versionshinweisen zu Amazon RDS for Oracle.

Wenn RDS die Aktualisierung plant, wird auf der Seite mit den Datenbankdetails auf der Registerkarte Wartung und Sicherungen die folgende Benachrichtigung angezeigt:

```
Automatic minor version upgrade to engine-version
```

Die vorherige Meldung weist darauf hin, dass RDS das Upgrade Ihrer DB-Engine für das nächste Wartungsfenster geplant hat.

Manchmal wird eine neue Nebenversion verfügbar, bevor RDS eine vorherige Nebenversion anwendet. Ihre Instance läuft beispielsweise, *minor-version-1* wenn *minor-version-2* sowohl als auch als Upgrade-Ziele verfügbar *minor-version-3* sind. Um unnötige Ausfallzeiten für Ihre DB-Instances zu vermeiden, plant RDS in dieser Situation das automatische Upgrade der Nebenversion auf die neueste Version, wobei das Upgrade auf die vorherige Version übersprungen wird. In diesem Beispiel aktualisiert RDS Ihre Instance von *minor-version-1* direkt auf*minor-version-3*.

Um eine gewisse Häufigkeit von Upgrades kleinerer Versionen sicherzustellen, können Sie Ihre Instances manuell aktualisieren, anstatt den automatischen Upgrade-Mechanismus zu verwenden. Um ein Upgrade für das nächste Wartungsfenster zu planen, geben Sie an, `--no-apply-immediately` wann Sie mit`modify-db-instance`. Um sofort ein Upgrade durchzuführen, geben Sie `--apply-immediately` stattdessen an. Weitere Informationen finden Sie unter Manuelles Upgraden der Engine-Version.

Verwaltung eines automatischen Upgrades für Nebenversionen in RDS für Oracle

Wenn eine neue Unterversion verfügbar wird, können Sie Ihre DB-Instance manuell auf diese Version aktualisieren. Im folgenden Beispiel wird die DB-Instance `orclinst1` sofort aktualisiert:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type immediate
```

Um ein automatisches Unterversion-Upgrade, das noch nicht geplant ist, zu deaktivieren, setzen Sie `--opt-in-type` auf `undo-opt-in`, wie im folgenden Beispiel:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
```

```
    --opt-in-type undo-opt-in
```

Wenn RDS bereits eine Aktualisierung für Ihre DB-Instance geplant hat, können Sie nicht `apply-pending-maintenance-action` verwenden, um dies zu stornieren. Sie können Ihre DB-Instance jedoch ändern und die automatische Funktion für Unterversion-Upgrades deaktivieren, was dazu führt, dass die Aktualisierung nicht mehr geplant ist.

Informationen zum Deaktivieren automatischer Unterversion-Upgrades finden Sie unter Automatisches Upgraden der Engine-Unterversion. Weitere Informationen zu apply-pending-maintenance-action finden Sie in der AWS CLI -Befehlsreferenz.

Überlegungen zu Oracle-Datenbank-Upgrades

Bevor Sie Ihre Oracle-Instance aktualisieren, sehen Sie sich die folgenden Informationen an.

Themen

- Überlegungen zu Oracle Multitenant
- Überlegungen zu Optionsgruppen
- Überlegungen zu Parametergruppen
- Überlegungen zur Zeitzone
- Überlegungen zum Spatial Patch Bundle (SPB)

Überlegungen zu Oracle Multitenant

In der folgenden Tabelle werden die Oracle-Datenbankarchitekturen beschrieben, die in verschiedenen Versionen unterstützt werden.

Oracle Database Version	RDS-Support-Status	Architektur
Oracle Database 21c	Unterstützt	Nur CDB
Oracle Database 19c	Unterstützt	CDB oder Nicht-CDB

In der folgenden Tabelle werden unterstützte und nicht unterstützte Upgrade-Pfade beschrieben.

Upgrade-Pfad	Unterstützt?
CDB zu CDB	Ja
Nicht-CDB zu CDB	Nein, aber Sie können eine Nicht-CDB in eine CDB konvertieren und sie dann aktualisieren
CDB zu Nicht-CDB	Nein

Weitere Informationen zu Oracle Multitenant in RDS for Oracle finden Sie unter Single-Tenant-Konfiguration der CDB-Architektur.

Überlegungen zu Optionsgruppen

Wenn Ihre DB-Instance eine benutzerdefinierte Optionsgruppe verwendet, kann Amazon RDS manchmal nicht automatisch eine neue Optionsgruppe zuweisen. Dies geschieht z.°B. beim Upgrade auf eine neue Hauptversion. Geben Sie in solchen Fällen beim Upgrade eine neue Optionsgruppe an. Wir empfehlen, dass Sie eine neue Optionsgruppe erstellen und dieser dieselben Optionen hinzufügen, über die auch Ihre bestehende benutzerdefinierte Optionsgruppe verfügt.

Weitere Informationen finden Sie unter Erstellen einer Optionsgruppe oder Kopieren einer Optionsgruppe.

Wenn Ihre DB-Instance eine benutzerdefinierte Optionsgruppe verwendet, die die `APEX-DEV` Optionen `APEX` und enthält, können Sie manchmal die Upgrade-Zeit reduzieren. Führen Sie dazu ein Upgrade Ihrer Version von Oracle APEX gleichzeitig mit Ihrer DB-Instance durch. Weitere Informationen finden Sie unter Die Oracle APEX-Version aktualisieren.

Überlegungen zu Parametergruppen

Wenn Ihre DB-Instance eine benutzerdefinierte Parametergruppe verwendet, kann Amazon RDS zuweilen Ihre DB-Instance nicht automatisch einer neuen Parametergruppe zuweisen. Dies geschieht z.°B. beim Upgrade auf eine neue Hauptversion. Stellen Sie in diesen Fällen sicher, dass Sie beim Upgrade eine neue Parametergruppe angeben. Wir empfehlen, dass Sie eine neue Parametergruppe erstellen und die Parameter so konfigurieren wie in Ihrer bestehenden benutzerdefinierten Parametergruppe.

Weitere Informationen finden Sie unter Erstellen einer DB-Parametergruppe in Amazon RDS oder Kopieren einer DB-Parametergruppe in Amazon RDS.

Überlegungen zur Zeitzone

Sie können die Zeitzonenoption verwenden, um die von Ihrer Oracle-DB-Instance verwendete Systemzeitzone zu ändern. Gründe, die Zeitzone einer DB-Instance zu ändern, sind beispielsweise Kompatibilitätsanforderungen der Umgebung an einem Standort oder eine veraltete Anwendung. Mit der Zeitzonenoption wird die Zeitzone auf der Ebene des Hosts geändert. Amazon RDS for Oracle aktualisiert die Systemzeitzone im Verlauf des Jahres automatisch. Weitere Informationen zur Systemzeitzone finden Sie unter Oracle-Zeitzone.

Wenn Sie eine Oracle DB-Instance erstellen, legt die Datenbank automatisch die Zeitzone der Datenbank fest. Die Zeitzone der Datenbank wird auch als Sommerzeit (DST, Daylight Saving Time) bezeichnet. Die Zeitzone der Datenbank unterscheidet sich von der Zeitzone des Systems.

Zwischen Oracle-Datenbankversionen können Patch-Sets oder einzelne Patches neue DST-Versionen enthalten. Diese Patches spiegeln die Änderungen der Übergangsregeln für verschiedene Zeitzonenregionen wider. Zum Beispiel kann sich eine Regierung ändern, wenn die Sommerzeit wirksam wird. Änderungen an DST-Regeln können sich auf vorhandene Daten des Datentyps `TIMESTAMP WITH TIME ZONE` auswirken.

Wenn Sie eine RDS-for-Oracle DB-Instance aktualisieren, aktualisiert Amazon RDS die Datenbank-Zeitzonendatei nicht automatisch. Um die Zeitzonendatei automatisch zu aktualisieren, können Sie die Option `TIMEZONE_FILE_AUTOUPGRADE` in die Optionsgruppe aufnehmen, die Ihrer DB-Instance während oder nach der Aktualisierung der Engine-Version zugewiesen ist. Weitere Informationen finden Sie unter Automatische Aktualisierung der Oracle-Zeitzonendatei.

Alternativ können Sie die Zeitzonendatei der Datenbank auch manuell aktualisieren, indem Sie eine neue Oracle DB-Instance mit dem gewünschten Sommerzeit-Patch erstellen. Wir empfehlen jedoch, dass Sie die Zeitzonendatei der Datenbank mit der Option `TIMEZONE_FILE_AUTOUPGRADE` aktualisieren.

Nachdem Sie die Zeitzonendatei aktualisiert haben, migrieren Sie die Daten von Ihrer aktuellen Instance zur neuen Instance. Sie können Daten mit verschiedenen Techniken migrieren, einschließlich der folgenden:

- AWS Database Migration Service
- Oracle GoldenGate
- Oracle Data Pump
- Ursprünglicher Export/Import (nicht länger für den allgemeinen Gebrauch unterstützt)

> **Note**
>
> Wenn Sie Daten mit Oracle Data Pump migrieren, löst das Dienstprogramm den Fehler ORA-39405 aus, wenn die Zielzeitzonenversion älter als die Quellzeitzonenversion ist.

Weitere Informationen finden Sie unter TIMESTAMP WITH TIMEZONE-Einschränkungen in der Oracle-Dokumentation.

Überlegungen zum Spatial Patch Bundle (SPB)

In RDS for Oracle ist das Release-Update (RU) eine kleinere Engine-Version, die Sicherheitskorrekturen, Bugfixes und neue Funktionen für Oracle Database enthält. Ein Spatial Patch Bundle (SPB) ist eine kleinere Engine-Version, die auch Patches enthält, die für die Oracle Spatial-Option entwickelt wurden. Beispielsweise ist 19.0.0.0.ru-2025-01.spb-1.r1 eine Engine-Nebenversion, die die RU-Patches in der Engine-Version 19.0.0.0.ru-2025-01.rur-2025-01.r1 sowie Spatial-Patches enthält.

Beachten Sie beim Upgrade SPBs Ihrer Datenbank auf Folgendes:

- SPBs werden nur für Oracle Database 19c unterstützt.
- In der Regel wird ein SPB 2—3 Wochen nach der entsprechenden vierteljährlichen RU veröffentlicht.
- Sie können Ihre DB-Instance auf ein SPB aktualisieren, auch wenn die Instance die Oracle Spatial-Option nicht verwendet, aber die Spatial-Patches in der Engine-Version gelten nur für Oracle Spatial. Sie können eine neue Instance auf einem SPB erstellen und die Oracle Spatial-Option später installieren.
- Wenn Sie das automatische Upgrade der Nebenversion für Ihre DB-Instance aktivieren, hängt Ihr Upgrade-Pfad davon ab, ob Ihre Instance derzeit ein SPB oder RU verwendet. Wenn Ihre Instance ein SPB verwendet, aktualisiert RDS Ihre Instance automatisch auf das neueste SPB. Wenn Ihre Instance eine RU verwendet, aktualisiert RDS Ihre Instance automatisch auf die neueste RU.
- Sie können Ihre DB-Instance nur dann manuell von einer RU auf eine SPB aktualisieren, wenn die SPB dieselbe Engine-Version oder eine höhere Version als Ihre aktuelle RU hat.
- Sie können Ihre DB-Instance nur dann manuell von einem SPB auf eine RU aktualisieren, wenn es sich bei der RU um eine höhere Version handelt.

Testen eines Oracle DB-Upgrades

Bevor Sie ein neues Hauptversions-Upgrade für Ihre DB-Instance durchführen, sollten Sie Ihre Datenbank und alle Anwendungen, die Zugriff auf die Datenbank haben, sorgfältig auf die Kompatibilität mit der neuen Version prüfen. Wir empfehlen Ihnen folgendes Vorgehen.

Um ein Hauptversions-Upgrade zu testen

1. Informieren Sie sich in der Upgrade-Dokumentation von Oracle über die neue Version der Datenbank-Engine, um zu prüfen, ob es Kompatibilitätsprobleme geben könnte, die sich auf Ihre Datenbank oder Anwendungen auswirken könnten. Weitere Informationen finden Sie unter Database Upgrade Guide in der Oracle-Dokumentation.
2. Wenn Ihre DB-Instance eine benutzerdefinierte Optionsgruppe verwendet, erstellen Sie eine neue Optionsgruppe, die kompatibel mit der neuen Version ist, auf die Sie upgraden. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.
3. Wenn Ihre DB-Instance eine benutzerdefinierte Parametergruppe verwendet, erstellen Sie eine neue Parametergruppe, die kompatibel mit der neuen Version ist, auf die Sie upgraden. Weitere Informationen finden Sie unter Überlegungen zu Parametergruppen.
4. Erstellen Sie einen DB-Snapshot der zu aktualisierenden DB-Instance. Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.
5. Stellen Sie den DB-Snapshot wieder her, um eine neue Test-DB-Instance zu erstellen. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.
6. Ändern Sie diese neue Test-DB-Instance mit den folgenden Methoden, um sie auf die neue Version upzugraden:

   - Konsole
   - AWS CLI
   - RDS-API
7. Test durchführen:

   - Führen Sie so viele Qualitätssicherungstests mit der upgegradeten DB-Instance durch, wie nötig, um sicherzustellen, dass Ihre Datenbank und Anwendung mit der neuen Version korrekt ausgeführt werden.
   - Führen Sie alle nötigen neuen Tests aus, um die Auswirkungen von Kompatibilitätsproblemen zu bewerten, die Sie in Schritt 1 bestimmt haben.
   - Testen Sie alle gespeicherten Prozeduren, Funktionen und Auslöser.

- Leiten Sie Testversionen Ihrer Anwendungen an die aktualisierte DB-Instance weiter. Überprüfen Sie, ob die Anwendungen mit dieser neuen Version korrekt ausgeführt werden.
- Beurteilen Sie den Speicherplatz, den die upgegradete Instance verwendet, um zu bestimmen, ob das Upgrade zusätzlichen Speicherplatz benötigt. Es kann sein, dass Sie eine größere Instance-Klasse auswählen müssen, um die neue Version bei der Produktion zu unterstützen. Weitere Informationen finden Sie unter .

8. Wenn alle Tests erfolgreich sind, aktualisieren Sie Ihre Produktions-DB-Instance. Es wird empfohlen zu bestätigen, dass die DB-Instance ordnungsgemäß funktioniert, bevor Sie Schreibvorgänge für die DB-Instance zulassen.

Aktualisierung der Version einer RDS für Oracle-DB-Instance

Um die DB-Engine-Version einer RDS for Oracle-DB-Instance manuell zu aktualisieren AWS Management Console, verwenden Sie die AWS CLI, oder die RDS-API. Allgemeine Informationen zu Datenbank-Upgrades in RDS finden Sie unterAktualisierung der Version einer RDS für Oracle-DB-Instance. Verwenden Sie den AWS CLI describe-db-engine-versionsBefehl, um gültige Upgrade-Ziele abzurufen.

Konsole

Um die Engine-Version einer RDS for Oracle-DB-Instance mithilfe der Konsole zu aktualisieren

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann die DB-Instance aus, die Sie upgraden möchten.
3. Wählen Sie Ändern aus.
4. Wählen Sie für die DB-Engine-Version eine höhere Datenbankversion aus.
5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen. Stellen Sie sicher, dass Sie die Auswirkungen eines Upgrades der Datenbankversion verstehen. Sie können eine aktualisierte DB-Instance nicht zurück in die vorherige Version konvertieren. Stellen Sie sicher, dass Sie sowohl Ihre Datenbank als auch Ihre Anwendung mit der neuen Version getestet haben, bevor Sie fortfahren.
6. Entscheiden Sie, wann Sie Ihr DB-Instance-Upgrade planen möchten. Wählen Sie Apply immediately, um die Änderungen sofort anzuwenden. Die Auswahl dieser Option kann in

einigen Fällen einen Ausfall verursachen. Weitere Informationen finden Sie unter Einstellung „Zeitplanänderungen“ verwenden.

7. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance (DB-Instance ändern) aus, um Ihre Änderungen zu speichern.

   Klicken Sie anderenfalls auf Zurück, um Ihre Änderungen zu bearbeiten, oder klicken Sie auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

Um die Engine-Version einer RDS for Oracle-DB-Instance zu aktualisieren, können Sie den modify-db-instanceCLI-Befehl verwenden. Geben Sie die folgenden Parameter an:

- `--db-instance-identifier`— der Name der RDS für Oracle-DB-Instance.
- `--engine-version`: die Versionsnummer der Datenbank-Engine, auf die das Upgrade durchgeführt wird

  Verwenden Sie den AWS CLI describe-db-engine-versionsBefehl, um Informationen zu gültigen Engine-Versionen zu erhalten.
- `--allow-major-version-upgrade`— um die DB-Engine-Version zu aktualisieren.
- `--no-apply-immediately`, um Änderungen im nächsten Wartungszeitraum anwenden. Verwenden Sie , um Änderungen sofort anzuwende `--apply-immediately`.

Example

Im folgenden Beispiel wird eine CDB-Instance mit dem Namen `myorainst` von von ihrer aktuellen Version `19.0.0.0.ru-2024-01.rur-2024-01.r1` auf Version `21.0.0.0.ru-2024-04.rur-2024-04.r1` aktualisiert.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier myorainst \
    --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 \
    --allow-major-version-upgrade \
    --no-apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myorainst ^
    --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 ^
    --allow-major-version-upgrade ^
    --no-apply-immediately
```

RDS-API

Verwenden Sie die Aktion ModifyDBInstance, um eine RDS for Oracle-DB-Instance zu aktualisieren. Geben Sie die folgenden Parameter an:

- `DBInstanceIdentifier` – der Name der DB-Instance, z. B *`myorainst`*.
- `EngineVersion`: die Versionsnummer der Datenbank-Engine, auf die das Upgrade durchgeführt wird Verwenden Sie den Vorgang DescribeDB, um Informationen zu gültigen Engine-Versionen zu erhalten. EngineVersions
- `AllowMajorVersionUpgrade`, um festzulegen, ob ein Hauptversions-Upgrade zugelassen wird. Setzen Sie hierzu den Wert auf `true`.
- `ApplyImmediately`: Änderungen sofort oder während des nächsten Wartungszeitraums anwenden Legen Sie den Wert auf fest, um Änderungen sofort anzuwende `true`. Legen Sie den Wert auf fest, um Änderungen im nächsten Wartungszeitraum durchzuführe `false`.

Aktualisieren eines Oracle-DB-Snapshots

Durch die Aktualisierung Ihrer Oracle-DB-Snapshots in Amazon RDS wird sichergestellt, dass Ihre Datenbank sicher, kompatibel und vollständig unterstützt bleibt. Wenn ältere Oracle-Versionen das Ende der Patchunterstützung erreichen, können Sie alle manuellen DB-Snapshots, die mit diesen Versionen verknüpft sind, aktualisieren, um potenzielle Sicherheitslücken oder Diensteinschränkungen zu vermeiden. Weitere Informationen finden Sie unter Oracle-Engine-Versionsverwaltung.

Amazon RDS unterstützt das Upgrade von Snapshots in allen AWS-Regionen.

Konsole

So aktualisieren Sie einen Oracle DB-Snapshot

1. Loggen Sie sich ein bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

2. Wählen Sie im Navigationsbereich Snapshots und wählen Sie dann den zu aktualisierenden DB-Snapshot aus.
3. Wählen Sie unter Actions (Aktionen) die Option Upgrade Snapshot (Snapshot aktualisieren). Die Seite Upgrade snapshot (Snapshot aktualisieren) erscheint.
4. Wählen Sie New engine version (Neue Engine-Version) aus, auf die der Snapshot aktualisiert werden soll.
5. (Optional) Wählen Sie für Optionsgruppe die Optionsgruppe für den aktualisierten DB-Snapshot aus. Beim Upgrade eines DB-Snapshot gelten die gleichen Überlegungen zur Optionsgruppe wie beim Upgrade einer DB-Instance. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.
6. Wählen Sie Save changes (Änderungen speichern) aus, um die Änderungen zu speichern.

   Während des Upgrades werden alle Snapshot-Aktionen für diesen DB-Snapshot deaktiviert. Außerdem wird der Status des DB-Snapshots von available (verfügbar) in upgraden... geändert. Wenn der Vorgang abgeschlossen wurde, wird der Status in active (aktiv) geändert. Wenn das Upgrade für den DB-Snapshot aufgrund einer Beschädigung des Snapshots nicht durchgeführt werden kann, wird der Status in unavailable (nicht verfügbar) geändert. Sie können den Snapshot aus diesem Zustand nicht wiederherstellen.

   > **Note**
   >
   > Wenn die Aktualisierung des DB-Snapshots fehlschlägt, wird der Snapshot wieder in seinen ursprünglichen Zustand zurückgebracht.

AWS CLI

Um einen Oracle-DB-Snapshot zu aktualisieren, verwenden Sie AWS CLI, rufen Sie den modify-db-snapshotBefehl mit den folgenden Parametern auf:

- `--db-snapshot-identifier`: der Name des DB-Snapshots
- `--engine-version`: die Version, auf die das Upgrade des Snapshots durchgeführt werden soll

Es könnte sein, dass Sie auch die folgenden Parameter einbeziehen müssen. Beim Upgrade eines DB-Snapshot gelten die gleichen Überlegungen zur Optionsgruppe wie beim Upgrade einer DB-Instance. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.

- `--option-group-name`: die Optionsgruppe für den aktualisierten DB-Snapshot

Example

Das folgende Beispiel führt ein Upgrade für einen DB-Snapshot aus.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier mydbsnapshot \
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \
    --option-group-name default:oracle-se2-19
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot ^
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 ^
    --option-group-name default:oracle-se2-19
```

RDS API

Um einen Oracle-DB-Snapshot mithilfe von Amazon zu aktualisieren RDSAPI, rufen Sie den odifyDBSnapshotM-Vorgang mit den folgenden Parametern auf:

- `DBSnapshotIdentifier`: der Name des DB-Snapshots
- `EngineVersion`: die Version, auf die das Upgrade des Snapshots durchgeführt werden soll

Es könnte sein, dass Sie auch den Parameter `OptionGroupName` einbeziehen müssen. Beim Upgrade eines DB-Snapshot gelten die gleichen Überlegungen zur Optionsgruppe wie beim Upgrade einer DB-Instance. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.

Verwenden von Drittanbietersoftware mit Ihrer RDS-for-Oracle-DB-Instance

Sie können eine RDS for Oracle DB-Instance hosten, die Tools und Software von Drittanbietern unterstützt.

Themen

- Verwenden von Oracle GoldenGate mit Amazon RDS for Oracle
- Verwenden des Oracle Repository Creation Utility (RCU) in RDS for Oracle
- Konfigurieren von Oracle Connection Manager auf einer Amazon-EC2-Instance
- Installieren einer Siebel-Datenbank auf Oracle auf Amazon RDS

Verwenden von Oracle GoldenGate mit Amazon RDS for Oracle

Oracle GoldenGate sammelt, repliziert und verwaltet Transaktionsdaten zwischen Datenbanken. Es handelt sich um ein Softwarepaket für die protokollbasierte Erfassung von Änderungsdaten (Change Data Capture, CDC) und Replikation, das mit Datenbanken für OLTP-Systeme (Online Transaction Processing) verwendet wird. Oracle GoldenGate erstellt Traildateien, die die zuletzt geänderten Daten aus der Quelldatenbank enthalten. Anschließend werden diese Dateien an den Server übertragen, wo ein Prozess die Pfaddatei in Standard-SQL konvertiert, um sie auf die Zieldatenbank anzuwenden.

Oracle GoldenGate mit RDS for Oracle unterstützt die folgenden Funktionen:

- Active-Active-Datenbank-Replik
- Notfallwiederherstellung
- Datenschutz
- In-Regionen- und regionsübergreifende Replikation
- Migration ohne Ausfallzeiten und Upgrades
- Datenreplikation zwischen einer DB-Instance von RDS für Oracle und einer Nicht-Oracle-Datenbank

> **Note**
>
> Eine Liste der unterstützten Datenbanken finden Sie unter Oracle Fusion Middleware Supported System Configurations in der Oracle-Dokumentation.

Sie können Oracle GoldenGate mit RDS for Oracle verwenden, um auf Hauptversionen von Oracle Database zu aktualisieren. Sie können Oracle beispielsweise verwenden, GoldenGate um ein Upgrade von einer lokalen Oracle Database 11g-Datenbank auf Oracle Database 19c auf einer Amazon RDS-DB-Instance durchzuführen.

Themen

- Unterstützte Versionen und Lizenzoptionen für Oracle GoldenGate
- Anforderungen und Einschränkungen für Oracle GoldenGate
- GoldenGate Oracle-Architektur
- Oracle einrichten GoldenGate

- Arbeiten mit den EXTRACT und REPLICAT Hilfsprogrammen von Oracle GoldenGate
- Überwachung von Oracle GoldenGate
- Fehlerbehebung bei Oracle GoldenGate

Unterstützte Versionen und Lizenzoptionen für Oracle GoldenGate

Sie können die Standard Edition 2 (SE2) oder die Enterprise Edition (EE) von RDS for Oracle mit Oracle GoldenGate Version 12c und höher verwenden. Sie können die folgenden GoldenGate Oracle-Funktionen verwenden:

- Oracle GoldenGate Remote Capture (Extrakt) wird unterstützt.
- Das Erfassen (Extrahieren) wird auf DB-Instances von RDS für Oracle unterstützt, die die herkömmliche Nicht-CDB-Datenbankarchitektur verwenden. Oracle GoldenGate Remote PDB Capture wird unterstützt, wenn Oracle Database 21c oder Oracle Database 19c Version 19.0.0.0.ru-2024-04.rur-2024-04.r1 oder höher CDBs ausgeführt wird.
- Oracle GoldenGate Remote Delivery (Replicat) wird auf RDS für Oracle-DB-Instances unterstützt, die entweder Nicht-CDB- oder CDB-Architekturen verwenden. Remote Delivery unterstützt Integrated Replicat, Parallel Replicat, Coordinated Replicat und Classic Replicat.
- RDS for Oracle unterstützt die Classic- und Microservices-Architekturen von Oracle. GoldenGate
- Die Replikation von Oracle GoldenGate DDL- und Sequence-Werten wird unterstützt, wenn der integrierte Erfassungsmodus verwendet wird.

Sie sind für die Verwaltung der GoldenGate Oracle-Lizenzierung (BYOL) für die Verwendung mit Amazon RDS insgesamt AWS-Regionen verantwortlich. Weitere Informationen finden Sie unter RDSfür Oracle-Lizenzoptionen.

Anforderungen und Einschränkungen für Oracle GoldenGate

Wenn Sie mit Oracle GoldenGate und RDS for Oracle arbeiten, sollten Sie die folgenden Anforderungen und Einschränkungen berücksichtigen:

- Sie sind verantwortlich für die Einrichtung und Verwaltung von Oracle GoldenGate für die Verwendung mit RDS for Oracle.
- Sie sind dafür verantwortlich, eine GoldenGate Oracle-Version einzurichten, die für die Quell- und die Zieldatenbank zertifiziert ist. Weitere Informationen finden Sie unter Oracle Fusion Middleware Supported System Configurations in der Oracle-Dokumentation.

- Sie können Oracle GoldenGate in vielen verschiedenen AWS Umgebungen für viele verschiedene Anwendungsfälle verwenden. Wenn Sie ein Support-Problem mit Oracle haben GoldenGate, wenden Sie sich an Oracle Support Services.
- Sie können Oracle GoldenGate auf RDS für Oracle-DB-Instances verwenden, die Oracle Transparent Data Encryption (TDE) verwenden. Um die Integrität der replizierten Daten aufrechtzuerhalten, konfigurieren Sie die Verschlüsselung auf dem GoldenGate Oracle-Hub mithilfe von Amazon EBS-verschlüsselten Volumes oder Traildateiverschlüsselung. Konfigurieren Sie auch die Verschlüsselung für Daten, die zwischen dem GoldenGate Oracle-Hub und den Quell- und Zieldatenbank-Instances gesendet werden. RDS-for-Oracle-DB-Instances unterstützen die Verschlüsselung mit Oracle Secure Sockets Layer oder Oracle Native Network Encryption.

GoldenGate Oracle-Architektur

Die GoldenGate Oracle-Architektur zur Verwendung mit Amazon RDS besteht aus den folgenden entkoppelten Modulen:

Quelldatenbank

Ihre Quelldatenbank kann entweder eine lokale Oracle-Datenbank, eine Oracle-Datenbank auf einer EC2 Amazon-Instance oder eine Oracle-Datenbank auf einer Amazon RDS DB-Instance sein.

Oracle-Hub GoldenGate

Ein GoldenGate Oracle-Hub verschiebt Transaktionsinformationen von der Quelldatenbank in die Zieldatenbank. Für Ihren Hub gibt es die folgenden zwei Möglichkeiten:

- Eine EC2 Amazon-Instance mit GoldenGate installierter Oracle-Datenbank und Oracle
- Eine lokale Oracle-Installation

Sie können mehr als einen EC2 Amazon-Hub haben. Wir empfehlen die Verwendung von zwei Hubs, wenn Sie Oracle GoldenGate für die regionsübergreifende Replikation verwenden.

Zieldatenbank

Ihre Zieldatenbank kann sich entweder auf einer Amazon RDS DB-Instance, einer EC2 Amazon-Instance oder einem lokalen Standort befinden.

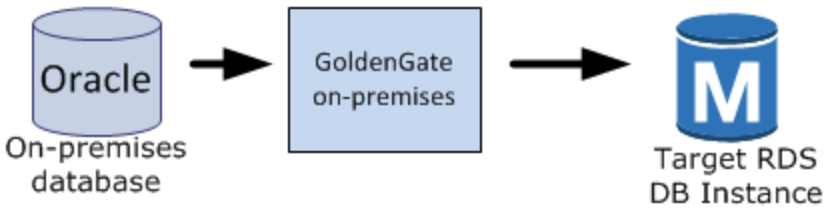
In den folgenden Abschnitten werden allgemeine Szenarien für Oracle GoldenGate auf Amazon beschriebenRDS.

Themen

- Lokale Quelldatenbank und Oracle-Hub GoldenGate
- Lokale Quelldatenbank und Amazon-Hub EC2
- RDSAmazon-Quelldatenbank und EC2 Amazon-Hub
- EC2Amazon-Quelldatenbank und EC2 Amazon-Hub
- EC2Amazon-Hubs in verschiedenen Regionen AWS

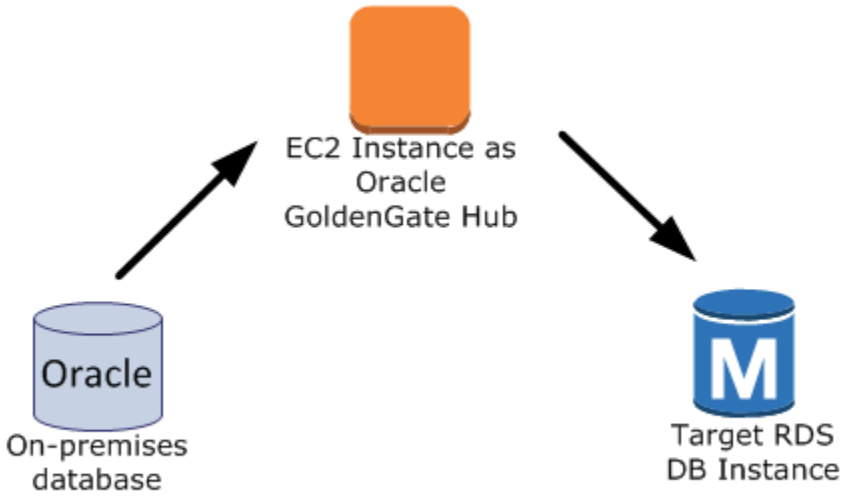
Lokale Quelldatenbank und Oracle-Hub GoldenGate

In diesem Szenario stellen eine lokale Oracle-Quelldatenbank und ein lokaler GoldenGate Oracle-Hub Daten für eine RDS Amazon-DB-Zielinstanz bereit.



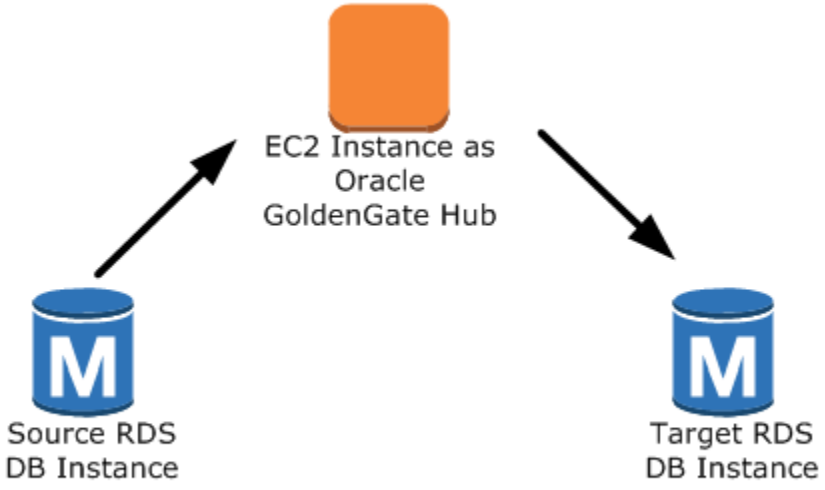
Lokale Quelldatenbank und Amazon-Hub EC2

In diesem Szenario fungiert eine lokale Oracle-Datenbank als Quelldatenbank. Es ist mit einem EC2 Amazon-Instance-Hub verbunden. Dieser Hub stellt Daten für ein Ziel RDS für eine Oracle-DB-Instance bereit.



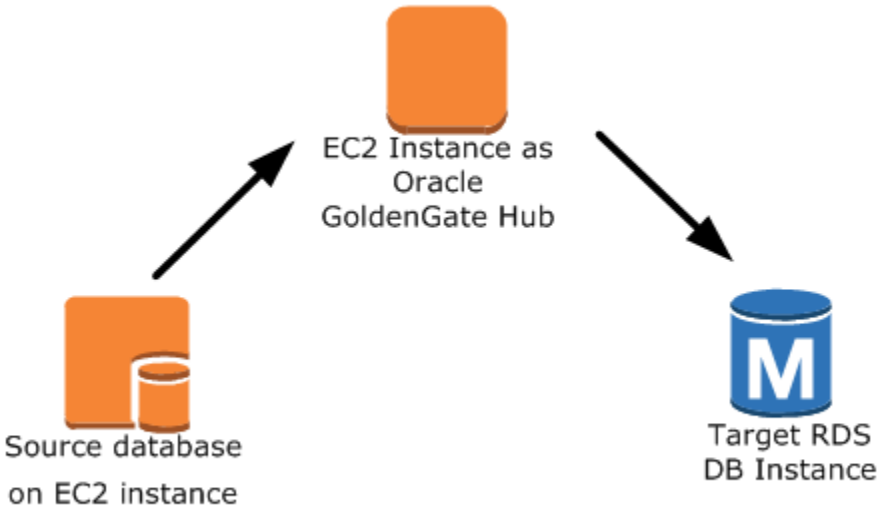
RDSAmazon-Quelldatenbank und EC2 Amazon-Hub

In diesem Szenario fungiert eine RDS For-Oracle-DB-Instance als Quelldatenbank. Es ist mit einem EC2 Amazon-Instance-Hub verbunden. Dieser Hub stellt Daten für ein Ziel RDS für eine Oracle-DB-Instance bereit.



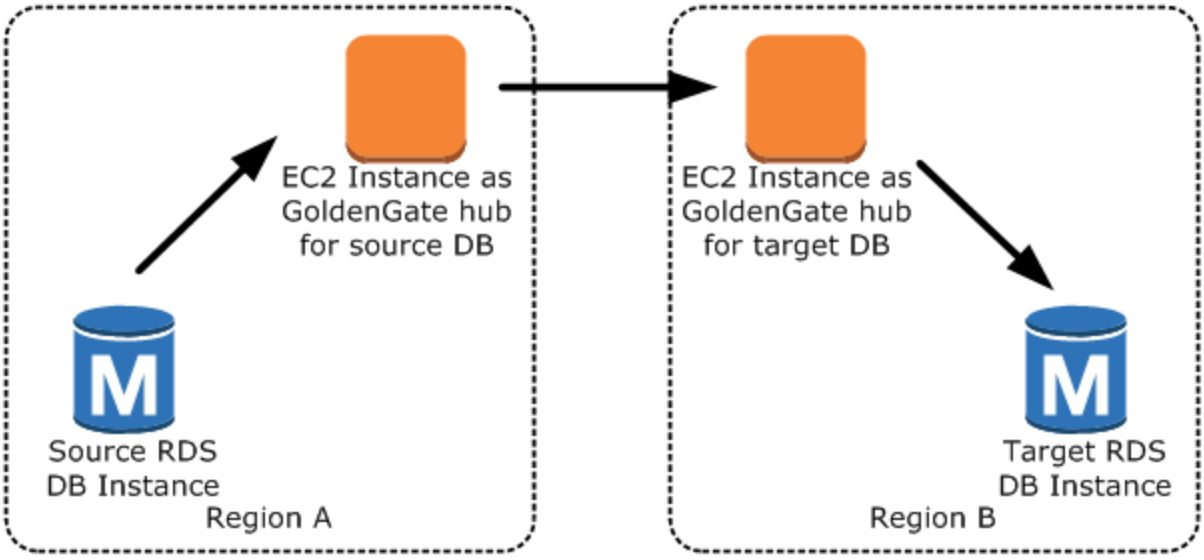
EC2Amazon-Quelldatenbank und EC2 Amazon-Hub

In diesem Szenario fungiert eine Oracle-Datenbank auf einer EC2 Amazon-Instance als Quelldatenbank. Es ist mit einem EC2 Amazon-Instance-Hub verbunden. Dieser Hub stellt Daten für ein Ziel RDS für eine Oracle-DB-Instance bereit.



EC2Amazon-Hubs in verschiedenen Regionen AWS

In diesem Szenario ist eine Oracle-Datenbank auf einer Amazon RDS DB-Instance mit einem EC2 Amazon-Instance-Hub in derselben AWS Region verbunden. Der Hub ist mit einem EC2 Amazon-Instance-Hub in einer anderen AWS Region verbunden. Dieser zweite Hub stellt dem Ziel Daten RDS für die Oracle-DB-Instance in derselben AWS Region bereit wie der zweite EC2 Amazon-Instance-Hub.



> **Note**
>
> Alle Probleme, die sich auf die Ausführung von Oracle GoldenGate in einer lokalen Umgebung auswirken, wirken sich auch auf die Ausführung von Oracle GoldenGate auf AWS aus. Es wird dringend empfohlen, den GoldenGate Oracle-Hub zu überwachen, um sicherzustellen, dass der `EXTRACT` Vorgang `REPLICAT` bei einem Failover wieder aufgenommen wird. Da der GoldenGate Oracle-Hub auf einer EC2 Amazon-Instance ausgeführt wird, verwaltet Amazon RDS den GoldenGate Oracle-Hub nicht und kann nicht sicherstellen, dass er läuft.

Oracle einrichten GoldenGate

Um Oracle GoldenGate mithilfe von Amazon einzurichtenRDS, konfigurieren Sie den Hub auf einer EC2 Amazon-Instance und anschließend die Quell- und Zieldatenbanken. Die folgenden Abschnitte enthalten ein Beispiel für die Einrichtung von Oracle GoldenGate für die Verwendung mit Amazon RDS for Oracle.

Themen

- Einrichtung eines GoldenGate Oracle-Hubs auf Amazon EC2
- Einrichtung einer Quelldatenbank für die Verwendung mit Oracle GoldenGate auf Amazon RDS
- Einrichtung einer Zieldatenbank für die Verwendung mit Oracle GoldenGate auf Amazon RDS

Einrichtung eines GoldenGate Oracle-Hubs auf Amazon EC2

Um einen GoldenGate Oracle-Hub auf einer EC2 Amazon-Instance zu erstellen, erstellen Sie zunächst eine EC2 Amazon-Instance mit einer vollständigen Client-Installation von OracleRDBMS. Auf der EC2 Amazon-Instance muss auch GoldenGate Oracle-Software installiert sein. Die GoldenGate Oracle-Softwareversionen hängen von den Quell- und Zieldatenbankversionen ab. Weitere Informationen zur Installation von Oracle GoldenGate finden Sie in der GoldenGateOracle-Dokumentation.

Die EC2 Amazon-Instance, die als GoldenGate Oracle-Hub dient, speichert und verarbeitet die Transaktionsinformationen aus der Quelldatenbank in Traildateien. Um diesen Prozess zu unterstützen, stellen Sie sicher, dass Sie die folgenden Bedingungen erfüllen:

- Sie haben genügend Speicherplatz für die Pfaddateien reserviert.

- Die EC2 Amazon-Instance verfügt über genügend Rechenleistung, um die Datenmenge zu verwalten.
- Die EC2 Instance verfügt über ausreichend Speicher, um die Transaktionsinformationen zu speichern, bevor sie in die Trail-Datei geschrieben werden.

So richten Sie einen Oracle GoldenGate Classic Architecture Hub auf einer EC2 Amazon-Instance ein

1. Erstellen Sie Unterverzeichnisse im GoldenGate Oracle-Verzeichnis.

   Starten `ggsci` Sie in der EC2 Amazon-Befehlszeilen-Shell den GoldenGate Oracle-Befehlsinterpreter. Mit dem Befehl `CREATE SUBDIRS` werden die Unterverzeichnisse im `/gg`-Verzeichnis für Parameter, Berichte und Prüfpunktdateien erstellt.

   ```
   prompt$ cd /gg
   prompt$ ./ggsci

   GGSCI> CREATE SUBDIRS
   ```

2. Konfigurieren Sie die `mgr.prm`-Datei.

   Im folgenden Beispiel werden der Datei `$GGHOME/dirprm/mgr.prm` Zeilen hinzugefügt.

   ```
   PORT 8199
   PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
   ```

3. Starten Sie den Manager.

   Im folgenden Beispiel wird `ggsci` gestartet und der Befehl `start mgr` ausgeführt.

   ```
   GGSCI> start mgr
   ```

Der GoldenGate Oracle-Hub ist jetzt einsatzbereit.

Einrichtung einer Quelldatenbank für die Verwendung mit Oracle GoldenGate auf Amazon RDS

Führen Sie die folgenden Aufgaben aus, um eine Quelldatenbank für die Verwendung mit Oracle einzurichten GoldenGate.

Einrichtungsschritte

- Schritt 1: Aktivieren zusätzlicher Protokollierung in der Quelldatenbank
- Schritt 2: Setzen Sie den Initialisierungsparameter ENABLE _ GOLDENGATE _ REPLICATION auf true
- Schritt 3: Festlegen des Protokollaufbewahrungszeitraums in der Quell-Datenbank
- Schritt 4: Erstellen Sie ein GoldenGate Oracle-Benutzerkonto in der Quelldatenbank
- Schritt 5: Erteilen von Berechtigungen für das Benutzerkonto in der Quelldatenbank
- Schritt 6: Fügen Sie einen TNS Alias für die Quelldatenbank hinzu

Schritt 1: Aktivieren zusätzlicher Protokollierung in der Quelldatenbank

Um die minimale zusätzliche Protokollierung auf Datenbankebene zu aktivieren, führen Sie die folgende PL/−Prozedur aus: SQL

```
EXEC rdsadmin.rdsadmin_util.alter_supplemental_logging(p_action => 'ADD')
```

Schritt 2: Setzen Sie den Initialisierungsparameter ENABLE _ GOLDENGATE _ REPLICATION auf true

Wenn Sie den `ENABLE_GOLDENGATE_REPLICATION`-Initialisierungsparameter auf `true` einstellen, können Datenbankdienste die logische Replikation unterstützen. Wenn sich Ihre Quelldatenbank auf einer RDS Amazon-DB-Instance befindet, stellen Sie sicher, dass Sie der DB-Instance eine Parametergruppe zugewiesen haben, deren `ENABLE_GOLDENGATE_REPLICATION` Initialisierungsparameter auf `true` gesetzt ist. Weitere Informationen zum Initialisierungsparameter `ENABLE_GOLDENGATE_REPLICATION` finden Sie in der Oracle-Database-Dokumentation.

Schritt 3: Festlegen des Protokollaufbewahrungszeitraums in der Quell-Datenbank

Erstellen Sie die Quelldatenbank unbedingt so, dass archivierte Redo-Protokolle aufbewahrt werden. Berücksichtigen Sie die folgenden Hinweise:

- Geben Sie die Dauer für die Protokollaufbewahrung in Stunden an. Der Mindestwert ist eine Stunde.
- Legen Sie die Dauer so fest, dass potenzielle Ausfallzeiten der Quell-Instance, jede mögliche Kommunikationsdauer und mögliche Dauer von Netzwerkproblemen für die Quell-Instance überschritten werden. Eine solche Dauer ermöglicht es Oracle, bei Bedarf Protokolle aus der Quell-Instance GoldenGate wiederherzustellen.
- Stellen Sie sicher, dass auf Ihrer Instance genügend Speicherplatz für die Dateien vorhanden ist.

Legen Sie beispielsweise den Aufbewahrungszeitraum für archivierte Redo-Protokolle auf 24 Stunden fest.

```
EXEC rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24)
```

Wenn die Protokollaufbewahrung nicht aktiviert ist oder wenn der Aufbewahrungswert zu klein ist, erhalten Sie eine Meldung ähnlich wie die folgende.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Da Ihre DB-Instance Ihre archivierten Redo-Protokolle aufbewahrt, stellen Sie sicher, dass Sie über ausreichend Speicherplatz für die Dateien verfügen. Um zu sehen, wie viel Speicherplatz Sie in der letzten Zeit verbraucht haben *num_hours* Stunden, führen Sie die folgende Abfrage aus und ersetzen Sie *num_hours* mit der Anzahl der Stunden.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) BYTES FROM V$ARCHIVED_LOG
   WHERE NEXT_TIME>=SYSDATE-num_hours/24 AND DEST_ID=1;
```

Schritt 4: Erstellen Sie ein GoldenGate Oracle-Benutzerkonto in der Quelldatenbank

Oracle GoldenGate wird als Datenbankbenutzer ausgeführt und benötigt die entsprechenden Datenbankberechtigungen, um auf die Redo- und archivierten Redo-Logs für die Quelldatenbank zuzugreifen. Um diese bereitzustellen, erstellen Sie ein Benutzerkonto in der Quelldatenbank. Weitere Informationen zu den Berechtigungen für ein GoldenGate Oracle-Benutzerkonto finden Sie in der Oracle-Dokumentation.

Mit folgenden Anweisungen wird ein Benutzerkonto mit dem Namen `oggadm1` erstellt.

```
CREATE TABLESPACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE ADMINISTRATOR TEMPORARY TABLESPACE TEMP;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Schritt 5: Erteilen von Berechtigungen für das Benutzerkonto in der Quelldatenbank

Bei dieser Aufgabe gewähren Sie den Datenbankbenutzern in Ihrer Quelldatenbank die erforderlichen Kontoberechtigungen.

So erteilen Sie Kontoberechtigungen in der Quelldatenbank

1. Erteilen Sie dem GoldenGate Oracle-Benutzerkonto mithilfe des SQL Befehls `grant` und der `rdsadmin.rdsadmin_util` Prozedur die erforderlichen Rechte`grant_sys_object`. Mit folgenden Anweisungen werden einem Benutzer mit dem Namen `oggadm1` Berechtigungen erteilt.

   ```
   GRANT CREATE SESSION, ALTER SESSION TO oggadm1;
   GRANT RESOURCE TO oggadm1;
   GRANT SELECT ANY DICTIONARY TO oggadm1;
   GRANT FLASHBACK ANY TABLE TO oggadm1;
   GRANT SELECT ANY TABLE TO oggadm1;
   GRANT SELECT_CATALOG_ROLE TO rds_master_user_name WITH ADMIN OPTION;
   EXEC rdsadmin.rdsadmin_util.grant_sys_object ('DBA_CLUSTERS', 'OGGADM1');
   GRANT EXECUTE ON DBMS_FLASHBACK TO oggadm1;
   GRANT SELECT ON SYS.V_$DATABASE TO oggadm1;
   GRANT ALTER ANY TABLE TO oggadm1;
   ```

2. Erteilen Sie die Rechte, die ein Benutzerkonto benötigt, um ein GoldenGate Oracle-Administrator zu sein. Führen Sie das folgende SQL PL/-Programm aus.

   ```
   EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
       grantee                 => 'OGGADM1',
       privilege_type          => 'capture',
       grant_select_privileges => true,
       do_grants               => TRUE);
   ```

   Um Berechtigungen zu widerrufen, verwenden Sie die Prozedur `revoke_admin_privilege` im selben Paket.

Schritt 6: Fügen Sie einen TNS Alias für die Quelldatenbank hinzu

Fügen Sie `$ORACLE_HOME/network/admin/tnsnames.ora` im Oracle-Standardverzeichnis den folgenden Eintrag hinzu, der vom `EXTRACT`-Prozess verwendet werden soll. Weitere Informationen zur Datei `tnsnames.ora` finden Sie in der Oracle-Dokumentation.

```
OGGSOURCE=
   (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-source.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Einrichtung einer Zieldatenbank für die Verwendung mit Oracle GoldenGate auf Amazon RDS

In dieser Aufgabe richten Sie eine Ziel-DB-Instance für die Verwendung mit Oracle ein GoldenGate.

Einrichtungsschritte

- Schritt 1: Setzen Sie den REPLICATION Initialisierungsparameter ENABLE GOLDENGATE _ _ auf true
- Schritt 2: Erstellen Sie ein GoldenGate Oracle-Benutzerkonto in der Zieldatenbank
- Schritt 3: Erteilen von Kontoberechtigungen in der Zieldatenbank
- Schritt 4: Fügen Sie einen TNS Alias für die Zieldatenbank hinzu

Schritt 1: Setzen Sie den REPLICATION Initialisierungsparameter ENABLE GOLDENGATE _ _ auf true

Wenn Sie den `ENABLE_GOLDENGATE_REPLICATION`-Initialisierungsparameter auf `true` einstellen, können Datenbankdienste die logische Replikation unterstützen. Wenn sich Ihre Quelldatenbank auf einer RDS Amazon-DB-Instance befindet, stellen Sie sicher, dass Sie der DB-Instance eine Parametergruppe zugewiesen haben, deren `ENABLE_GOLDENGATE_REPLICATION` Initialisierungsparameter auf `true` gesetzt ist. Weitere Informationen zum Initialisierungsparameter `ENABLE_GOLDENGATE_REPLICATION` finden Sie in der Oracle-Database-Dokumentation.

Schritt 2: Erstellen Sie ein GoldenGate Oracle-Benutzerkonto in der Zieldatenbank

Oracle GoldenGate wird als Datenbankbenutzer ausgeführt und benötigt die entsprechenden Datenbankberechtigungen. Um sicherzustellen, dass es über diese Berechtigungen verfügt, erstellen Sie ein Benutzerkonto in der Zieldatenbank.

Mit der folgenden Anweisung wird ein Benutzer mit dem Namen `oggadm1` erstellt.

```
CREATE TABLESPSACE administrator;
```

```
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE administrator
   TEMPORARY TABLESPACE temp;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Schritt 3: Erteilen von Kontoberechtigungen in der Zieldatenbank

Bei dieser Aufgabe gewähren Sie den Datenbankbenutzern in Ihrer Zieldatenbank die erforderlichen Kontoberechtigungen.

So erteilen Sie Kontoberechtigungen in der Zieldatenbank

1. Erteilen Sie dem GoldenGate Oracle-Benutzerkonto in der Zieldatenbank die erforderlichen Rechte. Im folgenden Beispiel erteilen Sie `oggadm1` Berechtigungen.

   ```
   GRANT CREATE SESSION        TO oggadm1;
   GRANT ALTER SESSION         TO oggadm1;
   GRANT CREATE CLUSTER        TO oggadm1;
   GRANT CREATE INDEXTYPE      TO oggadm1;
   GRANT CREATE OPERATOR       TO oggadm1;
   GRANT CREATE PROCEDURE      TO oggadm1;
   GRANT CREATE SEQUENCE       TO oggadm1;
   GRANT CREATE TABLE          TO oggadm1;
   GRANT CREATE TRIGGER        TO oggadm1;
   GRANT CREATE TYPE           TO oggadm1;
   GRANT SELECT ANY DICTIONARY TO oggadm1;
   GRANT CREATE ANY TABLE      TO oggadm1;
   GRANT ALTER ANY TABLE       TO oggadm1;
   GRANT LOCK ANY TABLE        TO oggadm1;
   GRANT SELECT ANY TABLE      TO oggadm1;
   GRANT INSERT ANY TABLE      TO oggadm1;
   GRANT UPDATE ANY TABLE      TO oggadm1;
   GRANT DELETE ANY TABLE      TO oggadm1;
   ```

2. Erteilen Sie die Rechte, die ein Benutzerkonto benötigt, um ein GoldenGate Oracle-Administrator zu sein. Führen Sie das folgende SQL PL/-Programm aus.

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
    grantee                 => 'OGGADM1',
    privilege_type          => 'apply',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

Um Berechtigungen zu widerrufen, verwenden Sie die Prozedur `revoke_admin_privilege` im selben Paket.

Schritt 4: Fügen Sie einen TNS Alias für die Zieldatenbank hinzu

Fügen Sie `$ORACLE_HOME/network/admin/tnsnames.ora` im Oracle-Standardverzeichnis den folgenden Eintrag hinzu, der vom `REPLICAT`-Prozess verwendet werden soll. Stellen Sie bei Oracle Multitenant-Datenbanken sicher, dass der TNS Alias auf den Dienstnamen von verweist. PDB Weitere Informationen zur Datei `tnsnames.ora` finden Sie in der Oracle-Dokumentation.

```
OGGTARGET=
    (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-target.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Arbeiten mit den EXTRACT und REPLICAT Hilfsprogrammen von Oracle GoldenGate

Die GoldenGate Oracle-Dienstprogramme `EXTRACT` und ich `REPLICAT` arbeiten zusammen, um die Quell- und Zieldatenbanken durch inkrementelle Transaktionsreplikation unter Verwendung von Traildateien synchron zu halten. Alle Änderungen, die an der Quelldatenbank vorgenommen werden`EXTRACT`, werden automatisch von der GoldenGate lokalen Oracle-Datenbank oder dem EC2 Amazon-Instance-Hub erkannt, formatiert und in Traildateien übertragen. Nach dem ersten Ladevorgang werden die Daten aus diesen Dateien gelesen und vom Dienstprogramm `REPLICAT` in die Zieldatenbank repliziert.

Das Oracle-Hilfsprogramm ausführen GoldenGate EXTRACT

Das `EXTRACT`-Dienstprogramm ruft Daten aus der Quelldatenbank ab, konvertiert sie und gibt sie in Trail-Dateien aus. Der grundlegende Prozess ist wie folgt:

1. EXTRACT leitet die Transaktionsdetails in den Speicher oder den temporären Festplattenspeicher weiter.
2. Die Quell-Datenbank führt einen Commit der aktuellen Transaktion durch.
3. EXTRACT schreibt die Transaktionsdetails in eine Trail-Datei.
4. Die Trail-Datei leitet diese Details an den GoldenGate lokalen Oracle-Hub oder den EC2 Amazon-Instance-Hub und dann an die Zieldatenbank weiter.

Mit den folgenden Schritten werden das Dienstprogramm EXTRACT gestartet, die Daten aus EXAMPLE.TABLE der Quelldatenbank OGGSOURCE erfasst und die Pfaddateien erstellt.

Um das Hilfsprogramm auszuführen EXTRACT

1. Konfigurieren Sie die EXTRACT Parameterdatei auf dem GoldenGate Oracle-Hub (lokal oder EC2 Amazon-Instance). Die folgende Liste zeigt eine beispielhafte EXTRACT-Parameterdatei mit dem Namen $GGHOME/dirprm/eabc.prm.

```
EXTRACT EABC

USERID oggadm1@OGGSOURCE, PASSWORD "my-password"
EXTTRAIL /path/to/goldengate/dirdat/ab

IGNOREREPLICATES
GETAPPLOPS
TRANLOGOPTIONS EXCLUDEUSER OGGADM1

TABLE EXAMPLE.TABLE;
```

2. Melden Sie sich auf dem GoldenGate Oracle-Hub bei der Quelldatenbank an und starten Sie die GoldenGate Oracle-Befehlszeilenschnittstelleggsci. Das folgende Beispiel zeigt das Format für die Anmeldung.

```
dblogin oggadm1@OGGSOURCE
```

3. Fügen Sie Transaktionsdaten hinzu, um die zusätzliche Protokollierung für die Datenbanktabelle zu aktivieren.

```
add trandata EXAMPLE.TABLE
```

4. Aktivieren Sie mithilfe der Befehlszeile `ggsci` das Dienstprogramm `EXTRACT` mit den folgenden Befehlen.

```
add extract EABC tranlog, INTEGRATED tranlog, begin now
add exttrail /path/to/goldengate/dirdat/ab
   extract EABC,
   MEGABYTES 100
```

5. Registrieren Sie das Dienstprogramm `EXTRACT` in der Datenbank, damit die archivierten Protokolle nicht gelöscht werden. Dank dieser Aufgabe können Sie alte, nicht festgeschriebene Transaktionen wiederherstellen, wenn dies erforderlich ist. Verwenden Sie den folgenden Befehl, um das Dienstprogramm `EXTRACT` in der Datenbank zu registrieren.

```
register EXTRACT EABC, DATABASE
```

6. Starten Sie das Dienstprogramm `EXTRACT` mit dem folgenden Befehl.

```
start EABC
```

Das GoldenGate REPLICAT Oracle-Hilfsprogramm ausführen

Das Dienstprogram `REPLICAT` sendet Transaktionsinformationen in Pfaddateien an die Zieldatenbank.

Mit den folgenden Schritten wird das Dienstprogramm `REPLICAT` aktiviert und gestartet, sodass es die erfassten Daten in die Tabelle `EXAMPLE.TABLE` der Zieldatenbank `OGGTARGET` replizieren kann.

Um das REPLICATE Hilfsprogramm auszuführen

1. Konfigurieren Sie die `REPLICAT` Parameterdatei auf dem GoldenGate Oracle-Hub (lokal oder EC2 Instanz). Die folgende Liste zeigt eine beispielhafte `REPLICAT`-Parameterdatei mit dem Namen `$GGHOME/dirprm/rabc.prm`.

```
REPLICAT RABC

USERID oggadm1@OGGTARGET, password "my-password"

ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

2. Melden Sie sich bei der Zieldatenbank an und starten Sie die GoldenGate Oracle-Befehlszeilenschnittstelle (`ggsci`). Das folgende Beispiel zeigt das Format für die Anmeldung.

```
dblogin userid oggadm1@OGGTARGET
```

3. Fügen Sie mithilfe der Befehlszeile `ggsci` eine Prüfpunkttabelle hinzu. Der angegebene Benutzer sollte das GoldenGate Oracle-Benutzerkonto sein, nicht der Besitzer des Zieltabellenschemas. Im folgenden Beispiel wird eine Prüfpunkttabelle mit dem Namen `gg_checkpoint` erstellt.

```
add checkpointtable oggadm1.oggchkpt
```

4. Verwenden Sie den folgenden Befehl, um das Dienstprogramm `REPLICAT` zu aktivieren.

```
add replicat RABC EXTTRAIL /path/to/goldengate/dirdat/ab CHECKPOINTTABLE
 oggadm1.oggchkpt
```

5. Starten Sie das Dienstprogramm `REPLICAT` mit dem folgenden Befehl.

```
start RABC
```

Überwachung von Oracle GoldenGate

Wenn Sie Oracle GoldenGate für die Replikation verwenden, stellen Sie sicher, dass der GoldenGate Oracle-Prozess läuft und die Quell- und Zieldatenbanken synchronisiert sind. Sie können die folgenden Überwachungstools verwenden:

- Amazon CloudWatch ist ein Überwachungsdienst, der in diesem Muster zur Überwachung von GoldenGate Fehlerprotokollen verwendet wird.
- Amazon SNS ist ein Nachrichtenbenachrichtigungsdienst, der in diesem Muster zum Senden von E-Mail-Benachrichtigungen verwendet wird.

Eine ausführliche Anleitung finden Sie unter Überwachen von GoldenGate Oracle-Protokollen mithilfe von Amazon CloudWatch.

Fehlerbehebung bei Oracle GoldenGate

In diesem Abschnitt werden die häufigsten Probleme bei der Verwendung von Oracle GoldenGate mit Amazon RDS for Oracle erläutert.

Themen

- Fehler beim Öffnen eines Online-Redo-Protokolls
- Oracle GoldenGate scheint richtig konfiguriert zu sein, aber die Replikation funktioniert nicht
- Integriert REPLICAT langsam, da die Abfrage aktiviert istSYS.“ _ DBA _ APPLY _ CDR _INFO“

Fehler beim Öffnen eines Online-Redo-Protokolls

Erstellen Sie die Datenbanken unbedingt so, dass archivierte Redo-Protokolle aufbewahrt werden. Berücksichtigen Sie die folgenden Hinweise:

- Geben Sie die Dauer für die Protokollaufbewahrung in Stunden an. Der Mindestwert ist eine Stunde.
- Legen Sie die Dauer so fest, dass potenzielle Ausfallzeiten der Quell-Instance, jede mögliche Kommunikationsdauer und mögliche Dauer von Netzwerkproblemen für die Quell-DB-Instance überschritten werden. Bei einer solchen Dauer kann Oracle bei Bedarf Protokolle aus der Quell-DB-Instance GoldenGate wiederherstellen.
- Stellen Sie sicher, dass auf Ihrer Instance genügend Speicherplatz für die Dateien vorhanden ist.

Wenn die Protokollaufbewahrung nicht aktiviert ist oder wenn der Aufbewahrungswert zu klein ist, erhalten Sie eine Meldung ähnlich wie die folgende.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Oracle GoldenGate scheint richtig konfiguriert zu sein, aber die Replikation funktioniert nicht

Bei bereits vorhandenen Tabellen müssen Sie angeben, mit welcher SCN Tabelle Oracle GoldenGate arbeitet.

So beheben Sie dieses Problem

1. Melden Sie sich bei der Quelldatenbank an und starten Sie die GoldenGate Oracle-Befehlszeilenschnittstelle (`ggsci`). Das folgende Beispiel zeigt das Format für die Anmeldung.

   ```
   dblogin userid oggadm1@OGGSOURCE
   ```

2. Richten Sie über die `ggsci` Befehlszeile den Start SCN für den `EXTRACT` Prozess ein. Im folgenden Beispiel wird der SCN auf 223274 für gesetzt. `EXTRACT`

   ```
   ALTER EXTRACT EABC SCN 223274
   start EABC
   ```

3. Melden Sie sich bei der Zieldatenbank an. Das folgende Beispiel zeigt das Format für die Anmeldung.

   ```
   dblogin userid oggadm1@OGGTARGET
   ```

4. Richten Sie über die `ggsci` Befehlszeile den Start SCN für den `REPLICAT` Prozess ein. Im folgenden Beispiel wird der SCN auf 223274 für gesetzt. `REPLICAT`

   ```
   start RABC atcsn 223274
   ```

Integriert REPLICAT langsam, da die Abfrage aktiviert istSYS.“ _ DBA _ APPLY _ CDR _INFO“

Oracle GoldenGate Conflict Detection and Resolution (CDR) bietet grundlegende Routinen zur Konfliktlösung. CDRKann beispielsweise einen eindeutigen Konflikt für eine `INSERT` Anweisung lösen.

Wenn eine Kollision CDR behoben wird, kann es `_DBA_APPLY_CDR_INFO` vorübergehend Datensätze in die Ausnahmetabelle einfügen. Integriertes `REPLICAT` löscht diese Datensätze später. In einem seltenen Szenario kann das integrierte `REPLICAT` eine große Anzahl von Kollisionen verarbeiten, aber ein neues integriertes `REPLICAT` ersetzt es nicht. Anstatt entfernt zu werden, sind die vorhandenen Zeilen in `_DBA_APPLY_CDR_INFO` verwaist. Alle neuen integrierten `REPLICAT`-Prozesse verlangsamen sich, da sie verwaiste Zeilen in `_DBA_APPLY_CDR_INFO` abfragen.

Verwenden Sie das RDS Amazon-Verfahren`_DBA_APPLY_CDR_INFO`, um alle Zeilen von zu entfernen`rdsadmin.rdsadmin_util.truncate_apply$_cdr_info`. Dieses Verfahren wird im

Rahmen des Release- und Patch-Updates vom Oktober 2020 veröffentlicht. Der Prozess ist in den folgenden Datenbankversionen verfügbar:

- Version 21.0.0.0.ru-2022-01.rur-2022-01.r1 und höher
- Version 19.0.0.0.ru-2020-10.rur-2020-10.r1 und höher

Im folgenden Beispiel wird die Tabelle abgeschnitten `_DBA_APPLY_CDR_INFO`.

```
SET SERVEROUTPUT ON SIZE 2000
EXEC rdsadmin.rdsadmin_util.truncate_apply$_cdr_info;
```

Verwenden des Oracle Repository Creation Utility (RCU) in RDS for Oracle

Sie können Amazon RDS zum Bereitstellen einer RDS-on-Oracle-DB-Instance verwenden, die Schemata zur Unterstützung Ihrer Oracle-Fusion-Middleware-Komponenten enthält. Bevor Sie Fusion-Middleware-Komponenten verwenden können, müssen Sie dafür Schemata in Ihrer Datenbank erstellen und mit Daten füllen. Sie erstellen und füllen die Schemata mithilfe des Oracle Repository Creation Utility (RCU).

Unterstützte Versionen und Lizenzoptionen für RCU

Amazon RDS unterstützt nur Oracle Repository Creation Utility (RCU) Version 12c. Sie können das RCU in folgenden Konfigurationen verwenden:

- RCU 12c mit Oracle-Datenbank 21c
- RCU 12c mit Oracle-Datenbank 19c

Bevor Sie RCU verwenden können, stellen Sie sicher, dass Sie folgendes tun:

- Besorgen Sie sich eine Lizenz für Oracle Fusion Middleware.
- Befolgen Sie die Oracle-Lizenzierungsrichtlinien für die Oracle-Datenbank, die das Repository hostet. Weitere Informationen finden Sie im Benutzerhandbuch mit Lizenzinformationen für Oracle Fusion Middleware in der Oracle-Dokumentation.

Fusion MiddleWare unterstützt Repositorys auf Oracle Database Enterprise Edition und Standard Edition 2. Oracle empfiehlt Enterprise Edition für Produktionsinstallationen, die Partitionierung und Installationen für Online-Indexwiederaufbau erfordern.

Bevor Sie Ihre RDS-for-Oracle-DB-Instance erstellen, vergewissern Sie sich, dass Ihre Oracle-Datenbankversion die Komponenten unterstützt, die Sie einsetzen möchten. Die Anforderungen für die Fusion Middleware-Komponenten und -Versionen, die Sie bereitstellen möchten, finden Sie in der Zertifizierungsmatrix. Weitere Informationen finden Sie unter Oracle Fusion Middleware Supported System Configurations in der Oracle-Dokumentation.

Amazon RDS unterstützt wie erforderlich Versions-Upgrades für Oracle-Datenbanken. Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Anforderungen und Einschränkungen für RCU

Um RCU verwenden zu können, benötigen Sie eine Amazon VPC. Ihre Amazon-RDS-DB-Instance muss nur für Ihre Fusion-Middleware-Komponenten verfügbar sein. Sie muss nicht vom öffentlichen Internet aus erreichbar sein. Hosten Sie Ihre Amazon-RDS-DB-Instance also in einem privaten Subnetz, welches ein höheres Sicherheitsniveau bietet. Sie benötigen auch eine RDS-for-Oracle-DB-Instance. Weitere Informationen finden Sie unter Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung.

Sie können die Schemata für beliebige Fusion Middleware-Komponenten in Ihrer Amazon RDS-DB-Instance speichern. Für die folgenden Schemata wurde eine korrekte Installation verifiziert:

- Analytics (ACTIVITIES)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Discussions (DISCUSSIONS)
- Metadata Services (MDS)
- Oracle Business Intelligence (BIPLATFORM)
- Oracle Platform Security Services (OPSS)
- Portal and Services (WEBCENTER)
- Portlet Producers (PORTLET)
- Service Table (STB)
- SOA-Infrastruktur (SOAINFRA)
- User Messaging Service (UCSUMS)
- WebLogic Dienste (WLS)

Richtlinien für das Arbeiten mit RCU

Empfehlungen für die Arbeit mit Ihrer DB-Instance in diesem Szenario:

- Wir empfehlen, dass Sie Multi-AZ für Produktionsworkloads nutzen. Weitere Informationen zum Arbeiten mit mehreren Availability Zones finden Sie unter Regionen, Availability Zones und Local Zones.

- Für zusätzliche Sicherheit empfiehlt Oracle die Verwendung von Transparent Data Encryption (TDE) zur Verschlüsselung Ihrer ruhenden Daten. Wenn Sie über eine Enterprise Edition-Lizenz verfügen, die die Advanced Security Option umfasst, können Sie die Verschlüsselung von ruhenden Daten mit der Option TDE aktivieren. Weitere Informationen finden Sie unter Oracle Transparent Data Encryption.

  Amazon RDS bietet auch eine Verschlüsselung ruhender Daten für alle Datenbank-Editionen. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
- Konfigurieren Sie Ihre VPC-Sicherheitsgruppen so, dass sie Kommunikation zwischen Ihren Anwendungsservern und Ihrer Amazon RDS-DB-Instance erlauben. Die Anwendungsserver, die die Fusion Middleware-Komponenten bereitstellen, können sich auf Amazon EC2 oder vor Ort befinden.

Ausführen von RCU

Verwenden Sie das Oracle Repository Creation Utility (RCU), um die Schemata zur Unterstützung Ihrer Fusion-Middleware-Komponenten zu erstellen und mit Daten zu füllen. Sie können RCU auf unterschiedliche Weise ausführen.

Themen

- Ausführen von RCU mithilfe der Befehlszeile in einem Schritt
- Ausführen von RCU mithilfe der Befehlszeile in mehreren Schritten
- Ausführen von RCU im interaktiven Modus

Ausführen von RCU mithilfe der Befehlszeile in einem Schritt

Wenn Sie keine Ihrer Schemata bearbeiten müssen, bevor Sie sie füllen, können Sie RCU in einem einzigen Schritt ausführen. Siehe anderenfalls den folgenden Abschnitt für die Ausführung von RCU in mehreren Schritten.

Sie können RCU mithilfe des Befehlszeilenparameters ohne Benutzereingriffe ausführe `-silent`. Wenn Sie RCU im unbeaufsichtigten Modus ausführen, können Sie die Eingabe von Kennwörtern in der Befehlszeile vermeiden, indem Sie eine Textdatei mit den Kennwörtern erstellen. Erstellen Sie eine Textdatei mit dem Passwort für `dbUser` in der ersten Zeile und dem Passwort für jede Komponente jeweils auf einer folgenden Zeile. Sie geben den Namen der Passwortdatei als letzten Parameter mit dem RCU-Befehl an.

Example

Das folgende Beispiel erstellt und füllt Schemata für die SOA-Infrastrukturkomponente (und ihre Abhängigkeiten) in einem einzigen Schritt.

Für LinuxmacOS, oderUnix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Weitere Informationen finden Sie unter Running Repository Creation Utility from the Command Line in der Oracle-Dokumentation.

Ausführen von RCU mithilfe der Befehlszeile in mehreren Schritten

Führen Sie RCU in mehreren Schritten aus, wenn Sie Ihre Schema-Skripts manuell bearbeiten möchten:

1. Führen Sie RCU im Modus Prepare Scripts for System Load aus, indem Sie den Befehlszeilenparameter `-generateScript` zur Erstellung der Skripts für Ihre Schemata angeben.
2. Bearbeiten Sie das generierte Skript manuell und führen Sie es au `script_systemLoad.sql`.
3. Führen Sie RCU erneut im Modus Perform Product Load aus, indem Sie den Befehlszeilenparameter `-dataLoad` zum Füllen der Schemata angeben.

4. Führen Sie das generierte Bereinigungsskript `script_postDataLoad.sql` aus.

Geben Sie den Befehlszeilenparameter `-silent` an, um RCU im Hintergrundmodus auszuführen. Wenn Sie RCU im silent-Modus ausführen, können Sie die Eingabe von Passwörtern in die Befehlszeile vermeiden, indem Sie eine Textdatei mit den Passwörtern anlegen. Erstellen Sie eine Textdatei mit dem Passwort für `dbUser` in der ersten Zeile und dem Passwort für jede Komponente jeweils auf einer folgenden Zeile. Geben Sie den Namen der Passwortdatei als letzten Parameter mit dem RCU-Befehl an.

Example

Im folgenden Beispiel werden Schema-Skripts für die SOA-Infrastrukturkomponente und deren Abhängigkeiten erstellt.

Für LinuxmacOS, oderUnix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-generateScript \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
[-encryptTablespace true] \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-scriptLocation /tmp/rcuscripts \
-f < /tmp/passwordfile.txt
```

Jetzt können Sie das generierte Skript bearbeiten, eine Verbindung zu Ihrer Oracle-DB-Instance aufbauen und das Skript ausführen. Das generierte Skript hat den Namen

`script_systemLoad.sql`. Weitere Information über das Verbinden mit Ihrer Oracle-DB-Instance finden Sie unter Schritt 3: Verbinden Ihres SQL-Clients mit einer Oracle-DB-Instance.

Das folgende Beispiel füllt die Schemata für die SOA-Infrastrukturkomponente (und deren Abhängigkeiten).

Für LinuxmacOS, oderUnix:

```
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-dataLoad \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Zum Abschluss bauen Sie eine Verbindung zu Ihrer Oracle-DB-Instance auf und führen das Bereinigungsskript aus. Das Skript hat den Namen `script_postDataLoad.sql`.

Weitere Informationen finden Sie unter Running Repository Creation Utility from the Command Line in der Oracle-Dokumentation.

Ausführen von RCU im interaktiven Modus

Führen Sie die RCU im interaktiven Modus aus, um die grafische RCU-Benutzeroberfläche zu verwenden. Fügen Sie den Parameter `-interactive` hinzu und schließen Sie den Parameter `-silent` aus. Weitere Informationen finden Sie unter Understanding Repository Creation Utility Screens in der Oracle-Dokumentation.

Example

Das folgende Beispiel startet RCU im interaktiven Modus und füllt die Verbindungsinformation vorab aus.

Für LinuxmacOS, oderUnix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-interactive \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal
```

Fehlerbehebung für RCU

Beachten Sie die folgenden Probleme.

Themen

- Von Oracle verwaltete Dateien (OMF)
- Objektberechtigung
- Enterprise Scheduler-Dienst

Von Oracle verwaltete Dateien (OMF)

Amazon RDS verwendet OMF-Datendateien zur einfacheren Speicherverwaltung. Sie können Tablespace-Attribute anpassen, z. B. "size" und "extent management". Wenn Sie jedoch beim Ausführen von RCU einen Datendateinamen angeben, schlägt der Tablespace-Code mit `ORA-20900` fehl. Sie können RCU wie folgt mit OMF verwenden:

- Verwenden Sie in RCU 12.2.1.0 und neuer den Befehlszeilenparameter `-honorOMF`.
- Verwenden Sie in RCU 12.1.0.3 und neuer mehrere Schritte und bearbeiten Sie das generierte Skript. Weitere Informationen finden Sie unter Ausführen von RCU mithilfe der Befehlszeile in mehreren Schritten.

Objektberechtigung

Da es sich bei Amazon RDS um einen verwalteten Service handelt, haben Sie keinen vollständigen SYSDBA-Zugriff auf Ihre RDS-für-Oracle-DB-Instance. Jedoch unterstützt RCU 12c Benutzer mit geringeren Berechtigungen. In den meisten Fällen reicht die Berechtigung des Masterbenutzers aus, um Repositorys zu erstellen.

Das Hauptkonto kann direkt Berechtigungen gewähren, die ihm bereits erteilt wurden `WITH GRANT OPTION`. In einigen Fällen kann RCU beim Versuch, SYS-Objektberechtigungen zu erteilen, mit `ORA-01031` fehlschlagen. Sie können es erneut versuchen und die `rdsadmin_util.grant_sys_object` gespeicherte Prozedur ausführen, wie im folgenden Beispiel gezeigt:

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('GV_$SESSION','MY_DBA','SELECT');
END;
/
```

Wenn Sie versuchen, SYS-Berechtigungen für das SCHEMA_VERSION_REGISTRY-Objekt zu erteilen, schlägt der Vorgang möglicherweise mit fehl. Sie können die Tabelle SCHEMA_VERSION_REGISTRY $ und die Anzeige SCHEMA_VERSION_REGISTRY mit dem Namen des Schemaeigentümers qualifizieren, der SYSTEM lautet, und den Vorgang erneut versuchen. Oder Sie können ein Synonym erstellen. Melden Sie sich als Hauptbenutzer an und führen Sie die folgenden Anweisungen aus:

```
CREATE OR REPLACE VIEW SYSTEM.SCHEMA_VERSION_REGISTRY
  AS SELECT * FROM SYSTEM.SCHEMA_VERSION_REGISTRY$;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY FOR
 SYSTEM.SCHEMA_VERSION_REGISTRY;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY$ FOR SCHEMA_VERSION_REGISTRY;
```

Enterprise Scheduler-Dienst

Wenn Sie mithilfe von RCU ein Enterprise Scheduler Service-Repository löschen, schlägt RCU eventuell mit `Error: Component drop check failed` fehl.

Konfigurieren von Oracle Connection Manager auf einer Amazon-EC2-Instance

Oracle Connection Manager (CMAN) ist ein Proxy-Server, der Verbindungsanforderungen an Datenbankserver oder andere Proxy-Server weiterleitet. Sie können mit CMAN Folgendes konfigurieren:

Zugriffskontrolle

Sie können Regeln erstellen, die von Benutzern angegebene Client-Anforderungen herausfiltern und andere akzeptieren.

Session-Multiplexing

Sie können mehrere Client-Sitzungen über eine Netzwerkverbindung zu einem gemeinsam genutzten Serverziel leiten.

In der Regel befindet sich CMAN auf einem Host, der vom Datenbankserver und den Client-Hosts getrennt ist. Weitere Informationen finden Sie unter Configuring Oracle Connection Manager in der Dokumentation zur Oracle Database.

Themen

- Unterstützte Versionen und Lizenzoptionen für CMAN
- Anforderungen und Einschränkungen für CMAN
- Konfigurieren von CMAN

Unterstützte Versionen und Lizenzoptionen für CMAN

CMAN unterstützt die Enterprise Edition aller Versionen von Oracle Database, die Amazon RDS unterstützt. Weitere Informationen finden Sie unter RDS für Oracle releases.

Sie können Oracle Connection Manager auf einem anderen Host als dem Host installieren, auf dem Oracle Database installiert ist. Sie benötigen keine separate Lizenz für den Host, der CMAN ausführt.

Anforderungen und Einschränkungen für CMAN

Um ein vollständig verwaltetes Benutzererlebnis zu bieten, schränkt Amazon RDS den Zugriff auf das Betriebssystem ein. Sie können keine Datenbankparameter ändern, die Betriebssystemzugriff

erfordern. Daher unterstützt Amazon RDS keine Funktionen von CMAN, für die Sie sich beim Betriebssystem anmelden müssen.

Konfigurieren von CMAN

Wenn Sie CMAN konfigurieren, führen Sie den größten Teil der Arbeit außerhalb Ihrer RDS-for-Oracle-Datenbank aus.

Themen

- Schritt 1: Konfigurieren von CMAN auf einer Amazon-EC2-Instance in derselben VPC wie die RDS-for-Oracle-Instance
- Schritt 2: Konfigurieren von Datenbankparametern für CMAN
- Schritt 3: Zuordnen Ihrer DB-Instance zur Parametergruppe

Schritt 1: Konfigurieren von CMAN auf einer Amazon-EC2-Instance in derselben VPC wie die RDS-for-Oracle-Instance

Um zu erfahren, wie man CMAN einrichtet, folgen Sie den detaillierten Anweisungen im Blogbeitrag Konfigurieren und Verwenden von Oracle Connection Manager auf Amazon EC2 für Amazon RDS for Oracle.

Schritt 2: Konfigurieren von Datenbankparametern für CMAN

Legen Sie für CMAN-Funktionen wie Traffic Director Mode und Session-Multiplexing den `REMOTE_LISTENER`-Parameter auf die Adresse der CMAN-Instance in einer DB-Parametergruppe fest. Betrachten Sie das folgenden Szenario:

- Die CMAN-Instance befindet sich auf einem Host mit der IP-Adresse `10.0.159.100` und nutzt Port `1521`.
- Die Datenbanken `orcla`, `orclb` und `orclc` befinden sich auf separaten RDS-for-Oracle-DB-Instances.

Die folgende Tabelle zeigt, wie der Wert `REMOTE_LISTENER` festgelegt wird. Der Wert `LOCAL_LISTENER` wird automatisch von Amazon RDS festgelegt.

Name der DB-Instance	DB-Instance-IP	Wert lokaler Listener (automatisch festgelegt)	Wert Remote-Listener (vom Benutzer festgelegt)
orcla	10.0.159.200	`( address= (protocol=tcp) (host=10.0.159.200) (port=1521) )`	10.0.159.100:1521
orclb	10.0.159.300	`( address= (protocol=tcp) (host=10.0.159.300) (port=1521) )`	10.0.159.100:1521
orclc	10.0.159.400	`( address= (protocol=tcp) (host=10.0.159.400) (port=1521) )`	10.0.159.100:1521

Schritt 3: Zuordnen Ihrer DB-Instance zur Parametergruppe

Erstellen oder ändern Sie Ihre DB-Instance zur Verwendung der Parametergruppe, die Sie in Schritt 2: Konfigurieren von Datenbankparametern für CMAN konfiguriert haben. Weitere Informationen finden Sie unter Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon.

Installieren einer Siebel-Datenbank auf Oracle auf Amazon RDS

Sie können Amazon RDS verwenden, um eine Siebel-Datenbank in einer Oracle-DB-Instance zu hosten. Die Siebel-Datenbank ist Teil der Siebel Customer Relationship Management (CRM)-Anwendungsarchitektur. Zur Illustration vgl. Allgemeine Architektur der Siebel Business-Anwendung.

Im folgenden Thema finden Sie Hilfe beim Einrichten einer Siebel-Datenbank auf einer Oracle DB-Instance auf Amazon RDS. Sie können sich auch informieren, wie Amazon Web Services verwendet werden kann, um Unterstützung für die von der Siebel-CRM-Anwendungsarchitektur erforderlichen Komponenten zu bieten.

> **Note**
>
> Sie müssen das Hauptbenutzerkonto verwenden, um eine Siebel-Datenbank auf Oracle in Amazon RDS zu installieren. Sie benötigen das Sonderrecht `SYSDBA` nicht, die Sonderrechte für den Hauptbenutzer sind ausreichend. Weitere Informationen finden Sie unter Berechtigungen von Hauptbenutzerkonten.

Lizenzierung und Versionen

Sie müssen Ihre eigene Lizenz für die Oracle-Datenbank und für Siebel besitzen, um eine Siebel-Datenbank auf Amazon RDS zu installieren. Sie müssen eine entsprechende Oracle-Datenbank-Lizenz für die DB-Instance-Klasse und die Oracle-Datenbank-Edition besitzen (mit der Lizenz für Software-Updates und Support). Weitere Informationen finden Sie unter RDSfür Oracle-Lizenzoptionen.

Oracle Database Enterprise Edition ist die einzige Edition, die von Siebel für dieses Szenario zertifiziert ist. Amazon RDS unterstützt Siebel CRM-Version 15.0 oder 16.0.

Amazon RDS unterstützt Versions-Upgrades für Datenbanken. Weitere Informationen finden Sie unter Upgrade der Engine-Version für eine DB-Instance.

Bevor Sie beginnen

Bevor Sie anfangen, benötigen Sie eine Amazon VPC. Da Ihre Amazon RDS-DB-Instance nur für Ihren Siebel Enterprise-Server und nicht für das öffentliche Internet verfügbar sein soll, wird Ihre Amazon RDS-DB-Instance in einem privaten Subnetz gehostet, was eine höhere Sicherheit bietet.

Weitere Informationen zum Erstellen einer Amazon VPC für die Verwendung mit Siebel CRM finden Sie unter Erstellen einer Oracle-DB-Instance und Herstellen einer Verbindung.

Bevor Sie beginnen, benötigen Sie auch eine Oracle-DB-Instance. Weitere Informationen zum Erstellen einer Oracle-DB-Instance für die Verwendung mit Siebel CRM finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Installieren und Konfigurieren einer Siebel-Datenbank

Nachdem Sie Ihre Oracle-DB-Instance erstellt haben, können Sie Ihre Siebel-Datenbank installieren. Für die Installation der Datenbank erstellen Sie Konten für Tabellenbesitzer und -Administratoren, installieren Sie gespeicherte Vorgänge und Funktionen und führen Sie den Siebel-Datenbank-Konfigurationsassistenten aus. Weitere Informationen finden Sie unter Installieren der Siebel-Datenbank auf dem RDBMS.

Sie müssen das Hauptbenutzerkonto verwenden, um den Siebel-Datenbank-Konfigurationsassistenten auszuführen. Sie benötigen das Sonderrecht `SYSDBA` nicht, die Sonderrechte für den Hauptbenutzer sind ausreichend. Weitere Informationen finden Sie unter Berechtigungen von Hauptbenutzerkonten.

Verwenden von anderen Amazon RDS-Funktionen mit einer Siebel-Datenbank

Nachdem Sie Ihre Oracle-DB-Instance erstellt haben, können Sie zusätzliche Amazon RDS-Funktionen verwenden und so Ihre Siebel-Datenbank anpassen.

Erfassen von Statistiken mit der Oracle-Statspack-Option

Sie können zu Ihrer DB-Instance Funktionen hinzufügen, indem Sie die Optionen in den DB-Optionsgruppen verwenden. Als Sie Ihre Oracle-DB-Instance erstellt haben, war für Sie die Standard-DB-Optionsgruppe eingestellt. Wenn Sie Funktionen zu Ihrer Datenbank hinzufügen möchten, können Sie eine neue Optionsgruppe für Ihre DB-Instance erstellen.

Wenn Sie Statistiken über Ihre Siebel-Datenbank sammeln möchten, können Sie die Funktion Oracle Statspack hinzufügen. Weitere Informationen finden Sie unter Oracle Statspack.

Einige Optionsänderungen in der DB-Instance werden sofort übernommen und einige während des nächsten Wartungszeitraums. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen. Sie können eine benutzerdefinierte Optionsgruppe erstellen, Ihre DB-Instance ändern und diese anfügen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Leistungsoptimierung mit Parametern

Sie können Ihre DB-Engine-Konfiguration über die Parametereinstellungen in Ihrer DB-Parametergruppe verwalten. Als Sie Ihre Oracle-DB-Instance erstellt haben, war für Sie die Standard-DB-Parametergruppe eingestellt. Wenn Sie Ihre Datenbank-Konfiguration anpassen möchten, können Sie eine neue Parametergruppe für Ihre DB-Instance erstellen.

Wenn Sie einen Parameter ändern, werden die Änderungen, je nach Parametertyp, sofort oder nach einem manuellen Neustart der DB-Instance übernommen. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS. Sie können eine benutzerdefinierte Parametergruppe erstellen, Ihre DB-Instance ändern und diese anfügen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sie können bestimmte Parameter anpassen, um Ihre Oracle-DB-Instance für Siebel CRM zu optimieren. Die folgende Tabelle zeigt einige empfohlene Parametereinstellungen. Weitere Informationen über Leistungsoptimierung von Siebel CRM finden Sie unter Siebel-CRM-Leistungsoptimierungsleitfaden.

Parametername	Standardwert	Anleitung für eine optimale Siebel CRM-Leistung
`_always_semi_join`	`CHOOSE`	`OFF`
`_b_tree_bitmap_plans`	`TRUE`	`FALSE`
`_like_with_bind_as_equality`	`FALSE`	`TRUE`
`_no_or_expansion`	`FALSE`	`FALSE`
`_optimizer_join_sel_sanity_check`	`TRUE`	`TRUE`

Parametername	Standardwert	Anleitung für eine optimale Siebel CRM-Leistung
`_optimizer_max_permutations`	2000	100
`_optimizer_sortmerge_join_enabled`	TRUE	FALSE
`_partition_view_enabled`	TRUE	FALSE
`open_cursors`	300	Mindestens **2000**.

Erstellen von Snapshots

Nachdem Sie Ihre Siebel-Datenbank erstellt haben, können Sie die Datenbank mithilfe der Snapshot-Funktionen von Amazon RDS kopieren. Weitere Informationen erhalten Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS und Wiederherstellung auf einer DB-Instance.

Unterstützung für andere Siebel CRM-Komponenten

Zusätzlich zu Ihrer Siebel-Datenbank können Sie auch Amazon Web Services verwenden, um anderen Komponenten in Ihrer Siebel CRM-Anwendungsarchitektur Unterstützung zu bieten. Weitere Informationen zur Unterstützung von Amazon AWS für zusätzliche Siebel CRM-Komponenten finden Sie in der folgenden Tabelle.

Siebel CRM-Komponente	AWS Amazon-Unterstützung
Siebel Enterprise (mit einem oder mehreren Siebel Servern)	Sie können Ihre Siebel-Server auf Amazon Elastic Compute Cloud (Amazon EC2)-Instances hosten. Mit Amazon EC2 können Sie so viele virtuelle Server wie

Siebel CRM-Komponente	AWS Amazon-Unterstützung
	nötig starten. Mithilfe von Amazon EC2 können Sie Ihre Nutzung einfach ausbauen oder reduzieren, um so erforderlichen Änderungen zu begegnen. Weitere Informationen finden Sie unter Was ist Amazon EC2? Sie können Ihre Server in der selben VPC wie Ihre DB-Instance anlegen und die VPC-Sicherheitsgruppe für Zugriffe auf die Datenbank verwenden. Weitere Informationen finden Sie unter Arbeiten mit einer DB-Instance in einer VPC.
Webserver (mit Siebel Web Server-Erweiterungen)	Sie können mehrere Web-Server auf mehreren EC2-Instance installieren. Sie können anschließend Elastic Load Balancing verwenden, um eingehenden Datenverkehr zwischen den Instances zu verteilen. Weitere Informationen finden Sie unter Was ist Elastic Load Balancing?
Siebel-Gateway-Namensserver	Sie können Ihren Siebel-Gateway-Namensserver auf einer EC2-Instance hosten. Anschließend können Sie Ihren Server in der selben VPC wie die DB-Instance anlegen und die VPC-Sicherheitsgruppe für Zugriffe auf die Datenbank verwenden. Weitere Informationen finden Sie unter Arbeiten mit einer DB-Instance in einer VPC.

Versionshinweise zur Oracle-Datenbank-Engine

Ihre Amazon RDS for Oracle DB-Instances bleiben mit Updates auf dem neuesten Stand. Wenn Sie die Updates anwenden, können Sie versichert sein, dass Ihre DB-Instance auf einer Version der Datenbank-Software ausgeführt wird, die sowohl von Oracle als auch von Amazon erfolgreich getestet wurde. Wir bieten keine Unterstützung von einmaligen Patches für einzelne RDS für Oracle-DB-Instances.

Sie können eine beliebige aktuell unterstützte Oracle Datenbank-Version festlegen, wenn Sie eine neue DB-Instance erstellen. Sie können die Hauptversionen, wie z. B. Oracle Database 19c sowie eine beliebige unterstützte Unterversion für die festgelegte Hauptversion festlegen. Wenn keine Version angegeben wird, verwendet Amazon RDS standardmäßig eine unterstützte Version - in der Regel die aktuelle Version. Wenn die Hauptversion, jedoch nicht die Unterversion, festgelegt ist, verwendet Amazon RDS standardmäßig den letzten Release der Hauptversion, die Sie festgelegt haben. Eine Liste aller unterstützten Versionen und Standardversionen für neu erstellte DB-Instances können Sie mit dem AWS CLI-Befehl `describe-db-engine-versions` aufrufen.

Weitere Informationen zu den Oracle-Datenbank-Versionen, die von Amazon RDS unterstützt werden, finden Sie in den Versionshinweisen zu Amazon RDS für Oracle.

Amazon RDS für PostgreSQL

Amazon RDS unterstützt DB-Instances für mehrere Versionen und Editionen von PostgreSQL. Eine Liste der verfügbaren Versionen finden Sie unter Verfügbare PostgreSQL-Datenbankversionen.

Sie können DB-Instances und DB-Snapshots, Wiederherstellungen und Backups erstellen. point-in-time DB-Instances mit PostgreSQL unterstützen Multi-AZ-Bereitstellungen, Read Replicas sowie bereitgestellte IOPS und können innerhalb einer Virtual Private Cloud (VPC) erstellt werden. Für die Verbindung zu einer DB-Instance mit PostgreSQL können Sie auch Secure Socket Layer (SSL) nutzen.

Bevor Sie eine DB-Instance erstellen, stellen Sie sicher, dass Sie die Schritte in Einrichtung Ihrer Amazon RDS-Umgebung ausführen.

Sie können eine beliebige Standard-SQL-Client-Anwendung verwenden, um von Ihrem Computer aus Befehle für die Instance auszuführen. Beispiele solcher Anwendungen: pgAdmin, ein beliebtes Open-Source-Verwaltungs- und Entwicklungstool für PostgreSQL, oder auch psql, ein Befehlszeilen-Dienstprogramm, das Teil einer PostgreSQL-Installation ist. Um eine verwaltete Service-Erfahrung zu bieten, ermöglicht Amazon RDS keinen Host-Zugriff auf DB-Instances. Eingeschränkt wird auch der Zugriff auf bestimmte Systemprozeduren und Tabellen, für die erweiterte Berechtigungen erforderlich sind. Amazon RDS unterstützt den Zugriff auf Datenbanken auf einer DB-Instance unter Verwendung jeder Standard-SQL-Client-Anwendung. Amazon RDS unterstützt keinen direkten Zugriff auf den Host auf eine DB-Instance mithilfe von Telnet oder Secure Shell (SSH).

Amazon RDS for PostgreSQL ist im Einklang mit vielen Industriestandards. Sie können beispielsweise Amazon-RDS-for-PostgreSQL-Datenbanken verwenden, um HIPAA-konforme Anwendungen zu erstellen und Zustandsdaten zu speichern. Dazu gehört die Speicherung geschützter Zustandsdaten (protected health information, PHI) im Rahmen eines abgeschlossenen Business Associate Agreement (BAA) mit AWS. Amazon RDS for PostgreSQL erfüllt auch die Anforderungen des Federal Risk and Authorization Management Program (FedRAMP). Amazon RDS for PostgreSQL hat eine vorläufige Betriebsgenehmigung (P-ATO) des FedRAMP Joint Authorization Board (JAB) für die FedRAMP HIGH Baseline in den Regionen erhalten. AWS GovCloud (US) Weitere Informationen über unterstützte Compliance-Standards finden Sie unter AWS Cloud-Compliance.

Um PostgreSQL-Daten in eine DB-Instance zu importieren, befolgen Sie bitte die Informationen im Abschnitt Daten in Postgre SQL auf Amazon importieren RDS.

> **Important**
>
> Wenn Sie auf ein Problem mit Ihrer RDS for PostgreSQL-DB-Instance stoßen, benötigt Ihr AWS Support-Mitarbeiter möglicherweise weitere Informationen zum Zustand Ihrer Datenbanken. Ziel ist es, sicherzustellen, dass der AWS Support die erforderlichen Informationen so schnell wie möglich erhält.
> Sie können PG Collector verwenden, um wertvolle Datenbankinformationen in einer konsolidierten HTML-Datei zu sammeln. Weitere Informationen zu PG Collector, zur Ausführung und zum Herunterladen des HTML-Berichts finden Sie unter PG Collector.
> Nach erfolgreichem Abschluss und sofern nicht anders angegeben, gibt das Skript die Ausgabe in einem lesbaren HTML-Format zurück. Das Skript ist so konzipiert, dass alle Daten oder Sicherheitsdetails aus dem HTML-Code ausgeschlossen werden, die Ihr Unternehmen gefährden könnten. Es nimmt auch keine Änderungen an Ihrer Datenbank oder ihrer Umgebung vor. Wenn Sie jedoch Informationen im HTML-Code finden, die Sie nicht teilen möchten, können Sie die problematischen Informationen entfernen, bevor Sie den HTML-Code hochladen. Wenn der HTML-Code akzeptabel ist, laden Sie ihn über den Abschnitt Anlagen in den Falldetails Ihres Support-Falls hoch.

Themen

- Allgemeine Verwaltungsaufgaben für Amazon RDS for Postgre SQL
- Arbeiten mit der Datenbank-Vorschauumgebung
- Verfügbare PostgreSQL-Datenbankversionen
- Den inkrementellen Release-Prozess von RDS für PostgreSQL verstehen
- Unterstützte SQL Postgre-Erweiterungsversionen
- Arbeiten mit SQL Postgre-Funktionen, die von Amazon RDS für Postgre unterstützt werden SQL
- Herstellen einer Verbindung zu einer DB-Instance, in der die PostgreSQL-Datenbank-Engine ausgeführt wird
- Verbindungen zu RDS für Postgre sichern SQL mit/SSLTLS
- Verwenden der Kerberos-Authentifizierung mit Amazon RDS for PostgreSQL
- Verwenden eines benutzerdefinierten DNS Servers für ausgehenden Netzwerkzugriff
- Upgrades der RDS for PostgreSQL-DB-Engine
- Aktualisierung einer SQL Postgre-DB-Snapshot-Engine-Version
- Arbeiten mit Read Replicas in Amazon RDS for PostgreSQL

- Verbesserung der Abfrageleistung für RDS für PostgreSQL mit Amazon-RDS-optimierten Lesevorgängen
- Daten in Postgre SQL auf Amazon importieren RDS
- Exportieren von Daten aus einem/einer RDS for PostgreSQL-DB-Instance zu Amazon S3
- Aufrufen einer AWS Lambda Funktion aus einem)
- Häufige DBA-Aufgaben für Amazon RDS for PostgreSQL
- Optimierung mit Warteereignissen für RDS für PostgreSQL
- Optimiert RDSfür Postgre SQL mit proaktiven Erkenntnissen von Amazon DevOps Guru
- Verwenden von PostgreSQL-Erweiterungen mit Amazon RDS for PostgreSQL
- Arbeiten mit den unterstützten Foreign Data Wrappern für RDS SQL
- Arbeiten mit Trusted Language Extensions für PostgreSQL

Allgemeine Verwaltungsaufgaben für Amazon RDS for Postgre SQL

Im Folgenden sind die allgemeinen Verwaltungsaufgaben aufgeführt, die Sie mit einer Amazon RDS for SQL Postgre-DB-Instance ausführen, mit Links zu der entsprechenden Dokumentation für jede Aufgabe.

Aufgabenbereich	Relevante Dokumentation
Amazon RDS für die erstmalige Nutzung einrichten Sie müssen einige Voraussetzungen erfüllen, um Ihre DB-Instance erstellen zu können. Beispielsweise werden DB-Instances standardmäßig mit einer Firewall erstellt, die den Zugriff auf die Instance verhindert. Sie müssen also eine Sicherheitsgruppe mit den korrekten IP-Adressen und der Netzwerkkonfiguration erstellen, um auf die DB-Instance zugreifen zu können.	Einrichtung Ihrer Amazon RDS-Umgebung
Grundlegendes zu Amazon RDS DB-Instances Wenn Sie eine DB-Instance für Produktionszwecke erstellen, sollten Sie wissen, wie Instance-Klassen, Speichertypen und Provisioned in Amazon IOPS RDS funktionieren.	Amazon RDS-Speichertypen Bereitgestellter IOPS SSD-Speicher

Aufgabenbereich	Relevante Dokumentation
Verfügbare SQL Postgre-Versionen finden Amazon RDS unterstützt mehrere Versionen von PostgreSQL.	Verfügbare PostgreSQL-Datenbankversionen
Einrichten von hoher Verfügbarkeit und Failover-Unterstützung Bei einer DB-Instance für die Produktion sollten Multi-AZ-Bereitstellungen eingesetzt werden. Multi-AZ-Bereitstellungen bieten eine erhöhte Verfügbarkeit, eine längere Lebensdauer von Daten sowie eine höhere Fehlertoleranz für DB-Instances.	Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS
Das Amazon Virtual Private Cloud (VPC) -Netzwerk verstehen Wenn Ihr AWS Konto eine Standardinstanz hatVPC, wird Ihre DB-Instance automatisch innerhalb der Standardinstanz erstelltV PC. In einigen Fällen hat Ihr Konto möglicherweise keinen StandardVPC, und Sie möchten die DB-Instance möglicher weise in einerVPC. In diesen Fällen müssen Sie die Gruppen VPC und die Subnetzgruppen erstellen, bevor Sie die DB-Instan ce erstellen.	Arbeiten mit einer DB-Instance in einer VPC
Daten in Amazon RDS Postgre importieren SQL Sie können verschiedene Tools verwenden, um Daten in Ihre SQL Postgre-DB-Instance bei Amazon RDS zu importieren.	Daten in Postgre SQL auf Amazon importieren RDS
Einrichten schreibgeschützter Read Replicas (primär und Standby) RDSfor Postgre SQL unterstützt Read Replicas sowohl in derselben AWS Region als auch in einer anderen AWS Region als die primäre Instance.	Arbeiten mit DB-Instance-Lesereplikaten Arbeiten mit Read Replicas in Amazon RDS for PostgreSQL Erstellen eines Lesereplikats in einer anderen AWS-Region

Aufgabenbereich	Relevante Dokumentation
Grundlagen zu Sicherheitsgruppen DB-Instances werden standardmäßig mit einer Firewall erstellt, die den Zugriff auf die Instance verhindert. Um den Zugriff über diese Firewall zu ermöglichen, bearbeiten Sie die Regeln für eingehende Nachrichten für die VPC Sicherheitsgruppe, die der VPC Hosting-DB-Instance zugeordnet ist.	Zugriffskontrolle mit Sicherheitsgruppen
Einrichten von Parametergruppen und -funktionen Um die Standardparameter für Ihre DB-Instance zu ändern, erstellen Sie eine benutzerdefinierte DB-Parametergruppe und ändern Sie die Einstellungen darauf. Wenn Sie dies tun, bevor Sie Ihre DB-Instance erstellen, können Sie Ihre benutzerdefinierte DB-Parametergruppe auswählen, wenn Sie die Instance erstellen.	Parametergruppen für Amazon RDS
Verbindung zu Ihrer Postgre-DB-Instance herstellen SQL Nachdem Sie eine Sicherheitsgruppe erstellt und sie einer DB-Instance zugeordnet haben, können Sie mit einer beliebigen SQL Standard-Client-Anwendung wie `psql` oder eine Verbindung zur DB-Instance herstellen. `pgAdmin`	Herstellen einer Verbindung zu einer DB-Instance, in der die PostgreSQL-Datenbank-Engine ausgeführt wird Verwenden von SSL mit einer PostgreSQL-DB-Instance
Sichern und Wiederherstellen Ihrer DB-Instance Sie können Ihre DB-Instance so konfigurieren, dass sie automatische Backups oder manuelle Snapshots vornimmt. Aus diesen Backups oder Snapshots können Sie dann Instances wiederherstellen.	Daten sichern, wiederherstellen und exportieren
Überwachen der Aktivität und Leistung Ihrer DB-Instance Sie können eine SQL Postgre-DB-Instance mithilfe von CloudWatch RDS Amazon-Metriken, Ereignissen und erweiterter Überwachung überwachen.	Metriken in der RDS Amazon-Konsole anzeigen RDSAmazon-Ereignisse anzeigen

Aufgabenbereich	Relevante Dokumentation
Aktualisierung der Postgre-Datenbankversion SQL Sie können sowohl Haupt- als auch Nebenversions-Upgrades für Ihre SQL Postgre-DB-Instance durchführen.	Upgrades der RDS for PostgreSQL-DB-Engine Auswahl einer Hauptversion für ein RDS for PostgreSQL-Upgrade
Arbeiten mit Protokolldateien Sie können auf die Protokolldateien für Ihre SQL Postgre-DB-Instance zugreifen.	Datenbank-Protokolldateien von RDS für PostgreSQL
Die Best Practices für Postgre-DB-Instances verstehen SQL Finden Sie einige der besten Methoden für die Arbeit mit Postgre SQL auf AmazonRDS.	Bewährte Methoden für die Arbeit mit PostgreSQL

Im Folgenden finden Sie eine Liste weiterer Abschnitte in diesem Handbuch, die Ihnen helfen können, wichtige Funktionen von RDS for SQL Postgre zu verstehen und zu verwenden:

- Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen
- Steuern des Benutzerzugriffs auf die PostgreSQL−Datenbank
- Arbeiten mit Parametern auf der DB-Instance von RDS for PostgreSQL
- Grundlegendes zu den von RDS for Postgre unterstützten Protokollierungsmechanismen SQL
-
- Verwenden eines benutzerdefinierten DNS Servers für ausgehenden Netzwerkzugriff

Arbeiten mit der Datenbank-Vorschauumgebung

Die SQL Postgre-Community veröffentlicht kontinuierlich neue SQL Postgre-Versionen und Erweiterungen, einschließlich Beta-Versionen. Dies gibt SQL Postgre-Benutzern die Möglichkeit, frühzeitig eine neue SQL Postgre-Version auszuprobieren. Weitere Informationen zum Beta-Release-Prozess der SQL Postgre-Community finden Sie unter Beta-Informationen in der Postgre-Dokumentation. SQL In ähnlicher Weise RDS stellt Amazon bestimmte SQL Postgre-Betaversionen als Vorschauversionen zur Verfügung. Auf diese Weise können Sie DB-Instances mit der Vorschauversion erstellen und ihre Funktionen in der Datenbank-Preview-Umgebung testen.

RDSfür SQL Postgre-DB-Instances in der Database Preview-Umgebung sind die Funktionen ähnlich wie andere RDS für Postgre-Instances. SQL Sie können eine Vorschauversion jedoch nicht für die Produktion einsetzen.

Beachten Sie folgende wichtige Einschränkungen:

- Alle DB-Instances werden 60 Tage nach Erstellung zusammen mit allen Backups und Snapshots gelöscht.
- Sie können eine DB-Instance nur in einer virtuellen privaten Cloud (VPC) erstellen, die auf dem VPC Amazon-Service basiert.
- Sie können nur Allzweckspeicher SSD und bereitgestellten IOPS SSD Speicher verwenden.
- Bei DB-Instances können Sie vom AWS Support keine Hilfe erhalten. Stattdessen können Sie Ihre Fragen in der von uns AWS verwalteten Q&A-Community re:POST stellen.AWS
- Sie können einen Snapshot einer DB-Instance nicht in eine Produktionsumgebung kopieren.

Die folgenden Optionen werden von der Vorschauversion unterstützt.

- Sie können nur DB-Instances mit den Instance-Typen M6i, R6i, M6g, M5, T3, R6g und R5 erstellen. Weitere Informationen zu RDS Instanzklassen finden Sie unter.
- Sie können Single-AZ- und Multi-AZ-Bereitstellungen verwenden.
- Sie können die standardmäßigen SQL Dump- und Ladefunktionen von Postgre verwenden, um Datenbanken aus der Database Preview-Umgebung zu exportieren oder Datenbanken in diese zu importieren.

Themen

- Nicht in der Datenbank-Vorschauumgebung unterstützte Features

- SQLPostgre-Version 17 in der Database Preview-Umgebung
- Erstellen einer neuen DB-Instance in der Datenbank-Vorschauumgebung

Nicht in der Datenbank-Vorschauumgebung unterstützte Features

Die folgenden Features sind in der Datenbank-Vorschauumgebung nicht verfügbar:

- Regionsübergreifende Snapshot-Kopie
- Regionsübergreifende Lesereplikate

SQLPostgre-Version 17 in der Database Preview-Umgebung

> **Note**
>
> Dies ist eine Vorschaudokumentation für Amazon RDS SQL Postgre-Version 17. Änderungen sind vorbehalten.

SQLPostgre-Version 17.0 ist jetzt in der Amazon RDS Database Preview-Umgebung verfügbar. SQLPostgre-Version 17.0 enthält mehrere Verbesserungen, die in der folgenden SQL Postgre-Dokumentation beschrieben werden: Postgre 17 veröffentlicht! SQL

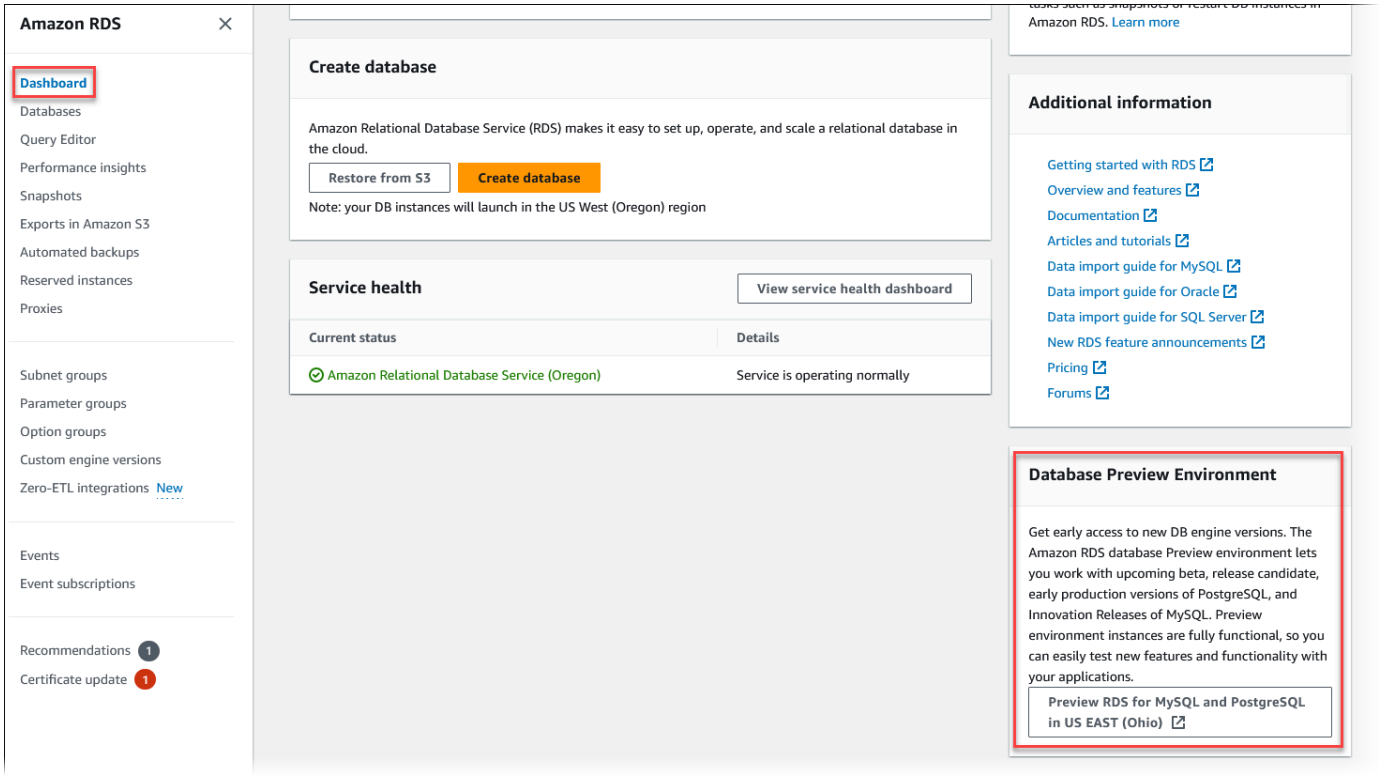
Weitere Informationen zur Database Preview-Umgebung finden Sie unter the section called “Arbeiten mit der Datenbank-Vorschauumgebung”. Um von der Konsole aus auf die Vorschau-Umgebung zuzugreifen, wählen Sie. https://console.aws.amazon.com/rds-preview/

Erstellen einer neuen DB-Instance in der Datenbank-Vorschauumgebung

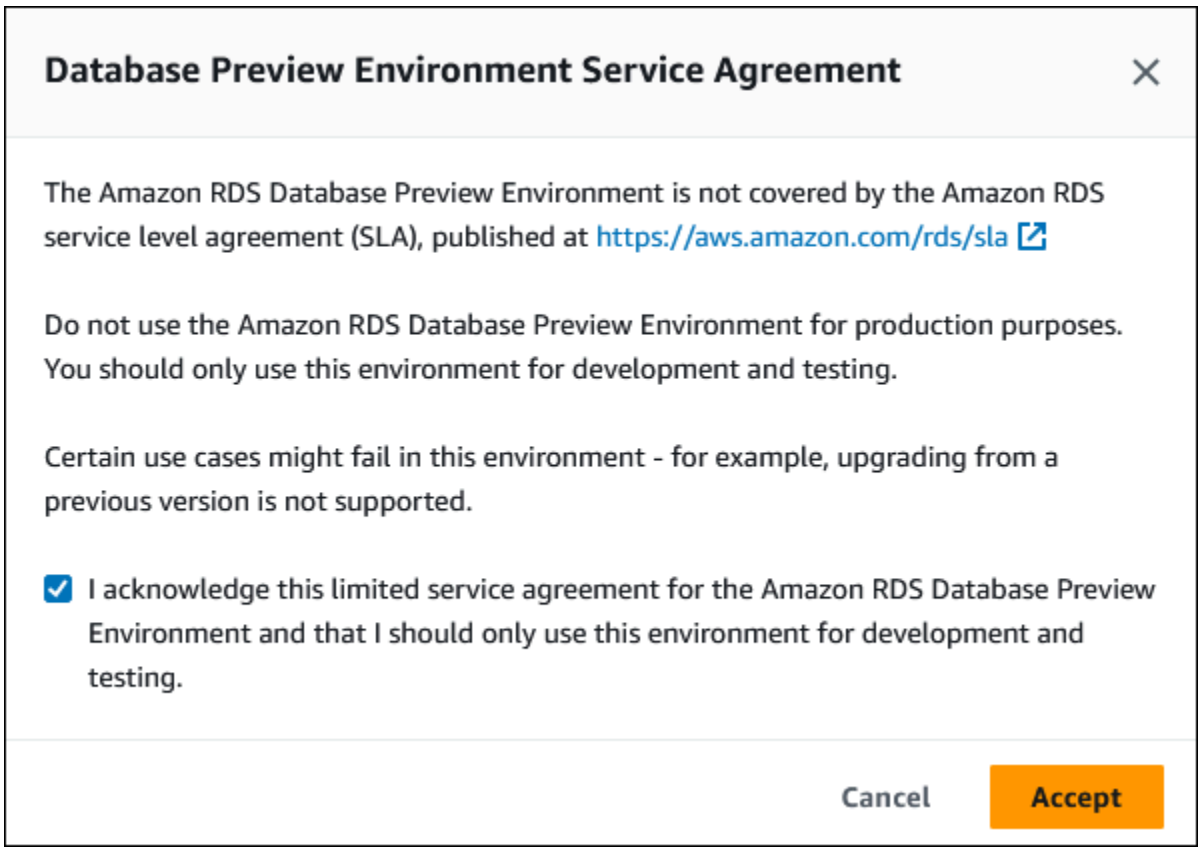
Erstellen Sie mit dem folgenden Verfahren eine DB-Instance in der Vorschauumgebung.

So erstellen Sie eine DB-Instance in der Datenbank-Vorschauumgebung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Dashboard aus.
3. Suchen Sie auf Dashboard-Seite nach Database Preview Environment (Datenbank-Preview-Umgebung), wie in der folgenden Abbildung gezeigt.



Sie können auch direkt zu Datenbank-Preview-Umgebung navigieren. Bevor Sie fortfahren können, müssen Sie die Einschränkungen bestätigen und akzeptieren.



4. Um die RDS für SQL Postgre-DB-Instance zu erstellen, gehen Sie genauso vor wie beim Erstellen einer beliebigen RDS Amazon-DB-Instance. Weitere Informationen finden Sie im Verfahren Konsole unter Erstellen einer DB-Instance.

Verwenden Sie den folgenden Endpunkt, um eine Instance in der Database Preview-Umgebung mit dem RDS API AWS CLI oder dem zu erstellen.

```
rds-preview.us-east-2.amazonaws.com
```

Verfügbare PostgreSQL-Datenbankversionen

Amazon RDS unterstützt DB-Instances für mehrere Editionen von PostgreSQL. Sie können eine beliebige aktuell verfügbare PostgreSQL-Version festlegen, wenn Sie eine DB-Instance erstellen. Sie können die Hauptversionen (z. B. PostgreSQL 14) sowie eine beliebige verfügbare Nebenversion für die angegebene Hauptversion festlegen. Wenn keine Version angegeben wird, verwendet Amazon RDS standardmäßig eine verfügbare Version – in der Regel die aktuelle Version. Wenn die Hauptversion, jedoch nicht die Unterversion, festgelegt ist, verwendet Amazon RDS standardmäßig den letzten Release der Hauptversion, die Sie festgelegt haben.

Verwenden Sie den `describe-db-engine-versions` AWS CLI Befehl, um eine Liste der verfügbaren Versionen sowie die Standardeinstellungen für neu erstellte DB-Instances anzuzeigen. Verwenden Sie zum Beispiel den folgenden Befehl, um die Standardversion der PostgreSQL-Engine anzuzeigen:

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Weitere Informationen zu den PostgreSQL-Versionen, die von Amazon RDS unterstützt werden, finden Sie in den Versionshinweisen zu Amazon RDS for PostgreSQL. Sie können auch Informationen zu den Support-Daten für wichtige Engine-Versionen anzeigen, indem Sie den AWS CLI Befehl describe-db-major-engine-versions ausführen oder den API-Vorgang Describe DBMajor EngineVersions RDS verwenden.

Wenn Sie nicht bereit sind, vor Ablauf des RDS-Standard-Supports manuell auf eine neue Hauptversion der Engine zu aktualisieren, registriert Amazon RDS Ihre Datenbanken nach Ablauf des RDS-Standard-Supportdatums automatisch bei Amazon RDS Extended Support. Anschließend können Sie weiterhin RDS für PostgreSQL Version 11 und höher ausführen. Weitere Informationen finden Sie unter Erweiterter Support für Amazon RDS mit Amazon RDS und Amazon RDS – Preise.

Veraltete Versionen für Amazon RDS for PostgreSQL

Beachten Sie die folgenden veralteten Versionen:

- RDS für PostgreSQL 10 wurde im Februar 2023 als veraltet eingestuft.
- RDS für PostgreSQL 9.6 wurde im März 2022 als veraltet eingestuft.
- RDS für PostgreSQL 9.5 wurde im März 2021 als veraltet eingestuft.

Weitere Informationen über die Verfallsrichtlinie für RDS for PostgreSQL finden Sie unter Amazon RDS. FAQs Weitere Informationen zu PostgreSQL-Versionen finden Sie unter Versioning-Richtlinie in der PostgreSQL-Dokumentation.

Den inkrementellen Release-Prozess von RDS für PostgreSQL verstehen

RDS for PostgreSQL bietet Sicherheitsupdates, Leistungsverbesserungen und neue Funktionen in Form von inkrementellen Releases, wobei die Kompatibilität kleinerer Versionen gewahrt bleibt. Diese Versionen sind als R1, R2, R3 usw. gekennzeichnet.

Namenskonvention für Release-Versionen

- R1 ist die erste Version einer Nebenversion. Es enthält gelegentlich neue Funktionen, Erweiterungen oder Upgrades vorhandener Erweiterungen.
- Zu den nachfolgenden Release-Versionen (R2, R3 und höher) gehören:
  - Sicherheits-Updates
  - Leistungsverbesserungen
  - Fehlerbehebungen
  - Aktualisierungen der Erweiterungen

Vorteile des inkrementellen Release-Prozesses von RDS für PostgreSQL

Der inkrementelle Freigabeprozess bietet die folgenden Vorteile:

- Schnelle Einführung neuer Versionen der PostgreSQL-Community bei gleichzeitiger getrennter Verwaltung von RDS-spezifischen Verbesserungen in nachfolgenden Versionen. Dies rationalisiert den Veröffentlichungsprozess und gewährleistet eine schnellere Bereitstellung kritischer Updates.
- Zugriff auf Bugfixes, neue Funktionen, Sicherheitsupdates und Erweiterungsupdates unter Beibehaltung der Kompatibilität mit der PostgreSQL-Nebenversion.

Verwaltung von Release-Updates

Amazon RDS informiert Sie über neue inkrementelle Versionen durch ausstehende Wartungsaktionen in der. AWS Management Console Sie können Ihre Datenbank mit einer der folgenden Methoden aktualisieren:

- Aktivieren Sie automatische Updates während der geplanten Wartungsfenster.
- Wenden Sie Updates manuell im Rahmen ausstehender Wartungsaktionen an.

- Verwenden Sie Blue/Green-Bereitstellungen mit physischer Replikation, um Ausfallzeiten zu minimieren. Weitere Informationen finden Sie unter Blue/Green Deployments unterstützen kleinere Versionsupgrades für RDS for PostgreSQL.

Bevor Sie Ihre Datenbank aktualisieren, sollten Sie die folgenden wichtigen Punkte berücksichtigen:

- Für Updates sind Datenbankneustarts erforderlich, es sei denn, Sie verwenden Blue/Green-Bereitstellungen mit physischer Replikation.
- Einige inkrementelle Versionen sind obligatorisch, insbesondere solche, die Sicherheitsupdates enthalten.

Weitere Informationen zur Aktualisierung Ihrer Amazon RDS-DB-Instance finden Sie unter Vertrauenswürdige SQL Postgre-Erweiterungen und apply-pending-maintenance-action.

Unterstützte SQL Postgre-Erweiterungsversionen

RDSfür Postgre SQL unterstützt viele SQL Postgre-Erweiterungen. Die SQL Postgre-Community bezeichnet diese manchmal als Module. Erweiterungen erweitern die von der Postgre-Engine bereitgestellten Funktionen. SQL Eine Liste der von Amazon unterstützten Erweiterungen finden Sie RDS in der Standard-DB-Parametergruppe für diese SQL Postgre-Version. Sie können sich auch die Liste der aktuellen Erweiterungen ansehen, wenn Sie `psql` verwenden, indem Sie den Parameter `rds.extensions` anzeigen, wie im folgenden Beispiel.

```
SHOW rds.extensions;
```

> **Note**
>
> Parameter, die in einer älteren Version hinzugefügt wurden, werden möglicherweise nicht richtig dargestellt, wenn der Parameter `rds.extensions` in `psql` verwendet wird.

Ab Postgre SQL 13 können bestimmte Erweiterungen von anderen Datenbankbenutzern als den installiert werden. RDS `rds_superuser` Diese sind bekannt alsvertrauenswürdige Erweiterungen. Weitere Informationen hierzu finden Sie unter Vertrauenswürdige SQL Postgre-Erweiterungen.

Bestimmte Versionen von RDS for Postgre SQL unterstützen den `rds.allowed_extensions` Parameter. Mit diesem Parameter können Sie die Erweiterungen `rds_superuser` einschränken, die in der RDS for SQL Postgre-DB-Instance installiert werden können. Weitere Informationen finden Sie unter Einschränkung der Installation von Postgre-Erweiterungen SQL.

Eine Liste der SQL Postgre-Erweiterungen und Versionen, die von jeder RDS für SQL Postgre-Version verfügbaren Versionen unterstützt werden, finden Sie unter SQLPostgre-Erweiterungen, die auf Amazon unterstützt werden, RDS in Amazon RDS für SQL Postgre-Versionshinweise.

Einschränkung der Installation von Postgre-Erweiterungen SQL

Sie können einschränken, welche Erweiterungen auf einer Postgre-DB-Instance installiert werden können. SQL Standardmäßig ist dieser Parameter nicht festgelegt, daher kann jede unterstützte Erweiterung hinzugefügt werden, wenn der Benutzer dazu berechtigt ist. Setzen Sie dazu den `rds.allowed_extensions`-Parameter auf eine Zeichenfolge von kommagetrennten Erweiterungsnamen. Indem Sie diesem Parameter eine Liste von Erweiterungen hinzufügen,

identifizieren Sie explizit die Erweiterungen, die Ihre RDS for SQL Postgre-DB-Instance verwenden kann. Nur diese Erweiterungen können dann in der SQL Postgre-DB-Instance installiert werden.

Die Standardzeichenfolge für den `rds.allowed_extensions`-Parameter ist '*', was bedeutet, dass jede für die Engine-Version verfügbare Erweiterung installiert werden kann. Das Ändern des `rds.allowed_extensions`-Parameters erfordert keinen Neustart der Datenbank, da es sich um einen dynamischen Parameter handelt.

Die SQL Postgre-DB-Instance-Engine muss eine der folgenden Versionen sein, damit Sie den `rds.allowed_extensions` Parameter verwenden können:

- Alle Postgre-16-Versionen SQL
- Postgre SQL 15 und alle höheren Versionen
- Postgre SQL 14 und alle höheren Versionen
- Postgre SQL 13.3 und höhere Nebenversionen
- Postgre SQL 12.7 und höhere Nebenversionen

Verwenden Sie den folgenden psql-Befehl, um zu sehen, welche Erweiterungsinstallationen zulässig sind.

```
postgres=> SHOW rds.allowed_extensions;
 rds.allowed_extensions
------------------------
 *
```

Wenn eine Erweiterung in der Liste im `rds.allowed_extensions`-Parameter installiert wurde, bevor sie ausgelassen wurde, kann die Erweiterung weiterhin normal verwendet werden, und Befehle wie `ALTER EXTENSION` und `DROP EXTENSION` funktionieren weiter. Nachdem eine Erweiterung jedoch eingeschränkt wurde, schlagen die `CREATE EXTENSION`-Befehle für die eingeschränkte Erweiterung fehl.

Die Installation von Erweiterungsabhängigkeiten mit `CREATE EXTENSION CASCADE` ist ebenfalls eingeschränkt. Die Erweiterung und ihre Abhängigkeiten müssen in angegeben werde `rds.allowed_extensions`. Wenn eine Installation der Erweiterungsabhängigkeit fehlschlägt, schlägt die gesamte `CREATE EXTENSION CASCADE`-Anweisung fehl.

Wenn eine Erweiterung nicht im `rds.allowed_extensions`-Parameter enthalten ist, wird ein Fehler wie der folgende angezeigt, wenn Sie versuchen, sie zu installieren.

```
ERROR: permission denied to create extension "extension-name"
HINT: This extension is not specified in "rds.allowed_extensions".
```

Vertrauenswürdige SQL Postgre-Erweiterungen

Für die Installation der meisten SQL Postgre-Erweiterungen sind Rechte erforderlich`rds_superuser`. Mit Postgre SQL 13 wurden vertrauenswürdige Erweiterungen eingeführt, die die Notwendigkeit reduzieren, regulären Benutzern `rds_superuser` Rechte zu gewähren. Mit dieser Funktion können Benutzer viele Erweiterungen installieren, wenn sie über die `CREATE` -Berechtigung für die aktuelle Datenbank anstatt der `rds_superuser`-Rolle verfügen. Weitere Informationen finden Sie in dem SQL CREATEEXTENSIONBefehl in der Postgre-Dokumentation. SQL

Im Folgenden werden die Erweiterungen aufgeführt, die von einem Benutzer installiert werden können, der über die `CREATE`-Berechtigung für die aktuelle Datenbank verfügt und die `rds_superuser`-Rolle nicht benötigt:

- bool_plperl
- btree_gin
- btree_gist
- citext
- cube
- dict_int
- fuzzystrmatch
- hstore
- intarray
- isn
- jsonb_plperl
- ltree
- pg_trgm
- pgcrypto
- plperl
- plpgsql
- pltcl

- tablefunc
- tsm_system_rows
- tsm_system_time
- unaccent
- uuid-ossp

Eine Liste der SQL Postgre-Erweiterungen und Versionen, die von jeder RDS für SQL Postgre-Version verfügbaren Versionen unterstützt werden, finden Sie unter SQLPostgre-Erweiterungen, die auf Amazon unterstützt werden, RDS in Amazon RDS für SQL Postgre-Versionshinweise.

Arbeiten mit SQL Postgre-Funktionen, die von Amazon RDS für Postgre unterstützt werden SQL

Amazon RDS for Postgre SQL unterstützt viele der gängigsten Postgre-FunktionenSQL. Postgre SQL verfügt beispielsweise über eine Autovacuum-Funktion, die routinemäßige Wartungsarbeiten an der Datenbank durchführt. Die Autovakuierungsfunktion ist standardmäßig aktiviert. Obwohl Sie diese Funktion deaktivieren können, empfehlen wir dringend, sie eingeschaltet zu lassen. Diese Funktion zu verstehen und zu verstehen, was Sie tun können, um sicherzustellen, dass sie ordnungsgemäß funktioniert, ist für jeden eine grundlegende Aufgabe. DBA Weitere Informationen zur Selbstbereinigung finden Sie unter . Um mehr über andere häufig auftretende DBA Aufgaben zu erfahren,Häufige DBA-Aufgaben für Amazon RDS for PostgreSQL.

RDSfor Postgre unterstützt SQL auch Erweiterungen, die der DB-Instance wichtige Funktionen hinzufügen. Sie können beispielsweise die GIS Post-Erweiterung verwenden, um mit räumlichen Daten zu arbeiten, oder die Erweiterung pg_cron verwenden, um Wartungsarbeiten innerhalb der Instanz zu planen. Weitere Hinweise zu Postgre-Erweiterungen finden Sie unter. SQL Verwenden von PostgreSQL-Erweiterungen mit Amazon RDS for PostgreSQL

Foreign Data Wrapper sind ein spezieller Erweiterungstyp, mit dem Ihre RDS for SQL Postgre-DB-Instance mit anderen kommerziellen Datenbanken oder Datentypen funktioniert. Weitere Hinweise zu Fremddaten-Wrappern, die von RDS for Postgre unterstützt werden, finden Sie unter. SQL Arbeiten mit den unterstützten Foreign Data Wrappern für RDS SQL

Im Folgenden finden Sie Informationen zu einigen anderen Funktionen, die von RDS for Postgre unterstützt werden. SQL

Themen

- Benutzerdefinierte Datentypen und Aufzählungen mit für Postgre RDS SQL
- Event-Trigger RDS für Postgre SQL
- Riesige Seiten RDS für Postgre SQL
- Ausführen der logischen Replikation für Amazon RDS for PostgreSQL
- RAMFestplatte für das stats_temp_directory
- Tablespaces für Postgre RDS SQL
- RDSfür SQL Postgre-Kollationen für EBCDIC und andere Mainframe-Migrationen
- Verwaltung der logischen Steckplatzsynchronisierung für RDS for PostgreSQL

Benutzerdefinierte Datentypen und Aufzählungen mit für Postgre RDS SQL

Postgre SQL unterstützt das Erstellen benutzerdefinierter Datentypen und das Arbeiten mit Aufzählungen. Weitere Informationen zum Erstellen von und Arbeiten mit Aufzählungen und anderen Datentypen finden Sie in der Postgre-Dokumentation unter Aufzählungstypen. SQL

Im Folgenden finden Sie ein Beispiel für das Erstellen eines Typs als Aufzählung und das anschließende Einfügen von Werten in eine Tabelle.

```
CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow', 'green', 'blue', 'purple');
CREATE TYPE
CREATE TABLE t1 (colors rainbow);
CREATE TABLE
INSERT INTO t1 VALUES ('red'), ( 'orange');
INSERT 0 2
SELECT * from t1;
colors
--------
red
orange
(2 rows)
postgres=> ALTER TYPE rainbow RENAME VALUE 'red' TO 'crimson';
ALTER TYPE
postgres=> SELECT * from t1;
colors
---------
crimson
orange
(2 rows)
```

Event-Trigger RDS für Postgre SQL

Alle aktuellen SQL Postgre-Versionen unterstützen Ereignisauslöser, ebenso wie alle verfügbaren Versionen von RDS for Postgre. SQL Sie können das Hauptbenutzerkonto nutzen (Standard, `postgres`), um Ereignisauslöser zu erstellen, zu ändern, umzubenennen und zu löschen. Ereignisauslöser befinden sich auf DB-Instance-Level und können so auf alle Datenbanken einer Instance angewendet werden.

Der folgende Code erstellt beispielsweise einen Event-Trigger, der den aktuellen Benutzer am Ende jedes Data Definition Language (DDL) -Befehls ausgibt.

```
CREATE OR REPLACE FUNCTION raise_notice_func()
    RETURNS event_trigger
    LANGUAGE plpgsql AS
$$
BEGIN
    RAISE NOTICE 'In trigger function: %', current_user;
END;
$$;

CREATE EVENT TRIGGER event_trigger_1
    ON ddl_command_end
EXECUTE PROCEDURE raise_notice_func();
```

Weitere Informationen zu SQL Postgre-Ereignisauslösern finden Sie unter Ereignisauslöser in der Postgre-Dokumentation. SQL

Es gibt mehrere Einschränkungen bei der Verwendung von SQL Postgre-Event-Triggern bei AmazonRDS. Diese umfassen u. a. folgende:

- Auf Read Replicas können keine Ereignisauslöser erstellt werden. Sie können jedoch Ereignisauslöser auf einer Read Replica-Quelle erstellen. Die Ereignisauslöser werden dann in die Read Replica kopiert. Die Ereignisauslöser auf der Read Replica werden nicht bei Änderungen, die von der Quelle ausgehen, ausgelöst. Wenn jedoch die Read Replica verwendet wird, werden die vorhandenen Ereignisauslöser bei Datenbank-Operationen ausgelöst.
- Um ein Hauptversions-Upgrade für eine SQL Postgre-DB-Instance durchzuführen, die Ereignisauslöser verwendet, stellen Sie sicher, dass Sie die Ereignisauslöser löschen, bevor Sie die Instance aktualisieren.

Riesige Seiten RDS für Postgre SQL

Huge Pages ist eine Arbeitsspeicher-Verwaltungsfunktion, die den Overhead reduziert, wenn eine DB-Instance mit großen, zusammenhängenden Arbeitsspeicherblöcken arbeitet, wie sie von gemeinsam genutzten Puffern verwendet werden. Diese SQL Postgre-Funktion wird von allen derzeit RDS für Postgre SQL verfügbaren Versionen unterstützt. Sie können große Seiten für Ihre Anwendung zuordnen, indem Sie Aufrufe des freigegebenen Speichers `mmap` oder `SYSV` verwenden. RDSfor Postgre SQL unterstützt sowohl Seitengrößen von 4 KB als auch 2 MB.

Sie können Huge Pages ein- oder ausschalten, indem Sie den Wert des `huge_pages`-Parameters ändern. Die Funktion ist standardmäßig für alle DB-Instance-Klassen aktiviert, außer Micro, Small und Medium.

RDSfür Postgre werden riesige Seiten SQL verwendet, die auf dem verfügbaren gemeinsamen Speicher basieren. Wenn die DB-Instance aufgrund von Einschränkungen des gemeinsamen Speichers keine riesigen Seiten verwenden kann, RDS verhindert Amazon, dass die DB-Instance gestartet wird. In diesem Fall RDS setzt Amazon den Status der DB-Instance auf einen Status mit inkompatiblen Parametern. In diesem Fall können Sie den `huge_pages` Parameter auf setzen, `off` damit Amazon RDS die DB-Instance starten kann.

Der Parameter `shared_buffers` ist wichtig für die Einstellung des gemeinsam genutzten Speicherpools, der für die Verwendung von großen Seiten erforderlich ist. Der Standardwert für den `shared_buffers`-Parameter verwendet ein Makro für Datenbankparameter. Dieses Makro legt einen Prozentsatz der insgesamt verfügbaren 8 KB an Seiten fest, die für den Speicher der DB-Instance verfügbar sind, verfügbar. Wenn Sie riesige Seiten verwenden, werden diese Seiten in den riesigen Seiten zusammengefasst zugeordnet. Amazon RDS versetzt eine DB-Instance in einen Zustand mit inkompatiblen Parametern, wenn die Shared-Memory-Parameter so eingestellt sind, dass sie mehr als 90 Prozent des DB-Instance-Speichers beanspruchen.

Weitere Informationen zur SQL Postgre-Speicherverwaltung finden Sie unter Ressourcenverbrauch in der Postgre-Dokumentation. SQL

Ausführen der logischen Replikation für Amazon RDS for PostgreSQL

Ab Version 10.4 unterstützt RDS für PostgreSQL die Veröffentlichung und das Abonnement von SQL-Syntax, die erstmals in PostgreSQL 10 eingeführt wurde. Weitere Informationen finden Sie unter Logische Replikation in der PostgreSQL-Dokumentation.

> **Note**
>
> Zusätzlich zu der in PostgreSQL 10 eingeführten nativen logischen Replikationsfunktion von PostgreSQL unterstützt RDS für PostgreSQL auch die `pglogical`-Erweiterung. Weitere Informationen finden Sie unter Verwenden von pglogical, um Daten zwischen Instances zu synchronisieren.

Im Folgenden erfahren Sie, wie Sie die logische Replikation für eine DB-Instance von RDS for PostgreSQL einrichten.

Themen

- Verständnis der logischen Replikation und logischen Decodierung
- Arbeiten mit logischen Replikations-Slots
- Replizieren von Daten auf Tabellenebene mithilfe logischer Replikation

Verständnis der logischen Replikation und logischen Decodierung

RDS for PostgreSQL unterstützt das Streaming von Write-Ahead-Log(WAL)-Änderungen mithilfe der Slots für logische Replikation von PostgreSQL. Es unterstützt auch die Verwendung logischer Decodierung. Sie können logische Replikations-Slots in Ihrer Instance einrichten und über diese Slots Datenbankänderungen auf einen Client wie z. B. streame `pg_recvlogical`. Sie erstellen logische Replikationsslots auf Datenbankebene, die Replikationsverbindungen zu einer einzelnen Datenbank unterstützen.

Die gängigsten Clients für die logische PostgreSQL-Replikation sind AWS Database Migration Service oder ein individuell verwalteter Host auf einer Amazon-Instance. EC2 Der logische Replikations-Slot enthält keine Informationen über den Empfänger des Streams. Es gibt auch keine Anforderung, dass das Ziel eine Replikatdatenbank sein muss. Wenn beim Einrichten eines Slots für die logische Replikation nicht vom Slot gelesen wird, können Daten in den Speicher Ihrer DB-Instance geschrieben werden und diesen schnell füllen.

Sie aktivieren die logische Replikation und die logische Decodierung PostgreSQL für Amazon RDS durch einen Parameter, einen neuen Replikationsverbindungstyp sowie eine Sicherheitsrolle. Beim Client für die logische Decodierung kann es sich um jeden beliebigen Client handeln, der eine Replikationsverbindung zu einer Datenbank in einer PostgreSQL-DB-Instance herstellen kann.

Logische Decodierung für eine DB-Instance von RDS for PostgreSQL aktivieren

1. Stellen Sie sicher, dass das Benutzerkonto, das Sie verwenden, folgende Rollen hat:
   - Die `rds_superuser`-Rolle, damit Sie die logische Replikation aktivieren können
   - Die `rds_replication`-Rolle erteilt Berechtigungen zur Verwaltung von logischen Slots und zum Streamen von Daten mithilfe von logischen Slots
2. Setzen Sie den statischen Parameter `rds.logical_replication` auf 1. Setzen Sie bei der Anwendung dieses Parameters auch die Parameter `wal_level`, `max_wal_senders`, `max_replication_slots` und `max_connections` fest. Diese Änderungen an den

Parametern können die WAL-Generierung steigern, legen Sie daher den Parameter `rds.logical_replication` nur dann fest, wenn Sie logische Slots verwenden.

3. Starten Sie die DB-Instance neu, damit der statische Parameter `rds.logical_replication` in Kraft tritt.
4. Erstellen Sie einen logischen Replikationsslot wie im nächsten Abschnitt erläutert. Für diesen Prozess ist es erforderlich, dass Sie ein Decodier-Plugin angeben. Derzeit unterstützt RDS for PostgreSQL die test_decoding- und wal2json-Ausgabe-Plugins, die mit PostgreSQL geliefert werden.

Weitere Informationen zur logischen Decodierung mit PostgreSQL finden Sie in der PostgreSQL-Dokumentation.

Arbeiten mit logischen Replikations-Slots

Sie können SQL-Befehle verwenden, um mit logischen Slots zu arbeiten. Beispiel: Der folgende Befehl erstellt einen logischen Slot mit dem Namen `test_slot` unter Verwendung des standardmäßigen Ausgangs-Plugin `test_decoding` von PostgreSQL.

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
slot_name    | xlog_position
-----------------+---------------
regression_slot | 0/16B1970
(1 row)
```

Mit dem folgenden Befehl können Sie die logischen Slots auflisten.

```
SELECT * FROM pg_replication_slots;
```

Mit dem folgenden Befehl können Sie einen logischen Slot entfernen.

```
SELECT pg_drop_replication_slot('test_slot');
pg_drop_replication_slot
----------------------
(1 row)
```

Weitere Beispiele zum Arbeiten mit logischen Replikations-Slots finden Sie unter Logical Decoding Examples in der PostgreSQL-Dokumentation.

Sobald Sie den logischen Replikationsslot erstellt haben, können Sie mit dem Streaming beginnen. Das folgende Beispiel zeigt, wie die logische Decodierung über das Streaming-Replikationsprotokoll gesteuert wird. Dieses Beispiel verwendet das Programm pg_recvlogical, das in der PostgreSQL-Distribution enthalten ist. Dazu muss die Client-Authentifizierung so eingerichtet sein, dass Replikationsverbindungen zugelassen werden.

```
pg_recvlogical -d postgres --slot test_slot -U postgres
    --host -instance-name.111122223333.aws-region.rds.amazonaws.com
    -f -  --start
```

Fragen Sie die Funktion `pg_replication_origin_status` ab, um den Inhalt der `pg_show_replication_origin_status` Ansicht anzuzeigen.

```
SELECT * FROM pg_show_replication_origin_status();
local_id | external_id | remote_lsn | local_lsn
----------+-------------+------------+-----------
(0 rows)
```

Replizieren von Daten auf Tabellenebene mithilfe logischer Replikation

Sie können logische Replikation verwenden, um Daten aus Quelltabellen in Zieltabellen in RDS for PostgreSQL zu replizieren. Bei der logischen Replikation werden zunächst vorhandene Daten aus den Quelltabellen geladen und anschließend die laufenden Änderungen weiter repliziert.

1. Erstellen Sie die Quelltabellen

   Connect zur Quelldatenbank in Ihrer RDS for PostgreSQL-DB-Instance her:

   ```
   source=> CREATE TABLE testtab (slno int primary key);
   CREATE TABLE
   ```

2. Fügen Sie Daten in die Quelltabellen ein

   ```
   source=> INSERT INTO testtab VALUES (generate_series(1,1000));
   INSERT 0 1000
   ```

3. Erstellen Sie eine Publikation für Quelltabellen

   - Erstellen Sie eine Publikation für die Quelltabellen:

```
source=> CREATE PUBLICATION testpub FOR TABLE testtab;
CREATE PUBLICATION
```

   - Verwenden Sie eine SELECT-Abfrage, um die Details der erstellten Publikation zu überprüfen:

```
source=> SELECT * FROM pg_publication;
  oid   | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
 pubtruncate | pubviaroot
--------+---------+----------+--------------+-----------+-----------+-----------
+-------------+------------
 115069 | testpub |    16395 | f            | t         | t         | t         |
 t           | f
(1 row)
```

   - Stellen Sie sicher, dass die Quelltabellen der Publikation hinzugefügt wurden:

```
source=> SELECT * FROM pg_publication_tables;
pubname | schemaname | tablename
---------+------------+-----------
 testpub | public     | testtab
(1 rows)
```

   - Um alle Tabellen in einer Datenbank zu replizieren, verwenden Sie:

```
CREATE PUBLICATION testpub FOR ALL TABLES;
```

   - Wenn die Publikation bereits für eine einzelne Tabelle erstellt wurde und Sie eine neue Tabelle hinzufügen müssen, können Sie die folgende Abfrage ausführen, um der vorhandenen Publikation neue Tabellen hinzuzufügen:

```
ALTER PUBLICATION <publication_name> add table <new_table_name>;
```

4. Connect zur Zieldatenbank her und erstellen Sie Zieltabellen

   - Connect zur Zieldatenbank in der Ziel-DB-Instance her. Erstellen Sie die Zieltabellen mit den gleichen Namen wie die Quelltabellen:

```
target=> CREATE TABLE testtab (slno int primary key);
CREATE TABLE
```

   - Stellen Sie sicher, dass in den Zieltabellen keine Daten vorhanden sind, indem Sie eine SELECT-Abfrage für die Zieltabellen ausführen:

```
target=> SELECT count(*) FROM testtab;
 count
-------
     0
(1 row)
```

5. Erstellen und überprüfen Sie das Abonnement in der Zieldatenbank

   - Erstellen Sie das Abonnement in der Zieldatenbank:

```
target=> CREATE SUBSCRIPTION testsub
CONNECTION 'host=<source RDS/host endpoint> port=5432 dbname=<source_db_name>
 user=<user> password=<password>'
PUBLICATION testpub;
NOTICE:  Created replication slot "testsub" on publisher
CREATE SUBSCRIPTION
```

   - Verwenden Sie eine SELECT-Abfrage, um zu überprüfen, ob das Abonnement aktiviert ist:

```
target=> SELECT oid, subname, subenabled, subslotname, subpublications FROM
 pg_subscription;
  oid  | subname | subenabled | subslotname | subpublications
-------+---------+------------+-------------+-----------------
 16434 | testsub | t          | testsub     | {testpub}
(1 row)
```

- Wenn das Abonnement erstellt wird, werden alle Daten aus den Quelltabellen in die Zieltabellen geladen. Führen Sie eine SELECT-Abfrage für die Zieltabellen aus, um zu überprüfen, ob die ursprünglichen Daten geladen werden:

```
target=> SELECT count(*) FROM testtab;
 count
-------
  1000
(1 row)
```

6. Überprüfen Sie den Replikationssteckplatz in der Quelldatenbank

   Durch die Erstellung eines Abonnements in der Zieldatenbank wird ein Replikationsslot in der Quelldatenbank erstellt. Überprüfen Sie die Details des Replikationssteckplatzes, indem Sie die folgende SELECT-Abfrage in der Quelldatenbank ausführen:

```
source=> SELECT * FROM pg_replication_slots;

slot_name |  plugin  | slot_type | datoid | database | temporary | active |
 active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn | wal_status
 | safe_wal_size
----------+----------+-----------+--------+----------+-----------+--------
+------------+------+--------------+-------------+---------------------
+------------+---------------
testsub   | pgoutput | logical   | 115048 | source   | f         | t      |
 846 |      |         6945 | 58/B4000568 | 58/B40005A0         | reserved   |
(1 row)
```

7. Die Replikation wird getestet

   - Testen Sie, ob Datenänderungen in den Quelltabellen in die Zieltabellen repliziert werden, indem Sie Zeilen in die Quelltabellen einfügen:

```
source=> INSERT INTO testtab VALUES(generate_series(1001,2000));
INSERT 0 1000

source=> SELECT count(*) FROM testtab;
 count
-------
  2000
(1 row)
```

   - Überprüfen Sie die Anzahl der Zeilen in den Zieltabellen, um sicherzustellen, dass neue Einfügungen repliziert werden:

```
target=> SELECT count(*) FROM testtab;
 count
-------
  2000
(1 row)
```

8. Aktualisierung des Abonnements nach dem Hinzufügen von Tabellen

   - Wenn Sie einer vorhandenen Publikation neue Tabellen hinzufügen, muss das Abonnement unbedingt aktualisiert werden, damit die Änderungen wirksam werden:

```
ALTER SUBSCRIPTION <subscription_name> REFRESH PUBLICATION;
```

   - Mit diesem Befehl werden fehlende Tabelleninformationen vom Herausgeber abgerufen und die Replikation für Tabellen gestartet, die seit der Erstellung oder letzten Aktualisierung des Abonnements zu den abonnierten Publikationen hinzugefügt wurden.

RAMFestplatte für das stats_temp_directory

Sie können den SQL Parameter RDS for Postgre verwenden, `rds.pg_stat_ramdisk_size` um den Systemspeicher anzugeben, der einer RAM Festplatte zum Speichern von Postgre zugewiesen ist. SQL `stats_temp_directory` Der RAM Festplattenparameter ist nur in RDS SQL Postgre-Version 14 und niedrigeren Versionen verfügbar.

Bei bestimmten Workloads kann durch die Einstellung dieses Parameters die Leistung verbessert und die I/O-Anforderungen können gesenkt werden. Weitere Informationen zu finden Sie in der `stats_temp_directory` Postgre-Dokumentation. SQL .

Um eine RAM Festplatte für Ihre einzurichten`stats_temp_directory`, setzen Sie den `rds.pg_stat_ramdisk_size` Parameter auf einen ganzzahligen Literalwert in der Parametergruppe, die von Ihrer DB-Instance verwendet wird. Dieser Parameter wird in MB angegeben, daher müssen Sie einen ganzzahligen Wert verwenden. Ausdrücke, Formeln und Funktionen sind für den Parameter `rds.pg_stat_ramdisk_size` nicht gültig. Starten Sie die DB-Instance neu, damit die Änderung wirksam wird. Weitere Informationen zum Festlegen von Parametern finden Sie unter Parametergruppen für Amazon RDS.

Mit dem folgenden AWS CLI Befehl wird beispielsweise der RAM Festplattenparameter auf 256 MB festgelegt.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name pg-95-ramdisk-testing \
    --parameters "ParameterName=rds.pg_stat_ramdisk_size, ParameterValue=256,
 ApplyMethod=pending-reboot"
```

Nach dem Neustart führen Sie den folgenden Befehl aus, um den Status des `stats_temp_directory` anzuzeigen:

```
postgres=> SHOW stats_temp_directory;
```

Der Befehl sollte Folgendes zurückgeben.

```
stats_temp_directory
---------------------------
/rdsdbramdisk/pg_stat_tmp
(1 row)
```

Tablespaces für Postgre RDS SQL

RDSfür Postgre SQL unterstützt Tablespaces aus Kompatibilitätsgründen. Da sich der gesamte Speicher auf einem einzigen logischen Volume befindet, können Sie keine Tablespaces für I/O-Splitting oder -Isolierung verwenden. Unsere Benchmarks und Erfahrung zeigen, dass ein einzelnes logisches Volume für die meisten Anwendungsfälle das beste Setup ist.

Um Tablespaces mit Ihrer RDS for Postgre-DB-Instance zu erstellen und zu verwenden, ist die Rolle erforderlichSQL. `rds_superuser` Das Hauptbenutzerkonto Ihrer RDS for SQL Postgre-DB-Instance (Standardname`postgres`) ist Mitglied dieser Rolle. Weitere Informationen finden Sie unter Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen.

Wenn Sie beim Erstellen eines Tablespace einen Dateinamen angeben, lautet das Pfadpräfix `/rdsdbdata/db/base/tablespace`. Im folgenden Beispiel werden Tablespace-Dateien in abgeleg `/rdsdbdata/db/base/tablespace/data`. In diesem Beispiel wird angenommen, dass ein `dbadmin`-Benutzer (Rolle) existiert und ihm die `rds_superuser`-Rolle gewährt wurde, die zur Arbeit mit Tablespaces benötigt wird.

```
postgres=> CREATE TABLESPACE act_data
  OWNER dbadmin
  LOCATION '/data';
CREATE TABLESPACE
```

Weitere Informationen zu Postgre-Tablespaces finden Sie unter SQL Tablespaces in der Postgre-Dokumentation. SQL

RDSfür SQL Postgre-Kollationen für EBCDIC und andere Mainframe-Migrationen

RDSZu SQL Postgre-Versionen 10 und höher gehört ICU Version 6.0.2, die auf Unicode 10.0 basiert und Kollationen aus dem Unicode Common Locale Data Repository 32 enthält. CLDR Diese Software-Internationalisierungsbibliotheken stellen sicher, dass Zeichenkodierungen unabhängig vom Betriebssystem oder der Plattform einheitlich dargestellt werden. Weitere Informationen zu Unicode CLDR -32 finden Sie in den Versionshinweisen zu Version CLDR 32 auf der Unicode-Website. CLDR Weitere Informationen zu den Internationalisierungskomponenten für Unicode (ICU) finden Sie auf der Website des ICUTechnical Committee (ICU-TC). Informationen zu ICU -60 finden Sie unter Download 60. ICU

Ab Version 14.3 enthält RDS für Postgre SQL auch Kollationen, die bei der Datenintegration und Konvertierung aus basierten Systemen helfen. EBCDIC Der erweiterte binärkodierte dezimale Austauschcode oder die EBCDICCodierung wird häufig von Mainframe-Betriebssystemen verwendet. Diese von Amazon RDS bereitgestellten Sortierungen sind eng definiert, sodass nur die Unicode-Zeichen sortiert werden, die direkt Codepages zugeordnet sind. EBCDIC Die Zeichen werden in der Reihenfolge der EBCDIC Codepunkte sortiert, um eine Datenvalidierung nach der Konvertierung zu ermöglichen. Diese Sortierungen enthalten weder denormalisierte Formen noch Unicode-Zeichen, die nicht direkt einem Zeichen auf der Quellcodepage zugeordnet sind. EBCDIC

Die Zeichenzuordnungen zwischen EBCDIC Codepages und Unicode-Codepunkten basieren auf Tabellen, die von veröffentlicht wurden. IBM Der komplette Satz steht IBM als komprimierte Datei zum Herunterladen zur Verfügung. RDSfor Postgre SQL verwendete diese Zuordnungen mit den von ihnen bereitgestellten Tools, ICU um die in den Tabellen in diesem Abschnitt aufgeführten Sortierungen zu erstellen. Die Sortierungsnamen enthalten eine Sprache und ein Land, wie es von der verlangt wird. ICU In EBCDIC Codepages werden jedoch keine Sprachen angegeben, und einige EBCDIC Codepages decken mehrere Länder ab. Das bedeutet, dass der Sprach- und Länderteil der Sortierungsnamen in der Tabelle willkürlich sind und nicht mit dem aktuellen Gebietsschema übereinstimmen müssen. Mit anderen Worten, die Codepage-Nummer ist der wichtigste Teil des Sortierungsnamens in dieser Tabelle. Sie können jede der in den folgenden Tabellen aufgelisteten Sortierungen in einer beliebigen RDS for Postgre-Datenbank verwenden. SQL

- Unicode to EBCDIC collations table— Einige Mainframe-Datenmigrationstools verwenden LATIN1 oder intern, um Daten zu kodieren und LATIN9 zu verarbeiten. Solche Tools verwenden Roundtrip-Schemata, um die Datenintegrität zu wahren und die umgekehrte Konvertierung zu unterstützen. Die Sortierungen in dieser Tabelle können von Tools verwendet werden, die Daten mithilfe von LATIN1 Kodierung verarbeiten, was keine besondere Behandlung erfordert.
- Unicode to LATIN9 collations table— Sie können diese Kollationen in jeder beliebigen Datenbank RDS für SQL Postgre verwenden.

In der folgenden Tabelle finden Sie Kollationen, die RDS für Postgre verfügbar sind und Codepages SQL EBCDIC Unicode-Codepunkten zuordnen. Es wird empfohlen, die Sortierungen in dieser Tabelle für die Anwendungsentwicklung zu verwenden, für die eine Sortierung auf der Grundlage der Reihenfolge der IBM Codepages erforderlich ist.

Name der Postgre-Sortierung SQL	Beschreibung der Code-Page-Zuordnung und Sortierreihenfolge
DA-DK-CP277-x-Intensivstation	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 277 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 277-Codepunkte sortiert
de-DE-cp273-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 273 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 273-Codepunkte sortiert
en-GB-cp285-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 285 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 285-Codepunkte sortiert
en-US-cp037-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 037 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 37-Codepunkte sortiert
es-ES-cp284-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 284 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 284-Codepunkte sortiert
fi-FI-cp278-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 278 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 278-Codepunkte sortiert
fr-FR-cp297-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 297 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 297-Codepunkte sortiert

Name der Postgre-Sortierung SQL	Beschreibung der Code-Page-Zuordnung und Sortierreihenfolge
it-IT-cp280-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 280 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 280-Codepunkte sortiert
nl-BE-cp500-x-icu	Unicode-Zeichen, die direkt der IBM EBCDIC Codepage 500 (gemäß den Konvertierungstabellen) zugeordnet sind, werden in der Reihenfolge der IBM CP 500-Codepunkte sortiert

Amazon RDS bietet eine Reihe zusätzlicher Sortierungen, die Unicode-Codepunkte, die LATIN9 Zeichen anhand der von veröffentlichten Tabellen zugeordnet sindIBM, in der Reihenfolge der ursprünglichen Codepunkte gemäß der EBCDIC Codepage der Quelldaten sortieren.

Name der Postgre-Sortierung SQL	Beschreibung der Code-Page-Zuordnung und Sortierreihenfolge
DA-DK-CP1142 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1142 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1142-Codepunkte sortiert IBM
DE-DE-CP1141 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1141 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1141-Codepunkte IBM sortiert
de-GB-CP1146 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1146 (gemäß den Konvertierungstabellen) konvertierten

Name der Postgre-Sortierung SQL	Beschreibung der Code-Page-Zuordnung und Sortierreihenfolge
	LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1146-Codepunkte sortiert IBM
en-US-CP1140 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1140 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in IBM der Reihenfolge der CP 1140-Codepunkte sortiert
es-ES-CP1145 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1145 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1145-Codepunkte IBM sortiert
fi-FI-CP1143 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1143 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1143-Codepunkte sortiert IBM
fr-FR-CP1147 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1147 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1147-Codepunkte sortiert IBM

Name der Postgre-Sortierung SQL	Beschreibung der Code-Page-Zuordnung und Sortierreihenfolge
it-IT-CP1144 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1144 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1144-Codepunkte IBM sortiert
nl-BE-CP1148 m-x-icu	Unicode-Zeichen, die den ursprünglich aus der IBM EBCDIC Codepage 1148 (gemäß den Konvertierungstabellen) konvertierten LATIN9 Zeichen zugeordnet sind, werden in der Reihenfolge der CP-1148-Codepunkte sortiert IBM

Im Folgenden finden Sie ein Beispiel für die Verwendung einer RDS for SQL Postgre-Kollatierung.

```
db1=> SELECT pg_import_system_collations('pg_catalog');
 pg_import_system_collations
----------------------------
                          36
db1=> SELECT '¤' < 'a' col1;
 col1
------
 t
db1=> SELECT '¤' < 'a' COLLATE "da-DK-cp277-x-icu" col1;
 col1
------
 f
```

Es wird empfohlen, die Sortierungen in den Unicode to EBCDIC collations table und in der für die Anwendungsentwicklung zu verwenden, Unicode to LATIN9 collations table für die eine Sortierung auf der Grundlage der Reihenfolge der IBM Codepages erforderlich ist. Die folgenden Sortierungen (mit dem Suffix „b“) sind auch in sichtbarpg_collation, sind aber für die Verwendung durch Mainframe-Datenintegrations- und Migrationstools vorgesehen. Sie ordnen Codepages bestimmten

Codepunktverschiebungen zu und erfordern eine besondere Behandlung bei AWS der Sortierung. Mit anderen Worten: Die folgenden Sortierungen werden nicht empfohlen.

- DA-DK-277 b-x-icu
- DA-DK-1142 b-x-icu
- DE-DE-CP273 b-x-icu
- DE-DE-CP1141 b-x-icu
- de-GB-CP1146 b-x-icu
- de-GB-CP285 b-x-icu
- de-US-CP037 b-x-icu
- de-US-CP1140 b-x-icu
- es-ES-CP1145 b-x-icu
- Es-ES-CP284 b-x-icu
- fi-FI-CP1143 b-x-icu
- fr-FR-CP1147 b-x-icu
- fr-FR-CP297 b-x-icu
- it-IT-CP1144 b-x-icu
- it-IT-CP280 b-x-icu
- NL-BE-CP1148 b-x-icu
- NL-BE-CP500 b-x-icu

Weitere Informationen zur Migration von Anwendungen aus Mainframe-Umgebungen zu finden Sie unter Was ist Mainframe-Modernisierung AWS? AWS .

Weitere Informationen zur Verwaltung von Kollationen in Postgre SQL finden Sie unter Collation Support in der Postgre-Dokumentation. SQL

Verwaltung der logischen Steckplatzsynchronisierung für RDS for PostgreSQL

Ab Community-PostgreSQL 17 wurde über den Parameter `sync_replication_slots` oder die zugehörige Funktion, die Slots bei der Ausführung manuell synchronisiert, eine neue Funktion `pg_sync_replication_slots()` zur automatischen Synchronisation logischer Replikationssteckplätze von Primär- zu Standby-Servern eingeführt.

Diese Funktionen sind ab RDS für PostgreSQL 17 verfügbar. Ein typisches Setup besteht aus einer primären Instanz und ihrer Read Replica sowie einem Abonnenten der primären Instanz für die logische Replikation.

Stellen Sie sicher, dass das Abonnement so erstellt wurde, dass die Failover-Option auf true gesetzt ist:

```
CREATE SUBSCRIPTION subname CONNECTION 'host=...' PUBLICATION pubname WITH (failover =
 true);
```

Dadurch wird ein logischer Steckplatz auf dem Herausgeber mit aktiviertem Failover erstellt.

```
postgres=> SELECT slot_name, slot_type, failover FROM pg_catalog.pg_replication_slots;
 slot_name | slot_type | failover
-----------+-----------+----------
 subname   | logical   | t
(1 row)
```

Durch die Aktivierung der Steckplatzsynchronisierung werden alle Steckplätze für die logische Failover-Replikation auf dem primären System automatisch auf den physischen Standby-Steckplätzen erstellt und regelmäßig synchronisiert. Stellen Sie sicher, dass die folgenden Werte über Parametergruppen festgelegt wurden:

- `rds.logical_replication`muss sein1, um die logische Replikation zu aktivieren
- `hot_standby_feedback`muss 1 im Standby-Modus sein
- `rds.logical_slot_sync_dbname`on the standby muss auf einen gültigen Datenbanknamen gesetzt sein

  Der Standardwert des Parameters ist`postgres`. Wenn die logische Veröffentlichungsinstanz über die `postgres` Datenbank verfügt, muss der Standardparameter nicht geändert werden.
- `synchronized_standby_slots`auf der Primärseite muss auf den physischen Replikationssteckplatz des Standby-Speichers eingestellt werden, der synchron sein soll
- `sync_replication_slots`muss sein, 1 um die automatische Synchronisation zu aktivieren

Mit einem Abonnement-Slot mit aktiviertem Failover und den oben genannten Parameterwerten kann der Abonnent beim Heraufstufen einer Standby-Instanz sein Abonnement für diese neu hochgestufte Instanz ändern und die logische Replikation nahtlos fortsetzen.

Herstellen einer Verbindung zu einer DB-Instance, in der die PostgreSQL-Datenbank-Engine ausgeführt wird

Wenn Amazon RDS Ihre DB-Instance bereitgestellt hat, können sie eine beliebige Standard-SQL-Client-Anwendung verwenden, um eine Verbindung zu der DB-Instance herzustellen. Bevor Sie eine Verbindung zu der DB-Instance herstellen können, muss diese verfügbar und zugänglich sein. Ob Sie sich von außerhalb der VPC mit der Instance verbinden können, hängt davon ab, wie Sie die Amazon-RDS-DB-Instance erstellt haben:

- Wenn Sie Ihre DB-Instance öffentlich erstellt haben, können Geräte und EC2 Amazon-Instances außerhalb der VPC eine Verbindung zu Ihrer Datenbank herstellen.
- Wenn Sie Ihre DB-Instance als privat erstellt haben, können nur EC2 Amazon-Instances und Geräte innerhalb der Amazon VPC eine Verbindung zu Ihrer Datenbank herstellen.

Um zu überprüfen, ob Ihre DB-Instance öffentlich oder privat ist, verwenden AWS Management Console Sie die Registerkarte Konnektivität und Sicherheit für Ihre Instance. Unter Security (Sicherheit) finden Sie den Wert „Publicly accessible“ (Öffentlich zugänglich), mit „No“ (Nein) für privat und „Yes“ (Ja) für öffentlich.

Weitere Informationen zu verschiedenen Amazon-RDS- und Amazon-VPC-Konfigurationen und deren Auswirkungen auf die Zugänglichkeit finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.

Inhalt

- Den PSQL-Client installieren
- Suchen der Verbindungsinformationen für eine RDS for PostgreSQL-DB-Instance
- Wird verwendet pgAdmin , um eine Verbindung zu einer RDS for SQL Postgre-DB-Instance herzustellen
- Verwenden Sie psql, um eine Verbindung zu Ihrer RDS for Postgre-DB-Instance herzustellen SQL
- Herstellen einer Verbindung zu RDS für Postgre SQL mit dem Amazon Web Services (AWS) - Treiber JDBC
- Herstellen einer Verbindung zu RDS für Postgre SQL mit dem Amazon Web Services (AWS) Python-Treiber
- Fehlerbehebung bei Verbindungen mit Ihrer RDS für PostgreSQL-Instance
  - Fehler – FATAL: Datenbank name existiert nicht

- Fehler – Keine Verbindung mit dem Server möglich: Zeitüberschreitung für die Verbindung
- Fehler bei Zugriffsregeln für Sicherheitsgruppen

Den PSQL-Client installieren

Um von einer Instance aus eine Verbindung zu Ihrer EC2 DB-Instance herzustellen, können Sie einen PostgreSQL-Client auf der EC2 Instance installieren. Führen Sie den folgenden Befehl aus, um die neueste Version des psql-Clients auf Amazon Linux 2023 zu installieren:

```
sudo dnf install postgresql<version number>
```

Um die neueste Version des psql-Clients auf Amazon Linux 2 zu installieren, führen Sie den folgenden Befehl aus:

```
sudo yum install -y postgresql
```

Führen Sie den folgenden Befehl aus, um die neueste Version des psql-Clients auf Ubuntu zu installieren:

```
sudo apt install -y postgresql-client
```

Suchen der Verbindungsinformationen für eine RDS for PostgreSQL-DB-Instance

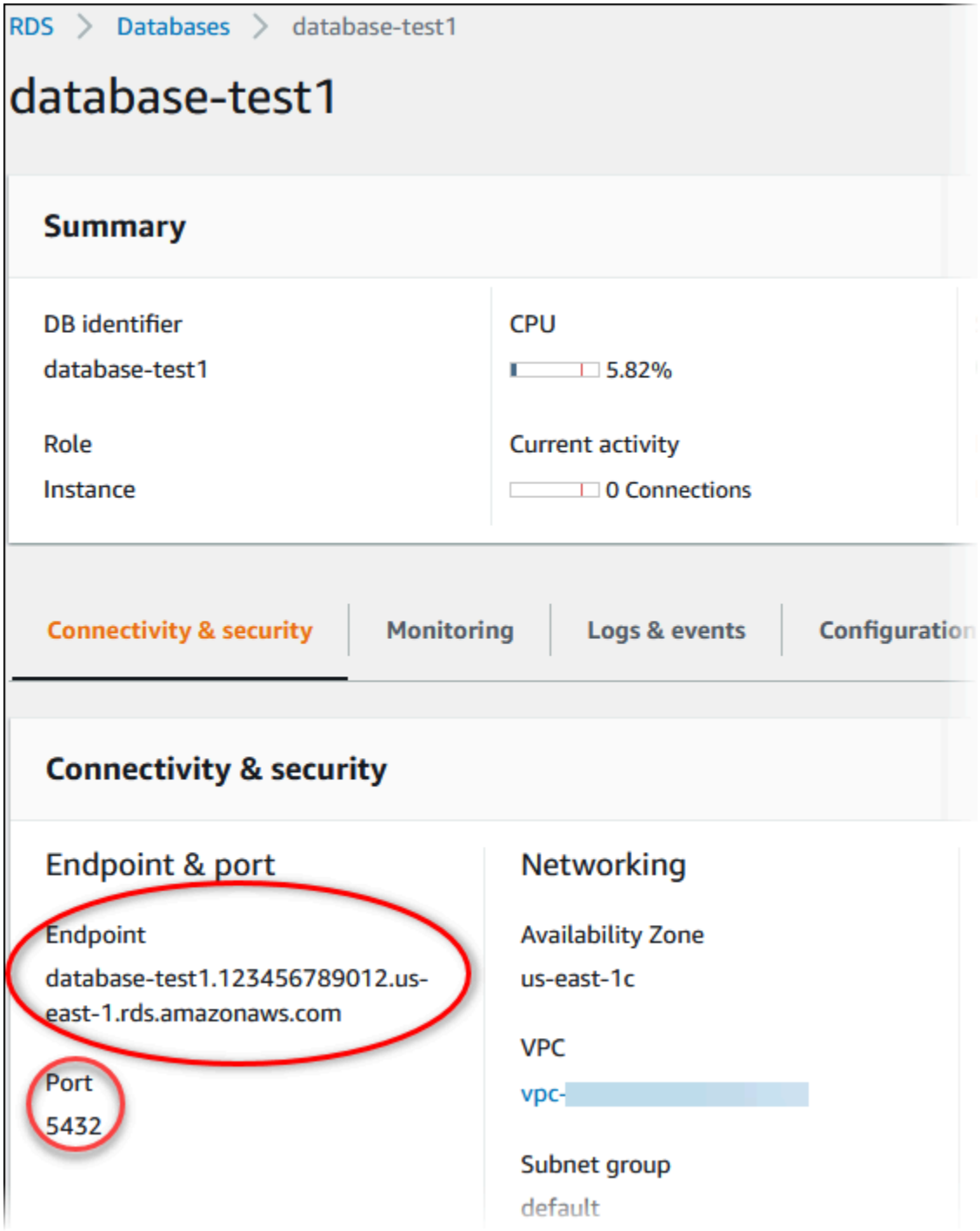
Wenn die DB-Instance verfügbar und zugänglich ist, können Sie eine Verbindung herstellen, indem Sie der SQL-Clientanwendung die folgenden Informationen bereitstellen:

- Der Endpunkt der DB-Instance, der als Hostname (DNS-Name) für die Instanz dient.
- Den Port, über den die DB-Instance kommuniziert. Der Standardport für PostgreSQL lautet 5432.
- Den Benutzernamen und das Passwort für die DB-Instanz. Der Standardwert „Haupt-Benutzername“ für PostgreSQL ist `postgres`.
- Der Name und das Passwort der Datenbank (DB-Name).

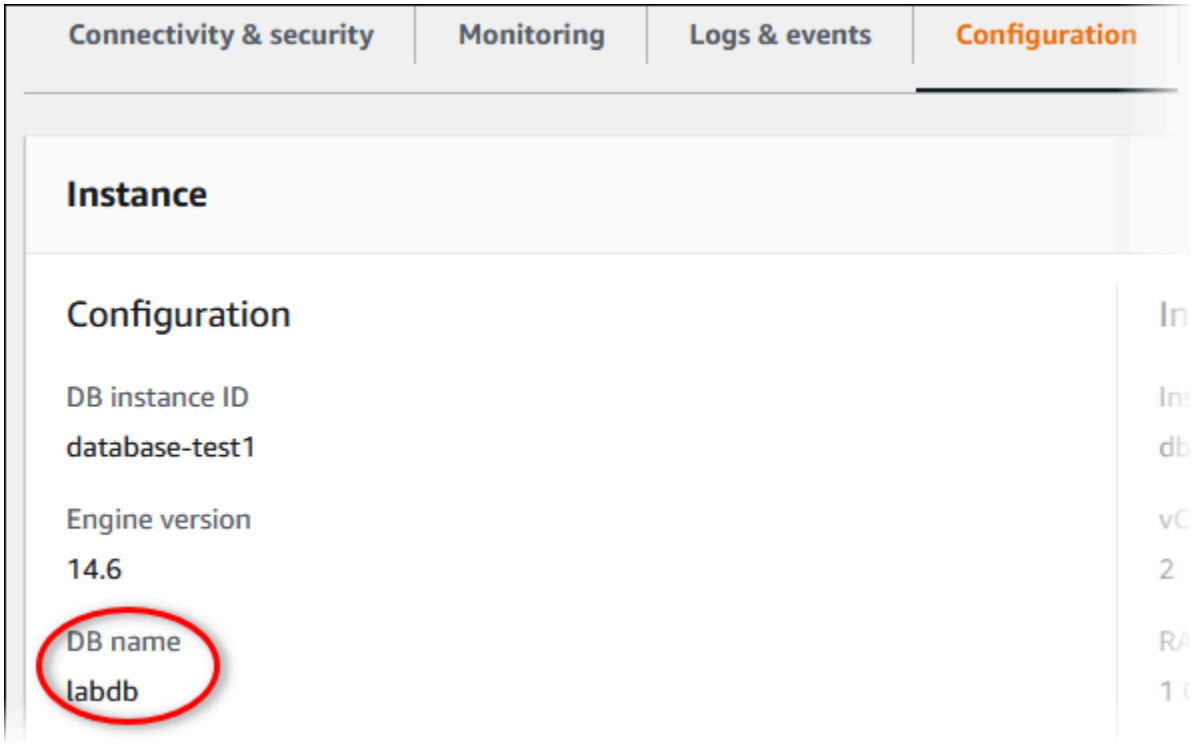
Sie können diese Details mithilfe des AWS CLI describe-db-instancesBefehls AWS Management Console, oder der Amazon DBInstancesRDS-API-Describe-Operation abrufen.

Um den Endpunkt, die Portnummer und den DB-Namen zu finden, verwenden Sie den AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Öffnen Sie die RDS-Konsole und wählen Sie Databases (Datenbanken), um eine Liste Ihrer DB-Instances anzuzeigen.
3. Wählen Sie den Namen der PostgreSQL-DB-Instance, um deren Details anzuzeigen.
4. Kopieren Sie auf der Registerkarte Connectivity & security (Anbindung und Sicherheit) den Endpunkt. Notieren Sie sich auch die Portnummer. Sie benötigen sowohl den Endpunkt als auch die Portnummer, um die Verbindung zur DB-Instance herzustellen.



5. Notieren Sie auf der Registerkarte Konfiguration den DB-Namen. Wenn Sie eine Datenbank erstellt haben, als Sie die RDS für PostgreSQL-Instance erstellt haben, wird der Name unter DB-Name aufgeführt. Wenn Sie keine Datenbank erstellt haben, zeigt der DB-Name einen Bindestrich (-) an.



Nachfolgend werden zwei Möglichkeiten gezeigt, eine Verbindung mit einer PostgreSQL-DB-Instance herzustellen. Im ersten Beispiel wird pgAdmin verwendet, ein beliebtes Open Source-Tool für die PostgreSQL-Administration und -Entwicklung. Im zweiten Beispiel wird psql verwendet, ein Befehlszeilen-Dienstprogramm, das in jeder PostgreSQL-Installation enthalten ist.

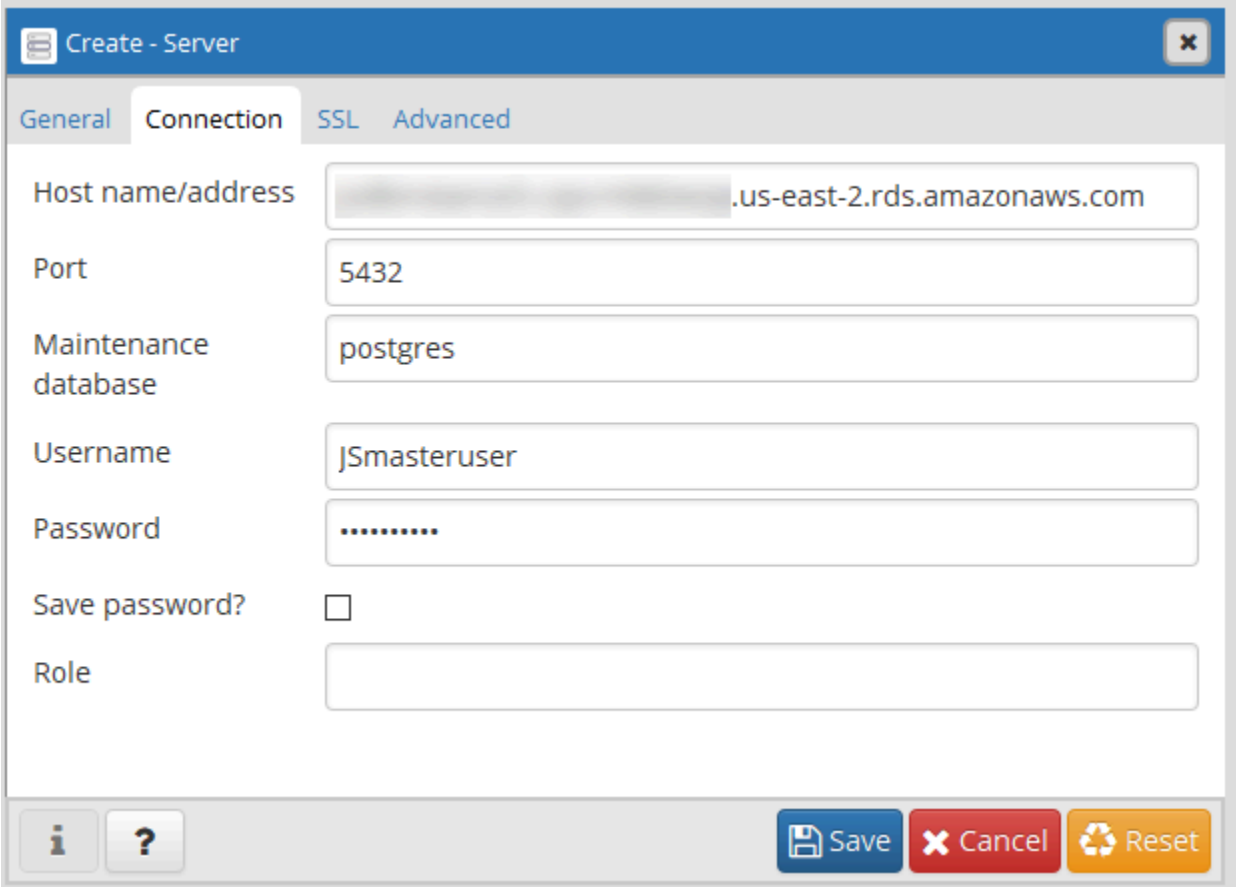
Wird verwendet pgAdmin , um eine Verbindung zu einer RDS for SQL Postgre-DB-Instance herzustellen

Sie können das Open-Source-Tool verwenden, um eine Verbindung pgAdmin zu Ihrer RDS for Postgre-DB-Instance herzustellen. SQL Sie können es pgAdmin von http://www.pgadmin.org/ herunterladen und installieren, ohne über eine lokale Postgre-Instanz SQL auf Ihrem Client-Computer zu verfügen.

Um eine Verbindung zu Ihrer RDS SQL Postgre-DB-Instance herzustellen, verwenden Sie pgAdmin

1. Starten Sie die pgAdmin Anwendung auf Ihrem Client-Computer.
2. Klicken Sie auf der Registerkarte Dashboard auf Add New Server (Neuen Server hinzufügen).
3. Geben Sie im Dialogfeld Create — Server auf der Registerkarte Allgemein einen Namen ein, unter dem der Server identifiziert werden sollpgAdmin.
4. Geben Sie auf der Registerkarte Verbindung die folgenden Informationen der DB-Instance ein:

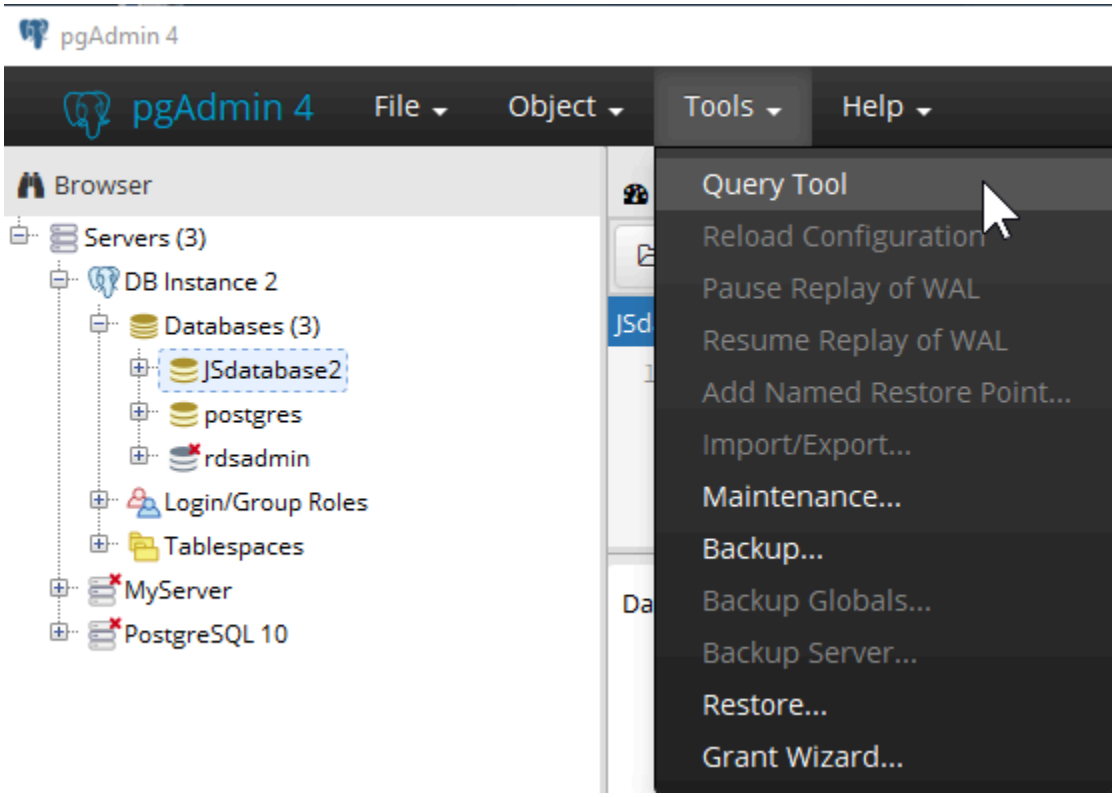
- Geben Sie unter Host den Endpunkt ein, z. B. `mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com`.
- Geben Sie in das Feld Port den zugehörigen Port ein.
- Geben Sie für Benutzername den Benutzernamen ein, den Sie beim Erstellen der DB-Instance eingegeben haben (wenn Sie den „Haupt-Benutzernamen“ vom Standardwert `postgres` geändert haben).
- Geben Sie unter Password (Passwort) das beim Erstellen der DB-Instance festgelegte Passwort ein.



5. Wählen Sie Save (Speichern) aus.

   Falls Probleme beim Herstellen der Verbindung auftreten, lesen Sie die Informationen unter Fehlerbehebung bei Verbindungen mit Ihrer RDS für PostgreSQL-Instance.

6. Um im pgAdmin Browser auf eine Datenbank zuzugreifen, erweitern Sie Server, DB-Instance und Datenbanken. Wählen Sie anschließend die gewünschte Datenbank in der DB-Instance aus.



7. Um ein Fenster zu öffnen, in dem Sie SQL Befehle eingeben können, wählen Sie Tools, Query Tool.

Verwenden Sie psql, um eine Verbindung zu Ihrer RDS for Postgre-DB-Instance herzustellen SQL

Sie können eine lokale Instanz des Befehlszeilenprogramms psql verwenden, um eine Verbindung zu einer RDS for SQL Postgre-DB-Instance herzustellen. Sie müssen entweder Postgre SQL oder den psql-Client auf Ihrem Client-Computer installiert haben.

Sie können den SQL Postgre-Client von der Postgre-Website herunterladen. SQL Folgen Sie den Anweisungen für Ihr Betriebssystem, um diese Version zu installieren.

Um mit psql eine Verbindung zu Ihrer RDS for SQL Postgre-DB-Instance herzustellen, müssen Sie host (DNS) -Informationen, Zugangsdaten und den Namen der Datenbank angeben.

Verwenden Sie eines der folgenden Formate, um eine Verbindung zu Ihrer RDS for Postgre-DB-Instance herzustellen. SQL Sie werden beim Verbinden zur Eingabe eines Passworts aufgefordert. Verwenden Sie in Stapelaufträgen oder Skripts die Option `--no-password`. Diese Option ist für die gesamte Sitzung festgelegt.

> **Note**
>
> Ein Verbindungsversuch mit `--no-password` schlägt fehl, wenn der Server eine Passwortauthentifizierung erfordert und ein Passwort aus anderen Quellen nicht verfügbar ist. Weitere Informationen finden Sie in der PSQL-Dokumentation.

Wenn Sie zum ersten Mal eine Verbindung zu dieser DB-Instance herstellen oder wenn Sie noch keine Datenbank für diese RDS für SQL Postgres-Instance erstellt haben, können Sie mit dem „Master-Benutzernamen“ und dem Passwort eine Verbindung zur Postgres-Datenbank herstellen.

Verwenden Sie unter Unix das folgende Format.

```
psql \
   --host=<DB instance endpoint> \
   --port=<port> \
   --username=<master username> \
   --password \
   --dbname=<database name>
```

Verwenden Sie unter Windows das folgende Format.

```
psql ^
   --host=<DB instance endpoint> ^
   --port=<port> ^
   --username=<master username> ^
   --password ^
   --dbname=<database name>
```

Der folgende Befehl stellt beispielsweise eine Verbindung zu einer Datenbank her, die `mypgdb` auf einer Postgre-DB-Instance aufgerufen wird, die mit fiktiven Anmeldeinformationen aufgerufen wird. SQL `mypostgresql`

```
psql --host=mypostgresql.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --
username=awsuser --password --dbname=mypgdb
```

Herstellen einer Verbindung zu RDS für Postgre SQL mit dem Amazon Web Services (AWS) -Treiber JDBC

Der Amazon Web Services (AWS) JDBC -Treiber ist als erweiterter JDBC Wrapper konzipiert. Dieser Wrapper ergänzt einen vorhandenen JDBC Treiber und erweitert dessen Funktionalität. Der Treiber ist Drop-In-kompatibel mit dem JDBC Community-PG-Treiber.

Um den AWS JDBC Treiber zu installieren, hängen Sie die AWS JDBC Treiber-.jar-Datei an (befindet sich in der Anwendung`CLASSPATH`) und behalten Sie die Verweise auf den jeweiligen Community-Treiber bei. Aktualisieren Sie das jeweilige URL Verbindungspräfix wie folgt:

- `jdbc:postgresql://` auf `jdbc:aws-wrapper:postgresql://`

Weitere Informationen zum AWS JDBC Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im JDBC GitHub Treiber-Repository von Amazon Web Services (AWS).

Herstellen einer Verbindung zu RDS für Postgre SQL mit dem Amazon Web Services (AWS) Python-Treiber

Der Amazon Web Services (AWS) Python-Treiber ist als fortschrittlicher Python-Wrapper konzipiert. Dieser Wrapper ergänzt den Open-Source-Treiber Psycopg und erweitert dessen Funktionalität. Der AWS Python-Treiber unterstützt Python-Versionen 3.8 und höher. Sie können das `aws-advanced-python-wrapper` Paket zusammen mit den `psycopg` Open-Source-Paketen mit dem `pip` Befehl installieren.

Weitere Informationen zum AWS Python-Treiber und vollständige Anweisungen zu seiner Verwendung finden Sie im GitHub Python-Treiber-Repository von Amazon Web Services (AWS).

Fehlerbehebung bei Verbindungen mit Ihrer RDS für PostgreSQL-Instance

Themen

- Fehler – FATAL: Datenbank name existiert nicht
- Fehler – Keine Verbindung mit dem Server möglich: Zeitüberschreitung für die Verbindung
- Fehler bei Zugriffsregeln für Sicherheitsgruppen

Fehler – FATAL: Datenbank *name* existiert nicht

Wenn Sie bei der Verbindung einen Fehler wie `FATAL: database name does not exist` erhalten, versuchen Sie, den Standard-Datenbanknamen postgres für die Option `--dbname` zu verwenden.

Fehler – Keine Verbindung mit dem Server möglich: Zeitüberschreitung für die Verbindung

Wenn die Verbindung mit der DB-Instance nicht hergestellt werden kann, wird meistens der Fehler `Could not connect to server: Connection timed out.` angezeigt. Ist dies der Fall, gehen Sie wie folgt vor:

- Prüfen Sie, ob der DB-Instance-Endpunkt als Hostname sowie die richtige Portnummer angegeben wurden.
- Stellen Sie sicher, dass die öffentliche Zugänglichkeit der DB-Instance auf Ja festgelegt ist, um externe Verbindungen zuzulassen. Informationen zum Ändern der Einstellung Öffentlicher Zugriff finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie sicher, dass der Benutzer, der sich mit der Datenbank verbindet, CONNECT-Zugriff darauf hat. Sie können folgende Abfrage verwenden, um den CONNECT-Zugriff auf die Datenbank bereitzustellen.

  ```
  GRANT CONNECT ON DATABASE database name TO username;
  ```

- Prüfen Sie, ob die der DB-Instance zugewiesene Sicherheitsgruppe die erforderlichen Regeln enthält, um den Zugriff durch alle vorhandenen Firewalls zu ermöglichen. Beispiel: Bei der Erstellung der DB-Instance wurde der Standardport 5432 festgelegt und die Firewall-Regeln des Unternehmens blockieren Verbindungen mit diesem Port von externen Unternehmensgeräten.

  Sie können dieses Problem beheben, indem Sie für die DB-Instance einen anderen Port verwenden. Stellen Sie außerdem sicher, dass die mit der DB-Instance verknüpfte Sicherheitsgruppe eingehende Verbindungen mit dem neuen Port zulässt. Informationen zum Ändern der Einstellung für Datenbank-Port finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Prüfen Sie, ob der Port, den Sie verwenden möchten, bereits von einer lokalen Instanz von PostgreSQL oder einem anderen Dienst belegt ist, der auf Ihrem Computer läuft. Wenn Sie beispielsweise eine lokale PostgreSQL-Datenbank auf demselben Port haben (Standard ist 5432), kann dies eine erfolgreiche Verbindung zur RDS for PostgreSQL-DB-Instance verhindern. Stellen

Sie sicher, dass der Port frei ist, oder versuchen Sie, wenn möglich, eine Verbindung mit einer anderen Portnummer herzustellen.

- Weitere Informationen finden Sie auch unter Fehler bei Zugriffsregeln für Sicherheitsgruppen.

Fehler bei Zugriffsregeln für Sicherheitsgruppen

Die bei Weitem häufigsten Verbindungsprobleme treten in Verbindung mit den Zugriffsregeln der Sicherheitsgruppe auf, die der DB-Instance zugewiesen wurde. Wenn Sie bei der Erstellung der DB-Instance die Standard-Sicherheitsgruppe verwendet haben, ist es sehr wahrscheinlich, dass die Regeln in der Sicherheitsgruppe den Zugriff auf die Instance nicht zulassen.

Damit die Verbindung möglich ist, muss die Sicherheitsgruppe, die Sie der DB-Instance bei der Erstellung zugewiesen haben, den Zugriff auf die DB-Instance zulassen. Wenn die DB-Instance beispielsweise in einer VPC erstellt wurde, muss sie über eine VPC-Sicherheitsgruppe verfügen, die die Verbindungen zulässt. Prüfen Sie, ob die DB-Instance mit einer Sicherheitsgruppe erstellt wurde, die keine Verbindungen von dem Gerät oder der EC2 Amazon-Instance autorisiert, auf dem die Anwendung ausgeführt wird.

Sie können eine Regel für eingehenden Datenverkehr in der Sicherheitsgruppe hinzufügen oder ändern. Die Auswahl der Option My IP (Meine IP) für Source (Quelle) ermöglicht Zugriff auf die DB-Instance von der IP-Adresse, die in Ihrem Browser erkannt wird. Weitere Informationen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

Wurde die DB-Instance außerhalb einer VPC erstellt, muss sie über eine Datenbank-Sicherheitsgruppe verfügen, die diese Verbindungen zulässt.

Weitere Informationen zu Amazon RDS-Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Verbindungen zu RDS für Postgre sichern SQL mit/SSLTLS

RDSfür Postgre SQL unterstützt die Secure Socket Layer (SSL) -Verschlüsselung für Postgre-DB-Instances. SQL Mithilfe SSL können Sie eine SQL Postgre-Verbindung zwischen Ihren Anwendungen und Ihren Postgre-DB-Instances verschlüsseln. SQL Sie können auch die Verwendung aller Verbindungen zu Ihrer SQL Postgre-DB-Instance erzwingen. SSL RDSfür Postgre unterstützt SQL auch Transport Layer Security (TLS), das Nachfolgeprotokoll von. SSL

Weitere Informationen über Amazon RDS und Datenschutz, einschließlich der Verschlüsselung von Verbindungen mitSSL/TLS, finden Sie unterDatenschutz in Amazon RDS.

Themen

- Verwenden von SSL mit einer PostgreSQL-DB-Instance
- Aktualisierung von Anwendungen für die Verbindung zu SQL Postgre-DB-Instances mithilfe neuer SSL TLS /-Zertifikate

Verwenden von SSL mit einer PostgreSQL-DB-Instance

Amazon RDS unterstützt die Secure Socket Layer (SSL)-Verschlüsselung für PostgreSQL-DB-Instances. Sie können die PostgreSQL-Verbindung zwischen Ihren Anwendungen und Ihren PostgreSQL-DB-Instances mit SSL verschlüsseln. Standardmäßig verwendet und erwartet RDS for PostgreSQL, dass sich alle Clients über SSL/TLS verbinden, aber Sie können dies auch verlangen. RDS für PostgreSQL unterstützt die Transport Layer Security (TLS) -Versionen 1.1, 1.2 und 1.3.

Allgemeine Informationen zum SSL-Support und zu PostgreSQL-Datenbanken finden Sie unter SSL-Support in der PostgreSQL-Dokumentation. Informationen zur Verwendung einer SSL-Verbindung über JDBC finden Sie unter Konfigurieren des Clients in der PostgreSQL-Dokumentation.

SSL-Unterstützung ist in allen AWS Regionen für PostgreSQL verfügbar. Amazon RDS erzeugt ein SSL-Zertifikat für Ihre DB-Instance, wenn die Instance erstellt wird. Wenn Sie die SSL-Zertifikatsverifizierung aktivieren, enthält das SSL-Zertifikat den Endpunkt der DB-Instance als Allgemeinen Namen (Common Name, CN) für das SSL-Zertifikat, sodass es vor Spoofing-Angriffen schützt.

Themen

- Herstellen einer Verbindung mit einer PostgreSQL-DB-Instance über SSL
- Erfordern einer SSL-Verbindung zu einer PostgreSQL-DB-Instance

- Bestimmen des SSL-Verbindungsstatus
- SSL-Verschlüsselungssammlungen in RDS for PostgreSQL

Herstellen einer Verbindung mit einer PostgreSQL-DB-Instance über SSL

So stellen Sie über SSL eine Verbindung zu einer PostgreSQL-DB-Instance her

1. Laden Sie das Zertifikat herunter.

   Informationen zum Herunterladen von Zertifikaten finden Sie unter .
2. Stellen Sie über SSL eine Verbindung zu einer PostgreSQL-DB-Instance her.

   Wenn Sie eine Verbindung über SSL herstellen, kann Ihr Client entscheiden, ob die Zertifikatskette überprüft werden soll. Wenn Ihre Verbindungsparameter `sslmode=verify-ca` oder `sslmode=verify-full` angeben, verlangt Ihr Client, dass sich die RDS CA-Zertifikate im Trust Store befinden oder von der Verbindungs-URL referenziert werden. Diese Anforderung dient zur Prüfung der Zertifikatskette, die Ihr Datenbankzertifikat signiert.

   Wenn ein Client wie psql oder JDBC mit SSL-Unterstützung konfiguriert ist, versucht dieser zunächst standardmäßig, die Verbindung zur Datenbank über SSL herzustellen. Wenn der Client keine Verbindung über SSL herstellen kann, stellt er die Verbindung ohne SSL her. Der für libpq-basierte Clients (wie psql) und JDBC verwendete `sslmode`-Standardmodus ist unterschiedlich. Die libpq-basierten und JDBC-Clients verwenden standardmäßig. `prefer`

   Verwenden Sie den Parameter `sslrootcert`, um auf das Zertifikat zu verweisen, beispielsweise `sslrootcert=rds-ssl-ca-cert.pem`.

Das folgende Beispiel veranschaulicht die Verwendung von `psql`, um eine Verbindung mit SSL und Zertifikatsüberprüfung mit einer PostgreSQL-DB-Instance herzustellen.

```
$ psql "host=db-name.555555555555.ap-southeast-1.rds.amazonaws.com
    port=5432 dbname=testDB user=testuser sslrootcert=rds-ca-rsa2048-g1.pem
 sslmode=verify-full"
```

Erfordern einer SSL-Verbindung zu einer PostgreSQL-DB-Instance

Mit dem Parameter `rds.force_ssl` können Sie es erforderlich machen, dass Verbindungen zu Ihrer PostgreSQL-DB-Instance SSL verwenden müssen. Der Standardwert des `rds.force_ssl`

Parameters ist 1 (ein) für RDS für PostgreSQL Version 15 und höher. Für alle anderen Hauptversionen von RDS für PostgreSQL 14 und älter ist der Standardwert dieses Parameters 0 (aus). Sie können den Parameter `rds.force_ssl` auf 1 (on) setzen, um SSL/TLS für Verbindungen zu Ihrem DB-Cluster zu verlangen. Sie können den Parameter `rds.force_ssl` auf 1 (ein) stellen und damit erforderlich machen, dass Verbindungen zu Ihrer PostgreSQL-DB-Instance SSL verwenden müssen.

Wenn Sie den Wert dieses Parameters ändern möchten, müssen Sie eine benutzerdefinierte DB-Parametergruppe erstellen. Anschließend ändern Sie den Wert für `rds.force_ssl` in Ihrer benutzerdefinierten DB-Parametergruppe in 1, um diese Funktion zu aktivieren. Wenn Sie die benutzerdefinierte DB-Parametergruppe vorbereiten, bevor Sie Ihre RDS-for-PostgreSQL-DB-Instance erstellen, können Sie sie während des Erstellungsprozesses (anstelle einer Standardparametergruppe) auswählen. Wenn Sie diesen Schritt ausführen, nachdem Ihre RDS-for-PostgreSQL-DB-Instance bereits ausgeführt wird, müssen Sie die Instance neu starten, damit diese die benutzerdefinierte Parametergruppe verwendet. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

Wenn die `rds.force_ssl`-Funktion auf Ihrer DB-Instance aktiv ist, werden Verbindungsversuche ohne SSL mit der folgenden Meldung abgelehnt:

```
$ psql -h db-name.555555555555.ap-southeast-1.rds.amazonaws.com port=5432 dbname=testDB
 user=testuser
psql: error: FATAL: no pg_hba.conf entry for host "w.x.y.z", user "testuser", database
 "testDB", SSL off
```

Bestimmen des SSL-Verbindungsstatus

Der verschlüsselte Status Ihrer Verbindung wird auf dem Anmelde-Banner angezeigt, wenn Sie sich mit der DB-Instance verbinden:

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.
postgres=>
```

Sie können auch die Erweiterung `sslinfo` laden und dann die Funktion `ssl_is_used()` aufrufen, um festzustellen, ob SSL verwendet wird. Die Funktion gibt `t` zurück, wenn für die Verbindung SSL genutzt wird. Andernfalls gibt sie `f` zurück.

```
postgres=> CREATE EXTENSION sslinfo;
CREATE EXTENSION
postgres=> SELECT ssl_is_used();
ssl_is_used
---------
t
(1 row)
```

Wenn Sie detailliertere Informationen erhalten möchten, können Sie die folgende Abfrage verwenden, um Informationen von `pg_settings` abzurufen:

```
SELECT name as "Parameter name", setting as value, short_desc FROM pg_settings WHERE
 name LIKE '%ssl%';
             Parameter name             |                  value                  |
                  short_desc
----------------------------------------+-----------------------------------------
+-------------------------------------------------------
 ssl                                    | on                                      |
 Enables SSL connections.
 ssl_ca_file                            | /rdsdbdata/rds-metadata/ca-cert.pem     |
 Location of the SSL certificate authority file.
 ssl_cert_file                          | /rdsdbdata/rds-metadata/server-cert.pem |
 Location of the SSL server certificate file.
 ssl_ciphers                            | HIGH:!aNULL:!3DES                       |
 Sets the list of allowed SSL ciphers.
 ssl_crl_file                           |                                         |
 Location of the SSL certificate revocation list file.
 ssl_dh_params_file                     |                                         |
 Location of the SSL DH parameters file.
 ssl_ecdh_curve                         | prime256v1                              |
 Sets the curve to use for ECDH.
 ssl_key_file                           | /rdsdbdata/rds-metadata/server-key.pem  |
 Location of the SSL server private key file.
 ssl_library                            | OpenSSL                                 |
 Name of the SSL library.
 ssl_max_protocol_version               |                                         |
 Sets the maximum SSL/TLS protocol version to use.
 ssl_min_protocol_version               | TLSv1.2                                 |
 Sets the minimum SSL/TLS protocol version to use.
 ssl_passphrase_command                 |                                         |
 Command to obtain passphrases for SSL.
 ssl_passphrase_command_supports_reload | off                                     |
 Also use ssl_passphrase_command during server reload.
```

```
 ssl_prefer_server_ciphers              | on                                      |
 Give priority to server ciphersuite order.
(14 rows)
```

Sie können auch alle Informationen über die SSL-Nutzung Ihrer RDS-for-PostgreSQL-DB-Instance nach Prozess, Client und Anwendung sammeln, indem Sie die folgende Abfrage verwenden:

```
SELECT datname as "Database name", usename as "User name", ssl, client_addr,
 application_name, backend_type
   FROM pg_stat_ssl
   JOIN pg_stat_activity
   ON pg_stat_ssl.pid = pg_stat_activity.pid
   ORDER BY ssl;
 Database name | User name | ssl |  client_addr   |    application_name    |
 backend_type
---------------+-----------+-----+----------------+------------------------
+------------------------------
               |           | f   |                |                        | autovacuum
 launcher
               | rdsadmin  | f   |                |                        | logical
 replication launcher
               |           | f   |                |                        | background
 writer
               |           | f   |                |                        |
 checkpointer
               |           | f   |                |                        | walwriter
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      |                        | client
 backend
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      | PostgreSQL JDBC Driver | client
 backend
 postgres      | postgres  | t   | 204.246.162.36 | psql                   | client
 backend
(8 rows)
```

Wenn Sie die Cipher identifizieren möchten, die für Ihre SSL-Verbindung verwendet wird, können Sie folgende Abfrage erstellen:

```
postgres=> SELECT ssl_cipher();
ssl_cipher
--------------------
DHE-RSA-AES256-SHA
(1 row)
```

Weitere Informationen zur Option `sslmode` finden Sie unter Datenbankverbindung-Steuerungsfunktionen in der PostgreSQL-Dokumentation.

SSL-Verschlüsselungssammlungen in RDS for PostgreSQL

Der PostgreSQL-Konfigurationsparameter ssl_ciphers gibt die Kategorien von Cipher Suites an, die für SSL-Verbindungen zur Datenbank zulässig sind, wenn TLS 1.2 und niedriger verwendet wird.

In RDS for PostgreSQL 16 und höher können Sie den `ssl_ciphers` Parameter so ändern, dass er bestimmte Werte aus den Chiffrier-Suites auf der Zulassungsliste verwendet. Dies ist ein dynamischer Parameter, für den kein Neustart der Datenbankinstanz erforderlich ist. Verwenden Sie entweder die Amazon RDS-Konsole oder den folgenden AWS CLI-Befehl, um die zugelassenen Cipher Suites anzuzeigen:

```
aws rds describe-db-parameters --db-parameter-group-name <your-parameter-group> --
region <region> --endpoint-url <endpoint-url> --output json | jq '.Parameters[] |
 select(.ParameterName == "ssl_ciphers")'
```

In der folgenden Tabelle sind sowohl die standardmäßigen Verschlüsselungssammlungen als auch die zulässigen Verschlüsselungssammlungen für Versionen aufgeführt, die benutzerdefinierte Konfigurationen unterstützen.

PostgreSQL-Engine-Version	Standardwerte für die ssl_cipher Suite	Zulässige benutzerdefinierte Werte der ssl_cipher-Suite

17	`HIGH:!aNULL:!3DES`	`TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
		`TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256`
		`TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
		`TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`

PostgreSQL-Engine-Version	Standardwerte für die ssl_cipher Suite	Zulässige benutzerdefinierte Werte der ssl_cipher-Suite
		`TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`

16	`HIGH:!aNULL:!3DES`	`TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
		`TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256`
		`TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
		`TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`
		`TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`

15	`HIGH:!aNULL:!3DES`	Benutzerdefinierte ssl_ciphers werden nicht unterstützt
14	`HIGH:!aNULL:!3DES`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt
13	`HIGH:!aNULL:!3DES`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt
12	`HIGH:!aNULL:!3DES`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt
11.4 und höhere Nebenversionen	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt

PostgreSQL-Engine-Version	Standardwerte für die ssl_cipher Suite	Zulässige benutzerdefinierte Werte der ssl_cipher-Suite
11,1, 11,2	`HIGH:MEDIUM:+3DES:!aNULL`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt
10.9 und höhere Nebenversionen	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt
10.7 und niedrigere Nebenversionen	`HIGH:MEDIUM:+3DES:!aNULL`	Benutzerdefinierte SSL_Ciphers werden nicht unterstützt

Um alle Instanzverbindungen für die Verwendung der `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384` Cipher Suite zu konfigurieren, ändern Sie Ihre Parametergruppe wie im folgenden Beispiel gezeigt:

```
aws rds modify-db-parameter-group --db-parameter-group-name <your-parameter-group> --
parameters
 "ParameterName='ssl_ciphers',ParameterValue='TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384',ApplyMet
```

In diesem Beispiel wird eine ECDSA-Chiffre verwendet, bei der Ihre Instance eine Zertifizierungsstelle mit Elliptischer Kurven-Kryptografie (ECC) verwenden muss, um eine Verbindung herzustellen. Informationen zu den von Amazon RDS bereitgestellten Zertifizierungsstellen finden Sie unter Zertifizierungsstellen.

Sie können die verwendeten Chiffren mithilfe der unter Bestimmung des SSL-Verbindungsstatus beschriebenen Methoden überprüfen.

Chiffren können je nach Kontext unterschiedliche Namen haben:

- Die Chiffren auf der Zulassungsliste, die Sie in Ihrer Parametergruppe konfigurieren können, werden mit ihren IANA-Namen bezeichnet.
- Die Banner `sslinfo` und `psql` logon verweisen auf Chiffren, die ihre OpenSSL-Namen verwenden.

Standardmäßig ist der Wert von `ssl_max_protocol_version` in RDS für PostgreSQL 16 und höher TLS v1.3. Sie müssen den Wert dieses Parameters auf TLS v1.2 setzen, da TLS v1.3 die im Parameter angegebenen Verschlüsselungskonfigurationen nicht verwendet. `ssl_ciphers`

Wenn Sie den Wert auf TLS v1.2 festlegen, verwenden Verbindungen nur die Chiffren, in denen Sie definiert haben. `ssl_ciphers`

```
aws rds modify-db-parameter-group --db-parameter-group-name <your-parameter-group> --
parameters
 "ParameterName='ssl_max_protocol_version',ParameterValue='TLSv1.2',ApplyMethod=immediate"
```

Um sicherzustellen, dass Datenbankverbindungen SSL verwenden, setzen Sie den Wert in Ihrer Parametergruppe `rds.force_ssl parameter` auf 1. Weitere Informationen zu Parametern und Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Aktualisierung von Anwendungen für die Verbindung zu SQL Postgre-DB-Instances mithilfe neuer SSL TLS /-Zertifikate

Zertifikate, die für Secure Socket Layer oder Transport Layer Security (SSL/TLS) verwendet werden, haben in der Regel eine festgelegte Lebensdauer. Wenn Dienstanbieter ihre Certificate-Authority(CA)-Zertifikate aktualisieren, müssen Clients ihre Anwendungen aktualisieren, um die neuen Zertifikate zu verwenden. Im Folgenden finden Sie Informationen darüber, wie Sie feststellen können, ob Ihre Client-AnwendungenSSL/verwenden, TLS um eine Verbindung zu Ihrer Amazon RDS for SQL Postgre-DB-Instance herzustellen. Sie finden auch Informationen darüber, wie Sie prüfen können, ob diese Anwendungen das Serverzertifikat überprüfen, wenn sie eine Verbindung herstellen.

> **Note**
>
> Eine Client-Anwendung, die so konfiguriert ist, dass sie das Serverzertifikat vor der TLS VerbindungSSL/verifiziert, muss über ein gültiges CA-Zertifikat im Trust Store des Clients verfügen. Aktualisieren Sie den Client-Truststore bei Bedarf für neue Zertifikate.

Nach der Aktualisierung der CA-Zertifikate in den Trust Stores Ihrer Client-Anwendung können Sie die Zertifikate auf Ihren DB-Instances rotieren. Es wird nachdrücklich empfohlen, diese Verfahren vor der Implementierung in Produktionsumgebungen in einer Nicht-Produktionsumgebung zu testen.

Weitere Informationen zur Zertifikatrotation finden Sie unter Rotieren Ihrer SSL/TLS-Zertifikate. Weitere Informationen zum Herunterladen von Zertifikaten finden Sie unter . Hinweise zur Verwendung vonSSL/TLSmit SQL Postgre-DB-Instances finden Sie unterVerwenden von SSL mit einer PostgreSQL-DB-Instance.

Themen

- Feststellen, ob Anwendungen mithilfe von Verbindungen zu SQL Postgre-DB-Instances herstellen SSL
- Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert
- Aktualisieren des Trust Stores Ihrer Anwendung
- Verwenden von SSL TLS /-Verbindungen für verschiedene Arten von Anwendungen

Feststellen, ob Anwendungen mithilfe von Verbindungen zu SQL Postgre-DB-Instances herstellen SSL

Prüfen Sie die DB-Instance-Konfiguration auf den Wert des Parameters `rds.force_ssl`. Standardmäßig ist der `rds.force_ssl` Parameter für DB-Instances, die SQL Postgre-Versionen vor Version 15 verwenden, auf `0` (aus) gesetzt. Standardmäßig `rds.force_ssl` ist er für DB-Instances, die SQL Postgre-Version 15 und spätere Hauptversionen verwenden, auf 1 (on) gesetzt. Wenn der `rds.force_ssl` Parameter auf 1 (on) gesetzt ist, müssen ClientsSSL/TLSfür Verbindungen verwenden. Weitere Informationen zu Parametergruppen finden Sie unter Parametergruppen für Amazon RDS.

Wenn Sie die Hauptversion von RDS Postgre SQL Version 9.5 oder höher verwenden und nicht auf 1 (on) eingestellt `rds.force_ssl` ist, fragen Sie die `pg_stat_ssl` Ansicht ab, mit SSL der Verbindungen überprüft werden sollen. Die folgende Abfrage gibt beispielsweise nur SSL Verbindungen und Informationen über die verwendeten SSL Clients zurück.

```
SELECT datname, usename, ssl, client_addr
  FROM pg_stat_ssl INNER JOIN pg_stat_activity ON pg_stat_ssl.pid =
 pg_stat_activity.pid
  WHERE ssl is true and usename<>'rdsadmin';
```

Nur Zeilen, die TLS Verbindungen vom TypSSL/verwenden, werden mit Informationen über die Verbindung angezeigt. Dies ist eine Beispielausgabe.

```
 datname  | usename | ssl | client_addr
----------+---------+-----+-------------
 benchdb  | pgadmin | t   | 53.95.6.13
 postgres | pgadmin | t   | 53.95.6.13
(2 rows)
```

Diese Abfrage zeigt nur die aktuellen Verbindungen zum Zeitpunkt der Abfrage an. Das Fehlen von Ergebnissen bedeutet nicht, dass keine Anwendungen SSL Verbindungen verwenden. Andere SSL Verbindungen werden möglicherweise zu einem anderen Zeitpunkt hergestellt.

Ermitteln, ob ein Client zum Herstellen von Verbindungen Zertifikatverifizierungen erfordert

Wenn ein Client wie psql oder mit SSL Support konfiguriert istJDBC, versucht der Client SSL standardmäßig zunächst, eine Verbindung mit der Datenbank herzustellen. Wenn der Client keine Verbindung herstellen kannSSL, kehrt er zu einer Verbindung ohne Verbindung zurück. SSL Der `sslmode` Standardmodus wird für beide libpq-basierten Clients (wie psql) verwendet und JDBC ist auf gesetzt. `prefer` Das Zertifikat auf dem Server wird nur verifiziert, wenn `sslrootcert` es auf oder `sslmode` gesetzt ist. `verify-ca` `verify-full` Wenn das Zertifikat ungültig ist, wird ein Fehler ausgelöst.

Wird verwendet`PGSSLROOTCERT`, um das Zertifikat anhand der `PGSSLMODE` Umgebungsvariablen zu verifizieren, deren `PGSSLMODE` Einstellung auf `verify-ca` oder gesetzt ist`verify-full`.

```
PGSSLMODE=verify-full PGSSLROOTCERT=/fullpath/ssl-cert.pem psql -h
 pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com -U masteruser -d postgres
```

Verwenden Sie das `sslrootcert` Argument, um das Zertifikat `sslmode` im Format einer Verbindungszeichenfolge zu verifizieren, wobei die `sslmode` Einstellung auf `verify-ca` oder gesetzt ist, `verify-full` um das Zertifikat zu überprüfen.

```
psql "host=pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com sslmode=verify-full
 sslrootcert=/full/path/ssl-cert.pem user=masteruser dbname=postgres"
```

Wenn Sie beispielsweise in vorherigen Fall ein ungültiges Stammzertifikat verwenden, wird Ihnen im Client einen Fehler ähnlich dem folgenden angezeigt.

```
psql: SSL error: certificate verify failed
```

Aktualisieren des Trust Stores Ihrer Anwendung

Informationen zur Aktualisierung des Vertrauensspeichers für SQL Postgre-Anwendungen finden Sie unter Sichere TCP /IP-Verbindungen mit SSL in der SQL Postgre-Dokumentation.

Informationen zum Herunterladen des Stammverzeichnisses finden Sie unter .

Beispiele für Skripte, die Zertifikate importieren, finden Sie unter Beispielskript für den Import von Zertifikaten in Ihren Trust Store.

> **Note**
>
> Wenn Sie den Trust Store aktualisieren, können Sie ältere Zertifikate beibehalten und die neuen Zertifikate einfach hinzufügen.

Verwenden von SSL TLS /-Verbindungen für verschiedene Arten von Anwendungen

Im Folgenden finden Sie Informationen zur Verwendung vonSSL/TLS-Verbindungen für verschiedene Arten von Anwendungen:

- psql

  Der Client wird über die Befehlszeile durch die Angabe von Optionen als Verbindungszeichenfolge oder Umgebungsvariablen aufgerufen. Für SSL TLS /-Verbindungen sind die relevanten Optionen `sslmode` (UmgebungsvariablePGSSLMODE), `sslrootcert` (UmgebungsvariablePGSSLROOTCERT).

  Die vollständige Liste der Optionen finden Sie unter Parameter-Schlüsselwörter in der SQL Postgre-Dokumentation. Die vollständige Liste der Umgebungsvariablen finden Sie unter Umgebungsvariablen in der Postgre-Dokumentation. SQL
- pgAdmin

  Dieser browserbasierte Client ist eine benutzerfreundlichere Oberfläche für die Verbindung zu einer SQL Postgre-Datenbank.

  Informationen zur Konfiguration von Verbindungen finden Sie in der pgAdmin Dokumentation.
- JDBC

  JDBCermöglicht Datenbankverbindungen mit Java-Anwendungen.

  Allgemeine Informationen zum Herstellen einer Verbindung zu einer SQL Postgre-Datenbank mit JDBC finden Sie unter Verbindung zur Datenbank herstellen in der Dokumentation zum SQL JDBC Postgre-Treiber. Informationen zum Herstellen einer Verbindung mitSSL/TLSfinden Sie unter Konfiguration des Clients in der Dokumentation zum Postgre-Treiber. SQL JDBC
- Python

Eine beliebte Python-Bibliothek für die Verbindung zu SQL Postgre-Datenbanken istpsycopg2.

Informationen zum Verwenden von psycopg2 finden Sie in der psycopg2-Dokumentation. Ein kurzes Tutorial zum Herstellen einer Verbindung zu einer SQL Postgre-Datenbank finden Sie im Psycopg2-Tutorial. Informationen zu den vom Verbindungsbefehl akzeptierten Optionen finden Sie unter psycopg2-Modulinhalte.

Important

Nachdem Sie festgestellt haben, dass Ihre DatenbankverbindungenSSL/verwenden, TLS und Ihren Application Trust Store aktualisiert haben, können Sie Ihre Datenbank so aktualisieren, dass sie die rds-ca-rsa 2048-g1-Zertifikate verwendet. Anweisungen hierzu finden Sie in Schritt 3 unter Aktualisierung Ihres CA-Zertifikats durch Änderung Ihrer DB-Instance oder Ihres Clusters.

Verwenden der Kerberos-Authentifizierung mit Amazon RDS for PostgreSQL

Sie können Kerberos verwenden, um Benutzer zu authentifizieren, wenn sie sich mit Ihrer DB-Instance mit PostgreSQL verbinden. Konfigurieren Sie dazu Ihre so, dass sie AWS Directory Service for Microsoft Active Directory für die Kerberos-Authentifizierung verwendet wird. AWS Directory Service for Microsoft Active Directory wird auch genannt. AWS Managed Microsoft AD Es ist eine Funktion, die mit verfügbar ist AWS Directory Service. Weitere Informationen finden Sie unter Was ist AWS Directory Service? im AWS Directory Service Administratorhandbuch.

Erstellen Sie zunächst ein AWS Managed Microsoft AD Verzeichnis zum Speichern von Benutzeranmeldeinformationen. Anschließend stellen Sie Ihrer PostgreSQL-DB-Instance die Active Directory Domain und weitere Informationen zur Verfügung. Wenn Benutzer sich mit PostgreSQL-DB-Instances authentifizieren, werden Authentifizierungsanforderungen an das AWS Managed Microsoft AD -Verzeichnis weitergeleitet.

Wenn Sie alle Ihre Anmeldeinformationen im selben Verzeichnis aufbewahren, können Sie Zeit und Mühe sparen. Sie haben einen zentralen Ort für die Speicherung und Verwaltung von Anmeldeinformationen für mehrere DB-Instances. Die Verwendung eines Verzeichnisses kann auch Ihr allgemeines Sicherheitsprofil verbessern.

Außerdem können Sie von Ihrem eigenen On-Premises Microsoft Active Directory auf Anmeldeinformationen zugreifen. Dazu erstellen Sie eine vertrauensvolle Domain-Beziehung, damit das AWS Managed Microsoft AD -Verzeichnis Ihrem On-Premises Microsoft Active Directory vertraut. Auf diese Weise können Ihre Benutzer auf Ihre PostgreSQL--Instances mit derselben Windows Single Sign-On-Oberfläche (SSO) zugreifen, die sie auch für den Zugriff auf Workloads in Ihrem lokalen Netzwerk verwenden.

Eine Datenbank kann die Kennwortauthentifizierung oder die Kennwortauthentifizierung mit Kerberos- oder (IAM) -Authentifizierung verwenden. AWS Identity and Access Management Weitere Informationen zur IAM-Authentifizierung finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

Themen

- Verfügbarkeit von Regionen und Versionen
- Übersicht über die Kerberos-Authentifizierung für PostgreSQL-DB-Instances
- Einrichten der Kerberos-Authentifizierung für PostgreSQL-DB-Instances

- Verwaltung eines RDS für PostgreSQL-DB-Instance) in einer Active Directory-Domäne
- Herstellen einer Verbindung zu PostgreSQL mit Kerberos-Authentifizierung

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen über die Verfügbarkeit von Versionen und Regionen von RDS für PostgreSQL mit Kerberos-Authentifizierung finden Sie unter Unterstützte Regionen und DB-Engines für die Kerberos-Authentifizierung in Amazon RDS.

Übersicht über die Kerberos-Authentifizierung für PostgreSQL-DB-Instances

Um die Kerberos-Authentifizierung für PostgreSQL-DB-Instances einzurichten, führen Sie die folgenden Schritte aus, die später näher erläutert werden:

1. Wird verwendet AWS Managed Microsoft AD , um ein Verzeichnis zu erstellen. AWS Managed Microsoft AD Sie können die AWS Management Console, oder die AWS Directory Service API verwenden AWS CLI, um das Verzeichnis zu erstellen. Stellen Sie sicher, dass Sie die relevanten ausgehenden Ports in der Verzeichnissicherheitsgruppe öffnen, damit das Verzeichnis mit der kommunizieren kann.
2. Erstellen Sie eine Rolle, die RDS-Zugriff gewährt, um Anrufe in Ihr AWS Managed Microsoft AD Verzeichnis zu tätigen. Erstellen Sie dazu eine AWS Identity and Access Management (IAM-) Rolle, die die verwaltete IAM-Richtlinie verwendet. `AmazonRDSDirectoryServiceAccess`

   Damit die IAM-Rolle Zugriff gewährt, muss der Endpunkt AWS Security Token Service (AWS STS) in der richtigen AWS Region für Ihr Konto aktiviert sein. AWS AWS STS Endpunkte sind standardmäßig in allen aktiv AWS-Regionen, und Sie können sie ohne weitere Aktionen verwenden. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Aktivierung und Deaktivierung AWS STS in einer AWS Region.
3. Erstellen und konfigurieren Sie Benutzer im AWS Managed Microsoft AD Verzeichnis mithilfe der Microsoft Active Directory-Tools. Weitere Informationen zum Erstellen von Benutzern in Ihrem Active Directory finden Sie unter Verwalten von Benutzern und Gruppen in AWS Managed Microsoft AD im AWS Directory Service Administratorhandbuch.
4. Wenn Sie planen, das Verzeichnis und die DB-Instance in verschiedenen AWS Konten oder virtuellen privaten Clouds (VPCs) zu lokalisieren, konfigurieren Sie VPC-Peering. Weitere Informationen finden Sie unter Was ist VPC Peering? im Amazon VPC Peering Guide.

5. Erstellen oder ändern Sie PostgreSQL-DB-Instances entweder über die Konsole, CLI oder RDS-API mit einer der folgenden Methoden:
   - Erstellen einer Amazon RDS-DB-Instance
   - Ändern einer Amazon RDS DB-Instance
   - Wiederherstellung auf einer DB-Instance
   - Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS

   Sie können die in derselben Amazon Virtual Private Cloud (VPC) wie das Verzeichnis oder in einem anderen AWS Konto oder einer anderen VPC lokalisieren. Wenn Sie die PostgreSQL-DB- erstellen oder ändern, gehen Sie wie folgt vor:
   - Geben Sie den Domänenbezeichner (`d-*`-Bezeichner) an, der beim Erstellen Ihres Verzeichnisses generiert wurde.
   - Geben Sie außerdem den Namen der IAM-Rolle an, die Sie erstellt haben.
   - Stellen Sie sicher, dass die Sicherheitsgruppe der DB-Instance eingehenden Datenverkehr von der Sicherheitsgruppe des Verzeichnisses empfangen kann.
6. Verwenden Sie die RDS-Master-Benutzer-Anmeldeinformationen, um sich mit PostgreSQL-DB-Instances zu verbinden. Erstellen Sie den Benutzer in PostgreSQL, der extern identifiziert werden soll. Extern identifizierte Benutzer können sich über die Kerberos-Authentifizierung bei der PostgreSQL-DB- anmelden.

Einrichten der Kerberos-Authentifizierung für PostgreSQL-DB-Instances

Um die Kerberos-Authentifizierung einzurichten, führen Sie die folgenden Schritte aus.

Themen

- Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD
- Schritt 2: (Optional) Erstellen einer Vertrauensbeziehung zwischen Ihrem On-Premises Active Directory und AWS Directory Service
- Schritt 3: Erstellen Sie eine IAM-Rolle für RDS für den Zugriff auf AWS Directory Service
- Schritt 4: Anlegen und Konfigurieren von Benutzern
- Schritt 5: Aktivieren des VPC-übergreifenden Datenverkehrs zwischen dem Verzeichnis und der DB-Instance
- Schritt 6: Erstellen oder Ändern von PostgreSQL-DB-
- Schritt 7: Erstellen von PostgreSQL-Benutzern für Ihre Kerberos-Prinzipale

- Schritt 8: Konfigurieren eines PostgreSQL-Clients

Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD

AWS Directory Service erstellt ein vollständig verwaltetes Active Directory in der AWS Cloud. Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, AWS Directory Service erstellt zwei Domänencontroller und DNS-Server für Sie. Die Verzeichnisserver werden in verschiedenen Subnetzen in einer VPC erstellt. Diese Redundanz trägt dazu bei, dass Ihr Verzeichnis auch im Fehlerfall erreichbar bleibt.

Wenn Sie ein AWS Managed Microsoft AD Verzeichnis erstellen, führt der AWS Directory Service die folgenden Aufgaben in Ihrem Namen aus:

- Richtet ein Active Directory in Ihrer VPC ein.
- Erstellt ein Konto für den Verzeichnisadministrator mit dem Benutzernamen `Admin` und dem angegebenen Passwort. Mit diesem Konto verwalten Sie das Verzeichnis.

  > **Important**
  >
  > Stellen Sie sicher, dass Sie dieses Passwort speichern. AWS Directory Service speichert dieses Passwort nicht und es kann nicht abgerufen oder zurückgesetzt werden.

- Erstellt eine Sicherheitsgruppe für die Verzeichniscontroller. Die Sicherheitsgruppe muss die Kommunikation mit der PostgreSQL-DB- zulassen.

AWS Erstellt beim Start AWS Directory Service for Microsoft Active Directory eine Organisationseinheit (OU), die alle Objekte Ihres Verzeichnisses enthält. Diese OU erhält den NetBIOS-Namen, den Sie beim Erstellen des Verzeichnisses eingegeben haben, und befindet sich im Domänenstamm. Der Domänenstamm gehört und wird von diesem verwaltet AWS.

Das `Admin` Konto, das mit Ihrem AWS Managed Microsoft AD Verzeichnis erstellt wurde, verfügt über Berechtigungen für die gängigsten Verwaltungsaktivitäten Ihrer Organisationseinheit:

- Erstellen, Aktualisieren oder Löschen von Benutzern
- Hinzufügen von Ressourcen zu Ihrer Domäne, etwa Datei- oder Druckserver, und anschließendes Gewähren der zugehörigen Ressourcenberechtigungen für Benutzer in der OU
- Zusätzliche OUs Container erstellen

- Delegieren von Befugnissen
- Wiederherstellen von gelöschten Objekten aus dem Active Directory-Papierkorb
- Führen Sie Active Directory- und DNS-Module (Domain Name Service) für Windows PowerShell im Active Directory-Webdienst aus

Das `Admin`-Konto hat auch die Berechtigung, die folgenden domänenweiten Aktivitäten durchzuführen:

- Verwalten von DNS-Konfigurationen (Hinzufügen, Entfernen oder Aktualisieren von Datensätzen, Zonen und Weiterleitungen)
- Aufrufen von DNS-Ereignisprotokollen
- Anzeigen von Sicherheitsereignisprotokollen

Um ein Verzeichnis zu erstellen mit AWS Managed Microsoft AD

1. Wählen Sie im Navigationsbereich AWS Directory Service -Konsole den Eintrag Directories (Verzeichnisse) und wählen Sie Set up directory (Verzeichnis einrichten) aus.
2. Wählen Sie AWS Managed Microsoft AD. AWS Managed Microsoft AD ist die einzige Option, die derzeit für die Verwendung mit RDS unterstützt wird.
3. Wählen Sie Weiter aus.
4. Geben Sie auf der Seite Enter directory information (Verzeichnisinformationen eingeben) die folgenden Informationen ein:

   Edition

   Wählen Sie die Edition aus, die Ihre Anforderungen erfüllt.

   DNS-Name des Verzeichnisses

   Den vollständig qualifizierten Namen für das Verzeichnis, z. B. **`corp.example.com`**.

   NetBIOS-Name des Verzeichnisses

   Ein optionaler Kurzname für das Verzeichnis, z. B. `CORP`.

   Verzeichnisbeschreibung

   Eine optionale Beschreibung des Verzeichnisses.

Administratorpasswort

Das Passwort für den Verzeichnisadministrator. Mit der Verzeichniserstellung wird ein Administratorkonto mit dem Benutzernamen `Admin` und diesem Passwort angelegt.

Das Passwort für den Verzeichnisadministrator darf nicht das Wort "admin" enthalten. Beachten Sie beim Passwort die Groß- und Kleinschreibung und es muss 8 bis 64 Zeichen lang sein. Zudem muss es mindestens ein Zeichen aus dreien der vier folgenden Kategorien enthalten:

- Kleinbuchstaben (a–z)
- Großbuchstaben (A–Z)
- Zahlen (0–9)
- Nicht-alphanumerische Zeichen (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Passwort bestätigen

Geben Sie das Administratorpasswort erneut ein.

> **Important**
>
> Stellen Sie sicher, dass Sie dieses Passwort speichern. AWS Directory Service speichert dieses Passwort nicht und es kann nicht abgerufen oder zurückgesetzt werden.

5. Wählen Sie Weiter aus.
6. Geben Sie auf der Seite Choose VPC and subnets (VPC und Subnetze wählen) die folgenden Informationen an.

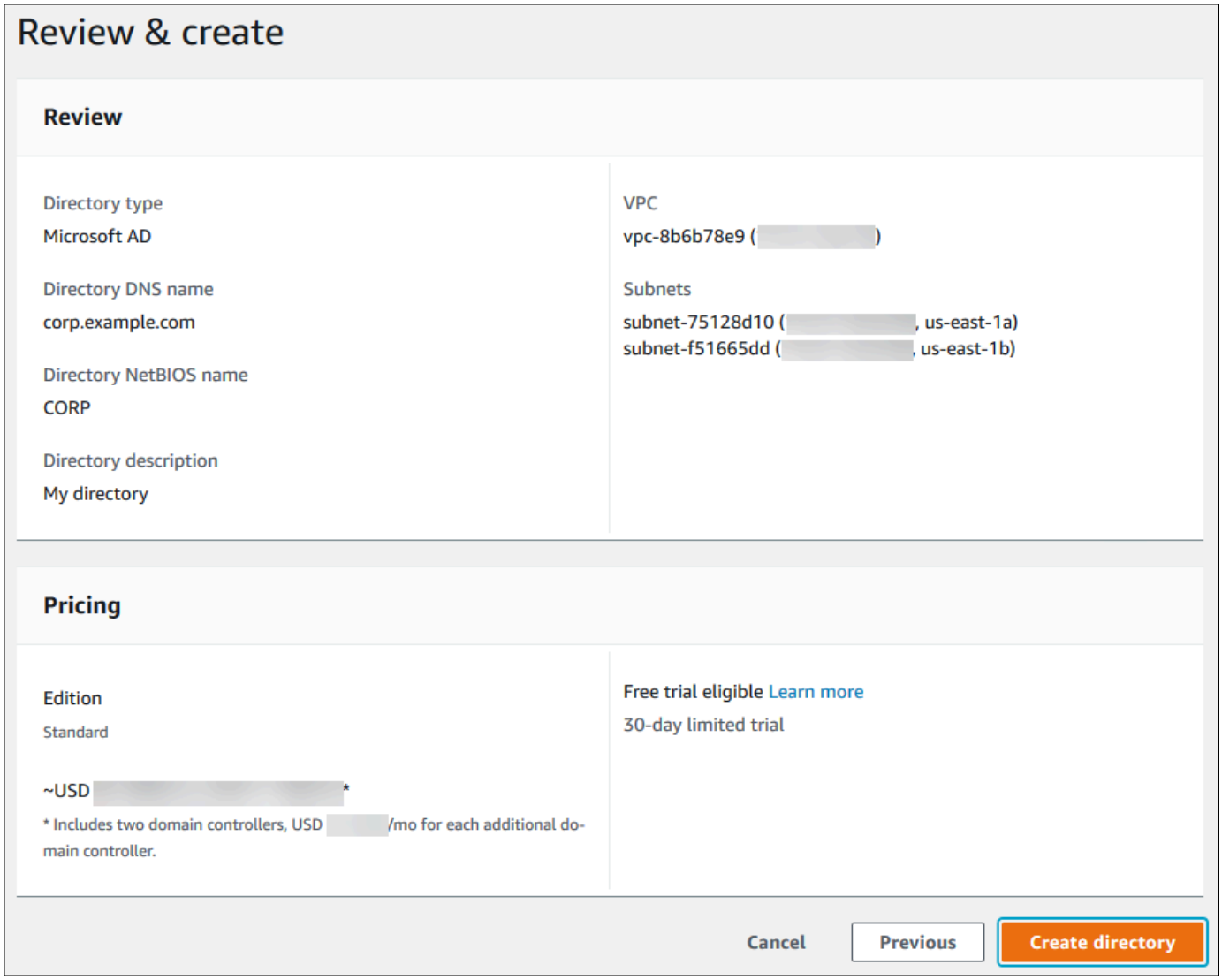
   VPC

   Wählen Sie die VPC für das Verzeichnis aus. Sie können PostgreSQL-DB-Instances in derselben VPC oder in einer anderen VPC erstellen.

   Subnetze

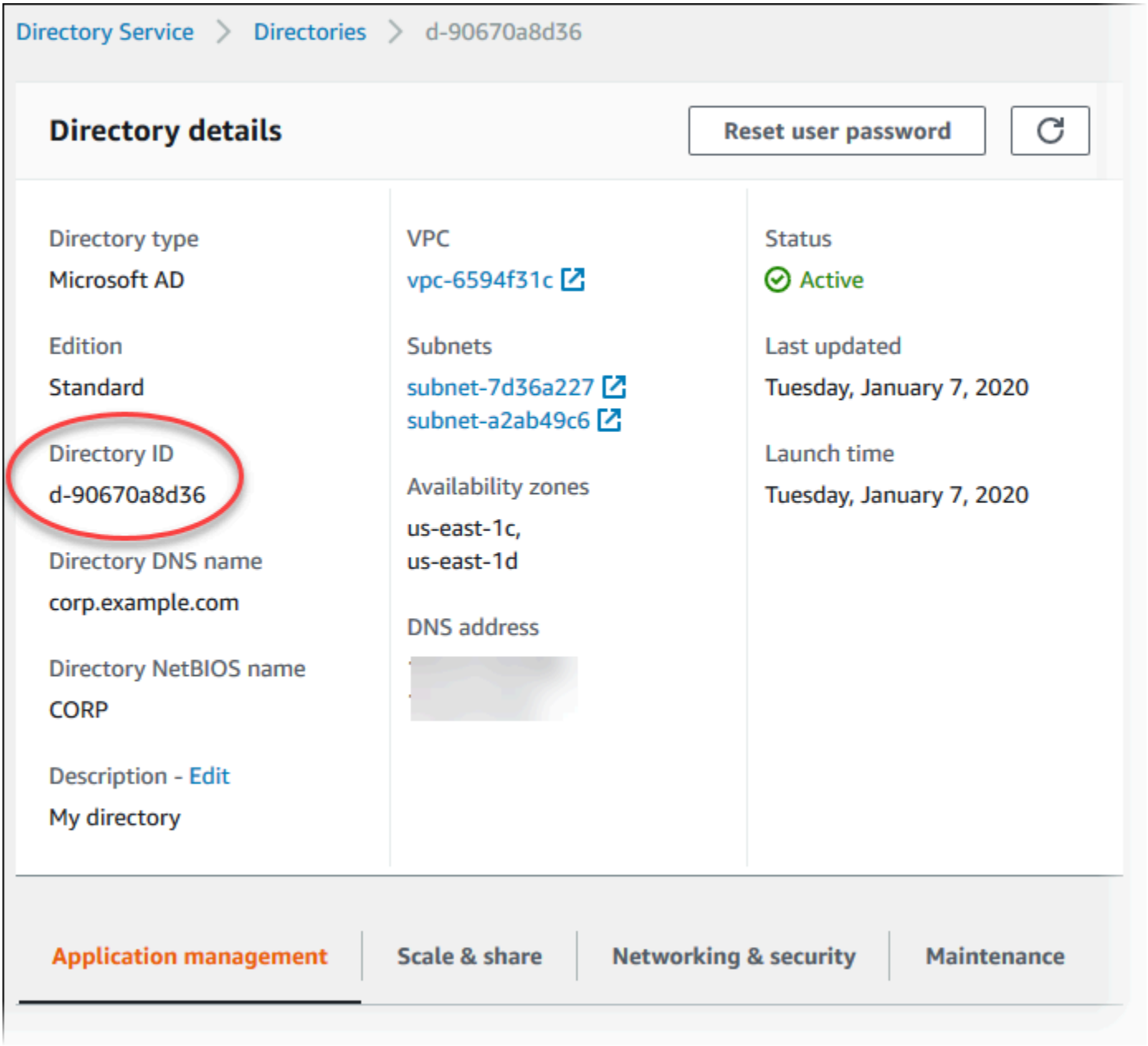
   Wählen Sie Subnetze für die Verzeichnis-Server aus. Die beiden Subnetze müssen zu verschiedenen Availability-Zonen gehören.
7. Wählen Sie Weiter aus.

8. Überprüfen Sie die Verzeichnisinformationen. Wenn Änderungen erforderlich sind, klicken Sie auf Previous (Zurück) und nehmen Sie die Änderungen vor. Wenn die Informationen richtig sind, wählen Sie Create directory (Verzeichnis erstellen).



Es dauert einige Minuten, bis das Verzeichnis erstellt wurde. Wenn es erfolgreich erstellt wurde, ändert sich der Wert Status in Active (Aktiv).

Um Informationen über das Verzeichnis anzuzeigen, wählen Sie die Verzeichnis-ID in der Verzeichnisauflistung aus. Notieren Sie sich den Wert Directory ID. Sie benötigen diesen Wert, wenn Sie Ihre PostgreSQL DB-Instance erstellen oder ändern.



Schritt 2: (Optional) Erstellen einer Vertrauensbeziehung zwischen Ihrem On-Premises Active Directory und AWS Directory Service

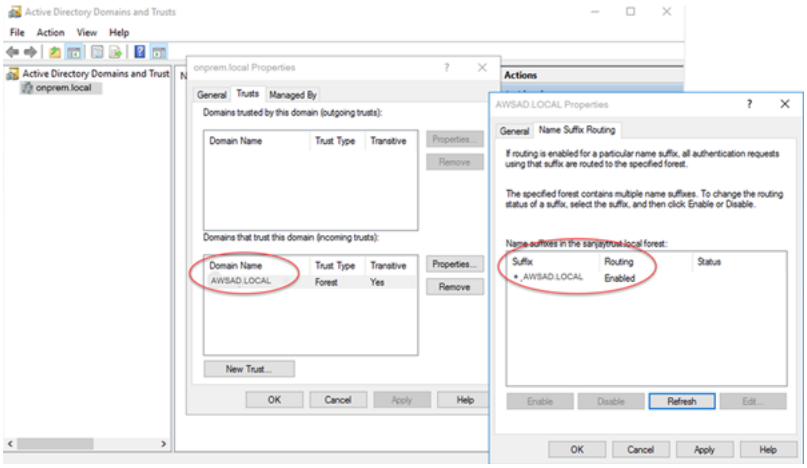
Wenn Sie Ihr eigenes lokales Microsoft Active Directory nicht verwenden möchten, fahren Sie mit for Schritt 3: Erstellen Sie eine IAM-Rolle für RDS für den Zugriff auf AWS Directory Service.

Um die Kerberos-Authentifizierung mit Ihrem lokalen Active Directory zu erhalten, müssen Sie eine vertrauensvolle Domänenbeziehung mithilfe einer Gesamtvertrauensstellung zwischen Ihrem lokalen Microsoft Active Directory und dem AWS Managed Microsoft AD Verzeichnis (erstellt in) einrichten. Schritt 1: Erstellen Sie ein Verzeichnis mit AWS Managed Microsoft AD Die Vertrauensstellung

kann unidirektional sein, wobei das AWS Managed Microsoft AD Verzeichnis dem lokalen Microsoft Active Directory vertraut. Die Vertrauensstellung kann auch bidirektional erfolgen, wobei beide Active Directories einander vertrauen. Weitere Informationen zum Einrichten von Vertrauensstellungen mithilfe von finden Sie unter Wann AWS Directory Service sollte eine Vertrauensstellung eingerichtet werden? im Administratorhandbuch.AWS Directory Service

> **Note**
>
> Wenn Sie ein lokales Microsoft Active Directory verwenden, stellen Windows-Clients eine Verbindung über den Domänennamen des AWS Directory Service im Endpunkt her und nicht über rds.amazonaws.com her. Weitere Informationen hierzu finden Sie unter Herstellen einer Verbindung zu PostgreSQL mit Kerberos-Authentifizierung.

Stellen Sie sicher, dass der lokale Microsoft Active Directory-Domänenname ein DNS-Suffix-Routing enthält, das der neu erstellten Vertrauensstellung entspricht. Im folgenden Screenshot wird ein Beispiel gezeigt.



Schritt 3: Erstellen Sie eine IAM-Rolle für RDS für den Zugriff auf AWS Directory Service

Damit Amazon RDS Sie anrufen AWS Directory Service kann, benötigt Ihr AWS Konto eine IAM-Rolle, die die verwaltete IAM-Richtlinie verwendet. `AmazonRDSDirectoryServiceAccess` Diese Rolle ermöglicht es Amazon RDS, Aufrufe von AWS Directory Service durchzuführen.

Wenn Sie mit dem eine DB-Instance erstellen AWS Management Console und Ihr Konsolen-Benutzerkonto über die `iam:CreateRole` entsprechende Berechtigung verfügt, erstellt die Konsole automatisch die benötigte IAM-Rolle. In diesem Fall lautet der Rollenname `rds-`

`directoryservice-kerberos-access-role`. Andernfalls müssen Sie die IAM-Rolle manuell erstellen. Wenn Sie diese IAM-Rolle erstellen`Directory Service`, wählen Sie die AWS verwaltete Richtlinie aus und fügen Sie `AmazonRDSDirectoryServiceAccess` sie ihr hinzu.

Weitere Informationen zum Erstellen von IAM-Rollen für einen Dienst finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

> **Note**
>
> Die für die Windows-Authentifizierung für RDS for Microsoft SQL Server verwendete IAM-Rolle kann nicht für Amazon RDS for PostgreSQL verwendet werden.

Alternativ können Sie Richtlinien mit den erforderlichen Berechtigungen erstellen, anstatt die verwaltete Richtlinie `AmazonRDSDirectoryServiceAccess` zu verwenden. In diesem Fall muss die IAM-Rolle die folgende IAM-Vertrauensrichtlinie haben.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Die Rolle muss auch über die folgende IAM-Rollenrichtlinie verfügen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Wenn Sie sich anmelden möchten AWS-Regionen, verwenden Sie regionsspezifische Dienstprinzipale in den Vertrauensrichtlinien für IAM-Rollen. Wenn Sie eine Vertrauensrichtlinie für Dienste in diesen Regionen erstellen, geben Sie den Regionalcode im Dienstprinzipal an.

Das folgende Beispiel zeigt eine Vertrauensrichtlinie, die regionsspezifische Dienstprinzipale umfasst:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.REGION-CODE.amazonaws.com",
          "rds.REGION-CODE.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

Ersetzen Sie REGION-CODE durch den Code für Ihre spezifische Region. Verwenden Sie beispielsweise die folgenden Service Principals für die Region Asien-Pazifik (Melbourne):

```
"Service": [
  "directoryservice.rds.ap-southeast-4.amazonaws.com",
  "rds.ap-southeast-4.amazonaws.com"
]
```

Schritt 4: Anlegen und Konfigurieren von Benutzern

Sie können Benutzer mit dem Tool „Active Directory Users and Computers“ erstellen. Dieses Tool ist ein Active Directory Domain Service und ein Active Directory Lightweight Directory Service. Weitere Informationen finden Sie unter Hinzufügen von Benutzern und Computern zur Active-Directory-Domain. In diesem Fall handelt es sich bei Benutzern um Einzelpersonen oder andere Entitäten, z. B. um ihre Computer, die Teil der Domain sind und deren Identitäten im Verzeichnis verwaltet werden.

Um Benutzer in einem AWS Directory Service Verzeichnis zu erstellen, müssen Sie mit einer Windows-basierten EC2 Amazon-Instance verbunden sein, die Mitglied des AWS Directory Service Verzeichnisses ist. Gleichzeitig müssen Sie als Benutzer angemeldet sein, der über Rechte zum Erstellen von Benutzern verfügt. Weitere Informationen finden Sie unter Erstellen eines Benutzers im AWS Directory Service Administration Guide.

Schritt 5: Aktivieren des VPC-übergreifenden Datenverkehrs zwischen dem Verzeichnis und der DB-Instance

Wenn Sie beabsichtigen, das Verzeichnis und die DB-Instance in derselben VPC zu platzieren, überspringen Sie diesen Schritt und fahren Sie mit Schritt 6: Erstellen oder Ändern von PostgreSQL-DB- fort.

Wenn Sie planen, das Verzeichnis und die DB-Instance unterschiedlich zu lokalisieren VPCs, konfigurieren Sie den VPC-übergreifenden Verkehr mithilfe von VPC-Peering oder Transit Gateway.AWS

Das folgende Verfahren ermöglicht den Datenverkehr zwischen VPCs mithilfe von VPC-Peering. Folgen Sie den Anweisungen unter Was ist VPC Peering? im Handbuch zu Amazon Virtual Private Cloud-Peering.

Aktivieren des VPC-übergreifenden Datenverkehrs mit VPC Peering

1. Richten Sie geeignete VPC-Routing-Regeln ein, um sicherzustellen, dass Netzwerk-Datenverkehr in beide Richtungen fließen kann.
2. Stellen Sie sicher, dass die Sicherheitsgruppe der DB-Instance eingehenden Datenverkehr von der Sicherheitsgruppe des Verzeichnisses empfangen kann.
3. Stellen Sie sicher, dass keine ACL-Regel (Network Access Control List) zum Blockieren des Datenverkehrs vorhanden ist.

Wenn ein anderes AWS Konto Eigentümer des Verzeichnisses ist, müssen Sie das Verzeichnis gemeinsam nutzen.

Um das Verzeichnis von mehreren AWS Konten gemeinsam zu nutzen

1. Beginnen Sie mit der gemeinsamen Nutzung des Verzeichnisses mit dem AWS Konto, unter dem die DB-Instance erstellt werden soll. Folgen Sie dazu den Anweisungen im AWS Directory Service Administratorhandbuch Tutorial: Teilen Ihres AWS verwalteten Microsoft AD-Verzeichnisses für einen nahtlosen EC2 Domänenbeitritt.
2. Melden Sie sich mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole an und stellen Sie sicher, dass die Domain den `SHARED` Status hat, bevor Sie fortfahren.
3. Notieren Sie sich den Wert der Verzeichnis-ID, während Sie mit dem Konto für die DB-Instance bei der AWS Directory Service Konsole angemeldet sind. Sie verwenden diese Verzeichnis-ID, um die DB-Instance mit der Domäne zu verbinden.

Schritt 6: Erstellen oder Ändern von PostgreSQL-DB-

Erstellen oder ändern Sie PostgreSQL-DB-Instances für die Verwendung mit Ihrem Verzeichnis. Sie können die Konsole, CLI oder RDS-API verwenden, um DB-Instances einem Verzeichnis zuzuordnen. Sie können dafür eine der folgenden Möglichkeiten auswählen:

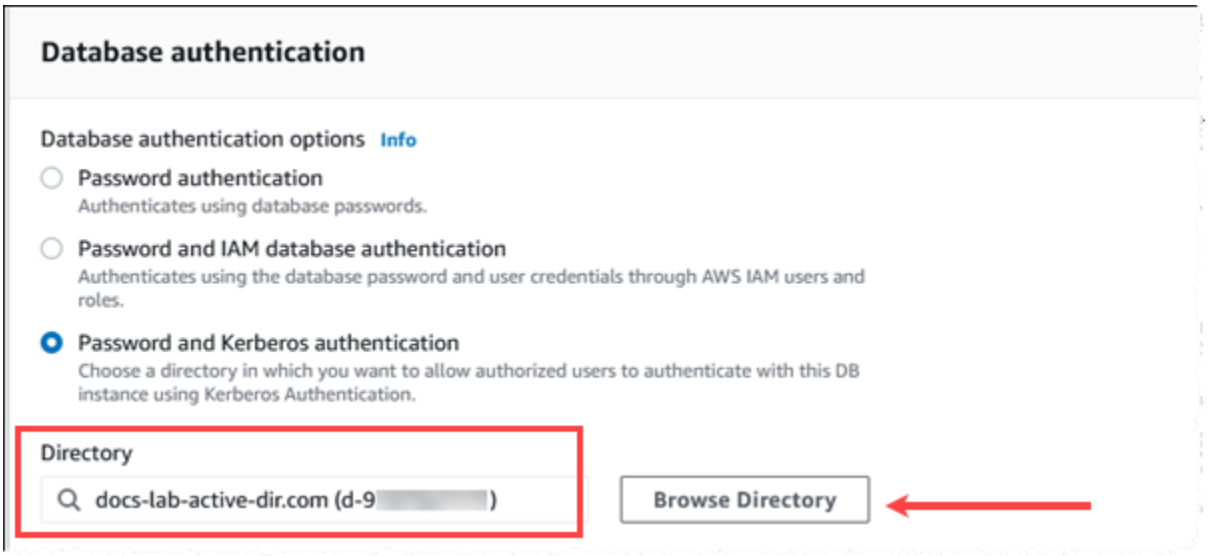
- Erstellen Sie eine neue PostgreSQL-DB-Instance mithilfe der Konsole, des create-db-instanceCLI-Befehls oder der Operation Create DBInstance RDS API. Detaillierte Anweisungen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

- Ändern Sie eine vorhandene PostgreSQL-DB-Instance mithilfe der Konsole, des modify-db-instanceCLI-Befehls oder der Operation Modify DBInstance RDS API. Detaillierte Anweisungen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Stellen Sie mithilfe der Konsole, des CLI-Befehls -db-snapshot oder des API-Vorgangs Restore From RDS eine restore-db-instance-fromPostgreSQL-DB-Instance aus einem DB-Snapshot wieder her DBInstance. DBSnapshot Detaillierte Anweisungen finden Sie unter Wiederherstellung auf einer DB-Instance.
- Stellen Sie eine PostgreSQL-DB-Instance point-in-time mithilfe der Konsole, des Befehls restore-db-instance-to- point-in-time CLI oder der Operation Restore DBInstance ToPointInTime RDS API in a wieder her. Detaillierte Anweisungen finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Die Kerberos-Authentifizierung wird nur für PostgreSQL-DB--Instances in einer VPC unterstützt. Der DB-Cluster kann sich in derselben VPC wie das Verzeichnis oder in einer anderen VPC befinden. Die DB-Instance muss eine Sicherheitsgruppe verwenden, die ausgehenden Datenverkehr innerhalb der VPC des Verzeichnisses zulässt, damit die DB-Instance mit dem Verzeichnis kommunizieren kann.

Konsole

Wenn Sie die Konsole verwenden, ändern oder wiederherstellen, um eine DB-Instance zu erstellen, wählen Sie im Abschnitt Datenbankauthentifizierung die Option Passwort- und Kerberos-Authentifizierung aus. Dann wählen Sie Verzeichnis durchsuchen. Wählen Sie das Verzeichnis aus oder wählen Sie Erstellen eines neuen Verzeichnisses, um den Directory Service zu verwenden.



AWS CLI

Wenn Sie den verwenden AWS CLI, sind die folgenden Parameter erforderlich, damit die das von Ihnen erstellte Verzeichnis verwenden kann:

- Für den `--domain`-Parameter verwenden Sie den Domänenbezeichner („d-*“-Bezeichner), der beim Erstellen des Verzeichnisses generiert wurde.
- Verwenden Sie für den `--domain-iam-role-name`-Parameter die von Ihnen erstellte Rolle, die die verwaltete IAM-Richtlinie `AmazonRDSDirectoryServiceAccess` verwendet.

Beispielsweise ändert der folgende CLI-Befehl eine DB-Instance zur Verwendung eines Verzeichnisses.

```
aws rds modify-db-instance --db-instance-identifier mydbinstance --domain d-Directory-ID --domain-iam-role-name role-name
```

> **Important**
>
> Wenn Sie eine DB-Instance ändern, um die Kerberos-Authentifizierung zu aktivieren, starten Sie die DB-Instance neu, nachdem Sie die Änderung vorgenommen haben.

Schritt 7: Erstellen von PostgreSQL-Benutzern für Ihre Kerberos-Prinzipale

Zu diesem Zeitpunkt ist Ihre DB-Instance von RDS für PostgreSQL mit der AWS Managed Microsoft AD -Domain verbunden. Die Benutzer, die Sie in dem Verzeichnis in Schritt 4: Anlegen und Konfigurieren von Benutzern erstellt haben, müssen als PostgreSQL-Datenbankbenutzer eingerichtet sein und über Berechtigungen verfügen, um sich bei der Datenbank anzumelden. Dazu melden Sie sich als Datenbankbenutzer mit `rds_superuser`-Rechten an. Wenn Sie beispielsweise beim Erstellen Ihrer DB-Instance von RDS für PostgreSQL die Standardeinstellungen akzeptiert haben, verwenden Sie `postgres`, wie in den folgenden Schritten gezeigt.

So erstellen Sie PostgreSQL-Datenbankbenutzer für Ihre Kerberos-Prinzipale

1. Verwenden Sie `psql`, um eine Verbindung mit dem DB-Instance-Endpunkt von RDS für PostgreSQL mit `psql` herzustellen. Im folgenden Beispiel wird das `postgres`-Standardkonto für die `rds_superuser`-Rolle verwendet.

```
psql --host=cluster-instance-1.111122223333.aws-region.rds.amazonaws.com --
port=5432 --username=postgres --password
```

2. Erstellen Sie einen Datenbankbenutzernamen für jeden Kerberos-Prinzipal (Active-Directory-Benutzername), der Zugriff auf die Datenbank haben soll. Verwenden Sie den kanonischen Benutzernamen (Identität), wie er in der Active-Directory-Instance definiert ist, d. h. einen `alias` in Kleinbuchstaben (Benutzername in Active Directory) und den Namen der Active-Directory-Domain für diesen Benutzernamen in Großbuchstaben. Der Active-Directory-Benutzername ist ein extern authentifizierter Benutzer. Setzen Sie den Namen daher in Anführungszeichen, wie im Folgenden gezeigt.

```
postgres=> CREATE USER "username@CORP.EXAMPLE.COM" WITH LOGIN;
CREATE ROLE
```

3. Weisen Sie dem Datenbankbenutzer die `rds_ad`-Rolle zu.

```
postgres=> GRANT rds_ad TO "username@CORP.EXAMPLE.COM";
GRANT ROLE
```

Nachdem Sie alle PostgreSQL-Benutzer für Ihre Active-Directory-Benutzeridentitäten erstellt haben, können Benutzer mit ihren Kerberos-Anmeldeinformationen auf die DB-Instance von RDS für PostgreSQL zugreifen.

Es ist erforderlich, dass die Datenbankbenutzer, die sich mit Kerberos authentifizieren, dies von Client-Computern aus tun, die Mitglieder der Active Directory-Domäne sind.

Datenbankbenutzer, denen die `rds_ad`-Rolle zugewiesen wurde, können nicht auch über die `rds_iam`-Rolle verfügen. Dies gilt auch für verschachtelte Mitgliedschaften. Weitere Informationen finden Sie unter IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

Schritt 8: Konfigurieren eines PostgreSQL-Clients

Gehen Sie folgendermaßen vor, um einen PostgreSQL-Client zu konfigurieren:

- Erstellen Sie eine krb5.conf-Datei (oder eine vergleichbare Datei), um auf die Domäne zu verweisen.

- Stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und fließen kann AWS Directory Service. Verwenden Sie ein Netzwerk-Dienstprogramm wie Netcat für die folgenden Aufgaben:
  - Überprüfen Sie den Datenverkehr über DNS für Port 53.
  - Überprüfen Sie den Datenverkehr TCP/UDP für Port 53 und für Kerberos, einschließlich der Ports 88 und 464 für. AWS Directory Service
- Stellen Sie sicher, dass der Datenverkehr zwischen dem Client-Host und der DB-Instance über den Datenbank-Port fließen kann. Verwenden Sie beispielsweise psql, um eine Verbindung herzustellen und auf die Datenbank zuzugreifen.

Im Folgenden finden Sie einen krb5.conf-Beispielinhalt für. AWS Managed Microsoft AD

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

Nachfolgend ein Beispiel für den Inhalt von krb5.conf für ein lokales Microsoft Active Directory.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
 ONPREM.COM = {
  kdc = onprem.com
  admin_server = onprem.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 .onprem.com = ONPREM.COM
```

```
onprem.com = ONPREM.COM
.rds.amazonaws.com = EXAMPLE.COM
.amazonaws.com.cn = EXAMPLE.COM
.amazon.com = EXAMPLE.COM
```

Verwaltung eines RDS für PostgreSQL-DB-Instance) in einer Active Directory-Domäne

Sie können die Konsole, die CLI oder die RDS-API verwenden, um Ihre und ihre Beziehung zu Ihrem Microsoft Active Directory zu verwalten. Sie können z. B. ein Active Directory zuordnen, um die Kerberos-Authentifizierung zu aktivieren. Sie können auch die Zuordnung für ein Active Directory entfernen, um die Kerberos-Authentifizierung zu deaktivieren. Sie können auch DB-Instances verschieben, die von einem Microsoft Active Directory zu einem anderen extern authentifiziert werden.

Sie können z. B. mithilfe der CLI Folgendes tun:

- Verwenden Sie den CLI-Befehl, um erneut zu versuchen, die Kerberos-Authentifizierung für eine fehlgeschlagene Mitgliedschaft zu aktivieren. modify-db-instance Geben Sie die Verzeichnis-ID der aktuellen Mitgliedschaft für die Option `--domain` an.
- Verwenden Sie den modify-db-instanceCLI-Befehl, um die Kerberos-Authentifizierung auf einer DB-Instance zu deaktivieren. Geben Sie `none` für die Option `--domain` an.
- Verwenden Sie den modify-db-instanceCLI-Befehl, um eine DB-Instance von einer Domain in eine andere zu verschieben. Geben Sie die Domänen-ID der neuen Domäne für die Option `--domain` an.

Grundlegendes zur Domänenmitgliedschaft

Nachdem Sie Ihre DB-Instance erstellt oder geändert haben, wird er zu einem Mitglied der Domäne. Sie können den Status der Domänenmitgliedschaft in der Konsole oder durch Ausführen des describe-db-instancesCLI-Befehls anzeigen. Der Status der DB-Instance kann einer der folgenden sein:

- `kerberos-enabled` – Für die DB-Instance ist die Kerberos-Authentifizierung aktiviert.
- `enabling-kerberos`— AWS ist dabei, die Kerberos-Authentifizierung für diese DB-Instance zu aktivieren.

- `pending-enable-kerberos` – Das Aktivieren der Kerberos-Authentifizierung ist für diese DB-Instance ausstehend.
- `pending-maintenance-enable-kerberos`— AWS wird versuchen, die Kerberos-Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu aktivieren.
- `pending-disable-kerberos` – Das Deaktivieren der Kerberos-Authentifizierung ist für diese DB-Instance ausstehend.
- `pending-maintenance-disable-kerberos`— AWS wird versuchen, die Kerberos-Authentifizierung auf der DB-Instance während des nächsten geplanten Wartungsfensters zu deaktivieren.
- `enable-kerberos-failed`— Ein Konfigurationsproblem AWS verhinderte die Aktivierung der Kerberos-Authentifizierung auf der DB-Instance. Beheben Sie das Konfigurationsproblem, bevor Sie den Befehl zum Ändern der DB-Instance erneut ausgeben.
- `disabling-kerberos`— AWS ist dabei, die Kerberos-Authentifizierung auf dieser DB-Instance zu deaktivieren.

Eine Anfrage zur Aktivierung der Kerberos-Authentifizierung kann wegen eines Netzwerkverbindungsproblems oder einer falschen IAM-Rolle fehlschlagen. In einigen Fällen kann der Versuch, die Kerberos-Authentifizierung zu aktivieren, fehlschlagen, wenn Sie eine DB-Instance erstellen oder ändern. Wenn dies passiert, stellen Sie sicher, dass Sie die richtige IAM-Rolle verwenden, und ändern Sie dann die DB-Instance, um der Domäne beizutreten.

> **Note**
>
> Nur die Kerberos-Authentifizierung mit RDS for PostgreSQL sendet Datenverkehr zu den DNS-Servern der Domäne. Alle anderen DNS-Anfragen werden als ausgehender Netzwerkzugriff auf Ihre DB-Instances mit PostgreSQL behandelt. Weitere Hinweise zum ausgehenden Netzwerkzugriff mit RDS for PostgreSQL finden Sie unter Verwenden eines benutzerdefinierten DNS Servers für ausgehenden Netzwerkzugriff.

Herstellen einer Verbindung zu PostgreSQL mit Kerberos-Authentifizierung

Sie können sich über die pgAdmin-Schnittstelle oder über eine Befehlszeilenschnittstelle wie psql per Kerberos-Authentifizierung mit PostgreSQL verbinden. Weitere Informationen zum Herstellen von Verbindungen finden Sie unter Herstellen einer Verbindung zu einer DB-Instance, in der die

PostgreSQL-Datenbank-Engine ausgeführt wird . Informationen zum Abrufen des Endpunkts, der Portnummer und anderer Details, die für die Verbindung benötigt werden, finden Sie unter Herstellen einer Verbindung zu einer PostgreSQL-DB-Instance.

> **Note**
>
> GSSAPI-Authentifizierung und Verschlüsselung in PostgreSQL werden von der Kerberos-Bibliothek implementiert. `libkrb5.so` Funktionen wie `postgres_fdw` diese nutzen dieselbe Bibliothek `dblink` auch für ausgehende Verbindungen mit Kerberos-Authentifizierung oder Verschlüsselung.

pgAdmin

Um pgAdmin für die Verbindung zu PostgreSQL mit Kerberos-Authentifizierung zu verwenden, führen Sie die folgenden Schritte aus:

1. Starten Sie die Anwendung pgAdmin auf Ihrem Client-Computer.
2. Klicken Sie auf der Registerkarte Dashboard auf Add New Server (Neuen Server hinzufügen).
3. Geben Sie im Dialogfeld (Erstellen – Server) auf der Registerkarte Allgemein einen Namen für den Server in pgAdmin ein.
4. Geben Sie auf der Registerkarte Connection (Verbindung) die folgenden Informationen aus der Datenbank von RDS für PostgreSQL ein:
   - Geben Sie für Host den Endpunkt für die DB-Instance von RDS für PostgreSQL Ein Endpunkt sieht in etwa wie folgt aus:

     ```
     RDS-DB-instance.111122223333.aws-region.rds.amazonaws.com
     ```

     Um von einem Windows-Client aus eine Verbindung zu einem lokalen Microsoft Active Directory herzustellen, verwenden Sie den Domänennamen des AWS Managed Active Directory anstelle des `rds.amazonaws.com` Host-Endpunkts. Nehmen wir zum Beispiel an, der Domänenname für das AWS verwaltete Active Directory lautet`corp.example.com`. Für Host würde der Endpunkt wie folgt angegeben werden:

     ```
     RDS-DB-instance.111122223333.aws-region.corp.example.com
     ```

   - Geben Sie unter Port den zugewiesenen Port ein.

- Geben Sie unter Wartungsdatenbank den Namen der initialen Datenbank ein, mit der sich der Client verbinden soll.
- Geben Sie unter Benutzername den Benutzernamen ein, den Sie für die Kerberos-Authentifizierung in Schritt 7: Erstellen von PostgreSQL-Benutzern für Ihre Kerberos-Prinzipale eingegeben haben.

5. Wählen Sie Speichern.

Psql

Um psql für die Verbindung mit PostgreSQL mit Kerberos-Authentifizierung zu verwenden, führen Sie die folgenden Schritte aus:

1. Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
kinit username
```

   Ersetzen Sie *username* durch den Benutzernamen. Geben Sie in der Eingabeaufforderung das im Microsoft Active Directory für den Benutzer gespeicherte Passwort ein.

2. Wenn die eine öffentlich zugängliche VPC verwendet, geben Sie die IP-Adresse für Ihren in Ihre `/etc/hosts` Datei auf dem Client ein. EC2 Die folgenden Befehle rufen beispielsweise die IP-Adresse ab und fügen sie dann in die `/etc/hosts`-Datei ein.

```
% dig +short PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com
;; Truncated, retrying in TCP mode.
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
34.210.197.118

% echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/
hosts
```

   Wenn Sie eine lokale Microsoft Active Directory von einem Windows-Client verwenden, müssen Sie eine Verbindung über einen speziellen Endpunkt herstellen. Anstatt die Amazon-Domain `rds.amazonaws.com` im Host-Endpunkt zu verwenden, verwenden Sie den Domainnamen des AWS Managed Active Directory.

Nehmen wir zum Beispiel an, der Domainname für Ihr AWS Managed Active Directory lautet`corp.example.com`. Verwenden Sie dann das Format *`PostgreSQL-endpoint.AWS-Region.corp.example.com`* für den Endpunkt und legen Sie es in der `/etc/hosts`-Datei ab.

```
% echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.corp.example.com" >> /etc/
hosts
```

3. Verwenden Sie den folgenden psql-Befehl, um sich bei einer PostgreSQL-DB-Instance anzumelden, der/die in Active Directory integriert ist.

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.rds.amazonaws.com postgres
```

Um sich beim PostgreSQL DB-Cluster von einem Windows-Client aus unter Verwendung eines lokalen Active Directory anzumelden, verwenden Sie den folgenden psql-Befehl mit dem Domänennamen aus dem vorhergehenden Schritt (`corp.example.com`):

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.corp.example.com postgres
```

Verwenden eines benutzerdefinierten DNS Servers für ausgehenden Netzwerkzugriff

RDSfor Postgre SQL unterstützt ausgehenden Netzwerkzugriff auf Ihre DB-Instances und ermöglicht die Auflösung von Domain Name Service (DNS) von einem benutzerdefinierten DNS Server aus, der dem Kunden gehört. Sie können nur vollständig qualifizierte Domainnamen von Ihrer RDS for SQL Postgre-DB-Instance über Ihren benutzerdefinierten Server auflösen. DNS

Themen

- Benutzerdefinierte DNS Auflösung aktivieren
- Deaktivieren der benutzerdefinierten Auflösung DNS
- Einen benutzerdefinierten DNS Server einrichten

Benutzerdefinierte DNS Auflösung aktivieren

Um die DNS Auflösung RDS für Ihren Kunden zu aktivierenVPC, ordnen Sie Ihrer SQL Postgre-Instance zunächst eine benutzerdefinierte DB-Parametergruppe zu. Aktivieren Sie dann den `rds.custom_dns_resolution`-Parameter, indem Sie ihn auf 1 setzen, und starten Sie dann die DB-Instance neu, damit die Änderungen durchgeführt werden können.

Deaktivieren der benutzerdefinierten Auflösung DNS

Um die DNS Auflösung bei Ihrem Kunden zu deaktivierenVPC, schalten Sie zunächst den `rds.custom_dns_resolution` Parameter Ihrer benutzerdefinierten DB-Parametergruppe aus, indem Sie ihn auf 0 setzen. Starten Sie dann die DB-Instance neu, damit die Änderungen durchgeführt werden können.

Einen benutzerdefinierten DNS Server einrichten

Nachdem Sie Ihren benutzerdefinierten DNS Nameserver eingerichtet haben, dauert es bis zu 30 Minuten, bis die Änderungen an Ihre DB-Instance weitergegeben werden. Nachdem die Änderungen an Ihre DB-Instance weitergegeben wurden, fragt der gesamte ausgehende Netzwerkverkehr, für den ein DNS Lookup erforderlich ist, Ihren DNS Server über Port 53 ab.

> **Note**
>
> Wenn Sie keinen benutzerdefinierten DNS Server einrichten und dieser Wert auf 1 gesetzt `rds.custom_dns_resolution` ist, werden Hosts mithilfe einer privaten Zone von Amazon Route 53 aufgelöst. Weitere Informationen finden Sie unter Arbeiten mit privat gehosteten Zonen.

Um einen benutzerdefinierten DNS Server RDS für Ihre SQL Postgre-DB-Instance einzurichten

1. Stellen Sie im Optionssatz des Dynamic Host Configuration Protocol (DHCP)VPC, der Ihrem beigefügt ist, die `domain-name-servers` Option auf die IP-Adresse Ihres DNS Nameservers ein. Weitere Informationen finden Sie unter DHCPOptionssätze.

   > **Note**
   >
   > Die `domain-name-servers` Option akzeptiert bis zu vier Werte, aber Ihre Amazon RDS DB-Instance verwendet nur den ersten Wert.

2. Stellen Sie sicher, dass Ihr DNS Server alle Suchanfragen lösen kann, einschließlich öffentlicher DNS Namen, EC2 privater DNS Amazon-Namen und kundenspezifischer DNS Namen. Wenn der ausgehende Netzwerkverkehr DNS Suchvorgänge enthält, die Ihr DNS Server nicht verarbeiten kann, müssen für Ihren DNS Server die entsprechenden DNS Upstream-Anbieter konfiguriert sein.
3. Konfigurieren Sie Ihren DNS Server so, dass er User Datagram Protocol (UDP) -Antworten von 512 Byte oder weniger erzeugt.
4. Konfigurieren Sie Ihren DNS Server so, dass er Transmission Control Protocol (TCP) -Antworten von 1.024 Byte oder weniger erzeugt.
5. Konfigurieren Sie Ihren DNS Server so, dass eingehender Datenverkehr von Ihren Amazon RDS DB-Instances über Port 53 zugelassen wird. Wenn sich Ihr DNS Server in einem Amazon befindetVPC, VPC muss er über eine Sicherheitsgruppe verfügen, die eingehende Regeln enthält, die den TCP Datenverkehr auf Port 53 zulassenUDP. Wenn sich Ihr DNS Server nicht in einem Amazon befindetVPC, muss er über entsprechende Firewall-Einstellungen verfügen, um UDP TCP eingehenden Datenverkehr auf Port 53 zuzulassen.

   Weitere Informationen finden Sie unter Sicherheitsgruppen für Sie VPC und Regeln hinzufügen und entfernen.

6. Konfigurieren Sie Ihre Amazon RDS DB-Instance so, dass ausgehender Datenverkehr über Port 53 zugelassen wird. VPC Ihre Sicherheitsgruppe VPC muss über eine Sicherheitsgruppe verfügen, die Regeln für ausgehenden TCP Datenverkehr auf Port 53 enthält. UDP

   Weitere Informationen finden Sie unter Sicherheitsgruppen für Sie VPC und Hinzufügen und Entfernen von Regeln im VPCAmazon-Benutzerhandbuch.
7. Stellen Sie sicher, dass der Routing-Pfad zwischen der RDS Amazon-DB-Instance und dem DNS Server korrekt konfiguriert ist, um DNS Datenverkehr zuzulassen.

   Wenn sich die RDS Amazon-DB-Instance und der DNS Server nicht in derselben befindenVPC, stellen Sie außerdem sicher, dass eine Peering-Verbindung zwischen ihnen eingerichtet ist. Weitere Informationen finden Sie unter Was ist VPC Peering? im Amazon VPC Peering Guide.

Upgrades der RDS for PostgreSQL-DB-Engine

Es gibt zwei Arten von Upgrades, die Sie für Ihre PostgreSQL-Datenbank verwalten können:

- Betriebssystem-Updates – Amazon RDS muss gelegentlich das Ihrer Datenbank zugrunde liegende Betriebssystem aktualisieren, um Sicherheitsmängel zu beheben oder Betriebssystemänderungen anzuwenden. Sie können mithilfe der RDS-Konsole AWS Command Line Interface (AWS CLI) oder der RDS-API entscheiden, wann Amazon RDS Betriebssystemupdates einführt. Weitere Informationen zu Betriebssystem-Aktualisierungen finden Sie unter Updates auf einen anwenden.
- Datenbank-Engine-Upgrades – Wenn Amazon RDS eine neue Version einer Datenbank-Engine unterstützt, können Sie Ihre Datenbanken auf die neue Version upgraden.

Eine Datenbank ist in diesem Zusammenhang eine DB-Instance von RDS für PostgreSQ oder ein Multi-AZ-DB-Cluster.

Es gibt zwei Arten von Engine-Upgrades für PostgreSQL-Datenbanken: Hauptversions-Upgrades und Nebenversions-Upgrades.

Hauptversions-Upgrades

Hauptversions-Upgrades können Datenbankänderungen enthalten, die nicht mit vorhandenen Anwendungen rückwärts kompatibel sind. Daher müssen Sie Hauptversions-Upgrades Ihrer Datenbanken manuell durchführen. Sie können ein Hauptversions-Upgrade starten, indem Sie Ihre DB-Instance oder Ihren Multi-AZ-DB-Cluster ändern. Bevor Sie ein Hauptversions-Upgrade durchführen, empfehlen wir Ihnen, die unter beschriebenen Schritte zu befolgen. Auswahl einer Hauptversion für ein RDS for PostgreSQL-Upgrade

Amazon RDS verarbeitet Multi-AZ-Hauptversions-Upgrades auf folgende Weise:

- Bereitstellung von Multi-AZ-DB-Instances — Amazon RDS aktualisiert gleichzeitig die primäre Instance und alle Standby-Instances. Ihre Datenbank ist möglicherweise einige Minuten lang nicht verfügbar, während das Upgrade abgeschlossen ist.
- Multi-AZ-DB-Cluster-Bereitstellung — Amazon RDS aktualisiert gleichzeitig die Reader- und Writer-Instances. Ihre Datenbank ist möglicherweise einige Minuten lang nicht verfügbar, während das Upgrade abgeschlossen ist.

Wenn Sie ein Upgrade für eine DB-Instance durchführen, die über regionsinterne Read Replicas verfügt, aktualisiert Amazon RDS die Replikate zusammen mit der primären DB-Instance.

Amazon RDS aktualisiert keine Lesereplikate von Multi-AZ-DB-Clustern. Wenn Sie ein Hauptversions-Upgrade eines Multi-AZ-DB-Clusters durchführen, ändert sich der Replikationsstatus der Read Replicas auf „Beendet“. Sie müssen die Lesereplikate nach Abschluss des Upgrades manuell löschen und neu erstellen.

> **Tip**
>
> Sie können die Ausfallzeit, die für ein Upgrade einer Hauptversion erforderlich ist, minimieren, indem Sie eine blaue/grüne Implementierung verwenden. Weitere Informationen finden Sie unter Verwendung von Blau/Grün-Bereitstellungen für Datenbankaktualisierungen.

Unterversion-Upgrades

Nebenversions-Upgrades enthalten dagegen nur Änderungen, die mit vorhandenen Anwendungen abwärtskompatibel sind. Sie können ein Nebenversions-Upgrade manuell starten, indem Sie Ihre Datenbank ändern. Sie können auch die Option Automatisches Upgrade der Nebenversion aktivieren, wenn Sie eine Datenbank erstellen oder ändern. Das bedeutet, dass Amazon RDS Ihre Datenbank automatisch aktualisiert, nachdem Sie die neue Version getestet und genehmigt haben.

Amazon RDS verarbeitet Multi-AZ-Nebenversions-Upgrades auf folgende Weise:

- Bereitstellung von Multi-AZ-DB-Instances — Amazon RDS aktualisiert gleichzeitig die primäre Instance und alle Standby-Instances. Ihre Datenbank ist möglicherweise einige Minuten lang nicht verfügbar, während das Upgrade abgeschlossen ist.
- Multi-AZ-DB-Cluster-Bereitstellung — Amazon RDS aktualisiert die Reader-DB-Instances nacheinander. Dann wird eine der Reader-DB-Instances zur neuen Writer-DB-Instance. Amazon RDS aktualisiert dann die alte Writer-Instance (die jetzt eine Reader-Instance ist). Multi-AZ-DB-Cluster reduzieren die Ausfallzeit bei kleineren Versions-Upgrades in der Regel auf etwa 35 Sekunden. Bei Verwendung mit RDS Proxy können sie die Ausfallzeiten weiter auf eine Sekunde oder weniger reduzieren. Weitere Informationen finden Sie unter Amazon-RDS-Proxy. Alternativ können Sie einen Open-Source-Datenbank-Proxy wie ProxySQL oder den AWS Advanced JDBC PgBouncerWrapper Driver verwenden.

Wenn Ihre Datenbank Read Replicas enthält, müssen Sie zuerst alle Read Replicas aktualisieren, bevor Sie die Quell-Instance oder den Quellcluster aktualisieren.

Weitere Informationen finden Sie unter Automatische Upgrades für Nebenversionen RDS für Postgre SQL. Informationen zur manuellen Durchführung eines Unterversionsupgrades finden Sie unter Manuelles Upgraden der Engine-Version.

Weitere Informationen über Datenbank-Engine-Versionen und die Richtlinie zur Ablehnung von Datenbank-Engine-Versionen finden Sie unter Datenbank-Engine-Versionen im Amazon RDS. FAQs

Themen

- Überlegungen zu PostgreSQL-Upgrades
- Suche nach gültigen Upgrade-Zielen
- SQLPostgre-Versionsnummern
- RDSVersionsnummern in RDS für Postgre SQL
- Auswahl einer Hauptversion für ein RDS for PostgreSQL-Upgrade
- So führen Sie ein Hauptversions-Upgrade für RDS for PostgreSQL durch
- Automatische Upgrades für Nebenversionen RDS für Postgre SQL
- Aktualisierung von SQL Postgre-Erweiterungen RDS für SQL Postgre-Datenbanken

Überlegungen zu PostgreSQL-Upgrades

Um Ihre Datenbanken sicher zu aktualisieren, verwendet Amazon RDS das in der PostgreSQL-Dokumentation beschriebene `pg_upgrade` Hilfsprogramm

Wenn Ihr Aufbewahrungszeitraum für Backups größer als 0 ist, erstellt Amazon RDS während des Upgrade-Prozesses zwei DB-Snapshots. Der erste DB-Snapshot gehört zur Datenbank, bevor Änderungen im Rahmen des Upgrades vorgenommen wurden. Wenn das Upgrade bei Ihren Datenbanken nicht funktioniert, können Sie diesen Snapshot wiederherstellen, um eine Datenbank zu erstellen, auf der die alte Version ausgeführt wird. Der zweite DB-Snapshot wird nach Abschluss des Upgrades übernommen. Diese DB-Snapshots werden automatisch gelöscht, sobald die Aufbewahrungsfrist für Backups abgelaufen ist.

> **Note**
>
> Amazon RDS nimmt während des Upgrade-Vorgangs nur dann DB-Snapshots auf, wenn Sie für den Aufbewahrungszeitraum des Backups Ihrer Datenbank einen Wert größer als 0 festgelegt haben. Informationen zum Ändern des Aufbewahrungszeitraums für Backups für eine DB-Instance finden Sie unter the section called “Ändern einer DB-Instance”. Sie können

keinen benutzerdefinierten Aufbewahrungszeitraum für Backups für Multi-AZ-DB-Cluster konfigurieren.

Wenn Sie ein Hauptversions-Upgrade einer DB-Instance durchführen, werden auch alle in der Region befindlichen Lesereplikate automatisch aktualisiert. Nach dem Start des Upgrade-Workflows warten die Lesereplikate auf den erfolgreichen Abschluss des `pg_upgrade` auf der primären DB-Instance. Dann wartet das Upgrade der primären DB-Instance auf den Abschluss der Upgrades der Lesereplikate. Bis das Upgrade abgeschlossen ist, treten Ausfälle auf. Bei Upgrades der Hauptversion eines Multi-AZ-DB-Clusters ändert sich der Replikationsstatus der Lesereplikate in Beendet.

Nachdem ein Upgrade abgeschlossen ist, können Sie nicht zur vorherigen Version der DB-Engine zurückkehren. Wenn Sie zur vorherigen Version zurückkehren möchten, stellen Sie den DB-Snapshot wieder her, der vor dem Upgrade erstellt wurde, um eine neue Datenbank zu erstellen.

Suche nach gültigen Upgrade-Zielen

Wenn Sie das verwenden, AWS Management Console um eine Datenbank zu aktualisieren, werden die gültigen Upgrade-Ziele für die Datenbank angezeigt. Sie können auch den folgenden AWS CLI Befehl verwenden, um die gültigen Upgrade-Ziele für eine Datenbank zu identifizieren:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Um beispielsweise die gültigen Upgrade-Ziele für eine PostgreSQL-Datenbank der Version 16.1 zu identifizieren, führen Sie den folgenden Befehl aus: AWS CLI

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version 16.1 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version 16.1 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

SQLPostgre-Versionsnummern

Die Reihenfolge der Versionsnummerierung für die SQL Postgre-Datenbank-Engine lautet wie folgt:

- Für SQL Postgre-Versionen 10 und höher hat die Engine-Versionsnummer das Format major.minor. Die Hauptversionsnummer ist der ganzzahlige Teil der Versionsnummer. Die Nebenversionsnummer ist der Nachkommabereich der Versionsnummer.

  Ein Upgrade der Hauptversion erhöht den ganzzahligen Teil der Versionsnummer, z. B. ein Upgrade von 10.Nebenversion auf 11.Nebenversion.
- Bei SQL Postgre-Versionen unter 10 hat die Engine-Versionsnummer das Format major.major.minor. Die Engine-Hauptversionsnummer ist sowohl die Ganzzahl als auch der erste Nachkommateil der Versionsnummer. 9.6 ist beispielsweise eine Hauptversion. Die Nebenversionsnummer ist der dritte Teil der Versionsnummer. Beispiel: Für Version 9.6.12 ist die 12 die Nebenversionsnummer.

  Ein Upgrade der Hauptversion erhöht den Hauptteil der Versionsnummer. Beispielsweise ist ein Upgrade von 9.6.12 auf 11.14 ein Upgrade der Hauptversion, wobei 9.6 und 11 die Hauptversionsnummern sind.

Informationen zur Versionsnummerierung von RDS Extended Support finden Sie unterVersionsbenennung für Amazon RDS Extended Support.

RDSVersionsnummern in RDS für Postgre SQL

RDSVersionsnummern verwenden das *major*.*minor*.*patch* Benennungsschema. Eine RDS Patch-Version enthält wichtige Bugfixes, die einer Nebenversion nach ihrer Veröffentlichung hinzugefügt wurden. Informationen zur Versionsnummerierung von RDS Extended Support finden Sie unterVersionsbenennung für Amazon RDS Extended Support.

Um die RDS Amazon-Versionsnummer Ihrer Datenbank zu ermitteln, müssen Sie zunächst die `rds_tools` Erweiterung mithilfe des folgenden Befehls erstellen:

```
CREATE EXTENSION rds_tools;
```

Ab der Veröffentlichung von Postgre SQL Version 15.2-R2 können Sie die RDS Versionsnummer Ihrer RDS for SQL Postgre-Datenbank mit der folgenden Abfrage ermitteln: SQL

```
postgres=> SELECT rds_tools.rds_version();
```

Wenn Sie beispielsweise eine RDS für Postgre SQL 15.2-Datenbank abfragen, wird Folgendes zurückgegeben:

```
rds_version
----------------
 15.2.R2
(1 row)
```

Auswahl einer Hauptversion für ein RDS for PostgreSQL-Upgrade

Hauptversions-Upgrades können Änderungen enthalten, die nicht mit früheren Versionen der Datenbank rückwärtskompatibel sind. Neue Funktionalität kann dazu führen, dass Ihre vorhandenen Anwendungen nicht mehr ordnungsgemäß funktionieren. Aus diesem Grund wendet Amazon-RDS-Hauptversions-Upgrades nicht automatisch an. Um ein Hauptversions-Upgrade durchzuführen, ändern Sie Ihre Datenbank manuell. Testen Sie alle Upgrades sorgfältig, um sicherzustellen, dass Ihre Anwendungen ordnungsgemäß funktionieren, bevor Sie das Upgrade auf Ihre Produktionsdatenbanken anwenden. Wenn Sie ein PostgreSQL-Hauptversions-Upgrade durchführen, werden jedoch die folgenden Schritte empfohlen, die unter beschrieben werde So führen Sie ein Hauptversions-Upgrade für RDS for PostgreSQL durch.

Wenn Sie eine Single-AZ-DB-Instance- oder eine Multi-AZ-DB-Instance-Bereitstellung von PostgreSQL auf die nächste Hauptversion aktualisieren, werden alle Lesereplikate, die mit der

Datenbank verknüpft sind, ebenfalls auf diese nächste Hauptversion aktualisiert. In einigen Fällen können Sie beim Upgrade auf eine höhere Hauptversion springen. Wenn das Upgrade eine Hauptversion überspringt, werden die Lesereplikate auch auf diese Ziel-Hauptversion aktualisiert. Upgrades auf Version 11, die andere Hauptversionen überspringen, haben gewisse Einschränkungen. Die Details finden Sie in den unter So führen Sie ein Hauptversions-Upgrade für RDS for PostgreSQL durch beschriebenen Schritten.

Die meisten PostgreSQL-Erweiterungen werden während eines PostgreSQL-Engine-Upgrades nicht aktualisiert. Diese müssen separat aktualisiert werden. Weitere Informationen finden Sie unter Aktualisierung von SQL Postgre-Erweiterungen RDS für SQL Postgre-Datenbanken.

Sie können herausfinden, welche Hauptversionen für Ihre RDS for PostgreSQL-Datenbank verfügbar sind, indem Sie die folgende AWS CLI Abfrage ausführen:

```
aws rds describe-db-engine-versions --engine postgres  --engine-version your-version
 --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

In der folgenden Tabelle finden Sie eine Zusammenfassung der verfügbaren Versionen. Ein Sternchen (*) auf der Versionsnummer bedeutet, dass die Version nicht mehr unterstützt wird. Wenn Ihre aktuelle Version nicht unterstützt wird, empfehlen wir Ihnen, auf das Upgrade-Ziel der neuesten Nebenversion oder auf eines der anderen verfügbaren Upgrade-Ziele für diese Version zu aktualisieren.

Aktuelle Quellversion	Upgrade-Ziele
17.6	Keine
17,5	17,6
17,4	17,6, 17,5
17,3*, 17,2	17,6 , 17,5, 17,4
17,1*	17,6 , 17,5, 17,4, 17,2
16,10	17,6
16,9	17,6, 17,5

Aktuelle Quellversion	Upgrade-Ziele
	16,10
16,8	17,6, 17,5, 17,4 16,10, 16,9
16,7*	17,6 , 17,5, 17,4 16,10, 16,9, 16,8
16,7	17,6, 17,5, 17,4 16,10, 16,9, 16,8
16,6	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8
16,5*, 16,4	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6
16,3	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6, 16,4
16,2*, 16,1*	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3
15,14	17,6 16,10
15,13	17,6, 17,5 16,10, 16,9 15,14

Aktuelle Quellversion	Upgrade-Ziele
15,12, 15,11*	17,6 , 17,5, 17,4 16,10, 16,9, 16,8 15,14, 15,13
15,10	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 15,14, 15,13, 15,12
15,9*	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 15,14, 15,13, 15,12, 15,10
15,8	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6, 16,4 15,14, 15,13, 15,12, 15,10
15,7	16,9 , 16,8, 16,7, 16,6, 16,5, 16,4, 16,3 15,13, 15,12, 15,11, 15,10, 15,9, 15,8
15,6*, 15,5*, 15,4*, 15,3*, 15,2*	17,6, 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3 15,14, 15,13, 15,12, 15,10, 15,8, 15,7
14,19	17,6 16,10 15,14

Aktuelle Quellversion	Upgrade-Ziele
14,18	17,6, 17,5 16,10, 16,9 15,14, 15,13 14,19
14,17, 14,16*	17,6 , 17,5, 17,4 16,10, 16,9, 16,8 15,14, 15,13, 15,12 14,19, 14,18
14,15	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 15,14, 15,13, 15,12, 15,10 14,19, 14,18, 14,17
14,14*	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 15,14, 15,13, 15,12, 15,10 14,19, 14,18, 14,17, 14,15
14,13	16,4 15,13 , 15,12, 15,11, 15,10, 15,9, 15,8 14,18, 14,17, 14,16, 14,15, 14,14

Aktuelle Quellversion	Upgrade-Ziele
14,12	16,3 15,13 , 15,12, 15,11, 15,10, 15,9, 15,8, 15,7 14,18, 14,17 , 14,16, 14,15, 14,14, 14,13
14,11*, 14,10*, 14,9*, 14,8*, 14,7*, 14,6*, 14,5*, 14,4*, 14,3*, 14,2*, 14,1*	17,6, 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3 15,14, 15,13, 15,12, 15,10, 15,8, 15,7 14,19, 14,18, 14,17, 14,15, 14,13, 14,12
13,22	17,6 16,10 15,14 14,19
13,21	17,6, 17,5 16,10, 16,9 15,14, 15,13 14,19, 14,18 13,22
13,20, 13,19*	17,6 , 17,5, 17,4 16,10, 16,9, 16,8 15,14, 15,13, 15,12 14,19, 14,18, 14,17 13,22, 13,21

Aktuelle Quellversion	Upgrade-Ziele
13,18, 13,17*	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 15,14, 15,13, 15,12, 15,10 14,19, 14,18, 14,17, 14,15 13,22, 13,21, 13,20
13,16	16,4 15,8 14,18 , 14,17, 14,15, 14,14, 14,13 13,21, 13,20 , 13,19, 13,18, 13,17
13,15	16,3 15,8, 15,7 14,18 , 14,17, 14,15, 14,14, 14,13, 14,12 13,21, 13,20 , 13,19, 13,18, 13,17, 13,16
13,14*, 13,13*, 13,1*, 13,1*, 13,10*, 13,9*, 13,8*, 13,7*, 13,6*, 13,5*, 13,4*, 13,3*, 13,2*, 13,1*	17,6, 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3 15,14, 15,13, 15,12, 15,10, 15,8, 15,7 14,19, 14,18, 14,17, 14,15, 14,13, 14,12 13,22, 13,21 , 13,20, 13,18, 13,16, 13,15

Aktuelle Quellversion	Upgrade-Ziele
12.22-rds.20250508	17,6, 17,5 16,10, 16,9 15,14, 15,13 14,19, 14,18 13,22, 13,21
12.22-rds.20250220	17,6, 17,5, 17,4 16,10, 16,9, 16,8 15,14, 15,13, 15,12 14,19, 14,18, 14,17 13,22, 13,21 , 13,20 12.22-rds.20250508
12,22, 12,21*	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 15,14, 15,13, 15,12, 15,10 14,19, 14,18, 14,17, 14,15 13,22, 13,21, 13,20 , 13,18 12.22-rds.20250220, 12.22-rds.20250508

Aktuelle Quellversion	Upgrade-Ziele
12,20*	17,6 , 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6, 16,4 15,14, 15,13, 15,12, 15,10, 15,8 14,19, 14,18, 14,17, 14,15, 14,13 13,22, 13,21 , 13,20, 13,18, 13,16 12.22, 12.22-rds.20250220, 12.22-rds .20250508
12,19*, 12,18*, 12,17*, 12,16*, 12,14*, 12,13*, 12,1*, 12,1*, 12,1*, 12,1*, 12,9*, 12,8*, 12,7*, 12,6*, 12,5*, 12,4*, 12,3*, 12,2*	17,6, 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3 15,14, 15,13, 15,12, 15,10, 15,8, 15,7 14,19, 14,18, 14,17, 14,15, 14,13, 14,12 13,22, 13,21 , 13,20, 13,18, 13,16, 13,15 12.22, 12.22-rds.20250220, 12.22-rds .20250508
11.22-rds.20250508	17,6, 17,5 16,10, 16,9 15,14, 15,13 14,19, 14,18 13,22, 13,21 12.22-rds.20250508

Aktuelle Quellversion	Upgrade-Ziele
11.22-rds.20250220	17,6, 17,5, 17,4 16,10, 16,9, 16,8 15,14, 15,13, 15,12 14,19, 14,18, 14,17 13,22, 13,21 , 13,20 12.22-rds.20250220, 12.22-rds.20250508 11.22-rds.20250508
11.22-rds.20240509	17,6, 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3 15,14, 15,13, 15,12, 15,10, 15,8, 15,7 14,19, 14,18, 14,17, 14,15, 14,13, 14,12 13,22, 13,21 , 13,20, 13,18, 13,16, 13,15 12.22, 12.22-rds.20250220, 12.22-rds .20250508 11.22-rds.20240808, 11.22-rds.20241121, 11.22-rds.20250220, 11.22-rds.20250508

Aktuelle Quellversion	Upgrade-Ziele
11,22, 11,21*, 11,20*, 11,19*, 11,17*, 11,16*, 11,15*, 11,14*, 11,13*, 11,1*, 11,1*, 11,1*, 11,1*, 11,9*, 11,8*, 11,7*, 11,6*, 11,5*, 11,4*, 11,2*, 11,1*	17,6, 17,5, 17,4, 17,2 16,10, 16,9, 16,8, 16,6 , 16,4, 16,3 15,14, 15,13, 15,12, 15,10, 15,8, 15,7 14,19, 14,18, 14,17, 14,15, 14,13, 14,12 13,22, 13,21 , 13,20, 13,18, 13,16, 13,15 12.22, 12.22-rds.20250220, 12.22-rds .20250508 11.22-rds.20240418 , 11.22-rds.20240509 , 11.22-rds.20240808, 11.22-rds.20241121, 11.22-rds.20250220, 11.22-rds.20250508

* Diese Version wird nicht mehr unterstützt.

So führen Sie ein Hauptversions-Upgrade für RDS for PostgreSQL durch

Wir empfehlen den folgenden Vorgang bei der Durchführung eines Hauptversions-Upgrades in einer Datenbank von Amazon RDS für PostgreSQL:

1. Bereithalten einer versionskompatiblen Parametergruppe – Wenn Sie eine benutzerdefinierte Parametergruppe verwenden, haben Sie zwei Optionen. Sie können eine Standardparametergruppe für die neue DB-Engine-Version angeben. Oder Sie können eine eigene benutzerdefinierte Parametergruppe für die neue DB-Engine-Version erstellen. Weitere Informationen erhalten Sie unter the section called “Parametergruppen” und the section called “DB-Cluster-Parametergruppen”.
2. Nach nicht unterstützten Datenbankklassen suchen – Überprüfen Sie, ob die Instance-Klasse Ihrer Datenbank mit der PostgreSQL-Version kompatibel ist, auf die Sie aktualisieren. Weitere Informationen finden Sie unter Unterstützte DB-Engines für DB-Instance-Klassen.
3. Auf nicht unterstützte Verwendung prüfen:
   - Vorbereitete Transaktionen: Übernehmen Sie oder machen Sie alle offenen vorbereiteten Transaktionen rückgängig, bevor Sie ein Upgrade durchführen.

Mit der folgenden Abfrage können Sie sicherstellen, dass für Ihre Datenbank keine geöffneten vorbereiteten Transaktionen vorhanden sind.

```
SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
```

- Reg*-Datentypen – Entfernen Sie alle Anwendungen der reg*-Datentypen, bevor Sie versuchen, einen Upgrade durchzuführen. Bis auf `regtype` und `regclass` ist kein Upgrade der reg*-Datentypen möglich. Das Dienstprogramm `pg_upgrade` kann diesen Datentyp nicht beibehalten, der von Amazon RDS für das Upgrade verwendet wird.

  Um zu überprüfen, dass keine Anwendungen der nicht unterstützten reg*-Datentypen vorhanden sind, geben Sie für jede Datenbank die folgende Abfrage aus.

```
SELECT count(*) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n,
 pg_catalog.pg_attribute a
  WHERE c.oid = a.attrelid
      AND NOT a.attisdropped
      AND a.atttypid IN ('pg_catalog.regproc'::pg_catalog.regtype,
                         'pg_catalog.regprocedure'::pg_catalog.regtype,
                         'pg_catalog.regoper'::pg_catalog.regtype,
                         'pg_catalog.regoperator'::pg_catalog.regtype,
                         'pg_catalog.regconfig'::pg_catalog.regtype,
                         'pg_catalog.regdictionary'::pg_catalog.regtype)
      AND c.relnamespace = n.oid
      AND n.nspname NOT IN ('pg_catalog', 'information_schema');
```

4. Suchen Sie nach ungültigen Datenbanken:

   - Stellen Sie sicher, dass keine ungültigen Datenbanken vorhanden sind. Die `datconnlimit` Spalte im `pg_database` Katalog enthält den Wert, `-2` um Datenbanken, die während eines `DROP DATABASE` Vorgangs unterbrochen wurden, als ungültig zu kennzeichnen.

     Verwenden Sie die folgende Abfrage, um nach ungültigen Datenbanken zu suchen:

```
SELECT datname FROM pg_database WHERE datconnlimit = - 2;
```

   - Die vorherige Abfrage gibt ungültige Datenbanknamen zurück. Sie können sie verwendenDROP `DATABASE` *`invalid_db_name`*`;`, um ungültige Datenbanken zu löschen. Sie können auch den folgenden Befehl verwenden, um ungültige Datenbanken zu löschen:

```
SELECT 'DROP DATABASE ' || quote_ident(datname) || ';' FROM pg_database WHERE
 datconnlimit = -2 \gexec
```

Weitere Informationen zu ungültigen Datenbanken finden Sie unter Grundlegendes zum Verhalten von Autovacuum bei ungültigen Datenbanken.

5. Umgang mit logischen Replikations-Slots – Ein Upgrade ist nicht möglich, wenn die Datenbank über logische Replikations-Slots verfügt. Logische Replikations-Slots werden normalerweise für die AWS DMS -Migration verwendet und zum Replizieren Datenbanktabellen in Data Lakes, BI-Tools und anderen Zielen. Stellen Sie vor dem Upgrade sicher, dass Sie den Zweck aller verwendeten logischen Replikations-Slots kennen, und bestätigen Sie, dass sie gelöscht werden können. Wenn die logischen Replikations-Slots noch verwendet werden, sollten Sie sie nicht löschen, und Sie können mit dem Upgrade nicht fortfahren.

   Wenn die logischen Replikations-Slots nicht benötigt werden, können Sie sie mit folgendem SQL-Befehl löschen:

   ```
   SELECT * FROM pg_replication_slots WHERE slot_type NOT LIKE 'physical';
   SELECT pg_drop_replication_slot(slot_name);
   ```

   Für die Einrichtung von logische Replikationsszenarien, die die `pglogical`-Erweiterung verwenden, müssen außerdem Slots gelöscht werden, damit ein Hauptversions-Upgrade erfolgreich durchgeführt werden kann. Informationen zum Identifizieren und Löschen von Slots, die mit der `pglogical`-Erweiterung erstellt wurden, finden Sie unter Verwaltung logischer Replikationssteckplätze für for Postgre SQL.

6. Behandlung von Lesereplikaten – Bei einem Upgrade Single-AZ-DB-Instance- oder Multi-AZ-DB-Instance-Bereitstellung werden neben der primären DB-Instance auch die Lesereplikate in der Region aktualisiert. Amazon RDS aktualisiert keine Lesereplikate von Multi-AZ-DB-Clustern.

   Lesereplikate können nicht separat aktualisiert werden. Wenn möglich, kann dies zu Situationen führen, in denen die primären und die Replikat-Datenbanken unterschiedliche PostgreSQL-Hauptversionen haben. Lesereplika-Upgrades können jedoch die Ausfallzeit der primären DB-Instance erhöhen. Um ein Lesereplikat-Upgrade zu verhindern, befördern Sie das Replikat zu einer eigenständigen Instance oder löschen Sie es, bevor Sie den Upgrade-Prozess starten.

   Der Upgrade-Prozess erstellt die Parametergruppe des Lesereplikats auf der Grundlage der aktuellen Parametergruppe des Lesereplikats neu. Sie können eine benutzerdefinierte

Parametergruppe erst dann auf ein Lesereplikat anwenden, wenn die Aktualisierung abgeschlossen ist, indem Sie das Lesereplikat modifizieren. Weitere Informationen über Lesereplikate finden Sie unter Arbeiten mit Read Replicas in Amazon RDS for PostgreSQL.

7. Handhabung von Zero-ETL-Integrationen — Wenn Sie bereits über eine Zero-ETL-Integration verfügen, löschen Sie diese, bevor Sie ein Upgrade auf eine Hauptversion durchführen. Erstellen Sie dann nach Abschluss des Upgrades die Integration neu.
8. Durchführen einer Sicherung – Wir empfehlen, vor dem Upgrade der Hauptversion eine Sicherung durchzuführen, damit Sie über einen bekannten Wiederherstellungspunkt für Ihre Datenbank verfügen. Wenn der Wert des Aufbewahrungszeitraums für Ihre Sicherung größer als 0 ist, erstellt der Upgrade-Vorgang vor und nach der Aktualisierung DB-Snapshots Ihrer Datenbank. Informationen über das Ändern Ihres Aufbewahrungszeitraums für Backups finden Sie unter Ändern einer Amazon RDS DB-Instance und the section called “Ändern eines Multi-AZ-DB-Clusters”.

   Informationen zum manuellen Durchführen der Sicherung finden Sie unter the section called “Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance” und the section called “Erstellen eines Multi-AZ-DB-Cluster-Snapshots”.
9. Aktualisieren bestimmter Erweiterungen vor dem Upgrade der Hauptversion – Wenn Sie planen, eine Hauptversion mit dem Upgrade zu überspringen, müssen Sie bestimmte Erweiterungen aktualisieren bevor Sie das Upgrade der Hauptversion durchführen. Zum Beispiel überspringt ein Upgrade von den Versionen 9.5.x oder 9.6.x auf die Versionen 11.x eine Hauptversion. Zu den zu aktualisierenden Erweiterungen gehören PostGIS und zugehörige Erweiterungen für die Verarbeitung räumlicher Daten.

   - `address_standardizer`
   - `address_standardizer_data_us`
   - `postgis_raster`
   - `postgis_tiger_geocoder`
   - `postgis_topology`

   Sie können aufgrund von Inkompatibilität nicht direkt auf PostgreSQL Version 17 aktualisieren, wenn Sie `rdkit` Version 4.6.0 und niedriger sowie PostgreSQL Version 16 und niedriger verwenden. `rdkit` Im Folgenden finden Sie die Upgrade-Optionen:

   - Wenn Sie PostgreSQL Version 13 und niedriger verwenden, müssen Sie zuerst ein Hauptversions-Upgrade auf Version 14.14 und höhere 14-Versionen, 15.9 und höhere 15-

Versionen oder 16.5 und höhere 16-Versionen durchführen und dann das Versionsupgrade auf PostgreSQL 17 durchführen.

- Wenn Sie PostgreSQL Version 14, 15 oder 16 verwenden, müssen Sie ein Nebenversions-Upgrade auf 14.14 und höhere 14-Versionen, 15.9 und höhere 15-Versionen oder 16.5 und höhere 16-Versionen durchführen und dann auf PostgreSQL Version 17 aktualisieren.

Führen Sie den folgenden Befehl für jede von Ihnen verwendete Erweiterung aus:

```
ALTER EXTENSION PostgreSQL-extension UPDATE TO 'new-version';
```

Weitere Informationen finden Sie unter Aktualisierung von SQL Postgre-Erweiterungen RDS für SQL Postgre-Datenbanken. Weitere Informationen über PostGIS-Upgrades finden Sie unter Schritt 6: Aktualisieren Sie die GIS Post-Erweiterung.

10 Entfernen bestimmter Erweiterungen vor dem Upgrade der Hauptversion – Ein Upgrade, bei dem eine Hauptversion auf Version 11.x übersprungen wird, unterstützt keine Aktualisierung der Erweiterung `pgRouting`. Ein Upgrade von den Versionen 9.4.x, 9.5.x oder 9.6.x auf 11.x-Versionen überspringt eine Hauptversion. Es ist sicher, die Erweiterung `pgRouting` zu verwerfen zu lassen und sie nach dem Upgrade wieder auf eine kompatible Version zu installieren. Für die Erweiterungsversionen, auf die Sie aktualisieren können, lesen Sie Unterstützte SQL Postgre-Erweiterungsversionen.

Die Erweiterungen `tsearch2` und `chkpass` werden für PostgreSQL ab Version 11 nicht mehr unterstützt. Wenn Sie auf Version 11.x aktualisieren, lassen Sie die Erweiterungen `tsearch2` und `chkpass` vor dem Upgrade weg.

11 Löschen unbekannter Datentypen – Löschen Sie `unknown`-Datentypen in Abhängigkeit von der Zielversion.

PostgreSQL-Version 10 unterstützt den `unknown`-Datentyp nicht mehr. Wenn eine Datenbank der Version 9.6 den Datentyp `unknown` verwendet, wird bei einem Upgrade auf eine Version 10 eine Fehlermeldung wie die folgende angezeigt:

```
Database instance is in a state that cannot be upgraded: PreUpgrade checks failed:
The instance could not be upgraded because the 'unknown' data type is used in user
 tables.
Please remove all usages of the 'unknown' data type and try again."
```

Verwenden Sie die folgende SQL, um den `unknown`-Datentyp in Ihrer Datenbank zu finden, damit Sie die betroffene Spalte entfernen oder in einen unterstützten Datentyp ändern können:

```
SELECT DISTINCT data_type FROM information_schema.columns WHERE data_type ILIKE
 'unknown';
```

12 Durchführen eines Upgrade-Trockenlaufs – Wir empfehlen dringend, ein Hauptversionsupgrade auf einem Duplikat Ihrer Produktionsdatenbank zu testen, bevor Sie das Upgrade auf Ihrer Produktionsdatenbank durchführen. Sie können die Ausführungspläne auf der duplizierten Testdatenbank auf mögliche Regressionen des Ausführungsplans überwachen und deren Leistung bewerten. Um eine doppelte Testinstanz zu erstellen, können Sie Ihre Datenbank entweder aus einem aktuellen Snapshot wiederherstellen oder Ihre Datenbank auf den letzten point-in-time wiederherstellbaren Zeitpunkt zurücksetzen.

Weitere Informationen finden Sie unter the section called “Wiederherstellung aus einem Snapshot” oder the section called “Point-in-time Wiederherstellung”. Informationen zu Multi-AZ-DB-Clustern finden Sie unter the section called “Wiederherstellen von einem Snapshot in einem Multi-AZ-DB-Cluster” oder the section called “Wiederherstellen eines Multi-AZ-DB-Clusters zu einer bestimmten Zeit”.

Weitere Einzelheiten zum Durchführen des Upgrades finden Sie unter the section called “Manuelles Upgraden der Engine-Version”.

Beachten Sie beim Upgrade einer Datenbank der Version 9.6 auf Version 10, dass PostgreSQL 10 standardmäßig parallele Abfragen aktiviert. Sie können die Auswirkungen von Parallelverarbeitung vor dem Upgrade testen, indem Sie den Parameter `max_parallel_workers_per_gather` in Ihrer Testdatenbank auf 2 ändern.

> **Note**
>
> Der Standardwert für `max_parallel_workers_per_gather`-Parameter in der `default.postgresql10`-DB-Parametergruppe lautet 2.

Weitere Informationen finden Sie unter Parallel Query in der PostgreSQL-Dokumentation. Um die Parallelverarbeitung in Version 10 zu deaktivieren, setzen Sie den Parameter `max_parallel_workers_per_gather` auf 0.

Während des Upgrades der Hauptversion werden die Datenbanken `public` und `template1` und das Schema `public` in jeder Datenbank vorübergehend umbenannt. Diese Objekte erscheinen in den Protokollen mit ihrem ursprünglichen Namen und einer zufälligen Zeichenfolge, die angehängt wird. Die Zeichenfolge wird so hinzugefügt, dass benutzerdefinierte Einstellungen wie `locale` und `owner` während des Upgrades der Hauptversion erhalten bleiben. Sobald das Upgrade abgeschlossen ist, werden die Objekte in ihre ursprünglichen Namen umbenannt.

> **Note**
>
> Während des Upgrade-Vorgangs für die Hauptversion können Sie keine point-in-time Wiederherstellung Ihrer DB-Instance oder Ihres Multi-AZ-DB-Clusters durchführen. Nachdem das Upgrade von Amazon RDS durchgeführt wurde, wird eine automatische Sicherung der Datenbank vorgenommen. Sie können eine point-in-time Wiederherstellung zu den Zeiten vor Beginn des Upgrades und nach Abschluss der automatischen Sicherung Ihrer Datenbank durchführen.

13. Wenn ein Upgrade mit Fehlern bei der Vorprüfung fehlschlägt, beheben Sie die Probleme – Während des Upgrades der Hauptversion führt Amazon RDS for PostgreSQL zunächst eine Vorprüfung durch, um alle Probleme zu identifizieren, die dazu führen könnten, dass das Upgrade fehlschlägt. Das Vorprüfverfahren überprüft alle potenziell nicht kompatiblen Bedingungen in allen Datenbanken der Instance.

    Wenn die Vorprüfung auf ein Problem stößt, wird ein Protokollereignis erstellt, das anzeigt, dass die Vorprüfung für das Upgrade fehlgeschlagen ist. Die Details des Vorprüfprozesses befinden sich in einem Upgrade-Protokoll mit dem Namen `pg_upgrade_precheck.log` für alle Datenbanken einer Datenbank. Amazon RDS hängt einen Zeitstempel an den Dateinamen an. Weitere Informationen zum Anzeigen von Protokollen finden Sie unter Überwachung von Amazon RDS Amazon.

    Wenn ein Lesereplikat-Upgrade bei der Vorprüfung fehlschlägt, wird die Replikation auf dem fehlgeschlagenen Lesereplikat unterbrochen und das Lesereplikat in den Status "beendet" versetzt. Löschen Sie das Lesereplikat und erstellen Sie ein neues Lesereplikat auf der Grundlage der aktualisierten primären DB-Instance.

    Beheben Sie alle im Vorprüfprotokoll identifizierten Probleme und versuchen Sie dann erneut das Hauptversionsupgrade. Im Folgenden finden Sie ein Beispiel für ein Vorprüfprotokoll.

```
---------------------------------------------------------------------------
Upgrade could not be run on Wed Apr 4 18:30:52 2018
---------------------------------------------------------------------------
The instance could not be upgraded from 9.6.11 to 10.6 for the following reasons.
Please take appropriate action on databases that have usage incompatible with the
 requested major engine version upgrade and try the upgrade again.

* There are uncommitted prepared transactions. Please commit or rollback all prepared
 transactions.* One or more role names start with 'pg_'. Rename all role names that
 start with 'pg_'.

* The following issues in the database 'my"million$"db' need to be corrected before
 upgrading:** The ["line","reg*"] data types are used in user tables. Remove all
 usage of these data types.
** The database name contains characters that are not supported by RDS for
 PostgreSQL. Rename the database.
** The database has extensions installed that are not supported on the target
 database version. Drop the following extensions from your database: ["tsearch2"].

* The following issues in the database 'mydb' need to be corrected before
 upgrading:** The database has views or materialized views that depend on
 'pg_stat_activity'. Drop the views.
```

14. Wenn ein Lesereplikat-Upgrade während des Upgrades der Datenbank fehlschlägt, beheben Sie das Problem – Ein fehlgeschlagenes Lesereplikat wird in den Zustand `incompatible-restore` versetzt und die Replikation auf der Datenbank wird beendet. Löschen Sie das Lesereplikat und erstellen Sie ein neues Lesereplikat auf der Grundlage der aktualisierten primären DB-Instance.

> **Note**
>
> Amazon RDS aktualisiert keine Lesereplikate für Multi-AZ-DB-Cluster. Wenn Sie ein Upgrade einer Hauptversion auf einem Multi-AZ-DB-Cluster durchführen, ändert sich der Replikationsstatus der zugehörigen Read Replicas in „Beendet“.

Ein Lesereplikat-Upgrade kann aus den folgenden Gründen fehlschlagen:

- Sie konnte die primäre DB-Instance auch nach einer Wartezeit nicht einholen.
- Es befand sich in einem beendeten oder inkompatiblen Lebenszyklusstatus, z. B. „Speicher voll“, „Inkompatible Wiederherstellung“ usw.

- Als das Upgrade der primären DB-Instance begann, lief auf der Lesereplikat bereits ein separates Upgrade der Unterversion.
- Das Lesereplikat verwendete inkompatible Parameter.
- Das Lesereplikat war nicht in der Lage, mit der primären DB-Instance zu kommunizieren, um den Datenordner zu synchronisieren.

15. Upgrade Ihrer Produktionsdatenbank – Wenn das Upgrade der Hauptversion erfolgreich durchgeführt wurde, sollten Sie in der Lage sein, Ihre Produktionsdatenbank sicher zu aktualisieren. Weitere Informationen finden Sie unter Manuelles Upgraden der Engine-Version.
16. Führen Sie die `ANALYZE`-Operation aus, um die Tabelle `pg_statistic` zu aktualisieren. Führen Sie diesen Schritt für jede Datenbank auf all Ihren PostgreSQL-Datenbanken durch. Optimizer-Statistiken werden während eines Hauptversionsupgrades nicht übertragen, daher müssen Sie alle Statistiken neu generieren, um Leistungsprobleme zu vermeiden. Führen Sie den Befehl ohne Parameter aus, um Statistiken für alle regulären Tabellen in der aktuellen Datenbank wie folgt zu generieren:

```
ANALYZE VERBOSE;
```

Das Flag `VERBOSE` ist optional, aber wenn Sie es verwenden, wird Ihnen der Fortschritt angezeigt. Weitere Informationen finden Sie unter ANALYZE in der PostgreSQL-Dokumentation.

Wenn Sie bestimmte Tabellen analysieren, anstatt ANALYZE VERBOSE zu verwenden, führen Sie den Befehl ANALYZE für jede Tabelle wie folgt aus:

```
ANALYZE table_name;
```

Analysieren Sie bei partitionierten Tabellen immer die übergeordnete Tabelle. Dieser Prozess:

- Automatisches Sampling von Zeilen auf allen Partitionen
- Aktualisiert die Statistiken für jede Partition rekursiv
- Verwaltet wichtige Planungsstatistiken auf übergeordneter Ebene

In übergeordneten Tabellen werden zwar keine tatsächlichen Daten gespeichert, ihre Analyse ist jedoch für die Abfrageoptimierung von entscheidender Bedeutung. Wenn ANALYZE nur auf einzelnen Partitionen ausgeführt wird, kann dies zu einer schlechten Abfrageleistung führen, da der Optimierer nicht über die umfassenden Statistiken verfügt, die für eine effiziente partitionsübergreifende Planung erforderlich sind.

> **Note**
>
> Führen Sie ANALYZE nach dem Upgrade auf Ihrem System aus, um Leistungsprobleme zu vermeiden.

Nachdem das Upgrade der Hauptversion abgeschlossen ist, empfehlen wir Folgendes:

- Bei einem PostgreSQL-Upgrade werden keine PostgreSQL-Erweiterungen aktualisiert. Informationen zum Upgrade von Erweiterungen finden Sie unter Aktualisierung von SQL Postgre-Erweiterungen RDS für SQL Postgre-Datenbanken.
- Verwenden Sie optional Amazon RDS, um zwei Protokolle anzuzeigen, die das Dienstprogramm `pg_upgrade` erstellt. Diese sind `pg_upgrade_internal.log` und `pg_upgrade_server.log`. Amazon RDS hängt einen Zeitstempel an den Dateinamen für diese Protokolle an. Sie können diese Protokolle wie jedes andere Protokoll einsehen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS Amazon.

  Sie können die Upgrade-Protokolle auch auf Amazon CloudWatch Logs hochladen. Weitere Informationen finden Sie unter PostgreSQL-Protokolle in Amazon Logs veröffentlichen CloudWatch .
- Um sicherzustellen, dass alles wie erwartet funktioniert, testen Sie Ihre Anwendung auf der aktualisierten Datenbank mit ähnlichem Workload. Nachdem das Upgrade bestätigt wurde, können Sie diese Testinstance löschen.

Automatische Upgrades für Nebenversionen RDS für Postgre SQL

Wenn Sie beim Erstellen oder Ändern einer DB-Instance oder eines Multi-AZ-DB-Clusters die Option Automatisches Unterversion-Upgrade aktivieren, können Sie Ihre Datenbank automatisch aktualisieren lassen.

RDSFür jede SQL Postgre-Hauptversion wird eine Nebenversion RDS als automatische Upgrade-Version bezeichnet. Nachdem eine Nebenversion von Amazon getestet und genehmigt wurdeRDS, erfolgt das Upgrade der Nebenversion automatisch während Ihres Wartungsfensters. RDSlegt nicht automatisch neuere veröffentlichte Nebenversionen als automatische Upgrade-Version fest. Bevor RDS eine neuere automatische Upgrade-Version bestimmt, werden mehrere Kriterien berücksichtigt, wie beispielsweise die folgenden:

- Bekannte Sicherheitsprobleme
- Fehler in der SQL Postgre-Community-Version
- Gesamtflottenstabilität seit Erscheinen der Minor-Version

Sie können den folgenden AWS CLI Befehl verwenden, um die aktuelle Zielversion für ein automatisches kleines Upgrade für eine bestimmte SQL Postgre-Nebenversion in einer bestimmten Version zu ermitteln. AWS-Region

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Der folgende AWS CLI Befehl bestimmt beispielsweise das automatische kleinere Upgrade-Ziel für die SQL Postgre-Nebenversion 16.1 in den USA Ost (Ohio) AWS-Region (us-east-2).

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version 16.1 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
```

```
--output table
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version 16.1 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

Ihre Ausgabe sieht Folgendem ähnlich.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  16.2           |
|  True        |  16.3           |
|  False       |  16.4           |
|  False       |  16.5           |
|  False       |  16.6           |
|  False       |  17.1           |
|  False       |  17.2           |
+--------------+-----------------+
```

In diesem Beispiel bezieht sich der `AutoUpgrade` Wert `True` auf SQL Postgre-Version 16.3. Das automatische kleinere Upgrade-Ziel ist also SQL Postgre-Version 16.3, die in der Ausgabe hervorgehoben ist.

Eine SQL Postgre-Datenbank wird während Ihres Wartungsfensters automatisch aktualisiert, wenn die folgenden Kriterien erfüllt sind:

- Für die Datenbank ist die Option Automatisches Unterversion-Upgrade aktiviert.
- Die Datenbank führt eine Unterversion der DB-Engine aus, die niedriger ist als die aktuelle Unterversion des automatischen Upgrades.

Weitere Informationen finden Sie unter Automatisches Upgraden der Engine-Unterversion.

> **Note**
>
> Ein SQL Postgre-Upgrade aktualisiert keine SQL Postgre-Erweiterungen. Informationen zum Upgrade von Erweiterungen finden Sie unter Aktualisierung von SQL Postgre-Erweiterungen RDS für SQL Postgre-Datenbanken.

Aktualisierung von SQL Postgre-Erweiterungen RDS für SQL Postgre-Datenbanken

Ein SQL Postgre-Engine-Upgrade aktualisiert die meisten Postgre-Erweiterungen nicht. SQL Um eine Erweiterung nach einem Upgrade auf eine Nebenversion zu aktualisieren, verwenden Sie den Befehl `ALTER EXTENSION UPDATE`.

> **Note**
>
> Informationen zur Aktualisierung der GIS Post-Erweiterung finden Sie unter Verwaltung von Geodaten mit der GIS Post-Erweiterung ()Schritt 6: Aktualisieren Sie die GIS Post-Erweiterung.
>
> Um die `pg_repack`-Erweiterung zu aktualisieren, entfernen Sie die Erweiterung und erstellen Sie eine neue Version in der aktualisierten Datenbank. Weitere Informationen finden Sie unter pg_repack installation in der `pg_repack`-Dokumentation.

Um eine Erweiterung zu aktualisieren, verwenden Sie den folgenden Befehl.

```
ALTER EXTENSION extension_name UPDATE TO 'new_version';
```

Eine Liste der unterstützten Versionen von SQL Postgre-Erweiterungen finden Sie unterUnterstützte SQL Postgre-Erweiterungsversionen.

Verwenden Sie den Postgre-Katalog SQL pg_extension im folgenden Befehl, um Ihre aktuell installierten Erweiterungen aufzulisten.

```
SELECT * FROM pg_extension;
```

Um eine Liste der spezifischen Erweiterungsversionen anzuzeigen, die für Ihre Installation verfügbar sind, verwenden Sie die Postgre-Ansicht SQL pg_available_extension_versions im folgenden Befehl.

```
SELECT * FROM pg_available_extension_versions;
```

Aktualisierung einer SQL Postgre-DB-Snapshot-Engine-Version

Mit Amazon RDS können Sie einen Storage Volume DB-Snapshot Ihrer SQL Postgre-DB-Instance erstellen. Wenn Sie einen DB-Snapshot erstellen, basiert der Snapshot auf der Engine-Version, die von Ihrer RDS Amazon-Instance verwendet wird. Sie können die Engine-Version für Ihre DB-Snapshots aktualisieren.

Wenn Sie einen DB-Snapshot wiederherstellen, der auf eine neue Engine-Version aktualisiert wurde, sollten Sie prüfen, ob das Upgrade erfolgreich durchgeführt wurde. Weitere Informationen zu größeren Versionsaktualisierungen finden Sie unter Upgrades der RDS for PostgreSQL-DB-Engine. Informationen zum Wiederherstellen eines DB-Snapshots finden Sie unter Wiederherstellung auf einer DB-Instance.

Sie können manuelle DB-Snapshots aktualisieren, die verschlüsselt oder nicht verschlüsselt sind.

Verwenden Sie das folgende AWS CLI Beispiel, um die verfügbaren Engine-Versionen RDS für Ihren SQL Postgre-DB-Snapshot anzuzeigen.

```
aws rds describe-db-engine-versions --engine postgres  --engine-version example-
engine-version --query "DBEngineVersions[*].ValidUpgradeTarget[*].
{EngineVersion:EngineVersion}" --output text --include-all
```

Weitere Informationen zu verfügbaren Engine-Versionen RDS für SQL Postgre-DB-Snapshots finden Sie unter. Auswahl einer Hauptversion für ein RDS for PostgreSQL-Upgrade

> **Note**
>
> Sie können keine Upgrades für DB-Snapshots durchführen, die durch automatisierte Sicherungsvorgänge erstellt wurden.

Konsole

So führen Sie ein Upgrade eines DB-Snapshots durch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den Snapshot für die Aktualisierung aus.

4. Wählen Sie unter Actions (Aktionen) die Option Upgrade Snapshot (Snapshot aktualisieren). Die Seite Upgrade snapshot (Snapshot aktualisieren) erscheint.
5. Wählen Sie zum Aktualisieren New engine version (Neue Engine-Version).
6. Wählen Sie Save changes (Änderungen speichern), um den Snapshot zu aktualisieren.

   Während des Upgrades werden alle Snapshot-Aktionen für diesen DB-Snapshot deaktiviert. Außerdem wird der Status des DB-Snapshots von available (verfügbar) in upgraden... geändert. Wenn der Vorgang abgeschlossen wurde, wird der Status in active (aktiv) geändert. Wenn das Upgrade für den DB-Snapshot aufgrund einer Beschädigung des Snapshots nicht durchgeführt werden kann, wird der Status in unavailable (nicht verfügbar) geändert. Sie können den Snapshot aus diesem Zustand nicht wiederherstellen.

   > **Note**
   >
   > Wenn die Aktualisierung des DB-Snapshots fehlschlägt, wird der Snapshot wieder in seinen ursprünglichen Zustand zurückgebracht.

AWS CLI

Verwenden Sie den AWS CLI modify-db-snapshotBefehl, um einen DB-Snapshot auf eine neue Version der Datenbank-Engine zu aktualisieren.

Parameter

- `--db-snapshot-identifier`: die Kennung des DB-Snapshots, für den das Upgrade durchgeführt werden soll Die Kennung muss ein eindeutiger Amazon-Ressourcenname (ARN) sein. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.
- `--engine-version`: Die Engine-Version, auf die das Upgrade des DB-Snapshots durchgeführt werden soll

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

RDS API

Um einen DB-Snapshot auf eine neue Version der Datenbank-Engine zu aktualisieren, rufen Sie den Amazon RDS API odifyDBSnapshotM-Vorgang auf.

- `DBSnapshotIdentifier`: die Kennung des DB-Snapshots, für den das Upgrade durchgeführt werden soll Die Kennung muss ein eindeutiger Amazon-Ressourcenname (ARN) sein. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.
- `EngineVersion`: Die Engine-Version, auf die das Upgrade des DB-Snapshots durchgeführt werden soll

Arbeiten mit Read Replicas in Amazon RDS for PostgreSQL

Sie können Lesevorgänge für Ihre Amazon RDS for PostgreSQL PostgreSQL-DB-Instances skalieren, indem Sie den Instances Read Replicas hinzufügen. Wie andere Amazon RDS-Datenbank-Engines verwendet RDS for PostgreSQL native Replikationsmechanismen von PostgreSQL, um Read Replicas über Änderungen in der Quell-DB auf dem neuesten Stand zu halten. Allgemeine Informationen zu Lesereplikaten und Amazon RDS finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Im Folgenden finden Sie spezifische Informationen zum Arbeiten mit Lesereplikaten in RDS for PostgreSQL.

Logische Dekodierung auf einer Read Replica

RDS for PostgreSQL unterstützt logische Replikation aus dem Standbymodus mit PostgreSQL 16.1. Auf diese Weise können Sie eine logische Dekodierung aus einer schreibgeschützten Standby-Instanz erstellen, wodurch die Belastung der primären DB-Instance reduziert wird. Sie können eine höhere Verfügbarkeit für Ihre Anwendungen erreichen, die Daten zwischen mehreren Systemen synchronisieren müssen. Diese Funktion steigert die Leistung Ihres Data Warehouse und Ihrer Datenanalyse.

Außerdem sorgen Replikationsslots in einem bestimmten Standby-Modus dafür, dass dieser Standby-Server zu einem Primärserver heraufgestuft wird. Das bedeutet, dass im Falle eines Failovers einer primären DB-Instance oder der Heraufstufung einer Standby-Instanz zur neuen primären Instanz die Replikationssteckplätze bestehen bleiben und die ehemaligen Standby-Abonnenten davon nicht betroffen sind.

Um eine logische Dekodierung auf einer Read Replica zu erstellen

1. Logische Replikation aktivieren — Um eine logische Dekodierung in einem Standby-Modus zu erstellen, müssen Sie die logische Replikation auf Ihrer Quell-DB-Instance und deren physischem Replikat aktivieren. Weitere Informationen finden Sie unter Konfiguration von Read Replicas mit PostgreSQL.

   - Um die logische Replikation für eine neu erstellte RDS for PostgreSQL-DB-Instance zu aktivieren, erstellen Sie eine neue benutzerdefinierte DB-Parametergruppe und setzen Sie den statischen Parameter `rds.logical_replication` auf. 1 Ordnen Sie dann diese DB-Parametergruppe der Quell-DB-Instance und ihrer physischen Read Replica zu. Weitere

Informationen finden Sie unter Zuordnen einer DB-Parametergruppe zu einer DB-Instance in Amazon RDS Amazon.

- Um die logische Replikation für eine bestehende RDS for PostgreSQL-DB-Instance zu aktivieren, ändern Sie die benutzerdefinierte DB-Parametergruppe der Quell-DB-Instance und ihrer physischen Read Replica, um den statischen Parameter auf zu setzen. `rds.logical_replication` 1 Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

> **Note**
>
> Sie müssen die DB-Instance neu starten, um diese Parameteränderungen zu übernehmen.

Sie können die folgende Abfrage verwenden, um die Werte für `wal_level` und `rds.logical_replication` auf der Quell-DB-Instance und ihrer physischen Read Replica zu überprüfen.

```
Postgres=>SELECT name,setting FROM pg_settings WHERE name IN
 ('wal_level','rds.logical_replication');

 name                    | setting
-------------------------+---------
 rds.logical_replication | on
 wal_level               | logical
(2 rows)
```

2. Eine Tabelle in der Quelldatenbank erstellen — Connect der Datenbank in Ihrer Quell-DB-Instance her. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu einer DB-Instance, in der die PostgreSQL-Datenbank-Engine ausgeführt wird.

   Verwenden Sie die folgenden Abfragen, um eine Tabelle in Ihrer Quelldatenbank zu erstellen und Werte einzufügen:

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

```
Postgres=>INSERT INTO LR_test VALUES (generate_series(1,10000));
INSERT 0 10000
```

3. Eine Publikation für die Quelltabelle erstellen — Verwenden Sie die folgende Abfrage, um eine Publikation für die Tabelle in der Quell-DB-Instance zu erstellen.

```
Postgres=>CREATE PUBLICATION testpub FOR TABLE LR_test;
CREATE PUBLICATION
```

   Verwenden Sie eine SELECT-Abfrage, um die Details der Publikation zu überprüfen, die sowohl auf der Quell-DB-Instance als auch auf der physischen Read Replica-Instance erstellt wurde.

```
Postgres=>SELECT * from pg_publication;

oid    | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
 pubtruncate | pubviaroot
-------+---------+----------+--------------+-----------+-----------+-----------
+-------------+------------
 16429 | testpub |    16413 | f            | t         | t         | t         | t
          | f
(1 row)
```

4. Erstellen Sie ein Abonnement aus einer logischen Replikatinstanz — Erstellen Sie eine weitere RDS for PostgreSQL-DB-Instance als logische Replikatinstanz. Stellen Sie sicher, dass die VPC korrekt eingerichtet ist, um sicherzustellen, dass diese logische Replikatinstanz auf die physische Read Replica-Instanz zugreifen kann. Weitere Informationen finden Sie unter Amazon VPC und Amazon RDS. Wenn Ihre Quell-DB-Instance inaktiv ist, können Verbindungsprobleme auftreten und die Primär-DB-Instance sendet die Daten nicht in den Standby-Modus.

```
Postgres=>CREATE SUBSCRIPTION testsub CONNECTION 'host=Physical replica host name
 port=port
                dbname=source_db_name user=user password=password'
 PUBLICATION testpub;
NOTICE:  created replication slot "testsub" on publisher
CREATE SUBSCRIPTION
```

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

Verwenden Sie eine SELECT-Abfrage, um die Details des Abonnements auf der logischen Replikat-Instance zu überprüfen.

```
Postgres=>SELECT oid,subname,subenabled,subslotname,subpublications FROM
 pg_subscription;

oid    | subname | subenabled | subslotname | subpublications
-------+---------+------------+-------------+-----------------
 16429 | testsub | t          | testsub     | {testpub}
(1 row)
postgres=> select count(*) from LR_test;
 count
-------
 10000
(1 row)
```

5. Überprüfen Sie den Status des logischen Replikationssteckplatzes — Sie können nur den physischen Replikationssteckplatz auf Ihrer Quell-DB-Instance sehen.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name                                    | slot_type | confirmed_flush_lsn
---------------------------------------------+-----------+---------------------
 rds_us_west_2_db_dhqfsmo5wbbjqrn3m6b6ivdhu4 | physical  |
(1 row)
```

Auf Ihrer Read Replica-Instance können Sie jedoch den Steckplatz für die logische Replikation sehen, und der `confirmed_flush_lsn` Wert ändert sich, wenn die Anwendung aktiv logische Änderungen verarbeitet.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
-----------+-----------+---------------------
 testsub   | logical   | 0/500002F0
(1 row)
```

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
-----------+-----------+---------------------
 testsub   | logical   | 0/5413F5C0
(1 row)
```

Read Replica-Beschränkungen unter PostgreSQL

Für PostgreSQL-Lesereplikate gelten folgende Beschränkungen:

- PostgreSQL-Lesereplikate sind schreibgeschützt. Auch wenn ein Lesereplikat keine beschreibbare DB-Instance ist, können Sie es zu einer eigenständigen RDS-for-PostgreSQL-DB-Instance hochstufen. Der Prozess kann jedoch nicht rückgängig gemacht werden.
- Sie können kein Lesereplikat aus einem anderen Lesereplikat erstellen, falls auf Ihrer RDS-for-PostgreSQL-DB-Instance eine frühere PostgreSQL-DB-Version als 14.1 ausgeführt wird. RDS for PostgreSQL unterstützt kaskadierende Lesereplikate nur in RDS for PostgreSQL Version 14.1 und höher. Weitere Informationen finden Sie unter Verwendung von kaskadierenden Lesereplikaten mit RDS for PostgreSQL.
- Wenn Sie ein PostgreSQL-Lesereplikat hochstufen, wird es zu einer beschreibbaren DB-Instance. Es empfängt keine Write-Ahead Log (WAL)-Dateien von einer DB-Quell-Instance mehr und ist keine schreibgeschützte Instance mehr. Sie können neue Lesereplikate aus der hochgestuften DB-Instance erstellen wie für jede andere RDS-for-PostgreSQL-DB-Instance. Weitere Informationen finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.
- Wenn Sie eine PostgreSQL-Read Replica innerhalb einer Replikationskette (einer Reihe von kaskadierenden Read Replicas) heraufstufen, empfangen alle vorhandenen Downstream-Read Replicas weiterhin automatisch WAL-Dateien von der hochgestuften Instanz. Weitere Informationen finden Sie unter Verwendung von kaskadierenden Lesereplikaten mit RDS for PostgreSQL.
- Wenn keine Benutzertransaktionen auf der Quell-DB-Instance laufen, meldet das zugehörige PostgreSQL-Lesereplikat eine Replikationsverzögerung von bis zu fünf Minuten. Die Replikatverzögerung wird berechnet als `currentTime - lastCommitedTransactionTimestamp`, was bedeutet, dass, wenn keine Transaktionen verarbeitet werden, der Wert der Replikatverzögerung für einen bestimmten Zeitraum steigt, bis

das Write-Ahead-Log (WAL)-Segment wechselt. Standardmäßig wechselt RDS für PostgreSQL das WAL-Segment alle 5 Minuten, was zu einem Transaktionsdatensatz und einer Verringerung der gemeldeten Verzögerung führt.

- Sie können keine automatisierten Backups für PostgreSQL-Lesereplikate für RDS for PostgreSQL Versionen vor 14.1 aktivieren. Automatisierte Backups für Lesereplikate werden nur für RDS for PostgreSQL 14.1 und höhere Versionen unterstützt. Erstellen Sie für RDS for PostgreSQL 13 und frühere Versionen einen Snapshot aus einem Lesereplikat, wenn Sie eine Sicherungskopie davon wünschen.
- Point-in-time Recovery (PITR) wird für Read Replicas nicht unterstützt. Sie können PITR nur mit einer primären (Writer-) Instance und nicht mit einem Lesereplikat verwenden. Weitere Informationen hierzu finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.
- Read Replicas für PostgreSQL-Versionen 12 und niedriger werden während des 60-90-tägigen Wartungsfensters automatisch neu gestartet, um die Passwortrotation anzuwenden. Wenn das Replikat vor dem geplanten Neustart die Verbindung zur Quelle verliert, wird es trotzdem neu gestartet, um die Replikation fortzusetzen. Bei PostgreSQL-Versionen 13 und höher kann es bei Read Replicas während der Passwortrotation zu kurzen Unterbrechungen und Wiederverbindungen bei der Replikation kommen.

Konfiguration von Read Replicas mit PostgreSQL

RDS for PostgreSQL verwendet die native PostgreSQL-Streaming-Replikation, um eine schreibgeschützte Kopie einer Quell-DB-Instance zu erstellen. Diese Read-Replica-DB-Instance ist eine asynchron erstellte physische Replikation der Quell-DB-Instance. Sie wird von einer speziellen Verbindung erstellt, die Write-Ahead Log (WAL)-Daten zwischen der Quell-DB-Instance und dem Lesereplikat übermittelt. Weitere Informationen finden Sie unter Streaming-Replikation in der PostgreSQL-Dokumentation.

PostgreSQL streamt Datenbankänderungen asynchron über diese sichere Verbindung, während sie auf der Quell-DB-Instance vorgenommen werden. Sie können die Kommunikation von Ihren Clientanwendungen zur Quell-DB-Instance oder zu allen Lesereplikaten verschlüsseln, indem Sie den Parameter `ssl` auf 1 festlegen. Weitere Informationen finden Sie unter Verwenden von SSL mit einer PostgreSQL-DB-Instance.

PostgreSQL verwendet eine Replikations-Rolle, um die Streaming-Replikation durchzuführen. Die Rolle besitzt Berechtigungen, kann jedoch nicht für das Ändern von Daten verwendet werden. PostgreSQL verfügt über einen Einzelvorgang für die Handhabung von Replikation.

Sie können ein PostgreSQL-Lesereplikat erstellen, ohne den Betrieb oder die Benutzer der Quell-DB-Instance zu beeinträchtigen. Amazon RDS legt die erforderlichen Parameter und Berechtigungen für die Quell-DB-Instance und das Lesereplikat ohne Auswirkungen auf den Service fest. Ein Snapshot von der Quell-DB-Instance wird erstellt und zum Erstellen des Lesereplikats verwendet. Wenn Sie das Lesereplikat irgendwann in der Zukunft löschen, tritt kein Ausfall auf.

Sie können bis zu 15 Lesereplikate von einer Quell-DB-Instance innerhalb derselben Region erstellen. Ab RDS for PostgreSQL 14.1 können Sie auch bis zu drei Ebenen von Lesereplikaten in einer Kette (Kaskade) aus einer Quell-DB-Instance erstellen. Weitere Informationen finden Sie unter Verwendung von kaskadierenden Lesereplikaten mit RDS for PostgreSQL. In allen Fällen müssen für die Quell-DB-Instance automatisierte Backups konfiguriert sein. Dazu legen Sie den Aufbewahrungszeitraum für Sicherungen in Ihrer DB-Instance auf einen anderen Wert als 0 fest. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats.

Sie können Lesereplikate für Ihre DB-Instance von RDS for PostgreSQL in derselben AWS-Region wie Ihre Quell-DB-Instance erstellen. Dies wird als regionale Replikation bezeichnet. Sie können Read Replicas auch in einer anderen AWS-Regionen Datenbank als der Quell-DB-Instance erstellen. Dies wird als regionsübergreifende Replikation bezeichnet. Weitere Informationen zum Einrichten von regionsübergreifenden Lesereplikaten finden Sie unter Erstellen eines Lesereplikats in einer anderen AWS-Region. Die verschiedenen Mechanismen, die den Prozess für die regionale und regionsübergreifende Replikation unterstützen, unterscheiden sich je nach RDS-for-PostgreSQL-Version geringfügig, wie in Funktionsweise der Streaming-Replikation für verschiedene RDS-for-PostgreSQL-Versionen erläutert.

Damit die Replikation effektiv durchgeführt werden kann, sollte jedes Lesereplikat über die selbe Menge an Rechen- und Speicherressourcen wie die Quell-DB-Instance verfügen. Wenn Sie die Quell-DB-Instance skalieren, skalieren Sie unbedingt auch die Lesereplikate.

Amazon RDS überschreibt alle nicht-kompatiblen Parameter in einem Lesereplikat, wenn diese das Lesereplikat vom Hochfahren abhalten. Nehmen Sie beispielsweise an, dass der Wert des Parameters `max_connections` auf der Quell-DB-Instance höher als auf dem Lesereplikat ist. In diesem Fall aktualisiert Amazon RDS den Parameter auf dem Lesereplikat, sodass er denselben Wert wie auf der Quell-DB-Instance hat.

Read Replicas von RDS for PostgreSQL haben Zugriff auf externe Datenbanken, die über Foreign Data Wrappers (FDWs) auf der Quell-DB-Instance verfügbar sind. Angenommen, Ihre RDS-for-PostgreSQL-DB-Instance verwendet den Wrapper `mysql_fdw` für den Zugriff auf Daten von RDS for MySQL. In diesem Fall können Ihre Lesereplikate auch auf diese Daten zugreifen. Zu den weiteren

unterstützten FDWs gehören, und. `oracle_fdw` `postgres_fdw` `tds_fdw` Weitere Informationen finden Sie unter Arbeiten mit den unterstützten Foreign Data Wrappern für RDS SQL.

Verwenden von RDS-for-PostgreSQL-Lesereplikate mit Multi-AZ-Konfigurationen

Sie können ein Lesereplikat aus einer Single-AZ- oder Multi-AZ-DB-Instance erstellen. Sie können Multi-AZ-Bereitstellungen verwenden, um die Haltbarkeit und Verfügbarkeit kritischer Daten mit einem Standby-Replikat zu verbessern. Ein Standby-Replikat ist ein dediziertes Lesereplikat, das die Workload übernehmen kann, wenn die Quell-DB ausfällt. Sie können Ihr Standby-Replikat nicht dazu verwenden, Leseverkehr bereitzustellen. Sie können jedoch Lesereplikate aus Multi-AZ-DB-Instances mit hohem Datenverkehr erstellen, um schreibgeschützte Abfragen auslagern zu können. Weitere Informationen zu Multi-AZ-Bereitstellungen finden Sie unter Multi-AZ-DB-Instance-Bereitstellungen für Amazon RDS.

Wenn die Quell-DB-Instance einer Multi-AZ-Bereitstellung ein Failover auf eine Standby-Instance vornimmt, wechseln die zugehörigen Lesereplikate zur Verwendung der Standby-Instance (jetzt primär) als Replikationsquelle. Die Lesereplikate müssen möglicherweise je nach RDS-for-PostgreSQL-Version wie folgt neu gestartet werden:

- PostgreSQL 13 und höhere Versionen – Ein Neustart ist nicht erforderlich. Die Lesereplikate werden automatisch mit der neuen Primär-Instance synchronisiert. In einigen Fällen legt Ihre Clientanwendung jedoch die Details des Domain Name Service (DNS) für Ihre Lesereplikate im Zwischenspeicher ab. Falls ja, setzen Sie den Wert time-to-live (TTL) auf weniger als 30 Sekunden. Damit wird verhindert, dass das Lesereplikat eine veraltete IP-Adresse beibehält (wodurch verhindert wird, dass es mit der neuen Primär-Instance synchronisiert wird). Weitere Informationen über diese und andere bewährte Methoden finden Sie unter Grundlegende Anleitungen für den Amazon RDS-Betrieb.
- PostgreSQL 12 und alle früheren Versionen – Die Lesereplikate werden nach einem Failover auf das Standby-Replikat automatisch neu gestartet, da der Standby (jetzt primär) eine andere IP-Adresse und einen anderen Instance-Namen hat. Durch den Neustart wird das Lesereplikat mit der neuen Primär-Instance synchronisiert.

Weitere Informationen zu Failover finden Sie unter Failover einer Multi-AZ-DB-Instance für Amazon RDS. Weitere Informationen dazu, wie Lesereplikate in einer Multi-AZ-Bereitstellung funktionieren, finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Wenn Sie Failover-Unterstützung für ein Lesereplikat bereitstellen möchten, können Sie ein Lesereplikat als Multi-AZ-DB-Instance erstellen. Amazon RDS erstellt dann eine Standby-Version

des Replikats in einer anderen Availability Zone. Das Erstellen Ihres Lesereplikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist.

Verwendung von kaskadierenden Lesereplikaten mit RDS for PostgreSQL

Ab Version 14.1 unterstützt RDS for PostgreSQL kaskadierende Lesereplikate. Mit kaskadierenden Lesereplikaten können Sie Lesereplikate skalieren, ohne dass Sie zusätzlichen Overhead für Ihre Quell-DB-Instance von RDS for PostgreSQL verursachen. Aktualisierungen des WAL-Protokolls werden von der Quell-DB-Instance nicht an jedes Lesereplikat gesendet. Stattdessen sendet jedes Lesereplikat Replica einer kaskadierenden Serie WAL-Log-Updates an das nächste Lesereplikat in der Reihe. Damit wird die Belastung der Quell-DB-Instance reduziert.

Bei kaskadierenden Lesereplikaten sendet Ihre DB-Instance von RDS for PostgreSQL WAL-Daten an das erste Lesereplikat in der Kette. Dieses Lesereplikat sendet dann WAL-Daten an das zweite Replikat in der Kette usw. Das Endergebnis ist, dass alle Lesereplikate in der Kette die Änderungen von der DB-Instance von RDS for PostgreSQL aufweisen, jedoch ohne Overhead ausschließlich auf der Quell-DB-Instance zu verursachen.

Sie können eine Reihe von bis zu drei Lesereplikaten in einer Kette von einer Quell-DB-Instance von RDS for PostgreSQL erstellen. Angenommen, Sie haben eine DB-Instance von RDS for PostgreSQL 14.1, `rpg-db-main`. Sie haben die folgenden Möglichkeiten:

- Beginnend mit `rpg-db-main` erstellen Sie das erste Lesereplikat in der Kette `read-replica-1`.
- Als Nächstes erstellen Sie ab `read-replica-1` das nächste Lesereplikat in der Kette `read-replica-2`.
- Schließlich erstellen Sie ab `read-replica-2` das nächste Lesereplikat in der Kette `read-replica-3`.

Sie können kein weiteres Lesereplikat über dieses dritte kaskadierende Lesereplikat hinaus in der Reihe für `rpg-db-main` erstellen. Eine vollständige Reihe von Instances aus einer Quell-DB-Instance von RDS for PostgreSQL bis zum Ende einer Reihe kaskadierender Lesereplikate kann aus höchstens vier DB-Instances bestehen.

Damit die Kaskadierung von Lesereplikaten funktioniert, aktivieren Sie automatische Backups auf Ihrem RDS for PostgreSQL. Erstellen Sie zuerst das Lesereplikat und aktivieren Sie dann automatische Backups auf der DB-Instance von RDS for PostgreSQL. Der Prozess ist der gleiche wie bei anderen Amazon-RDS-DB-Engines. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats.

Wie bei jedem Lesereplikat können Sie ein Lesereplikat, das Teil einer Kaskade ist, hochstufen. Wenn Sie ein Lesereplikat aus einer Kette von Lesereplikaten hochstufen, wird dieses Replikat aus der Kette entfernt. Angenommen, Sie möchten einen Teil der Workload von Ihrer `rpg-db-main`-DB-Instance zu einer neuen Instance verschieben, die nur von der Buchhaltung verwendet wird. Ausgehend von der Kette von drei Lesereplikaten aus dem Beispiel entscheiden Sie sich, `read-replica-2` hochzustufen. Die Kette ist wie folgt betroffen:

- Durch Hochstufen von `read-replica-2` wird es aus der Replikationskette entfernt.
  - Es ist jetzt eine vollständige DB-Instance mit Lese-/Schreibzugriff.
  - Die Replizierung auf `read-replica-3` wird fortgesetzt wie vor der Hochstufung.
- Ihre `rpg-db-main` setzt die Replizierung auf `read-replica-1` fort.

Weitere Informationen über das Hochstufen von Lesereplikaten finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.

> **Note**
>
> - RDS for PostgreSQL unterstützt keine Hauptversions-Upgrades für kaskadierende Replikate. Bevor Sie ein Hauptversions-Upgrade durchführen, müssen Sie kaskadierende Replikate entfernen. Sie können sie nach Abschluss des Upgrades auf Ihrer Quell-DB-Instance und den First-Level-Replikaten neu erstellen.
> - Für kaskadierende Lesereplikate unterstützt RDS für PostgreSQL 15 Lesereplikate für jede Quell-DB-Instance auf der ersten Replikationsebene und 5 Lesereplikate für jede Quell-DB-Instance auf der zweiten und dritten Replikationsebene.

Erstellen von regionsübergreifenden kaskadierenden Read Replicas mit RDS für PostgreSQL

RDS für PostgreSQL unterstützt regionsübergreifende kaskadierende Read Replicas. Sie können aus der Quell-DB-Instance ein regionsübergreifendes Replikat und anschließend daraus Repliken derselben Region erstellen. Sie können auch ein regionsübergreifendes Replikat aus der Quell-DB-Instance und anschließend regionsübergreifende Replikate daraus erstellen.

Erstellen Sie ein regionsübergreifendes Replikat und anschließend regionsübergreifende Replikate

Sie können eine RDS for PostgreSQL-DB-Instance mit Version 14.1 oder höher verwenden`rpg-db-main`, um Folgendes zu tun:

1. Beginnen Sie mit `rpg-db-main` (US-EAST-1) und erstellen Sie die erste regionsübergreifende Read Replica in der Kette, (US-WEST-2). `read-replica-1`
2. Erstellen Sie mit der ersten regionsübergreifenden `read-replica-1` (US-WEST-2) die zweite Read Replica in der Kette, (US-WEST-2). `read-replica-2`
3. Verwenden Sie`read-replica-2`, um das dritte Read Replica in der Kette (US-WEST-2) zu erstellen. `read-replica-3`

Erstellen Sie ein Replikat derselben Region und anschließend regionsübergreifende Replikate

Sie können eine RDS for PostgreSQL-DB-Instance mit Version 14.1 oder höher verwenden`rpg-db-main`, um Folgendes zu tun:

1. Beginnen Sie mit `rpg-db-main` (US-EAST-1) und erstellen Sie das erste Read Replica in der Kette, (US-EAST-1). `read-replica-1`
2. Erstellen Sie mit `read-replica-1` (US-EAST-1) die erste regionsübergreifende Read Replica in der Kette, (US-WEST-2). `read-replica-2`
3. Erstellen Sie mit `read-replica-2` (US-WEST-2) das dritte Read Replica in der Kette, (US-WEST-2). `read-replica-3`

Einschränkungen bei der Erstellung regionsübergreifender Read Replicas

- Eine regionsübergreifende kaskadierende Kette von Datenbankreplikaten kann sich über maximal zwei Regionen mit maximal vier Ebenen erstrecken. Die vier Ebenen umfassen die Datenbankquelle und drei Read Replicas.

Vorteile der Verwendung von kaskadierenden Read Replicas

- Verbesserte Leseskalierbarkeit — Durch die Verteilung von Leseabfragen auf mehrere Replikate trägt die kaskadierende Replikation zum Lastenausgleich bei. Dies verbessert die Leistung, insbesondere bei leseintensiven Anwendungen, indem die Belastung der Writer-Datenbank verringert wird.

- Geografische Verteilung — Kaskadierende Replikate können sich an verschiedenen geografischen Standorten befinden. Dies reduziert die Latenz für Benutzer, die sich weit von der Primärdatenbank entfernt befinden, und bietet eine lokale Lesereplik, wodurch die Leistung und das Benutzererlebnis verbessert werden.
- Hochverfügbarkeit und Disaster Recovery — Im Falle eines Ausfalls des Primärservers können Replikate auf Primärserver hochgestuft werden, wodurch die Kontinuität gewährleistet ist. Durch die kaskadierende Replikation wird dies noch weiter verbessert, indem mehrere Ebenen von Failover-Optionen bereitgestellt werden, wodurch die allgemeine Ausfallsicherheit des Systems verbessert wird.
- Flexibilität und modulares Wachstum — Wenn das System wächst, können neue Replikate auf verschiedenen Ebenen hinzugefügt werden, ohne dass die Primärdatenbank umfassend neu konfiguriert werden muss. Dieser modulare Ansatz ermöglicht ein skalierbares und überschaubares Wachstum der Replikationseinrichtung.

Bewährte Methode für die Verwendung von regionsübergreifenden Read Replicas

- Bevor Sie ein Replikat heraufstufen, erstellen Sie zusätzliche Replikate. Das spart Zeit und ermöglicht eine effiziente Bewältigung der Arbeitslast.

Funktionsweise der Streaming-Replikation für verschiedene RDS-for-PostgreSQL-Versionen

Wie in Konfiguration von Read Replicas mit PostgreSQL erläutert, verwendet RDS for PostgreSQL das native Streaming-Replikationsprotokoll von PostgreSQL, um WAL-Daten von der Quell-DB-Instance zu senden. Es sendet Quell-WAL-Daten an Lesereplikate sowohl für regionale als auch regionsübergreifende Lesereplikate. Mit Version 9.4 führte PostgreSQL physische Replikationsslots als unterstützenden Mechanismus für den Replikationsprozess ein.

Ein physischer Replikationsslot verhindert, dass eine Quell-DB-Instance WAL-Daten entfernt, bevor sie von allen Lesereplikaten verbraucht werden. Jedes Lesereplikat hat einen eigenen physischen Slot in der Quell-DB-Instance. Der Slot verfolgt das älteste WAL (nach logischer Sequenznummer, LSN), das möglicherweise vom Replikat benötigt wird. Nachdem alle Slots und DB-Verbindungen über eine bestimmte WAL (LSN) hinausgegangen sind, wird diese LSN zum Kandidaten für die Entfernung am nächsten Checkpunkt.

Amazon RDS verwendet Amazon S3, um WAL-Daten zu archivieren. Bei regionalen Lesereplikaten können Sie diese archivierten Daten verwenden, um das Lesereplikat bei Bedarf wiederherzustellen. Dies kann beispielsweise erforderlich werden, wenn die Verbindung zwischen Quell-DB und Lesereplikat aus irgendeinem Grund unterbrochen wird.

In der folgenden Tabelle finden Sie eine Zusammenfassung der Unterschiede zwischen PostgreSQL-Versionen und den unterstützenden Mechanismen für regional und regionsübergreifend, die von RDS for PostgreSQL verwendet werden.

Version	Regional	Regionsübergreifend
PostgreSQL 14.1 und höhere Versionen	• Replikationsslots • Amazon-S3-Archiv	• Replikationsslots
PostgreSQL 13 und niedrigere Versionen	• Amazon-S3-Archiv	• Replikationsslots

Weitere Informationen finden Sie unter Überwachen und Optimieren des Replikationsprozesses.

Grundlegendes zu Parametern, die die PostgreSQL-Replikation steuern

Die folgenden Parameter beeinflussen den Replikationsprozess und bestimmen, wie gut Lesereplikate mit der Quell-DB-Instance auf dem neuesten Stand bleiben:

max_wal_senders

Der Parameter `max_wal_senders` gibt die maximale Anzahl von Verbindungen an, die die Quell-DB-Instance gleichzeitig über das Streaming-Replikationsprotokoll unterstützen kann.

Der Standardwert variiert für RDS-Versionen für PostgreSQL:

- Für die Versionen 13, 14 und 15 ist der Standardwert 20.
- Für die Versionen 16 und höher ist der Standardwert 35.

Dieser Parameter sollte etwas höher als die tatsächliche Anzahl der Lesereplikate eingestellt werden. Wenn dieser Parameter für die Anzahl der Lesereplikate zu niedrig eingestellt ist, wird die Replikation beendet.

Weitere Informationen dazu finden Sie im Abschnitt max_wal_senders der PostgreSQL-Dokumentation.

> **Note**
>
> `max_wal_sendersist` ein statischer Parameter, der einen Neustart der DB-Instance erfordert, damit die Änderungen wirksam werden.

wal_keep_segments

Der Parameter `wal_keep_segments` gibt die Anzahl der Write-Ahead Log (WAL)-Dateien an, die die Quell-DB-Instance im `pg_wal`-Verzeichnis speichert. Die Standardeinstellung ist 32.

Wenn `wal_keep_segments` nicht auf einen ausreichend großen Wert für Ihre Bereitstellung festgelegt ist, kann ein Lesereplikat so weit zurückbleiben, dass das Streaming der Replikation anhält. In diesem Fall generiert Amazon RDS einen Replikationsfehler und beginnt mit der Wiederherstellung des Lesereplikats. Dies geschieht, indem es die archivierten WAL-Daten der Quell-DB-Instance von Amazon S3 wiedergibt. Dieser Wiederherstellungsprozess wird fortgeführt, bis das Lesereplikat ausreichend nah am aktuellen Stand angekommen ist, um mit dem Streaming der Replikation fortfahren zu können. Sie können diesen Prozess unter Beispiel: Wiederherstellen eines Lesereplikats nach einer Replikationsunterbrechungen in Aktion sehen, wie er vom PostgreSQL-Protokoll erfasst wurde.

> **Note**
>
> In PostgreSQL Version 13 wird der Parameter `wal_keep_segments` als `wal_keep_size` bezeichnet. Er dient dem gleichen Zweck wie `wal_keep_segments`, der Standardwert wird jedoch in Megabyte (MB) (2 048 MB) und nicht als Anzahl der Dateien angegeben. Weitere Informationen finden Sie unter wal_keep_segments und wal_keep_size in der PostgreSQL-Dokumentation.

max_slot_wal_keep_size

Der Parameter `max_slot_wal_keep_size` steuert die Menge an WAL-Daten, die die RDS-for-PostgreSQL-DB-Instance im Verzeichnis `pg_wal` für Slots speichert. Dieser Parameter wird für Konfigurationen verwendet, die Replikationsslots nutzen. Der Standardwert für diesen Parameter ist `-1`, was bedeutet, dass es keine Begrenzung gibt, wie viele WAL-Daten auf der Quell-DB-Instance gespeichert werden. Weitere Informationen zur Überwachung Ihrer Replikationsslots finden Sie unter Überwachen von Replikationsslots für Ihre RDS-for-PostgreSQL-DB-Instance.

Weitere Informationen zu diesem Parameter finden Sie unter max_slot_wal_keep_size in der PostgreSQL-Dokumentation.

Sobald ein Stream, der WAL-Daten an ein Lesereplikat leitet, unterbrochen wird, wechselt PostgreSQL in den Wiederherstellungsmodus. Es stellt die gelesene Replik mithilfe archivierter WAL-Daten von Amazon S3 oder mithilfe der WAL-Daten wieder her, die dem Replikationsslot zugeordnet sind. Sobald dieser Vorgang abgeschlossen ist, nimmt PostgreSQL das Streaming der Replikation wieder auf.

Beispiel: Wiederherstellen eines Lesereplikats nach einer Replikationsunterbrechungen

Im folgenden Beispiel finden Sie die Protokolldetails, die den Wiederherstellungsprozess für ein Lesereplikat veranschaulichen. Das Beispiel stammt von einer RDS for PostgreSQL-DB-Instance, auf der PostgreSQL Version 12.9 in derselben Version AWS-Region wie die Quell-DB ausgeführt wird, sodass keine Replikationsslots verwendet werden. Der Wiederherstellungsprozess ist derselbe wie für andere RDS-for-PostgreSQL-DB-Instances, die eine frühere PostgreSQL-Version als 14.1 mit regionalen Lesereplikaten ausführen.

Wenn das Lesereplikat den Kontakt zur Quell-DB-Instance verloren hat, zeichnet Amazon RDS das Problem im Protokoll als `FATAL: could not receive data from WAL stream`-Nachricht zusammen mit dem `ERROR: requested WAL segment ... has already been removed` auf. Wie in der fett gedruckten Zeile gezeigt, stellt Amazon RDS das Replikat wieder her, indem es eine archivierte WAL-Datei wiedergibt.

```
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG:  switched WAL source from archive to stream
 after failure
2014-11-07 19:01:10 UTC::@:[11575]:LOG: started streaming WAL from primary at 1A/
D3000000 on timeline 1
2014-11-07 19:01:10 UTC::@:[11575]:FATAL: could not receive data from WAL stream:
ERROR:  requested WAL segment 000000010000001A000000D3 has already been removed
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG: could not restore file "00000002.history"
 from archive: return code 0
2014-11-07 19:01:15 UTC::@:[23180]:DEBUG: switched WAL source from stream to archive
 after failure recovering 000000010000001A000000D3
2014-11-07 19:01:16 UTC::@:[23180]:LOG:  restored log file "000000010000001A000000D3"
 from archive
```

Wenn Amazon RDS genügend archivierte WAL-Daten wiedergegeben hat, damit das Replikat aufholt, wird das Streaming an das Lesereplikat fortgesetzt. Wenn das Streaming fortgesetzt wird, schreibt Amazon RDS einen Eintrag ähnlich wie folgenden in die Protokolldatei.

```
2014-11-07 19:41:36 UTC::@:[24714]:LOG:started streaming WAL from primary at 1B/
B6000000 on timeline 1
```

Festlegen der Parameter, die den gemeinsam genutzten Speicher steuern

Die von Ihnen festgelegten Parameter bestimmen die Größe des gemeinsam genutzten Speichers für die Nachverfolgung von Transaktionen IDs, Sperren und vorbereiteten Transaktionen. Die Struktur des gemeinsam genutzten Speichers einer Standby-Instance muss der einer primären Instance entsprechen oder größer sein. Dadurch wird sichergestellt, dass der gemeinsam genutzte Speicher der Ersteren während der Wiederherstellung nicht knapp wird. Wenn die Parameterwerte des Replikats niedriger sind als die Parameterwerte der primären Instance, passt Amazon RDS die Replikatparameter automatisch an und startet die Engine neu.

Die betroffenen Parameter sind:

- max_connections
- max_worker_processes
- max_wal_senders
- max_prepared_transactions
- max_locks_per_transaction

Um RDS-Neustarts von Replikaten aufgrund von unzureichendem Speicher zu vermeiden, empfehlen wir, die Parameteränderungen in Form eines fortlaufenden Neustarts auf jedes Replikat anzuwenden. Sie müssen die folgenden Regeln anwenden, wenn Sie die Parameter festlegen:

- Erhöhen der Parameterwerte:
  - Sie sollten immer zuerst die Parameterwerte aller Lesereplikate erhöhen und einen fortlaufenden Neustart aller Replikate durchführen. Wenden Sie dann die Parameteränderungen auf die primäre Instance an und führen Sie dann einen Neustart aus.
- Verringern der Parameterwerte:
  - Sie sollten zuerst die Parameterwerte der primären Instance verringern und einen Neustart durchführen. Wenden Sie dann die Parameteränderungen auf alle zugehörigen Lesereplikate an und führen Sie einen fortlaufenden Neustart durch.

Überwachen und Optimieren des Replikationsprozesses

Wir empfehlen dringend, Ihre RDS-for-PostgreSQL-DB-Instance und Lesereplikate routinemäßig zu überwachen. Sie müssen sicherstellen, dass Ihre Lesereplikate mit den Änderungen an der Quell-DB-Instance Schritt halten. Amazon RDS stellt Ihre Lesereplikate transparent wieder her, wenn Unterbrechungen des Replikationsprozesses auftreten. Es ist jedoch am besten, wenn keine Wiederherstellung erforderlich ist. Die Wiederherstellung mit Replikationsslots ist schneller als die Verwendung des Amazon-S3-Archivs, aber jeder Wiederherstellungsprozess kann die Leseleistung beeinträchtigen.

Um festzustellen, wie gut Ihre Lesereplikate mit der Quell-DB-Instance Schritt halten, können Sie folgende Schritte ausführen:

- Prüfen Sie den Wert für **`ReplicaLag`** zwischen Quell-DB-Instance und Replikaten. Replikatverzögerung ist die Zeit in Sekunden, die ein Lesereplikat hinter seiner Quell-DB-Instance zurückbleibt. Diese Metrik gibt das Ergebnis der folgenden Abfrage zurück.

  ```
  SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS "ReplicaLag";
  ```

  Die Replikatverzögerung ist ein Hinweis darauf, wie gut ein Lesereplikat mit der Quell-DB-Instance Schritt hält. Dies ist die Latenz zwischen der Quell-DB-Instance und einem bestimmten Lesereplikat. Ein hoher Wert für die Replikatverzögerung kann auf eine Nichtübereinstimmung zwischen den DB-Instance-Klassen oder Speichertypen (oder beiden) hinweisen, die von der Quell-DB-Instance und ihren Lesereplikaten verwendet werden. Die DB-Instance-Klasse und die Speichertypen für die DB-Quellinstance und alle Lesereplikate sollten identisch sein.

  Replikatverzögerung kann auch das Ergebnis intermittierender Verbindungsprobleme sein. Sie können die Replikationsverzögerung in Amazon überwachen, CloudWatch indem Sie sich die Amazon `ReplicaLag` RDS-Metrik ansehen. Für weitere Informationen über `ReplicaLag` und andere Metriken für Amazon RDS finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS.
- Im PostgreSQL-Protokoll finden Sie Informationen, mit denen Sie Ihre Einstellungen anpassen können. Das PostgreSQL-Protokoll erfasst an jedem Checkpoint die Anzahl der recycelten Transaktionsprotokolldateien, wie im folgenden Beispiel gezeigt.

  ```
  2014-11-07 19:59:35 UTC::@:[26820]:LOG:  checkpoint complete: wrote 376 buffers
   (0.2%);
  ```

```
0 transaction log file(s) added, 0 removed, 1 recycled; write=35.681 s, sync=0.013 s,
 total=35.703 s;
sync files=10, longest=0.013 s, average=0.001 s
```

Sie können diese Informationen verwenden, um herauszufinden, wie viele Transaktionsdateien in einem bestimmten Zeitraum recycelt werden. Sie können die Einstellung für `wal_keep_segments` bei Bedarf ändern. Angenommen, das PostgreSQL-Protokoll zeigt bei `checkpoint complete` für ein 5-Minuten-Intervall `35 recycled` an. In diesem Fall reicht der `wal_keep_segments`-Standardwert 32 nicht aus, um mit der Streaming-Aktivität Schritt zu halten, und wir empfehlen Ihnen, den Wert dieses Parameters zu erhöhen.

- Verwenden Sie Amazon CloudWatch , um Metriken zu überwachen, mit denen Replikationsprobleme vorhergesagt werden können. Anstatt das PostgreSQL-Protokoll direkt zu analysieren, können Sie Amazon verwenden, CloudWatch um die gesammelten Metriken zu überprüfen. Sie können beispielsweise den Wert der Metrik `TransactionLogsGeneration` überprüfen, um zu sehen, wie viele WAL-Daten von der Quell-DB-Instance generiert werden. In einigen Fällen kann die Worklaod Ihrer DB-Instance WAL-Daten in großer Menge erzeugen. In diesem Fall müssen Sie möglicherweise die DB-Instance-Klasse für Ihre Quell-DB-Instance und Lesereplikate ändern. Die Verwendung einer Instance-Klasse mit hoher Netzwerkleistung (10 Gbit/s) kann die Verzögerung von Replikaten verringern.

Überwachen von Replikationsslots für Ihre RDS-for-PostgreSQL-DB-Instance

Alle Versionen von RDS for PostgreSQL verwenden Replikationsslots für regionsübergreifende Lesereplikate. RDS for PostgreSQL 14.1 und höhere Versionen verwenden Replikationsslots für regionale Lesereplikate. Regionale Lesereplikate nutzen Amazon S3 auch dazu, WAL-Daten zu archivieren. Mit anderen Worten, wenn Ihre DB-Instance und Lesereplikate PostgreSQL 14.1 oder höher ausführen, stehen sowohl Replikationsslots als auch Amazon-S3-Archive für die Wiederherstellung des Lesereplikats zur Verfügung. Die Wiederherstellung eines Lesereplikats Replica mit seinem Replikationsslots ist schneller als die Wiederherstellung aus dem Amazon-S3-Archiv. Wir empfehlen daher, die Replikationsslots und zugehörige Metriken zu überwachen.

Sie können die Replikationsslots auf Ihren RDS-for-PostgreSQL-DB-Instances anzeigen, indem Sie die Ansicht `pg_replication_slots` wie folgt abfragen.

```
postgres=> SELECT * FROM pg_replication_slots;
slot_name                 | plugin | slot_type | datoid | database | temporary |
 active | active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn |
 wal_status | safe_wal_size | two_phase
```

```
---------------------------+--------+----------+--------+----------+-----------
+--------+------------+------+-------------+-------------+--------------------
+------------+--------------+-----------
rds_us_west_1_db_555555555 |        | physical  |        |          | f         | t
  |      13194 |      |              | 23/D8000060 |                    | reserved   |
               | f
(1 row)
```

Der `wal_status` `reserved` bedeutet, dass die Menge der vom Slot gespeicherten WAL-Daten innerhalb der Grenzen des Parameters `max_wal_size` liegt. Mit anderen Worten: Der Replikationsslots ist angemessen dimensioniert. Folgende Statuswerte sind außerdem möglich:

- `extended` – Der Slot überschreitet die Einstellung `max_wal_size`, die WAL-Daten werden jedoch beibehalten.
- `unreserved` – Der Slot verfügt nicht mehr über alle erforderlichen WAL-Daten. Ein Teil davon wird am nächsten Checkpoint entfernt.
- `lost` – Einige erforderliche WAL-Daten wurden entfernt. Der Slot ist nicht mehr nutzbar.

Der Status `unreserved` und der `lost` Status von `wal_status` werden nur angezeigt, wenn der Wert nicht negativ `max_slot_wal_keep_size` ist.

Die Ansicht `pg_replication_slots` zeigt Ihnen den aktuellen Status Ihrer Replikationsslots an. Um die Leistung Ihrer Replikationsslots zu bewerten, können Sie Amazon verwenden CloudWatch und die folgenden Metriken überwachen:

- **`OldestReplicationSlotLag`**— Zeigt die Menge an Write-Ahead Log (WAL) -Daten auf der Quelle an, die nicht vom Replikat mit der größten Verzögerung verbraucht wurde.
- **`TransactionLogsDiskUsage`** – Zeigt an, wie viel Speicher für WAL-Daten verwendet wird. Wenn ein Lesereplikat erheblich zurückbleibt, kann der Wert dieser Metrik erheblich steigen.

Weitere Informationen zur Verwendung von Amazon CloudWatch und seinen Metriken für RDS for PostgreSQL finden Sie unter. Überwachung von Amazon RDS mit Amazon CloudWatch Weitere Informationen zur Überwachung des Streamings der Replikation auf Ihren RDS-for-PostgreSQL-DB-Instances finden Sie unter Bewährte Methoden für die Amazon-RDS-PostgreSQL-Replikation im AWS -Datenbank-Blog.

Problembehandlung für RDS for PostgreSQL Read Replica

Im Folgenden finden Sie Ideen zur Fehlerbehebung für einige häufig auftretende Probleme mit Read Replica in RDS for PostgreSQL.

Beenden Sie die Abfrage, die den Read Replica-Lag verursacht hat

Transaktionen, die im Transaktionsstatus entweder aktiv oder inaktiv sind und über einen längeren Zeitraum in der Datenbank ausgeführt werden, können den WAL-Replikationsprozess stören und dadurch die Replikationsverzögerung erhöhen. Achten Sie daher darauf, die Laufzeit dieser Transaktionen mit der `pg_stat_activity` PostgreSQL-Ansicht zu überwachen.

Führen Sie eine Abfrage auf der primären Instanz ähnlich der folgenden aus, um die Prozess-ID (PID) der Abfrage zu ermitteln, die über einen längeren Zeitraum ausgeführt wird:

```
SELECT datname, pid,usename, client_addr, backend_start,
xact_start, current_timestamp - xact_start AS xact_runtime, state,
backend_xmin FROM pg_stat_activity WHERE state='active';
```

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Nachdem Sie die PID der Abfrage identifiziert haben, können Sie wählen, ob Sie die Abfrage beenden möchten.

Führen Sie eine Abfrage auf der primären Instanz aus, die der folgenden ähnelt, um die Abfrage zu beenden, die für eine lange Zeit läuft:

```
SELECT pg_terminate_backend(PID);
```

Verbesserung der Abfrageleistung für RDS für PostgreSQL mit Amazon-RDS-optimierten Lesevorgängen

Mit Amazon-RDS-optimierten Lesevorgängen können Sie eine schnellere Abfrageverarbeitung für RDS für PostgreSQL erreichen. Eine DB-Instance von RDS für PostgreSQL oder ein Multi-AZ-DB-Cluster, die bzw. der RDS-optimierte Lesevorgänge verwendet, kann eine doppelt so schnelle Abfrageverarbeitung erreichen als eine DB-Instance bzw. ein DB-Cluster, die bzw. der diese Funktion nicht verwendet.

Themen

- Übersicht über RDS-optimierte Lesevorgänge in PostgreSQL
- Anwendungsfälle für RDS Optimized Reads
- Bewährte Methoden für RDS Optimized Reads
- Verwenden von RDS Optimized Reads
- Überwachen von DB-Instances, die RDS Optimized Reads verwenden
- Einschränkungen für RDS-optimierte Lesevorgänge in PostgreSQL

Übersicht über RDS-optimierte Lesevorgänge in PostgreSQL

Optimized Reads ist standardmäßig auf RDS für PostgreSQL-Versionen 15.2 und höher, 14.7 und höher und 13.10 und höher verfügbar, wenn basierte DB-Instance-Klassen verwendet NVMe werden. Hardwarespezifikationen, die angeben, welche Instances verwendet werden, finden Sie unter. NVMe Hardware-Spezifikationen für DB-Instance-Klassen

Wenn Sie eine RDS for PostgreSQL-DB-Instance oder einen Multi-AZ-DB-Cluster verwenden, für die RDS-optimierte Lesevorgänge aktiviert sind, wird mit dem lokalen, auf Non-Volatile Memory Express (NVMe) basierenden Solid-State-Drive-Speicher (SSD) auf Blockebene eine um bis zu 50% schnellere Abfrageleistung erzielt. Sie können eine schnellere Abfrageverarbeitung erreichen, indem Sie die von PostgreSQL generierten temporären Tabellen im lokalen Speicher ablegen, wodurch der Datenverkehr zu Elastic Block Storage (EBS) über das Netzwerk reduziert wird.

In PostgreSQL werden temporäre Objekte einem temporären Namespace zugewiesen, der am Ende der Sitzung automatisch gelöscht wird. Der temporäre Namespace entfernt beim Löschen alle Objekte, die von der Sitzung abhängig sind, einschließlich schemaqualifizierter Objekte wie Tabellen, Funktionen, Operatoren oder sogar Erweiterungen.

In RDS für PostgreSQL ist der Parameter `temp_tablespaces` für diesen temporären Arbeitsbereich konfiguriert, in dem die temporären Objekte gespeichert sind.

Die folgenden Abfragen geben den Namen des Tablespace und seinen Speicherort zurück.

```
postgres=> show temp_tablespaces;
temp_tablespaces
---------------------
rds_temp_tablespace
(1 row)
```

Das `rds_temp_tablespace` ist ein von RDS konfigurierter Tablespace, der auf den lokalen Speicher verweist. NVMe Sie können jederzeit zurück zum Amazon EBS-Speicher wechseln, indem Sie diesen Parameter ändern, indem `Parameter group` Sie verwenden, AWS Management Console um auf einen anderen Tablespace als zu verweisen. `rds_temp_tablespace` Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS. Sie können den SET-Befehl auch verwenden, um den Wert des Parameters `temp_tablespaces` mit dem SET-Befehl auf Sitzungsebene in `pg_default` zu ändern. Durch Ändern des Parameters wird der temporäre Arbeitsbereich wieder an Amazon EBS umgeleitet. Der Wechsel zurück zu Amazon EBS hilft, wenn der lokale Speicher für Ihre RDS-Instance oder Ihren Cluster nicht ausreicht, um eine bestimmte SQL-Operation auszuführen.

```
postgres=> SET temp_tablespaces TO 'pg_default';
SET
```

```
postgres=> show temp_tablespaces;

 temp_tablespaces
------------------
 pg_default
```

Anwendungsfälle für RDS Optimized Reads

Im Folgenden sind einige Anwendungsfälle aufgeführt, für die optimierte Lesevorgänge von Vorteil sein können:

- Analytische Abfragen, die Common Table Expressions (CTEs), abgeleitete Tabellen und Gruppierungsoperationen beinhalten.

- Lesereplikate, die die nicht optimierten Abfragen für eine Anwendung verarbeiten.
- Bedarfsgesteuerte oder dynamische Berichtsabfragen mit komplexen Operationen wie GROUP BY und ORDER BY, für die nicht immer die entsprechenden Indizes verwendet werden können.
- Andere Workloads, die interne temporäre Tabellen verwenden.
- `CREATE INDEX`oder `REINDEX` Operationen zum Sortieren.

Bewährte Methoden für RDS Optimized Reads

Nutzen Sie die folgenden bewährten Methoden für RDS Optimized Reads:

- Fügen Sie Wiederholungslogik für schreibgeschützte Abfragen hinzu, falls diese aufgrund eines Fehlers wegen eines vollen Instance-Speichers während der Ausführung fehlschlagen.
- Überwachen Sie anhand der CloudWatch Metrik den verfügbaren Speicherplatz im Instance-Speicher`FreeLocalStorage`. Wenn der Instance-Speicher aufgrund der Workload der DB-Instance oder des Multi-AZ-DB-Clusters sein Limit erreicht, ändern Sie die DB-Instance bzw. den DB-Cluster und verwenden Sie eine größere DB-Instance-Klasse.

Verwenden von RDS Optimized Reads

Wenn Sie eine RDS for PostgreSQL-DB-Instance mit einer der NVMe basierten DB-Instance-Klassen in einer Single-AZ-DB-Instance-Bereitstellung, einer Multi-AZ-DB-Instance-Bereitstellung oder einer Multi-AZ-DB-Cluster-Bereitstellung bereitstellen, verwendet die DB-Instance automatisch RDS-optimierte Lesevorgänge.

Weitere Informationen zur Multi-AZ-Bereitstellung finden Sie unter. Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS

Führen Sie einen der folgenden Schritte aus, um RDS Optimized Reads zu aktivieren:

- Erstellen Sie eine RDS for PostgreSQL-DB-Instance oder einen Multi-AZ-DB-Cluster mithilfe einer der NVMe basierten DB-Instance-Klassen. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Ändern Sie eine bestehende RDS for PostgreSQL-DB-Instance oder einen Multi-AZ-DB-Cluster, um eine der NVMe basierten DB-Instance-Klassen zu verwenden. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

RDS Optimized Reads ist überall verfügbar AWS-Regionen , wo eine oder mehrere DB-Instance-Klassen mit lokalem NVMe SSD-Speicher unterstützt werden. Weitere Informationen finden Sie unter .

Um zu einer Instance mit nicht RDS-optimierten Lesevorgängen zurückzukehren, ändern Sie die DB-Instance-Klasse Ihrer RDS-Instance oder Ihres Clusters auf eine ähnliche Instance-Klasse, die nur EBS-Speicher für Ihre Datenbank-Workloads unterstützt. Wenn die aktuelle DB-Instance-Klasse beispielsweise db.r6gd.4xlarge ist, wählen Sie db.r6g.4xlarge aus, um zurückzuwechseln. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.

Überwachen von DB-Instances, die RDS Optimized Reads verwenden

Sie können DB-Instances, die RDS-optimierte Lesevorgänge verwenden, anhand der folgenden CloudWatch Metriken überwachen:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Diese Metriken liefern Daten über den verfügbaren Instance-Speicher, die IOPS und den Durchsatz. Weitere Informationen zu diesen Metriken finden Sie unter Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS.

Um die aktuelle Nutzung Ihres lokalen Speichers zu überwachen, melden Sie sich bei Ihrer Datenbank an und führen Sie die folgende Abfrage aus:

```
SELECT
    spcname AS "Name",
    pg_catalog.pg_size_pretty(pg_catalog.pg_tablespace_size(oid)) AS "size"
FROM
    pg_catalog.pg_tablespace
WHERE
    spcname IN ('rds_temp_tablespace');
```

Weitere Hinweise zu den temporären Dateien und ihrer Verwendung finden Sie unterVerwalten temporärer Dateien mit PostgreSQL.

Einschränkungen für RDS-optimierte Lesevorgänge in PostgreSQL

Für RDS-optimierte Lesevorgänge gilt die folgende Einschränkung in PostgreSQL:

- Transaktionen können fehlschlagen, wenn der Instance-Speicher voll ist.

Daten in Postgre SQL auf Amazon importieren RDS

Angenommen, Sie haben eine bestehende SQL Postgre-Bereitstellung, die Sie zu Amazon RDS migrieren möchten. Wie komplex diese Aufgabe ist, hängt von der Größe Ihrer Datenbank und den Arten von Datenbankobjekten ab, die Sie übertragen. Nehmen wir beispielsweise eine Datenbank mit mehreren Gigabyte an Datensätzen, einschließlich der gespeicherten Prozeduren und Auslöser. Eine solche Datenbank ist komplizierter als eine einfache Datenbank mit nur wenigen Megabyte an Testdaten und ohne Auslöser oder gespeicherten Prozeduren.

Wir empfehlen, dass Sie native Tools für die Migration von SQL Postgre-Datenbanken unter den folgenden Bedingungen verwenden:

- Sie möchten eine homogene Migration durchführen, bei der die Ausgangsdatenbank denselben Datenbank-Engine hat wie die Zieldatenbank.
- Sie möchten eine komplette Datenbank migrieren.
- Die nativen Tools erlauben Ihnen, Ihr System mit einer minimalen Ausfallzeit zu migrieren.

In den meisten anderen Fällen ist die Durchführung einer Datenbankmigration mit dem AWS Database Migration Service (AWS DMS) der beste Ansatz. AWS DMSkann Datenbanken ohne Ausfallzeiten migrieren und bei vielen Datenbank-Engines die fortlaufende Replikation fortsetzen, bis Sie bereit sind, zur Zieldatenbank zu wechseln. Sie können entweder zu derselben Datenbank-Engine oder zu einer anderen Datenbank-Engine migrieren, indem Sie AWS DMS. Wenn Sie zu einer anderen Datenbank-Engine als Ihrer Quelldatenbank migrieren, können Sie die AWS Schema Conversion Tool (AWS SCT) verwenden. Sie verwenden AWS SCT es, um Schemaobjekte zu migrieren, die nicht von migriert wurden. AWS DMS Weitere Informationen zu finden Sie AWS DMS unter Was ist AWS Database Migration Service?

Modifizieren Sie die DB-Parametergruppe so, dass sie nur die folgenden Einstellungen für Ihren Import enthält. Sie sollten die Parametereinstellungen testen, um festzustellen, welches die effizientesten Einstellungen für Ihre DB-Instance-Größe sind. Nach der Import abgeschlossen wurde, müssen Sie auch diese Parameter auf die Produktionswerte zurücksetzen.

Ändern Sie die Einstellungen für Ihre DB-Instance wie folgt:

- Deaktivieren Sie die DB-Instance-Backups (Sicherungsaufbewahrung auf 0 setzen).
- Deaktivieren Sie Multi-AZ.

Modifizieren Sie die DB-Parametergruppe so, dass sie die folgenden Einstellungen enthält. Sie sollten beim Importieren von Daten nur diese Einstellungen verwenden. Sie sollten die Parametereinstellungen testen, um festzustellen, welches die effizientesten Einstellungen für Ihre DB-Instance-Größe sind. Nach der Import abgeschlossen wurde, müssen Sie auch diese Parameter auf die Produktionswerte zurücksetzen.

Parameter	Empfohlene Werte beim Importieren	Beschreibung
`maintenance_work_mem`	524288, 1048576, 2097152 oder 4194304 (in KB). Diese Einstellungen sind vergleichbar mit 512 MB, 1 GB, 2 GB und 4 GB.	Der Wert für diese Einstellung hängt von der Größe Ihres Hosts ab. Dieser Parameter wird während CREATE INDEX Anweisungen verwendet und jeder parallel Befehl kann so viel Speicher beanspruchen. Berechnen Sie den besten Wert, sodass Sie diesen Wert nicht zu hoch einstellen und Gefahr laufen, keinen Speicherplatz mehr zu haben.
`max_wal_size`	256 (für Version 9.6), 4096 (für Version 10 und höher)	Maximale Größe, damit der bei automatischen Checkpoints WAL wachsen kann. Eine Erhöhung dieses Parameters kann den Zeitaufwand für die Wiederherstellung nach dem Absturz erhöhen. Dieser Parameter ersetzt `checkpoint_segments` Postgre SQL 9.6 und höher. Für SQL Postgre-Version 9.6 ist dieser Wert in 16-MB-Einheiten angegeben. Bei späteren Versionen liegt der Wert bei 1 MB Einheiten . In Version 9.6 bedeutet das beispielsweise 128 Blöcke mit einer Größe von jeweils 16 MB. In Version 12.4 bedeutet das 2 048 Blöcke mit einer Größe von jeweils 1 MB.
`checkpoint_timeout`	1800	Der Wert für diese Einstellung ermöglicht eine seltenere WAL Rotation.

Parameter	Empfohlene Werte beim Importieren	Beschreibung
`synchronous_commit`	Aus	Deaktivieren Sie diese Einstellungen, um Schreibvorgänge zu beschleunigen. Das Deaktivieren dieses Parameters kann das Risiko eines Datenverlusts bei einem Serverabsturz erhöhen (nicht ausschalt enFSYNC).
`wal_buffers`	8192	Dieser Wert wird in 8 KB-Einheiten angegeben . Dies wiederum trägt zur Geschwindigkeit Ihrer WAL Generierung bei
`autovacuum`	0	Deaktivieren Sie den SQL Postgre-Autovakuum -Parameter, während Sie Daten laden, damit er keine Ressourcen verbraucht

Verwenden Sie den Befehl `pg_dump -Fc` (komprimiert) oder den Befehl `pg_restore -j` (parallel) mit diesen Einstellungen.

> **Note**
>
> Der SQL Befehl Postgre `pg_dumpall` erfordert Super_User-Berechtigungen, die beim Erstellen einer DB-Instance nicht gewährt werden, sodass er nicht für den Import von Daten verwendet werden kann.

Themen

- Import einer SQL Postgre-Datenbank aus einer Amazon-Instance EC2
- Verwenden des Befehls\ copy zum Importieren von Daten in eine Tabelle auf einer SQL Postgre-DB-Instance
- Importieren von Amazon S3 in eine DB-Instance von RDS für PostgreSQL
- Transport von SQL Postgre-Datenbanken zwischen

Import einer SQL Postgre-Datenbank aus einer Amazon-Instance EC2

Wenn Sie Daten auf einem SQL Postgre-Server auf einer EC2 Amazon-Instance haben und diese auf eine SQL Postgre-DB-Instance verschieben möchten, können Sie diesen Prozess befolgen, um die Daten zu migrieren.

1. Erstellen Sie eine Datei mit pg_dump, die zu ladenden Daten enthält
2. Erstellen Sie die DB-Ziel-Instance
3. Verwenden Sie psql, um die Datenbank auf der DB-Instance zu erstellen und die Daten zu laden
4. Erstellen Sie einen DB-Snapshot der DB-Instance

In den folgenden Abschnitten finden Sie weitere Informationen zu den einzelnen oben aufgeführten Schritten.

Schritt 1: Erstellen einer Datei, die zu ladenden Daten enthält, mit pg_dump

Das `pg_dump` Hilfsprogramm verwendet den COPY Befehl, um ein Schema und einen Datendump einer Postgre-Datenbank zu erstellen. SQL Das durch `pg_dump` erzeugte Dump-Skript lädt Daten in eine Datenbank mit demselben Namen und erstellt die Tabellen, Indizes und Fremdschlüssel. Sie können den Befehl `pg_restore` und den Parameter `-d` verwenden, um die Daten in einer Datenbank mit einem anderen Namen wiederherzustellen.

Bevor Sie die Daten-Dumpdatei erstellen, sollten Sie die zu dumpenden Tabellen abfragen, um die Zeilenanzahl zu ermitteln und diese in der DB-Ziel-Instance zu bestätigen.

Der folgende Befehl erstellt eine Dump-Datei mit dem Namen mydb2dump.sql für eine Datenbank mit dem Namen mydb2.

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

Schritt 2: Erstellen einer DB-Ziel-Instance

Erstellen Sie die SQL Ziel-Postgre-DB-Instance entweder mit der RDS Amazon-Konsole AWS CLI, oderAPI. Achten Sie darauf, dass beim Erstellen der Instance die Einstellung der Sicherungsaufbewahrung auf 0 festgelegt ist und deaktivieren Sie Multi-AZ. Dadurch wird der Datenimport beschleunigt. Bevor Sie die Daten dumpen können, müssen Sie eine Datenbank auf der Instance erstellen. Die Datenbank kann den gleichen Namen wie die in den Dumpdaten enthaltene Datenbank haben. Alternativ können Sie auch eine Datenbank mit einem anderen Namen erstellen.

In diesem Fall verwenden Sie den Befehl pg_restore und den Parameter -d, um die Daten in einer Datenbank mit einem neuen Namen wiederherzustellen.

Beispielsweise können die folgenden Befehle für das Dumpen, Wiederherstellen und Umbenennen einer Datenbank verwendet werden.

```
pg_dump -Fc -v -h [endpoint of instance] -U [master username] [database]
 > [database].dump
createdb [new database name]
pg_restore -v -h [endpoint of instance] -U [master username] -d [new database
 name] [database].dump
```

Schritt 3: Verwenden von psql zum Erstellen der Datenbank auf der DB-Instance und zum Laden der Daten

Sie können dieselbe Verbindung nutzen, die Sie für die Ausführung des Befehls pg_dump verwendet haben, um eine Verbindung mit der DB-Ziel-Instance herzustellen und die Datenbank neu zu erstellen. Mit psql können Sie den Masterbenutzername und das Masterpasswort verwenden, um die Datenbank auf der DB-Instance zu erstellen

Das folgende Beispiel verwendet psql und eine Dump-Datei namens mydb2dump.sql, um eine Datenbank namens mydb2 auf einer SQL Postgre-DB-Instance namens mypginstance zu erstellen:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
psql \
   -f mydb2dump.sql \
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com \
   --port 8199 \
   --username myawsuser \
   --password password \
   --dbname mydb2
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
psql ^
   -f mydb2dump.sql ^
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com ^
   --port 8199 ^
   --username myawsuser ^
```

```
    --password password ^
    --dbname mydb2
```

> **Note**
>
> Geben Sie aus Sicherheitsgründen ein anderes Passwort als hier angegeben an.

Schritt 4: Erstellen Sie einen DB-Snapshot der DB-Instance

Sobald Sie überprüft haben, dass die Daten in Ihre DB-Instance geladen wurden, empfehlen wir Ihnen, einen DB-Snapshot der SQL Ziel-Postgre-DB-Instance zu erstellen. -DB-Snapshots sind vollständige Backups von Ihrer DB-Instance, die für eine Backup Ihrer DB-Instance auf einen bekannten Zustand verwendet werden können. Ein DB-Snapshot, der sofort nach einem Ladevorgang erstellt wird, bewahrt Sie davor, die Daten bei einem Fehler erneut laden zu müssen. Einen solchen Snapshot können Sie auch als Ausgangspunkt für neue DB-Instances verwenden. Weitere Informationen zum Erstellen eines DB-Snapshots finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

Verwenden des Befehls\ copy zum Importieren von Daten in eine Tabelle auf einer SQL Postgre-DB-Instance

Der SQL `\copy` Befehl Postgre ist ein Metabefehl, der im `psql` interaktiven Client-Tool verfügbar ist. Sie können ihn verwenden`\copy`, um Daten in eine Tabelle auf Ihrer SQL Postgre-DB-Instance RDS zu importieren. Um den `\copy`-Befehl zu verwenden, müssen Sie zuerst die Tabellenstruktur auf der Ziel-DB-Instance erstellen, damit `\copy` ein Ziel für die zu kopierenden Daten hat.

Sie können `\copy` es verwenden, um Daten aus einer Datei mit kommagetrennten Werten (CSV) zu laden, z. B. aus einer Datei, die exportiert und auf Ihrer Client-Workstation gespeichert wurde.

Um die CSV Daten in das Ziel RDS für die SQL Postgre-DB-Instance zu importieren, stellen Sie zunächst eine Verbindung mit der Ziel-DB-Instance her. `psql`

```
psql --host=db-instance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=target-db
```

Führen Sie dann den `\copy`-Befehl mit den folgenden Parametern aus, um das Ziel für die Daten und ihr Format zu identifizieren.

- `target_table`— Der Name der Tabelle, die die aus der CSV Datei kopierten Daten erhalten soll.
- `column_list` – Spaltenspezifikationen für die Tabelle.
- `'filename'`— Der vollständige Pfad zur CSV Datei auf Ihrer lokalen Workstation.

```
\copy target_table from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV;
```

Wenn Ihre CSV Datei Informationen zur Spaltenüberschrift enthält, können Sie diese Version des Befehls und der Parameter verwenden.

```
\copy target_table (column-1, column-2, column-3, ...)
    from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV HEADER;
```

Wenn der `\copy` Befehl fehlschlägt, gibt Postgre SQL Fehlermeldungen aus.

Erstellen einer neuen DB-Instance in der Database Preview-Umgebung mithilfe eines `psql` Befehls mit dem `\copy` Meta-Befehl, wie in den folgenden Beispielen gezeigt. In diesem Beispiel wird source-table als Name für die Quelltabelle verwendet, source-table.csv für die CSV-Datei und target-db für die Zieldatenbank:

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
$psql target-db \
    -U <admin user> \
    -p <port> \
    -h <DB instance name> \
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
$psql target-db ^
    -U <admin user> ^
    -p <port> ^
    -h <DB instance name> ^
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Vollständige Informationen zum `\copy` Befehl finden Sie auf der psql-Seite in der SQL Postgre-Dokumentation im Abschnitt Meta-Befehle.

Importieren von Amazon S3 in eine DB-Instance von RDS für PostgreSQL

Sie können Daten, die mit Amazon Simple Storage Service gespeichert wurden, in eine Tabelle auf einer RDS für PostgreSQL DB-Instance importieren. Um dies zu tun, installieren Sie zuerst die `aws_s3`-Erweiterung von RDS für PostgreSQL. Diese Erweiterung stellt die Funktionen bereit, die Sie zum Importieren von einem Amazon S3 Bucket verwenden. Ein Bucket ist ein Amazon S3 Container für Objekte und Dateien. Die Daten können sich in einer Datei mit kommagetrennten Werten (CSV), einer Textdatei oder einer komprimierten Datei (GZIP) befinden. Im Folgenden erfahren Sie, wie Sie die Erweiterung installieren und Daten aus Amazon S3 in eine Tabelle importieren.

Um von Simple Storage Service (Amazon S3) in RDS for PostgreSQL zu importieren, muss Ihre Datenbank PostgreSQL Version 10.7 oder höher verwenden.

Wenn Sie keine Daten in Amazon S3 gespeichert haben, müssen Sie zunächst einen Bucket erstellen und die Daten speichern. Weitere Informationen finden Sie in den folgenden Themen im Benutzerhandbuch zum Amazon Simple Storage Service.

- Erstellen Sie einen Bucket
- Hinzufügen eines Objekts zu einem Bucket

Das kontoübergreifende Importieren aus Amazon S3 wird unterstützt. Weitere Informationen finden Sie unter Gewähren kontoübergreifender Berechtigungen im Benutzerhandbuch zu Amazon Simple Storage Service.

Sie können den vom Kunden verwalteten Schlüssel für die Verschlüsselung verwenden, wenn Sie Daten aus S3 importieren. Weitere Informationen finden Sie unter In AWS KMS gespeicherte KMS-Schlüssel im Benutzerhandbuch zu Amazon Simple Storage Service.

Themen

- Installieren der Erweiterung aws_s3
- Übersicht über den Import von Daten aus Amazon S3-Daten
- Einrichten des Zugriffs auf einen Amazon S3-Bucket
- Importieren von Daten aus Amazon S3 in Ihren RDS für PostgreSQL DB-Instance
- Funktionsreferenz

Installieren der Erweiterung aws_s3

Bevor Sie Amazon S3 mit Ihrem für die SQL Postgre-DB-Instance verwenden können, müssen Sie die `aws_s3` Erweiterung installieren. Diese Erweiterung bietet Funktionen zum Importieren von Daten aus einem Amazon S3. Es bietet auch Funktionen für den Export von Daten aus und RDSfür eine SQL Postgre-DB-Instance in einen Amazon S3 S3-Bucket. Weitere Informationen finden Sie unter Exportieren von Daten aus einem/einer RDS for PostgreSQL-DB-Instance zu Amazon S3. Die Erweiterung `aws_s3` hängt von einigen Hilfsfunktionen in der Erweiterung `aws_commons` ab, die bei Bedarf automatisch installiert wird.

So installieren Sie die Erweiterung **`aws_s3`**

1. Verwenden Sie psql (oderpgAdmin), um als Benutzer mit Rechten eine Verbindung zur für SQL Postgre-DB-Instance herzustellen. `rds_superuser` Wenn Sie beim Einrichten den Standardnamen beibehalten haben, stellen Sie eine Verbindung als `postgres` her.

   ```
   psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Führen Sie den folgenden Befehl aus, um die Erweiterung zu installieren.

   ```
   postgres=> CREATE EXTENSION aws_s3 CASCADE;
   NOTICE: installing required extension "aws_commons"
   CREATE EXTENSION
   ```

3. Wenn Sie überprüfen möchten, ob die Erweiterung installiert wurde, können Sie psql-Metabefehl `\dx` verwenden.

   ```
   postgres=> \dx
          List of installed extensions
       Name     | Version |   Schema   |                 Description
   -------------+---------+------------+---------------------------------------------
    aws_commons | 1.2     | public     | Common data types across AWS services
    aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
    plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
   (3 rows)
   ```

Die Funktionen zum Importieren von Daten aus Amazon S3 und Exportieren von Daten nach Amazon S3 sind jetzt verfügbar.

Übersicht über den Import von Daten aus Amazon S3-Daten

S3-Daten in Amazon RDS importieren

Sammeln Sie zunächst die Details, die Sie der Funktion zur Verfügung stellen müssen. Dazu gehören der Name der Tabelle auf der Ihre RDS for PostgreSQL-DB-Instance sowie der Bucket-Name, der Dateipfad, der Dateityp und der AWS-Region Speicherort der Amazon S3 S3-Daten. Weitere Informationen finden Sie unter Kopieren von Objekten im Benutzerhandbuch zu Amazon Simple Storage Service.

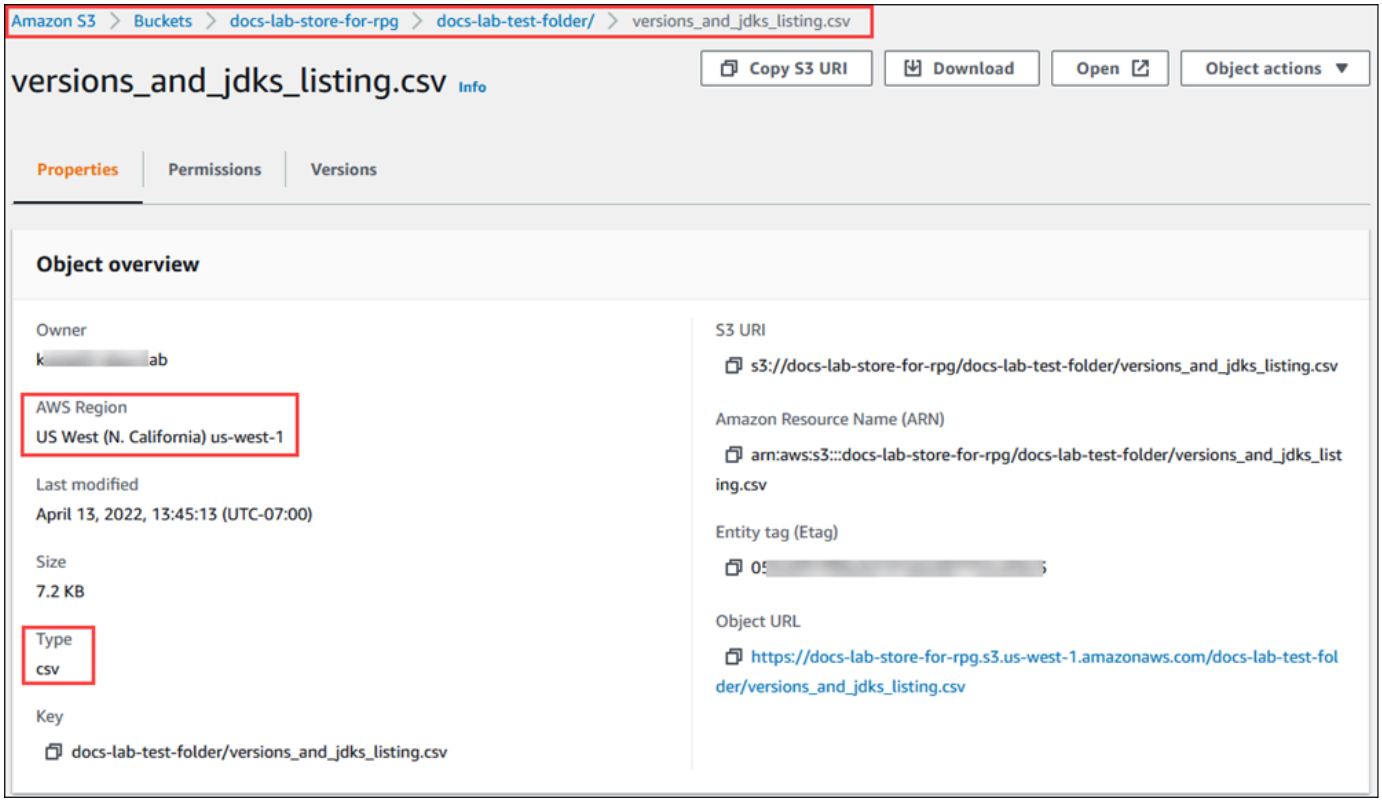
> **Note**
>
> Der mehrteilige Datenimport aus Amazon S3 wird derzeit nicht unterstützt.

1. Ermittelt den Namen der Tabelle, in die die `aws_s3.table_import_from_s3`-Funktion die Daten importieren soll. Mit dem folgenden Befehl wird beispielsweise eine Tabelle `t1` erstellt, die in späteren Schritten verwendet werden kann.

   ```
   postgres=> CREATE TABLE t1
       (col1 varchar(80),
       col2 varchar(80),
       col3 varchar(80));
   ```

2. Rufen Sie die Details zum Amazon-S3-Bucket und die zu importierenden Daten ab. Öffnen Sie dazu die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/und wählen Sie Buckets. Suchen Sie den Bucket, der Ihre Daten enthält, in der Liste. Wählen Sie den Bucket aus, öffnen Sie die Seite Objektübersicht und wählen Sie dann Properties (Eigenschaften).

   Notieren Sie sich den Namen, den Pfad AWS-Region, den und den Dateityp des Buckets. Sie benötigen den Amazon-Ressourcenname (ARN) später, um den Zugriff auf Amazon S3 über eine IAM-Rolle einzurichten. Weitere Informationen finden Sie unter Einrichten des Zugriffs auf einen Amazon S3-Bucket. In der folgenden Abbildung sehen Sie ein Beispiel.



3. Sie können den Pfad zu den Daten im Amazon S3 S3-Bucket überprüfen, indem Sie den AWS CLI Befehl verwenden`aws s3 cp`. Wenn die Informationen korrekt sind, lädt dieser Befehl eine Kopie der Amazon S3-Datei herunter.

```
aws s3 cp s3://amzn-s3-demo-bucket/sample_file_path ./
```

4. Richten Sie Berechtigungen auf Ihrer DB-Instance von RDS für PostgreSQL ein, um den Zugriff auf die Datei im Amazon-S3-Bucket zu gestatten. Dazu verwenden Sie entweder eine AWS Identity and Access Management (IAM-) Rolle oder Sicherheitsanmeldedaten. Weitere Informationen finden Sie unter Einrichten des Zugriffs auf einen Amazon S3-Bucket.

5. Geben Sie den Pfad und andere gesammelte Amazon S3-Objektdetails (siehe Schritt 2) an die `create_s3_uri`-Funktion zum Erstellen eines Amazon S3-URI-Objekts. Weitere Informationen zu dieser Funktion finden Sie unter aws_commons.create_s3_uri. Es folgt ein Beispiel für die Erstellung dieses Objekts während einer psql-Sitzung.

```
postgres=> SELECT aws_commons.create_s3_uri(
   'docs-lab-store-for-rpg',
   'versions_and_jdks_listing.csv',
   'us-west-1'
) AS s3_uri \gset
```

Im nächsten Schritt übergeben Sie dieses Objekt (`aws_commons._s3_uri_1`) an die `aws_s3.table_import_from_s3`-Funktion, um die Daten in die Tabelle zu importieren.

6. Rufen Sie die `aws_s3.table_import_from_s3`-Funktion zum Importieren der Daten aus Amazon S3 in Ihre Tabelle auf. Referenz-Informationen finden Sie unter aws_s3.table_import_from_s3. Beispiele finden Sie unter Importieren von Daten aus Amazon S3 in Ihren RDS für PostgreSQL DB-Instance.

Einrichten des Zugriffs auf einen Amazon S3-Bucket

Um Daten aus einer Amazon S3-Datei zu importieren, erteilen Sie dem/der RDS for PostgreSQL-DB-Instance die Berechtigung, auf den Amazon S3-Bucket zuzugreifen, in dem sich die Datei befindet. Sie können den Zugriff auf einen Amazon S3-Bucket auf zwei Arten erlaubt, wie in den folgenden Themen beschrieben.

Themen

- Verwenden einer IAM-Rolle für den Zugriff auf einen Amazon S3-Bucket
- Verwenden von Sicherheitsanmeldeinformationen für den Zugriff auf einen Amazon S3-Bucket
- Fehlerbehebung beim Zugriff auf Amazon S3

Verwenden einer IAM-Rolle für den Zugriff auf einen Amazon S3-Bucket

Bevor Sie Daten aus einer Amazon S3-Datei laden, geben Sie Ihrer RDS for PostgreSQL DB-Instance die Berechtigung, auf den Amazon S3-Bucket der Datei zuzugreifen. Auf diese Weise müssen Sie keine zusätzlichen Anmeldeinformationen verwalten oder im aws_s3.table_import_from_s3-Funktionsaufruf angeben.

Erstellen Sie dazu eine IAM-Richtlinie, die den Zugriff auf den Amazon S3-Bucket ermöglicht. Erstellen Sie eine IAM-Rolle und hängen Sie die Richtlinie an die Rolle an. Weisen Sie dann die IAM-Rolle Ihrer DB Instance zu.

Einer RDS-for-PostgreSQL-DB-Instance über eine IAM-Rolle Zugriff auf Amazon S3 gewähren

1. Erstellen Sie eine IAM-Richtlinie.

   Diese Richtlinie enthält die Bucket- und Objektberechtigungen, die Ihrer RDS for PostgreSQL DB-Instance den Zugriff auf Amazon S3 ermöglichen.

Nehmen Sie die folgenden erforderlichen Aktionen in die Richtlinie auf, um die Übertragung von Dateien von einem Amazon S3-Bucket nach Amazon RDS zu ermöglichen:

- `s3:GetObject`
- `s3:ListBucket`

Nehmen Sie die folgenden Ressourcen in die Richtlinie auf, um den Amazon S3-Bucket und Objekte im Bucket zu identifizieren. Dies zeigt das Amazon Resource Name (ARN) Format für den Zugriff auf Amazon S3 an.

- arn:aws:s3::: *amzn-s3-demo-bucket*
- arn:aws:s3::: /* *amzn-s3-demo-bucket*

Weitere Informationen zum Erstellen einer IAM-Richtlinie für RDS für PostgreSQL finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff. Siehe auch Tutorial: Erstellen und Anfügen Ihrer ersten vom Kunden verwalteten Richtlinie im IAM-Benutzerhandbuch.

Der folgende AWS CLI Befehl erstellt eine IAM-Richtlinie mit diesen Optionen. `rds-s3-import-policy` Sie gewährt Zugriff auf einen Bucket namens *amzn-s3-demo-bucket*.

> **Note**
>
> Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Richtlinie, der vom Befehl zurückgegeben wurde. Sie benötigen den ARN in einem nachfolgenden Schritt, in dem Sie die Richtlinie an eine IAM-Rolle anhängen.

Example

Für LinuxmacOS, oderUnix:

```
aws iam create-policy \
   --policy-name rds-s3-import-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
```

```
      {
        "Sid": "s3import",
        "Action": [
          "s3:GetObject",
          "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::amzn-s3-demo-bucket",
          "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ]
      }
    ]
  }'
```

Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-import-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3import",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket",
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

2. Erstellen Sie eine IAM-Rolle.

   Sie tun dies, damit Amazon RDS in Ihrem Namen diese IAM-Rolle übernehmen kann, um auf Ihre Amazon S3-Buckets zuzugreifen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.

Wir empfehlen die Verwendung von `aws:SourceArn` und `aws:SourceAccount` globaler Bedingungskontext-Schlüssel in ressourcenbasierten Richtlinien, um die Berechtigungen des Dienstes auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Wenn Sie sowohl globale Kontextschlüssel nutzen und der `aws:SourceArn`-Wert enthält die Konto-ID, muss der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert die gleiche Konto-ID verwenden, wenn er in der gleichen Richtlinienanweisung verwendet wird.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Verwenden Sie in der Richtlinie den `aws:SourceArn` globalen Kontextschlüssel mit dem vollständigen ARN der Ressource. Das folgende Beispiel zeigt, wie Sie dazu den AWS CLI Befehl verwenden, um eine Rolle mit dem Namen zu erstellen`rds-s3-import-role`.

Example

Für LinuxmacOS, oderUnix:

```
aws iam create-role \
   --role-name rds-s3-import-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
```

```
    ]
  }'
```

Windows:

```
aws iam create-role ^
   --role-name rds-s3-import-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

3. Fügen Sie die erstellte IAM-Richtlinie der IAM-Rolle an, die Sie erstellt haben.

   Mit dem folgenden AWS CLI Befehl wird die im vorherigen Schritt erstellte Richtlinie der Rolle `rds-s3-import-role` Replace *`your-policy-arn`* mit dem Richtlinien-ARN zugeordnet, den Sie in einem früheren Schritt notiert haben.

   Example

   Für LinuxmacOS, oderUnix:

```
aws iam attach-role-policy \
   --policy-arn your-policy-arn \
   --role-name rds-s3-import-role
```

   Windows:

```
aws iam attach-role-policy ^
   --policy-arn your-policy-arn ^
   --role-name rds-s3-import-role
```

4. Fügen Sie die IAM-Rolle der DB Instance hinzu.

   Sie tun dies, indem Sie das AWS Management Console oder verwenden AWS CLI, wie im Folgenden beschrieben.

Konsole

So fügen Sie eine IAM-Rolle für eine PostgreSQL DB--Instance über die Konsole hinzu:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie den Namen der PostgreSQL DB--Instance aus, um ihre Details anzuzeigen.
3. Wählen Sie auf der Registerkarte Connectivity & Security im Bereich Manage IAM roles (IAM-Rollen verwalten) die Rolle aus, die unter Add IAM roles (IAM-Rollen hinzufügen) zu diesen Instances hinzugefügt werden soll.
4. Wählen Sie unter Feature (Funktion) die Option s3Import aus.
5. Wählen Sie Rolle hinzufügen.

AWS CLI

So fügen Sie eine IAM-Rolle für eine PostgreSQL-DB-Instance mithilfe der CLI hinzu

- Verwenden Sie den folgenden CLI-Befehl, um die IAM-Rolle zur RDS for PostgreSQL DB-Instance mit dem Namen `my-db-instance` hinzuzufügen. Ersetzen Sie *`your-role-arn`* durch den Rollen-ARN, den Sie im vorherigen Schritt notiert haben. Verwenden Sie `s3Import` für den Wert der `--feature-name`-Option.

  Example

  Für LinuxmacOS, oderUnix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier my-db-instance \
   --feature-name s3Import \
```

```
    --role-arn your-role-arn   \
    --region your-region
```

Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier my-db-instance ^
    --feature-name s3Import ^
    --role-arn your-role-arn ^
    --region your-region
```

RDS-API

Um eine IAM-Rolle für eine mithilfe der Amazon RDS-API hinzuzufügen, rufen Sie den Vorgang auf.

Verwenden von Sicherheitsanmeldeinformationen für den Zugriff auf einen Amazon S3-Bucket

Wenn Sie es vorziehen, können Sie Sicherheitsanmeldeinformationen verwenden, um den Zugriff auf einen Amazon S3-Bucket zu ermöglichen, anstatt den Zugriff mit einer IAM-Rolle zu ermöglichen. Dazu geben Sie die `credentials`-Parameter im aws_s3.table_import_from_s3-Funktionsaufruf an.

Der `credentials` Parameter ist eine Struktur vom Typ`aws_commons._aws_credentials_1`, die Anmeldeinformationen enthält. AWS Verwenden Sie die Funktion aws_commons.create_aws_credentials, um den Zugriffs- und den Geheimschlüssel in einer `aws_commons._aws_credentials_1`-Struktur festzulegen, wie nachfolgend dargestellt.

```
postgres=> SELECT aws_commons.create_aws_credentials(
   'sample_access_key', 'sample_secret_key', '')
AS creds \gset
```

Nachdem Sie die `aws_commons._aws_credentials_1` -Struktur erstellt haben, verwenden Sie die Funktion aws_s3.table_import_from_s3 mit dem Parameter `credentials`, um die Daten zu importieren, wie nachfolgend gezeigt.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   :'creds'
);
```

Sie können auch den Funktionsaufruf aws_commons.create_aws_credentials in den Funktionsaufruf `aws_s3.table_import_from_s3` einbinden.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')
);
```

Fehlerbehebung beim Zugriff auf Amazon S3

Wenn Sie beim Versuch, Daten aus Amazon S3 zu importieren, auf Verbindungsprobleme stoßen, finden Sie im Folgenden Empfehlungen:

- Fehlerbehebung für Amazon RDS-Identität und -Zugriff
- Fehlerbehebung bei Amazon S3 im Entwicklerhandbuch für Amazon Simple Storage Service
- Fehlerbehebung bei Amazon S3 und IAM im IAM-Benutzerhandbuch

Importieren von Daten aus Amazon S3 in Ihren RDS für PostgreSQL DB-Instance

Sie importieren Daten aus Ihrem Amazon-S3-Bucket mithilfe der `table_import_from_s3`-Funktion der aws_s3-Erweiterung. Referenz-Informationen finden Sie unter aws_s3.table_import_from_s3.

> **Note**
>
> Die folgenden Beispiele verwenden die IAM-Rollen-Methode, um den Zugriff auf den Amazon-S3-Bucket zu ermöglichen. Daher enthalten die `aws_s3.table_import_from_s3`-Funktionsaufrufe keine Berechtigungsnachweisparameter.

Nachfolgend ist ein typisches Beispiel aufgeführt.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't1',
   '',
   '(format csv)',
   :'s3_uri'
);
```

Es werden folgende Parameter verwendet:

- `t1` – Der Name für die Tabelle in der PostgreSQL DB Instance, in die die Daten kopiert werden.
- `''` – Eine optionale Liste mit Spalten in der Datenbanktabelle. Mithilfe dieses Parameters können Sie angeben, welche Spalten der S3-Daten in die Tabellenspalten übernommen werden sollen. Wenn keine Spalten angegeben sind, werden alle Spalten in die Tabelle kopiert. Ein Beispiel zum Verwenden einer Spaltenliste finden Sie unter Importieren einer Amazon S3-Datei, die ein benutzerdefiniertes Trennzeichen verwendet.
- `(format csv)` – PostgreSQL COPY-Argumente. Der Kopiervorgang verwendet die Argumente und das Format des PostgreSQL-Befehls COPY, um die Daten zu importieren. Zu den Optionen für das Format gehören kommagetrennte Werte (CSV), Text und Binärwerte. Der Standard ist Text.
- `s3_uri` – Eine Struktur mit den Informationen zum Identifizieren der Amazon S3-Datei. Ein Beispiel für die Verwendung der Funktion aws_commons.create_s3_uri zum Erstellen einer `s3_uri`-Struktur finden Sie unter Übersicht über den Import von Daten aus Amazon S3-Daten.

Weitere Informationen zu dieser Funktion finden Sie unter aws_s3.table_import_from_s3.

Die Funktion gibt `aws_s3.table_import_from_s3` zurück. Weitere Informationen zum Angeben von anderen Dateien für den Import aus einem Amazon S3-Bucket finden Sie in einem der folgenden Beispiele.

> **Note**
>
> Beim Importieren einer Datei mit 0 Byte tritt ein Fehler auf.

Themen

- Importieren einer Amazon S3-Datei, die ein benutzerdefiniertes Trennzeichen verwendet
- Importieren einer Amazon S3-komprimierten Datei (gzip)
- Importieren einer kodierten Amazon S3-Datei

Importieren einer Amazon S3-Datei, die ein benutzerdefiniertes Trennzeichen verwendet

Das folgende Beispiel zeigt, wie man eine Datei importiert, die ein benutzerdefiniertes Trennzeichen verwendet. Außerdem wird veranschaulicht, wie mit dem `column_list`-Parameter der Funktion aws_s3.table_import_from_s3 kontrolliert wird, wo die Daten in der Datenbanktabelle platziert werden.

Für dieses Beispiel wird angenommen, dass die folgenden Informationen in durch Pipe-Zeichen getrennte Spalten in der Amazon S3-Datei angeordnet sind.

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
...
```

So importieren Sie eine Datei, die ein benutzerdefiniertes Trennzeichen verwendet:

1. Erstellen Sie eine Tabelle in der Datenbank für die importierten Daten.

   ```
   postgres=> CREATE TABLE test (a text, b text, c text, d text, e text);
   ```

2. Verwenden Sie die folgende Form der Funktion aws_s3.table_import_from_s3, um Daten aus der Amazon S3-Datei zu importieren.

   Zur Angabe der Datei können Sie auch den Funktionsaufruf aws_commons.create_s3_uri in den Funktionsaufruf `aws_s3.table_import_from_s3` einbinden.

   ```
   postgres=> SELECT aws_s3.table_import_from_s3(
      'test',
      'a,b,d,e',
      'DELIMITER ''|''',
      aws_commons.create_s3_uri('amzn-s3-demo-bucket', 'pipeDelimitedSampleFile', 'us-
   east-2')
   );
   ```

Die Daten befinden sich nun in den folgenden Spalten der Tabelle.

```
postgres=> SELECT * FROM test;
a | b | c | d | e
---+------+---+---+------+-----------
1 | foo1 | | bar1 | elephant1
2 | foo2 | | bar2 | elephant2
3 | foo3 | | bar3 | elephant3
4 | foo4 | | bar4 | elephant4
```

Importieren einer Amazon S3-komprimierten Datei (gzip)

Das folgende Beispiel zeigt, wie eine mit gzip komprimierte Datei aus Amazon S3 importiert wird. Die Datei, die Sie importieren, muss die folgenden Amazon-S3-Metadaten aufweisen:

- Schlüssel: `Content-Encoding`
- Wert: `gzip`

Wenn Sie die Datei mit dem hochladen AWS Management Console, werden die Metadaten in der Regel vom System übernommen. Informationen zum Hochladen von Dateien auf Amazon S3 mithilfe der AWS Management Console AWS CLI, der oder der API finden Sie unter Hochladen von Objekten im Amazon Simple Storage Service-Benutzerhandbuch.

Weitere Informationen zu Amazon-S3-Metadaten und zu vom System bereitgestellten Metadaten finden Sie unter Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole im Benutzerhandbuch für Amazon Simple Storage Service.

Importieren Sie die gzip-Datei folgendermaßen in Ihrer -RDS for PostgreSQL DB-Instance.

```
postgres=> CREATE TABLE test_gzip(id int, a text, b text, c text, d text);
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_gzip', '', '(format csv)',
 'amzn-s3-demo-bucket', 'test-data.gz', 'us-east-2'
);
```

Importieren einer kodierten Amazon S3-Datei

Das folgende Beispiel zeigt, wie eine Datei aus Amazon S3 mit Windows-1252-Kodierung importiert wird.

```
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_table', '', 'encoding ''WIN1252''',
 aws_commons.create_s3_uri('amzn-s3-demo-bucket', 'SampleFile', 'us-east-2')
);
```

Funktionsreferenz

Funktionen

- aws_s3.table_import_from_s3

- aws_commons.create_s3_uri
- aws_commons.create_aws_credentials

aws_s3.table_import_from_s3

Importiert Amazon S3 S3-Daten in eine . Die Erweiterung `aws_s3` stellt die Funktion `aws_s3.table_import_from_s3` bereit. Der Rückgabewert ist Text.

Syntax

Die erforderlichen Parameter sind `table_name`, `column_list` und `options`. Diese Parameter identifizieren die Datenbanktabelle und geben an, wie die Daten in die Tabelle kopiert werden.

Sie können auch die folgenden Parameter verwenden:

- Die zu importierende Amazon S3-Datei wird mit dem Parameter `s3_info` übergeben. Wenn Sie diesen Parameter verwenden, wird der Zugriff auf Amazon S3 durch eine IAM Rolle für die SQL bereitgestellt.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1
)
```

- Die Anmeldeinformationen für den Zugriff auf Amazon S3 werden mit dem Parameter `credentials` übergeben. Wenn Sie diesen Parameter verwenden, verwenden Sie keine IAM Rolle.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1,
   credentials aws_commons._aws_credentials_1
)
```

Parameter

table_name

Eine erforderliche Textzeichenfolge, die den Namen der SQL Postgre-Datenbanktabelle enthält, in die die Daten importiert werden sollen.

column_list

Eine erforderliche Textzeichenfolge, die eine optionale Liste der Spalten der SQL Postgre-Datenbanktabelle enthält, in die die Daten kopiert werden sollen. Wenn die Zeichenfolge leer ist, werden alle Spalten der Tabelle verwendet. Ein Beispiel finden Sie unter Importieren einer Amazon S3-Datei, die ein benutzerdefiniertes Trennzeichen verwendet.

options

Eine erforderliche Textzeichenfolge mit Argumenten für den Postgre-Befehl. SQL `COPY` Diese Argumente geben an, wie die Daten in die SQL Postgre-Tabelle kopiert werden sollen. Weitere Informationen finden Sie in der Postgre-Dokumentation. SQL COPY

s3_info

Ein zusammengesetzter `aws_commons._s3_uri_1`-Typ mit den folgenden Informationen zum S3-Objekt:

- `bucket` – Der Name des Amazon S3-Buckets, der die Datei enthält.
- `file_path` – Der Amazon S3-Dateiname einschließlich des Pfads der Datei.
- `region`— Das AWS Region, in der sich die Datei befindet. Für eine Liste von AWS Regionsnamen und zugehörige Werte finden Sie unterRegionen, Availability Zones und Local Zones.

Anmeldedaten

Ein zusammengesetzter `aws_commons._aws_credentials_1`-Typ mit den folgenden Anmeldeinformationen, die für den Importvorgang verwendet werden sollen:

- Zugriffsschlüssel
- Geheimschlüssel
- Sitzungs-Token

Hinweise zum Erstellen einer zusammengesetzten `aws_commons._aws_credentials_1`-Struktur finden Sie unter aws_commons.create_aws_credentials.

Alternative Syntax

Zum Testen können Sie statt der Parameter `s3_info` und `credentials` eine erweiterte Gruppe von Parametern verwenden. Nachfolgend sind weitere Syntaxvariationen für die Funktion `aws_s3.table_import_from_s3` aufgeführt:

- Statt den Parameter `s3_info` zum Identifizieren einer Amazon S3-Datei zu verwenden, nutzen Sie die Kombination aus den Parametern `bucket`, `file_path` und `region`. Bei dieser Form der Funktion wird der Zugriff auf Amazon S3 über eine IAM Rolle in der SQL Postgre-DB-Instance bereitgestellt.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text
)
```

- Statt den Parameter `credentials` zum Angeben einer Amazon S3-Datei zu verwenden, nutzen Sie die Kombination aus den Parametern `access_key`, `session_key` und `session_token`.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text,
   access_key text,
   secret_key text,
   session_token text
)
```

Alternative Parameter

bucket

Eine Textzeichenfolge mit den Namen des Amazon S3-Buckets, der die Datei enthält.

file_path

Eine Textzeichenfolge, die den Amazon S3-Dateinamen einschließlich des Pfades der Datei enthält.

Region

Eine Textzeichenfolge, die identifiziert AWS-Region Speicherort der Datei. Für eine Liste von AWS-Region Namen und zugehörige Werte finden Sie unterRegionen, Availability Zones und Local Zones.

access_key

Eine Textzeichenfolge mit dem Zugriffsschlüssel, der für den Importvorgang verwendet werden soll. Die Standardeinstellung istNULL.

secret_key

Eine Textzeichenfolge mit dem Geheimschlüssel, der für den Importvorgang verwendet werden soll. Die Standardeinstellung istNULL.

session_token

(Optional) Eine Textzeichenfolge mit dem Sitzungsschlüssel, der für den Importvorgang verwendet werden soll. Die Standardeinstellung istNULL.

aws_commons.create_s3_uri

Erstellt eine `aws_commons._s3_uri_1`-Struktur für die Amazon S3-Dateiinformationen. Die Ergebnisse der Funktion `aws_commons.create_s3_uri` werden im Parameter `s3_info` der Funktion aws_s3.table_import_from_s3 verwendet.

Syntax

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Parameter

bucket

Eine erforderliche Textzeichenfolge mit dem Namen des Amazon S3-Buckets für die Datei.

file_path

Eine erforderliche Textzeichenfolge, die den Amazon S3-Dateinamen einschließlich des Pfads der Datei enthält.

Region

Eine erforderliche Textzeichenfolge, die Folgendes enthält AWS-Region in dem sich die Datei befindet. Für eine Liste von AWS-Region Namen und zugehörige Werte finden Sie unterRegionen, Availability Zones und Local Zones.

aws_commons.create_aws_credentials

Legt einen Zugriffs- und einen Geheimschlüssel in einer `aws_commons._aws_credentials_1`-Struktur fest. Die Ergebnisse der Funktion `aws_commons.create_aws_credentials` werden im Parameter `credentials` der Funktion aws_s3.table_import_from_s3 verwendet.

Syntax

```
aws_commons.create_aws_credentials(
   access_key text,
   secret_key text,
   session_token text
)
```

Parameter

access_key

Eine erforderliche Textzeichenfolge mit dem Zugriffsschlüssel, der zum Importieren einer Amazon S3-Datei verwendet werden soll. Die Standardeinstellung istNULL.

secret_key

Eine erforderliche Textzeichenfolge mit dem Geheimschlüssel, der zum Importieren einer Amazon S3-Datei verwendet werden soll. Die Standardeinstellung istNULL.

session_token

Eine erforderliche Textzeichenfolge mit dem Sitzungs-Token, der zum Importieren einer Amazon S3-Datei verwendet werden soll. Die Standardeinstellung istNULL. Wenn Sie ein optionales `session_token` angeben, können Sie temporäre Anmeldeinformationen verwenden.

Transport von SQL Postgre-Datenbanken zwischen

Durch die Verwendung SQL übertragbarer Postgre-Datenbanken für Amazon RDS können Sie eine SQL Postgre-Datenbank zwischen zwei DB-Instances verschieben. Dies ist eine sehr schnelle Möglichkeit, große Datenbanken zwischen verschiedenen DB-Instances zu migrieren. Um diesen Ansatz zu verwenden, müssen Ihre DB-Instances beide dieselbe Hauptversion von Postgre ausführen. SQL

Diese Funktion erfordert, dass Sie die `pg_transport`-Erweiterung sowohl für die Quell- als auch die Ziel-DB-Instance installieren. Die `pg_transport`-Erweiterung bietet einen physischen Transportmechanismus, der die Datenbankdateien mit minimaler Verarbeitung verschiebt. Dieser Mechanismus bewegt Daten viel schneller als herkömmliche Abfragen- und Ladeprozesse mit weniger Ausfallzeiten.

> **Note**
>
> SQLPortable Postgre-Datenbanken sind RDS für Postgre SQL 11.5 und höher sowie für Postgre Version 10.10 und RDS höher verfügbar. SQL

Um eine SQL Postgre-DB-Instance von einer RDS für Postgre SQL DB-Instance zu einer anderen zu transportieren, richten Sie zunächst die Quell- und Ziel-Instances ein, wie unter beschrieben. Einrichten einer DB-Instance für den Transport Anschließend können Sie die Datenbank mit der unter Transport einer Postgre-Datenbank SQL beschriebenen Funktion transportieren.

Themen

- Was passiert beim Datenbanktransport?
- Einschränkungen bei der Verwendung von portablen SQL Postgre-Datenbanken
- Einrichtung für den Transport einer Postgre-Datenbank SQL
- Transport einer SQL Postgre-Datenbank von der Quelle zum Ziel
- Funktionsreferenz für transportable Datenbanken

- Parameterreferenz für transportable Datenbanken

Was passiert beim Datenbanktransport?

Die Funktion für SQL transportable Postgre-Datenbanken verwendet ein Pull-Modell, um die Datenbank von der Quell-DB-Instance in das Ziel zu importieren. Die `transport.import_from_server`-Funktion erstellt die In-Transit-Datenbank auf der Ziel-DB-Instance. Die In-Transit-Datenbank ist auf der Ziel-DB-Instance für die Dauer des Transports nicht zugänglich.

Wenn der Transport beginnt, werden alle aktuellen Sitzungen auf der Quelldatenbank beendet. Alle anderen Datenbanken als die Quelldatenbank auf der Quell-DB-Instance sind vom Transport nicht betroffen.

Die Quelldatenbank wird in einen speziellen schreibgeschützten Modus versetzt. Während es sich in diesem Modus befindet, können Sie sich mit der Quelldatenbank verbinden und schreibgeschützte Abfragen durchführen. Allerdings sind schreibende Abfragen und einige andere Arten von Befehlen blockiert. Von diesen Einschränkungen ist nur die spezifische Quelldatenbank betroffen, die transportiert wird.

Während des Transports können Sie für die Ziel-DB-Instance keine Point-in-time-Wiederherstellung durchführen. Dies liegt daran, dass der Transport nicht transaktional ist und das SQL Postgre-Write-Ahead-Protokoll nicht zum Aufzeichnen von Änderungen verwendet wird. Wenn in der Ziel-DB-Instance das automatische Backup aktiviert ist, wird nach Abschluss des Transports automatisch ein Backup durchgeführt. Point-in-timeWiederherstellungen sind für Zeiten nach Abschluss des Backups verfügbar.

Wenn der Transport fehlschlägt, versucht die `pg_transport`-Erweiterung, alle Änderungen an den Quell- und Ziel-DB-Instances rückgängig zu machen. Dazu gehört auch das Entfernen der teilweise transportierten Datenbank auf dem Ziel. Abhängig von der Art des Fehlers kann es vorkommen, dass die Quelldatenbank weiterhin schreibende Abfragen ablehnt. Verwenden Sie in diesem Fall den folgenden Befehl, um schreibende Abfragen zu ermöglichen.

```
ALTER DATABASE db-name SET default_transaction_read_only = false;
```

Einschränkungen bei der Verwendung von portablen SQL Postgre-Datenbanken

Transportdatenbanken haben folgende Einschränkungen:

- Lesereplikate – Sie können keine Transportdatenbanken für Lesereplikate oder übergeordnete Instances von Lesereplikaten verwenden.
- Nicht unterstützte Spaltentypen – Sie können die `reg`-Datentypen nicht in Datenbanktabellen verwenden, die Sie mit dieser Methode transportieren möchten. Diese Typen hängen von den Systemkatalogobjekten IDs (OIDs) ab, die sich während des Transports häufig ändern.
- Tablespaces – Alle Quelldatenbankobjekte müssen sich im Standard-Tablespace `pg_default` befinden.
- Kompatibilität — Sowohl die Quell- als auch die Ziel-DB-Instance müssen dieselbe Hauptversion von Postgre SQL ausführen.
- Erweiterungen – Auf der Quell-DB-Instance kann nur das `pg_transport` installiert sein.
- Rollen und ACLs — Die Zugriffsberechtigungen und Besitzinformationen der Quelldatenbank werden nicht auf die Zieldatenbank übertragen. Alle Datenbankobjekte werden erstellt und gehören dem lokalen Zielbenutzer des Transports.
- Gleichzeitige Transporte – Eine einzelne DB-Instance kann bis zu 32 gleichzeitige Transporte unterstützen, einschließlich Importe und Exporte, wenn Worker-Prozesse ordnungsgemäß konfiguriert wurden.
- RDSNur für SQL Postgre-DB-Instances — SQL Transportable Postgre-Datenbanken werden nur RDS für Postgre-DB-Instances unterstützt. SQL Sie können es nicht mit lokalen Datenbanken oder Datenbanken verwenden, die auf Amazon EC2 laufen.

Einrichtung für den Transport einer Postgre-Datenbank SQL

Bevor Sie beginnen, stellen Sie sicher, dass Ihre vier RDS SQL Postgre-DB-Instances die folgenden Anforderungen erfüllen:

- Auf den RDS SQL For-Postgre-DB-Instances für Quelle und Ziel muss dieselbe Version von Postgre ausgeführt werden. SQL
- Die Ziel-DB darf keine Datenbank mit dem gleichen Namen wie die Quell-DB haben, die Sie transportieren möchten.
- Das Konto, das Sie zum Ausführen des Transports verwenden, benötigt `rds_superuser`-Berechtigungen sowohl für die Quell-DB als auch für die Ziel-DB.
- Die Sicherheitsgruppe für die Quell-DB-Instance muss eingehenden Zugriff von der Ziel-DB-Instance zulassen. Dies ist möglicherweise bereits der Fall, wenn sich Ihre Quell- und Ziel-DB-Instances in der befinden. VPC Weitere Informationen zu Sicherheitsgruppen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Für den Transport von Datenbanken von einer Quell-DB-Instance zu einer Ziel-DB-Instance sind mehrere Änderungen an der DB-Parametergruppe erforderlich, die mit jeder Instance verknüpft ist. Das bedeutet, dass Sie eine benutzerdefinierte DB-Parametergruppe für die Quell-DB-Instance erstellen und eine benutzerdefinierte DB-Parametergruppe für die Ziel-DB-Instance erstellen müssen.

> **Note**
>
> Wenn Ihre DB-Instances bereits mit benutzerdefinierten DB-Parametergruppen konfiguriert sind, können Sie mit Schritt 2 im folgenden Verfahren beginnen.

Konfigurieren Sie die benutzerdefinierten DB-Gruppenparameter für den Transport von Datenbanken wie folgt:

Verwenden Sie für die folgenden Schritte ein Konto mit `rds_superuser`-Berechtigungen.

1. Wenn die Quell- und Ziel-DB-Instances eine Standard-DB-Parametergruppe verwenden, müssen Sie eine benutzerdefinierte DB-Parametergruppe mit der entsprechenden Version für Ihre Instances erstellen. Sie tun dies, damit Sie Werte für mehrere Parameter ändern können. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
2. Ändern Sie in der benutzerdefinierten DB-Parametergruppe die Werte für die folgenden Parameter:

   - `shared_preload_libraries` – Fügen Sie `pg_transport` zur Liste der Bibliotheken hinzu.
   - `pg_transport.num_workers` – Der Standardwert ist 3. Erhöhen oder reduzieren Sie diesen Wert nach Bedarf für Ihre Datenbank. Für eine 200-GB-Datenbank empfehlen wir nicht größer als 8. Beachten Sie, dass Sie, wenn Sie den Standardwert für diesen Parameter erhöhen, auch den Wert von `max_worker_processes` erhöhen sollten.
   - `pg_transport.work_mem`— Der Standardwert ist entweder 128 MB oder 256 MB, abhängig von der SQL Postgre-Version. Die Standardeinstellung kann normalerweise unverändert bleiben.
   - `max_worker_processes` – Der Wert dieses Parameters muss unter Verwendung der folgenden Berechnung festgelegt werden:

     ```
     (3 * pg_transport.num_workers) + 9
     ```

Dieser Wert ist am Ziel erforderlich, um verschiedene Hintergrund-Worker-Prozesse abzuwickeln, die am Transport beteiligt sind. Weitere Informationen dazu finden Sie `max_worker_processes`, unter Ressourcenverbrauch in der Postgre-Dokumentation. SQL

Weitere Informationen zu `pg_transport`-Parametern finden Sie unter Parameterreferenz für transportable Datenbanken .

3. Starten Sie die Quelle RDS für die SQL Postgre-DB-Instance und die Ziel-Instance neu, damit die Einstellungen für die Parameter wirksam werden.
4. Connect zu Ihrer RDS for SQL Postgre-Quell-DB-Instance her.

```
psql --host=source-instance.111122223333.aws-region.rds.amazonaws.com --port=5432
 --username=postgres --password
```

5. Entfernen Sie fremde Erweiterungen aus dem öffentlichen Schema der DB-Instance. Während des eigentlichen Transportvorgangs ist nur die `pg_transport`-Erweiterung zulässig.
6. Installieren Sie die `pg_transport`-Erweiterung wie folgt:

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

7. Connect zu Ihrer SQL Ziel-DB-Instance RDS für Postgre her. Entfernen Sie alle fremden Erweiterungen und installieren Sie dann die `pg_transport`-Erweiterung.

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

Transport einer SQL Postgre-Datenbank von der Quelle zum Ziel

Nachdem Sie den in Einrichtung für den Transport einer Postgre-Datenbank SQL beschriebenen Prozess abgeschlossen haben, können Sie den Transport starten. Führen Sie dazu die Funktion `transport.import_from_server` auf der Ziel-DB-Instance aus. In der folgenden Syntax finden Sie die Funktionsparameter.

```
SELECT transport.import_from_server(
   'source-db-instance-endpoint',
    source-db-instance-port,
```

```
   'source-db-instance-user',
   'source-user-password',
   'source-database-name',
   'destination-user-password',
   false);
```

Der im Beispiel gezeigte `false`-Wert teilt der Funktion mit, dass es sich nicht um einen Trockenlauf handelt. Um die Transporteinrichtung zu testen, können Sie `true` für die Option `dry_run` angeben, wenn Sie die Funktion aufrufen, wie im Folgenden gezeigt:

```
postgres=> SELECT transport.import_from_server(
    'docs-lab-source-db.666666666666aws-region.rds.amazonaws.com', 5432,
    'postgres', '********', 'labdb', '******', true);
INFO:  Starting dry-run of import of database "labdb".
INFO:  Created connections to remote database        (took 0.03 seconds).
INFO:  Checked remote cluster compatibility          (took 0.05 seconds).
INFO:  Dry-run complete                          (took 0.08 seconds total).
 import_from_server
--------------------

(1 row)
```

Die INFO Zeilen werden ausgegeben, weil der `pg_transport.timing` Parameter auf seinen Standardwert gesetzt ist,`true`. Legen Sie `dry_run` auf `false` fest, wenn Sie den Befehl ausführen und die Quelldatenbank wie folgt in das Ziel importiert wird:

```
INFO:  Starting import of database "labdb".
INFO:  Created connections to remote database        (took 0.02 seconds).
INFO:  Marked remote database as read only           (took 0.13 seconds).
INFO:  Checked remote cluster compatibility          (took 0.03 seconds).
INFO:  Signaled creation of PITR blackout window     (took 2.01 seconds).
INFO:  Applied remote database schema pre-data       (took 0.50 seconds).
INFO:  Created connections to local cluster          (took 0.01 seconds).
INFO:  Locked down destination database              (took 0.00 seconds).
INFO:  Completed transfer of database files          (took 0.24 seconds).
INFO:  Completed clean up                            (took 1.02 seconds).
INFO:  Physical transport complete               (took 3.97 seconds total).
import_from_server
--------------------
(1 row)
```

Für diese Funktion müssen Sie Passwörter für Datenbankbenutzer angeben. Daher empfehlen wir Ihnen, die Passwörter der Benutzerrollen, die Sie für den Transportvorgang verwendet haben, nach Abschluss des Transports zu ändern. Oder Sie können SQL Bind-Variablen verwenden, um temporäre Benutzerrollen zu erstellen. Verwenden Sie diese temporären Rollen für den Transport und entfernen Sie die Rollen anschließend.

Falls Ihr Transport nicht erfolgreich ist, wird möglicherweise eine Fehlermeldung ähnlich der folgenden angezeigt:

```
pg_transport.num_workers=8 25% of files transported failed to download file data
```

Die Fehlermeldung „Dateidaten konnte nicht heruntergeladen werden“ zeigt an, dass die Anzahl der Worker-Prozesse für die Größe der Datenbank nicht korrekt festgelegt ist. Möglicherweise müssen Sie den für `pg_transport.num_workers` festgelegten Wert erhöhen oder verringern. Jeder Fehler meldet den Prozentsatz der Fertigstellung, sodass Sie die Auswirkungen Ihrer Änderungen sehen können. Wenn Sie beispielsweise die Einstellung in einem Fall von 8 auf 4 ändern, führte dies zu Folgendem:

```
pg_transport.num_workers=4 75% of files transported failed to download file data
```

Beachten Sie, dass der Parameter `max_worker_processes` auch während des Transportvorgangs berücksichtigt wird. Mit anderen Worten, Sie müssen möglicherweise sowohl `pg_transport.num_workers` als auch `max_worker_processes` ändern, um die Datenbank erfolgreich zu transportieren. Das gezeigte Beispiel hat endlich funktioniert, als die `pg_transport.num_workers` wurde auf 2 eingestellt wurde:

```
pg_transport.num_workers=2 100% of files transported
```

Weitere Hinweise zur `transport.import_from_server`-Funktion und ihren Parametern finden Sie unter Funktionsreferenz für transportable Datenbanken.

Funktionsreferenz für transportable Datenbanken

Die `transport.import_from_server` Funktion transportiert eine SQL Postgre-Datenbank, indem sie sie von einer Quell-DB-Instance in eine Ziel-DB-Instance importiert. Dies geschieht durch die Verwendung eines physischen Datenbankverbindungs-Transportmechanismus.

Vor dem Start des Transports überprüft diese Funktion, ob die Quell- und Ziel-DB-Instances dieselbe Version haben und für die Migration kompatibel sind. Es bestätigt auch, dass die Ziel-DB-Instance über genügend Speicherplatz für die Quelle verfügt.

Syntax

```
transport.import_from_server(
   host text,
   port int,
   username text,
   password text,
   database text,
   local_password text,
   dry_run bool
)
```

Rückgabewert

Keine.

Parameter

Die Beschreibung der Parameter der `transport.import_from_server`-Funktion finden Sie in der folgenden Tabelle.

Parameter	Beschreibung
`host`	Der Endpunkt der Quell-DB-Instance.
`port`	Eine ganze Zahl, die den Port der Quell-DB-Instance darstellt. SQLPostgre-DB-Instances verwenden häufig Port 5432.
`username`	Der Benutzer der Quell-DB-Instance. Dieser Benutzer muss Mitglied der Rolle `rds_superuser` sein.
`password`	Das Benutzerpasswort der Quell-DB-Instance.
`database`	Der Name der Datenbank in der zu transportierenden Quell-DB-Instance.
`local_password`	Das lokale Passwort des aktuellen Benutzers für die Ziel-DB-Instance. Dieser Benutzer muss Mitglied der Rolle `rds_superuser` sein.

Parameter	Beschreibung
`dry_run`	Ein optionaler boolescher Wert, der angibt, ob ein Trockenlauf durchgeführt werden soll. Die Voreinstellung ist `false`, was bedeutet, dass der Transport durchgeführt wird. Um die Kompatibilität zwischen der Quell- und der Ziel-DB-Instance zu bestätige n, ohne den eigentlichen Transport durchzuführen, legen Sie `dry_run` auf `true` fest.

Beispiel

Ein Beispiel finden Sie unter Transport einer SQL Postgre-Datenbank von der Quelle zum Ziel.

Parameterreferenz für transportable Datenbanken

Mehrere Parameter steuern das Verhalten der `pg_transport`-Erweiterung. Im Folgenden finden Sie Beschreibungen dieser Parameter.

`pg_transport.num_workers`

Die Anzahl der für den Transportprozess zu verwendenden Worker. Die Voreinstellung ist 3. Gültige Werte sind 1–32. Selbst die größten Datenbanktransporte erfordern in der Regel weniger als acht Worker. Der Wert dieser Einstellung für die Ziel-DB-Instance wird während des Transports sowohl vom Ziel als auch von der Quelle verwendet.

`pg_transport.timing`

Gibt an, ob während des Transports Zeitinformationen übermittelt werden sollen. Der Standardwert ist `true`, was bedeutet, dass Zeitinformationen gemeldet werden. Es wird empfohlen, diesen Parameter auf `true` festzulegen, damit Sie den Fortschritt überwachen können. Eine Beispielausgabe finden Sie unter Transport einer SQL Postgre-Datenbank von der Quelle zum Ziel.

`pg_transport.work_mem`

Die maximale Menge an Speicher, die für jeden Worker zugewiesen werden kann. Die Standardeinstellung ist 131072 Kilobyte (KB) oder 262144 KB (256 MB), abhängig von der Postgre-Version. SQL Der Mindestwert beträgt 64 Megabyte (65536 KB). Gültige Werte sind in Kilobyte (KBs) als binäre Basis-2-Einheiten angegeben, wobei 1 KB = 1024 Byte ist.

Der Transport benötigt möglicherweise weniger Speicher als in diesem Parameter angegeben. Selbst große Datenbanktransporte benötigen normalerweise weniger als 256 MB (262144 KB) Speicher pro Worker.

Exportieren von Daten aus einem/einer RDS for PostgreSQL-DB-Instance zu Amazon S3

Sie können Daten aus einer DB-Instance von RDS für PostgreSQL abfragen und direkt in Dateien exportieren, die in einem Amazon-S3-Bucket gespeichert sind. Dazu installieren Sie zuerst die Erweiterung von RDS für PostgreSQL `aws_s3`. Diese Erweiterung stellt die Funktionen bereit, die Sie zum Exportieren von Abfrageergebnissen nach Amazon S3 verwenden. Im Folgenden erfahren Sie, wie Sie die Erweiterung installieren und Daten nach Amazon S3 exportieren.

> **Note**
>
> Kontoübergreifender Export nach Amazon S3 wird nicht unterstützt.

Alle derzeit verfügbaren Versionen von RDS für PostgreSQL unterstützen den Export von Daten nach Amazon Simple Storage Service. Ausführliche Versionsinformationen finden Sie unter Aktualisierungen von Amazon RDS für PostgreSQL im Abschnitt Versionshinweise für Amazon RDS für PostgreSQL.

Wenn Sie keinen Bucket für Ihren Export eingerichtet haben, lesen Sie die folgenden Themen im Benutzerhandbuch von Amazon Simple Storage Service.

- Einrichten von Amazon S3
- Einen Bucket erstellen

Standardmäßig verwenden die von RDS for PostgreSQL nach Amazon S3 exportierten Daten serverseitige Verschlüsselung mit einem. Von AWS verwalteter Schlüssel Wenn Sie die Bucket-Verschlüsselung verwenden, muss der Amazon S3 S3-Bucket mit einem AWS Key Management Service (AWS KMS) -Schlüssel (SSE-KMS) verschlüsselt werden. Derzeit werden Buckets, die mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt sind, nicht unterstützt.

> **Note**
>
> Sie können DB-Snapshot-Daten mit der AWS Management Console AWS CLI, oder Amazon RDS-API in Amazon S3 speichern. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS.

Themen

- Installieren der Erweiterung aws_s3
- Übersicht über das Exportieren von Daten zu Amazon S3
- Angeben des Amazon S3-Dateipfads für den Export
- Einrichten des Zugriffs auf einen Amazon S3-Bucket
- Exportieren von Abfragedaten mithilfe der Funktion aws_s3.query_export_to_s3
- Funktionsreferenz
- Fehlerbehebung beim Zugriff auf Amazon S3

Installieren der Erweiterung aws_s3

Bevor Sie Amazon Simple Storage Service mit Ihrer DB-Instance von RDS für PostgreSQL verwenden können, müssen Sie die Erweiterung `aws_s3` installieren. Diese Erweiterung bietet Funktionen zum Exportieren von Daten aus einer DB-Instance von RDS für PostgreSQL in einen Amazon-S3-Bucket Sie stellt außerdem Funktionen zum Importieren von Daten aus Amazon S3 bereit. Weitere Informationen finden Sie unter Importieren von Amazon S3 in eine DB-Instance von RDS für PostgreSQL. Die Erweiterung `aws_s3` hängt von einigen Hilfsfunktionen in der Erweiterung `aws_commons` ab, die bei Bedarf automatisch installiert wird.

So installieren Sie die Erweiterung **`aws_s3`**

1. Verwenden Sie psql (oder pgAdmin), um eine Verbindung mit der DB-Instance von RDS für PostgreSQL als Benutzer mit `rds_superuser`-Berechtigungen herzustellen. Wenn Sie beim Einrichten den Standardnamen beibehalten haben, stellen Sie eine Verbindung als `postgres` her.

   ```
   psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Führen Sie den folgenden Befehl aus, um die Erweiterung zu installieren.

   ```
   postgres=> CREATE EXTENSION aws_s3 CASCADE;
   NOTICE: installing required extension "aws_commons"
   CREATE EXTENSION
   ```

3. Wenn Sie überprüfen möchten, ob die Erweiterung installiert wurde, können Sie psql-Metabefehl `\dx` verwenden.

```
postgres=> \dx
       List of installed extensions
    Name     | Version |   Schema   |                Description
-------------+---------+------------+-------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

Die Funktionen zum Importieren von Daten aus Amazon S3 und Exportieren von Daten nach Amazon S3 sind jetzt verfügbar.

Überprüfen, ob Ihre Version von RDS für PostgreSQL Exporte nach Amazon S3 unterstützt

Sie können überprüfen, ob Ihre Version von RDS für PostgreSQL den Export nach Amazon S3 unterstützt, indem Sie den Befehl `describe-db-engine-versions` verwenden. Im folgenden Beispiel wird die Unterstützung für Version 10.14 überprüft.

```
aws rds describe-db-engine-versions --region us-east-1
--engine postgres --engine-version 10.14 | grep s3Export
```

Wenn die Ausgabe die Zeichenfolge `"s3Export"` enthält, unterstützt die Engine Amazon S3-Exporte. Ansonsten unterstützt die Engine sie nicht.

Übersicht über das Exportieren von Daten zu Amazon S3

Verwenden Sie das folgende Verfahren, um in einer Datenbank gespeicherte Daten in einen Amazon S3 Bucket zu exportieren.

So exportieren Sie Daten nach S3

1. Identifizieren Sie einen Amazon S3-Dateipfad, der zum Exportieren von Daten verwendet werden soll. Weitere Informationen zu diesem Prozess finden Sie unter Angeben des Amazon S3-Dateipfads für den Export.
2. Erteilen Sie die Berechtigung für den Zugriff auf den Amazon S3-Bucket.

Um Daten in eine Amazon S3-Datei zu exportieren, erteilen Sie dem/der RDS for PostgreSQL-DB-Instance die Berechtigung, auf den Amazon S3-Bucket zuzugreifen, den der Export für die Speicherung verwendet. Dazu gehören die folgenden Schritte:

1. Erstellen Sie eine IAM-Richtlinie, die Zugriff auf einen Amazon S3-Bucket bietet, in den Sie exportieren möchten.
2. Erstellen Sie eine IAM-Rolle.
3. Fügen Sie die erstellte Richtlinie an die erstellte Rolle an.
4. Fügen Sie diese IAM-Rolle zu Ihrem Ihrer DB-Instance hinzu.

Weitere Informationen zu diesem Prozess finden Sie unter Einrichten des Zugriffs auf einen Amazon S3-Bucket.

3. Identifizieren Sie eine Datenbankabfrage, um die Daten abzurufen. Exportieren Sie die Abfragedaten, indem Sie die Funktion `aws_s3.query_export_to_s3` aufrufen.

   Nachdem Sie die vorangegangenen Vorbereitungsaufgaben abgeschlossen haben, verwenden Sie die aws_s3.query_export_to_s3-Funktion, um Abfrageergebnisse in Amazon S3 zu exportieren. Weitere Informationen zu diesem Prozess finden Sie unter Exportieren von Abfragedaten mithilfe der Funktion aws_s3.query_export_to_s3.

Angeben des Amazon S3-Dateipfads für den Export

Geben Sie die folgenden Informationen an, um den Speicherort in Amazon S3 zu identifizieren, in den Sie Daten exportieren möchten:

- Bucket-Name – Ein Bucket ist ein Container für Amazon S3-Objekte oder -Dateien.

  Weitere Informationen zum Speichern von Daten mit Amazon S3 finden Sie unter Erstellen eines Buckets und Arbeiten mit Objekten im Amazon Simple Storage Service-Benutzerhandbuch.
- Dateipfad – Der Dateipfad gibt an, wo der Export im Amazon S3-Bucket gespeichert wird. Der Dateipfad besteht aus Folgendem:
  - Ein optionales Pfadpräfix, das einen Pfad für virtuelle Ordner identifiziert.
  - Ein Dateipräfix, das eine oder mehrere Dateien identifiziert, die gespeichert werden sollen. Größere Exporte werden in mehreren Dateien gespeichert, jeweils mit einer maximalen Größe

von ca. 6 GB. Die zusätzlichen Dateinamen haben das gleiche Dateipräfix, aber mit `_part`*XX* angefügt. *XX* stellt 2, dann 3 usw. dar.

Beispielsweise ist ein Dateipfad mit einem `exports`-Ordner und einem `query-1-export`-Dateipräfix `/exports/query-1-export`.

- AWS Region (optional) — Die AWS Region, in der sich der Amazon S3 S3-Bucket befindet. Wenn Sie keinen AWS Regionswert angeben, speichert Amazon RDS Ihre Dateien in Amazon S3 in derselben AWS Region wie die exportierende .

> **Note**
>
> Derzeit muss die AWS Region mit der Region der exportierenden identisch sein.

Eine Liste der AWS Regionsnamen und der zugehörigen Werte finden Sie unterRegionen, Availability Zones und Local Zones.

Um die Amazon S3-Dateiinformationen darüber zu speichern, wo der Export gespeichert werden soll, können Sie die aws_commons.create_s3_uri-Funktion verwenden, um eine zusammengesetzte `aws_commons._s3_uri_1`-Struktur wie folgt zu erstellen.

```
psql=> SELECT aws_commons.create_s3_uri(
   'amzn-s3-demo-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Sie geben diesen `s3_uri_1`-Wert später als Parameter im Aufruf der aws_s3.query_export_to_s3-Funktion an. Beispiele finden Sie unter Exportieren von Abfragedaten mithilfe der Funktion aws_s3.query_export_to_s3.

Einrichten des Zugriffs auf einen Amazon S3-Bucket

Um Daten zu Amazon S3 zu exportieren, erteilen Sie Ihrer PostgreSQL-DB-Instance die Berechtigung, auf den Amazon S3-Bucket zuzugreifen, in den die Dateien aufgenommen werden sollen.

Führen Sie dazu die folgenden Schritte aus.

So erteilen Sie einer PostgreSQL-DB-Instance Zugriff auf Amazon S3 über eine IAM-Rolle

1. Erstellen Sie eine IAM-Richtlinie.

   Diese Richtlinie enthält die Bucket- und Objektberechtigungen, die Ihrer PostgreSQL-DB-Instance den Zugriff auf Amazon S3 ermöglichen.

   Führen Sie beim Erstellen dieser Richtlinie die folgenden Schritte aus:

   a. Nehmen Sie die folgenden erforderlichen Aktionen in die Richtlinie auf, um die Übertragung von Dateien aus Ihrer PostgreSQL-DB-Instance in einen Amazon S3-Bucket zu gestatten:

      - `s3:PutObject`
      - `s3:AbortMultipartUpload`

   b. Geben Sie den Amazon-Ressourcennamen (ARN) ein, der den Amazon S3-Bucket und die Objekte im Bucket identifiziert. Das ARN-Format für den Zugriff auf Amazon S3 lautet: `arn:aws:s3:::amzn-s3-demo-bucket/*`

   Weitere Informationen zum Erstellen einer IAM-Richtlinie für Amazon RDS for PostgreSQL finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff. Siehe auch Tutorial: Erstellen und Anfügen Ihrer ersten vom Kunden verwalteten Richtlinie im IAM-Benutzerhandbuch.

   Mit dem folgenden AWS CLI Befehl wird eine IAM-Richtlinie `rds-s3-export-policy` mit diesen Optionen erstellt. Sie gewährt Zugriff auf einen Bucket namens *amzn-s3-demo-bucket*.

   > **Warning**
   >
   > Es wird empfohlen, die Datenbank innerhalb einer privaten VPC einzurichten, deren Endpunktrichtlinien für den Zugriff auf bestimmte Buckets konfiguriert sind. Weitere Informationen finden Sie unter Verwenden von Endpunktrichtlinien für Amazon S3 im Amazon VPC Benutzerhandbuch.
   > Es wird dringend empfohlen, keine Richtlinie mit Zugriff auf alle Ressourcen zu erstellen. Dieser Zugriff kann eine Bedrohung für die Datensicherheit darstellen.
   > Wenn Sie eine Richtlinie erstellen, die `S3:PutObject` Zugriff auf alle Ressourcen mit `"Resource":"*"` gewährt, kann ein Benutzer mit Exportberechtigungen Daten in alle

Buckets in Ihrem Konto exportieren. Darüber hinaus kann der Benutzer Daten in jeden öffentlich beschreibbaren Bucket in Ihrer AWS -Region exportieren.

Notieren Sie nach dem Erstellen der Richtlinie den Amazon-Ressourcennamen (ARN) der Richtlinie. Sie benötigen den ARN für einen nachfolgenden Schritt, in dem Sie die Richtlinie an eine IAM-Rolle anhängen.

```
aws iam create-policy  --policy-name rds-s3-export-policy  --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3export",
         "Action": [
           "s3:PutObject*",
           "s3:ListBucket",
           "s3:GetObject*",
           "s3:DeleteObject*",
           "s3:GetBucketLocation",
           "s3:AbortMultipartUpload"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket/*"
         ]
       }
     ]
   }'
```

2. Erstellen Sie eine IAM-Rolle.

   Sie tun dies, damit Amazon RDS in Ihrem Namen diese IAM-Rolle übernehmen kann, um auf Ihre Amazon S3-Buckets zuzugreifen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer im IAM-Benutzerhandbuch.

   Wir empfehlen die Verwendung von `aws:SourceArn` und `aws:SourceAccount` globaler Bedingungskontext-Schlüssel in ressourcenbasierten Richtlinien, um die Berechtigungen des Dienstes auf eine bestimmte Ressource zu beschränken. Dies ist der effektivste Weg, um sich vor dem verwirrtes Stellvertreterproblem zu schützen.

Wenn Sie sowohl globale Kontextschlüssel nutzen und der `aws:SourceArn`-Wert enthält die Konto-ID, muss der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert die gleiche Konto-ID verwenden, wenn er in der gleichen Richtlinienanweisung verwendet wird.

- Verwenden von `aws:SourceArn` wenn Sie einen serviceübergreifenden Zugriff für eine einzelne Ressource wünschen.
- Verwenden von `aws:SourceAccount` wenn Sie zulassen möchten, dass eine Ressource in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft wird.

Verwenden Sie in der Richtlinie den `aws:SourceArn` globalen Kontextschlüssel mit dem vollständigen ARN der Ressource. Das folgende Beispiel zeigt, wie Sie dazu den AWS CLI Befehl verwenden, um eine Rolle mit dem Namen `rds-s3-export-role` zu erstellen.

Example

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws iam create-role  \
    --role-name rds-s3-export-role  \
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws iam create-role  ^
    --role-name rds-s3-export-role  ^
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

3. Fügen Sie die erstellte IAM-Richtlinie der IAM-Rolle an, die Sie erstellt haben.

   Mit dem folgenden AWS CLI Befehl wird die zuvor erstellte Richtlinie an die Rolle `rds-s3-export-role`. Replace *`your-policy-arn`* mit dem Richtlinien-ARN angehängt, den Sie in einem früheren Schritt notiert haben.

   ```
   aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-export-role
   ```

4. Fügen Sie die IAM-Rolle der DB Instance hinzu. Sie tun dies, indem Sie das AWS Management Console oder verwenden AWS CLI, wie im Folgenden beschrieben.

Konsole

So fügen Sie eine IAM-Rolle für eine PostgreSQL DB--Instance über die Konsole hinzu:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie den Namen der PostgreSQL DB--Instance aus, um ihre Details anzuzeigen.

3. Wählen Sie auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) im Bereich Manage IAM roles (IAM-Rollen verwalten) die Rolle aus, die unter Add IAM roles to this instance (IAM-Rollen zu dieser Instance hinzufügen) hinzugefügt werden soll.
4. Wählen Sie unter Funktion die Option s3Export aus.
5. Wählen Sie Rolle hinzufügen.

AWS CLI

So fügen Sie eine IAM-Rolle für eine PostgreSQL-DB-Instance mithilfe der CLI hinzu

- Verwenden Sie den folgenden CLI-Befehl, um die IAM-Rolle zur RDS for PostgreSQL DB-Instance mit dem Namen `my-db-instance` hinzuzufügen. Ersetzen Sie *`your-role-arn`* durch den Rollen-ARN, den Sie im vorherigen Schritt notiert haben. Verwenden Sie `s3Export` für den Wert der `--feature-name`-Option.

  Example

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

  ```
  aws rds add-role-to-db-instance \
     --db-instance-identifier my-db-instance \
     --feature-name s3Export \
     --role-arn your-role-arn   \
     --region your-region
  ```

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

  ```
  aws rds add-role-to-db-instance ^
     --db-instance-identifier my-db-instance ^
     --feature-name s3Export ^
     --role-arn your-role-arn ^
     --region your-region
  ```

Exportieren von Abfragedaten mithilfe der Funktion aws_s3.query_export_to_s3

Exportieren Sie Ihre PostgreSQL-Daten nach Amazon S3, indem Sie die aws_s3.query_export_to_s3-Funktion aufrufen.

Themen

- Voraussetzungen
- Aufruf von aws_s3.query_export_to_s3
- Exportieren in eine CSV-Datei, die ein benutzerdefiniertes Trennzeichen verwendet
- Exportieren in eine Binärdatei mit Codierung

Voraussetzungen

Bevor Sie die `aws_s3.query_export_to_s3`-Funktion verwenden, müssen Sie die folgenden Voraussetzungen erfüllen:

- Installieren Sie die erforderlichen PostgreSQL-Erweiterungen, wie unter beschriebe Übersicht über das Exportieren von Daten zu Amazon S3.
- Legen Sie fest, wohin die Daten nach Amazon S3 exportiert werden sollen, wie unter beschriebe Angeben des Amazon S3-Dateipfads für den Export.
- Stellen Sie sicher, dass die Exportzugriff auf Amazon S3 hat (wie unter Einrichten des Zugriffs auf einen Amazon S3-Bucket beschrieben).

In den folgenden Beispielen wird eine Datenbanktabelle namens verwende `sample_table`. In diesen Beispielen werden die Daten in einen Bucket namens exportier *`amzn-s3-demo-bucket`*. Die Beispieltabelle und die Daten werden mit den folgenden SQL-Anweisungen in psql erstellt.

```
psql=> CREATE TABLE sample_table (bid bigint PRIMARY KEY, name varchar(80));
psql=> INSERT INTO sample_table (bid,name) VALUES (1, 'Monday'), (2,'Tuesday'), (3,
 'Wednesday');
```

Aufruf von aws_s3.query_export_to_s3

Im Folgenden werden die grundlegenden Möglichkeiten zum Aufrufen der aws_s3.query_export_to_s3-Funktion dargestellt.

In diesen Beispielen wird die Variable `s3_uri_1` zum Identifizieren einer Struktur verwendet, die die Informationen zur Identifizierung der Amazon S3-Datei enthält. Verwenden Sie die Funktion aws_commons.create_s3_uri, um die Struktur zu erstellen.

```
psql=> SELECT aws_commons.create_s3_uri(
   'amzn-s3-demo-bucket',
```

```
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Obwohl die Parameter für die folgenden zwei `aws_s3.query_export_to_s3`-Funktionsaufrufe unterschiedlich sind, sind die Ergebnisse für diese Beispiele identisch. Alle Zeilen der `sample_table`-Tabelle werden in einen Bucket namens *`amzn-s3-demo-bucket`* exportiert.

```
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1');

psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1', options :='format text');
```

Die Parameter werden wie folgt beschrieben:

- `'SELECT * FROM sample_table'` – Der erste Parameter ist eine erforderliche Textzeichenfolge, die eine SQL-Abfrage enthält. Die PostgreSQL-Engine führt diese Abfrage aus. Die Ergebnisse der Abfrage werden in den S3-Bucket kopiert, der in anderen Parametern identifiziert wurde.
- `:'s3_uri_1'` – Dieser Parameter ist eine Struktur, die die Datei Amazon S3 identifiziert. In diesem Beispiel wird die zuvor erstellte Struktur anhand einer Variablen identifiziert. Sie können die Struktur stattdessen erstellen, indem Sie den Funktionsaufruf `aws_commons.create_s3_uri` wie folgt inline in den Funktionsaufruf `aws_s3.query_export_to_s3` einschließen.

  ```
  SELECT * from aws_s3.query_export_to_s3('select * from sample_table',
     aws_commons.create_s3_uri('amzn-s3-demo-bucket', 'sample-filepath', 'us-west-2')
  );
  ```

- `options :='format text'` – Der `options`-Parameter ist eine optionale Textzeichenfolge, die `COPY`-PostgreSQL-Argumente enthält. Beim Kopiervorgang werden die Argumente und das Format des PostgreSQL COPY-Befehls verwendet.

Wenn die angegebene Datei nicht im Amazon S3-Bucket vorhanden ist, wird sie erstellt. Wenn die Datei bereits vorhanden ist, wird sie überschrieben. Die Syntax für den Zugriff auf die exportierten Daten in Amazon S3 ist die folgende.

```
s3-region://bucket-name[/path-prefix]/file-prefix
```

Größere Exporte werden in mehreren Dateien gespeichert, jeweils mit einer maximalen Größe von ca. 6 GB. Die zusätzlichen Dateinamen haben das gleiche Dateipräfix, aber mit `_part`*XX* angefügt. *XX* stellt 2, dann 3 usw. dar. Angenommen, Sie geben den Pfad, in dem Sie Datendateien speichern, wie folgt an.

```
s3-us-west-2://amzn-s3-demo-bucket/my-prefix
```

Wenn der Export drei Datendateien anlegen muss, enthält der Amazon S3-Bucket die folgenden Datendateien.

```
s3-us-west-2://amzn-s3-demo-bucket/my-prefix
s3-us-west-2://amzn-s3-demo-bucket/my-prefix_part2
s3-us-west-2://amzn-s3-demo-bucket/my-prefix_part3
```

Die vollständige Referenz für diese Funktion und weitere Aufrufmöglichkeiten finden Sie unter aws_s3.query_export_to_s3. Weitere Informationen zum Zugriff auf Dateien in Amazon S3 finden Sie unter View an object im Amazon Simple Storage Service User Guide.

Exportieren in eine CSV-Datei, die ein benutzerdefiniertes Trennzeichen verwendet

Das folgende Beispiel zeigt, wie die aws_s3.query_export_to_s3-Funktion zum Exportieren von Daten in eine Datei aufgerufen wird, die ein benutzerdefiniertes Trennzeichen verwendet. Im Beispiel werden Argumente des Befehls PostgreSQL COPY verwendet, um das CSV-Dateiformat (durch Kommas getrennte Werte) und ein Doppelpunkt (:)-Trennzeichen anzugeben.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format csv, delimiter $$:$$');
```

Exportieren in eine Binärdatei mit Codierung

Das folgende Beispiel zeigt, wie die aws_s3.query_export_to_s3-Funktion zum Exportieren von Daten in eine Binärdatei mit Windows-1253-Codierung aufgerufen wird.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format binary, encoding WIN1253');
```

Funktionsreferenz

Funktionen

- aws_s3.query_export_to_s3
- aws_commons.create_s3_uri

aws_s3.query_export_to_s3

Exportiert ein PostgreSQL-Abfrageergebnis in einen Amazon S3-Bucket. Die Erweiterung `aws_s3` stellt die Funktion `aws_s3.query_export_to_s3` bereit.

Die zwei erforderlichen Parameter sind `query` und `s3_info`. Diese definieren die zu exportierende Abfrage und identifizieren den Amazon S3-Bucket, in den exportiert werden soll. Ein optionaler Parameter namens `options` ermöglicht die Definition verschiedener Exportparameter. Beispiele für die Verwendung der `aws_s3.query_export_to_s3`-Funktion finden Sie unter Exportieren von Abfragedaten mithilfe der Funktion aws_s3.query_export_to_s3.

Syntax

```
aws_s3.query_export_to_s3(
    query text,
    s3_info aws_commons._s3_uri_1,
    options text,
    kms_key text
)
```

Eingabeparameter

query

Eine erforderliche Textzeichenfolge, die eine SQL-Abfrage enthält, die von der PostgreSQL-Engine ausgeführt wird. Die Ergebnisse dieser Abfrage werden in einen S3-Bucket kopiert, der im `s3_info`-Parameter identifiziert wurde.

s3_info

Ein zusammengesetzter `aws_commons._s3_uri_1`-Typ mit den folgenden Informationen zum S3-Objekt:

- `bucket` – Der Name des Amazon S3-Buckets, der die Datei enthalten soll.
- `file_path` – Der Amazon S3-Dateiname und der -Pfad.
- `region`— Die AWS Region, in der sich der Bucket befindet. Eine Liste der AWS Regionsnamen und der zugehörigen Werte finden Sie unterRegionen, Availability Zones und Local Zones.

Derzeit muss dieser Wert dieselbe AWS Region wie die der exportierenden sein. Die Standardeinstellung ist die AWS Region der exportierenden .

Informationen zum Erstellen einer zusammengesetzten `aws_commons._s3_uri_1`-Struktur finden Sie in der aws_commons.create_s3_uri-Funktion.

options

Eine optionale Textzeichenfolge mit Argumenten für den PostgreSQL `COPY`-Befehl. Diese Argumente geben an, wie die Daten beim Exportieren kopiert werden sollen. Weitere Informationen finden Sie in der PostgreSQL COPY-Dokumentation.

kms_key-Text

Eine optionale Textzeichenfolge, die den vom Kunden verwalteten KMS-Schlüssel des S3-Buckets enthält, in den die Daten exportiert werden sollen.

Alternative Eingabeparameter

Zum Testen können Sie statt des Parameters `s3_info` eine erweiterte Gruppe von Parametern verwenden. Nachfolgend sind weitere Syntaxvariationen für die Funktion `aws_s3.query_export_to_s3` aufgeführt.

Statt den Parameter `s3_info` zum Identifizieren einer Amazon S3-Datei zu verwenden, nutzen Sie die Kombination aus den Parametern `bucket`, `file_path` und `region`.

```
aws_s3.query_export_to_s3(
    query text,
    bucket text,
    file_path text,
    region text,
    options text,
    kms_key text
)
```

query

Eine erforderliche Textzeichenfolge, die eine SQL-Abfrage enthält, die von der PostgreSQL-Engine ausgeführt wird. Die Ergebnisse dieser Abfrage werden in einen S3-Bucket kopiert, der im `s3_info`-Parameter identifiziert wurde.

bucket

Eine erforderliche Textzeichenfolge mit dem Namen des Amazon S3-Buckets, der die Datei enthält.

file_path

Eine erforderliche Textzeichenfolge, die den Amazon S3-Dateinamen einschließlich des Pfads der Datei enthält.

Region

Eine optionale Textzeichenfolge, die die AWS Region enthält, in der sich der Bucket befindet. Eine Liste der AWS Regionsnamen und der zugehörigen Werte finden Sie unterRegionen, Availability Zones und Local Zones.

Derzeit muss dieser Wert dieselbe AWS Region wie die der exportierenden sein. Die Standardeinstellung ist die AWS Region der exportierenden .

options

Eine optionale Textzeichenfolge mit Argumenten für den PostgreSQL `COPY`-Befehl. Diese Argumente geben an, wie die Daten beim Exportieren kopiert werden sollen. Weitere Informationen finden Sie in der PostgreSQL COPY-Dokumentation.

kms_key-Text

Eine optionale Textzeichenfolge, die den vom Kunden verwalteten KMS-Schlüssel des S3-Buckets enthält, in den die Daten exportiert werden sollen.

Ausgabeparameter

```
aws_s3.query_export_to_s3(
    OUT rows_uploaded bigint,
    OUT files_uploaded bigint,
    OUT bytes_uploaded bigint
)
```

rows_uploaded

Die Anzahl der Tabellenzeilen, die für die angegebene Abfrage erfolgreich in Amazon S3 hochgeladen wurden.

files_uploaded

Die Anzahl der in Amazon S3 hochgeladenen Dateien. Dateien werden in Größen von ca. 6 GB erstellt. Jeder weiteren erstellten Datei wird `_part`*XX* an den Namen angehängt. *XX* stellt 2, dann 3 usw. nach Bedarf dar.

bytes_uploaded

Die Gesamtanzahl der in Amazon S3 hochgeladenen Bytes.

Beispiele

```
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'amzn-s3-demo-bucket', 'sample-filepath');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'amzn-s3-demo-bucket', 'sample-filepath','us-west-2');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'amzn-s3-demo-bucket', 'sample-filepath','us-west-2','format text');
```

aws_commons.create_s3_uri

Erstellt eine `aws_commons._s3_uri_1`-Struktur für die Amazon S3-Dateiinformationen. Die Ergebnisse der Funktion `aws_commons.create_s3_uri` werden im Parameter `s3_info` der Funktion aws_s3.query_export_to_s3 verwendet. Ein Beispiel für die Verwendung der `aws_commons.create_s3_uri`-Funktion finden Sie unter Angeben des Amazon S3-Dateipfads für den Export.

Syntax

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Eingabeparameter

bucket

Eine erforderliche Textzeichenfolge mit dem Namen des Amazon S3-Buckets für die Datei.

file_path

Eine erforderliche Textzeichenfolge, die den Amazon S3-Dateinamen einschließlich des Pfads der Datei enthält.

Region

Eine erforderliche Textzeichenfolge, die die AWS Region enthält, in der sich die Datei befindet. Eine Liste der AWS Regionsnamen und der zugehörigen Werte finden Sie unterRegionen, Availability Zones und Local Zones.

Fehlerbehebung beim Zugriff auf Amazon S3

Wenn beim Versuch, Daten nach Amazon S3 zu exportieren, Verbindungsprobleme auftreten, bestätigen Sie zunächst, dass die Regeln für den ausgehenden Zugriff für die mit Ihrer DB-Instance verknüpfte VPC-Sicherheitsgruppe Netzwerkkonnektivität zulassen. Insbesondere muss die Sicherheitsgruppe über eine Regel verfügen, die es der DB-Instance ermöglicht, TCP-Verkehr an Port 443 und an beliebige IPv4 Adressen (0.0.0.0/0) zu senden. Weitere Informationen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

Empfehlungen finden Sie im Folgenden:

- Fehlerbehebung für Amazon RDS-Identität und -Zugriff
- Fehlerbehebung bei Amazon S3 im Entwicklerhandbuch für Amazon Simple Storage Service
- Fehlerbehebung bei Amazon S3 und IAM im IAM-Benutzerhandbuch

Aufrufen einer AWS Lambda Funktion aus einem)

AWS Lambda ist ein ereignisgesteuerter Rechendienst, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Es ist für die Verwendung mit vielen AWS Diensten verfügbar, einschließlich . Sie können beispielsweise Lambda-Funktionen verwenden, um Ereignisbenachrichtigungen aus einer Datenbank zu verarbeiten oder Daten aus Dateien zu laden, wann immer eine neue Datei in Amazon S3 hochgeladen wird. Weitere Informationen zu Lambda finden Sie unter Was ist AWS Lambda? im AWS Lambda Entwicklerhandbuch.

> **Note**
>
> Das Aufrufen einer AWS Lambda Funktion wird in diesen RDS-Versionen für PostgreSQL unterstützt:
>
> - Alle PostgreSQL 17-Versionen
> - Alle PostgreSQL 16-Versionen
> - Alle PostgreSQL 15-Versionen
> - PostgreSQL 14.1 und höhere Unterversionen
> - PostgreSQL 13.2 und höhere Unterversionen
> - PostgreSQL 12.6 und höhere Unterversionen

Die Einrichtung von für die Arbeit mit Lambda-Funktionen ist ein mehrstufiger Prozess, der IAM AWS Lambda, Ihre VPC und Ihre umfasst. Im Folgenden finden Sie Zusammenfassungen der notwendigen Schritte.

Weitere Informationen über Lambda-Funktionen finden Sie unter Erste Schritte mit Lambda und Grundlagen von AWS Lambda im AWS Lambda -Entwicklerhandbuch.

Themen

- Schritt 1: Konfigurieren Sie Ihren für ausgehende Verbindungen zu AWS Lambda
- Schritt 2: Konfigurieren Sie IAM für Ihren und AWS Lambda
- Schritt 3: Installieren der aws_lambda-Erweiterung für eine RDS-for-PostgreSQL-DB-Instance
- Schritt 4: Verwenden von Lambda-Hilfsfunktionen mit Ihrer RDS-for-PostgreSQL-DB-Instance (optional)
- Schritt 5: Aufrufen einer Lambda-Funktion von Ihrem RDS-for-PostgreSQL-DB-Instance

- Schritt 6: Erteilen der Berechtigung, Lambda-Funktionen aufzurufen, für andere Benutzer
- Beispiele: Aufrufen von Lambda-Funktionen aus Ihrem SQL
- Fehlermeldungen von Lambda-Funktionen
- AWS Lambda -Funktion und Parameterreferenz

Schritt 1: Konfigurieren Sie Ihren für ausgehende Verbindungen zu AWS Lambda

Lambda-Funktionen werden immer in einer Amazon-VPC ausgeführt, die dem AWS Lambda Service gehört. Lambda wendet Netzwerkzugriffs- und Sicherheitsregeln auf diese VPC an und pflegt und überwacht die VPC automatisch. Ihre DB-Instance von RDS für PostgreSQL muss Netzwerkdatenverkehr an die VPC des Lambda-Services senden. Wie Sie dies konfigurieren, hängt davon ab, ob die primäre DB-Instance Ihres öffentlich oder privat ist.

- Öffentlicher RDS für PostgreSQL-DB-Instance — ist öffentlich, wenn sie sich in einem öffentlichen Subnetz auf Ihrer VPC befindet und wenn die Eigenschaft "" der Instance lautet. PubliclyAccessible `true` Um den Wert dieser Eigenschaft zu ermitteln, können Sie den Befehl verwenden. describe-db-instances AWS CLI Alternativ können Sie über die AWS Management Console die Registerkarte Connectivity & security (Konnektivität und Sicherheit) öffnen und prüfen, ob Publicly accessible (Öffentlich zugänglich) auf Yes (Ja) eingestellt ist. Wenn Sie überprüfen möchten, ob sich die Instance im öffentlichen Subnetz Ihrer VPC befindet, können Sie die AWS Management Console oder die AWS CLI verwenden.

  Um den Zugriff auf Lambda einzurichten, verwenden Sie AWS Management Console oder, AWS CLI um eine ausgehende Regel für die Sicherheitsgruppe Ihrer VPC zu erstellen. Die ausgehende Regel gibt an, dass TCP Port 443 verwenden kann, um Pakete an beliebige IPv4 Adressen (0.0.0.0/0) zu senden.
- Private RDS für PostgreSQL-DB-Instance — In diesem Fall befindet sich die Eigenschaft "PubliclyAccessible" der Instance in einem privaten Subnetz `false` oder sie befindet sich in einem privaten Subnetz. Damit die Instance mit Lambda arbeiten kann, können Sie ein Network Address Translation (NAT)-Gateway verwenden. Weitere Informationen finden Sie unter NAT-Gateways. Sie können Ihre VPC auch mit einem VPC-Endpunkt für Lambda konfigurieren. Weitere Informationen finden Sie unter VPC-Endpunkte im Amazon-VPC-Benutzerhandbuch. Der Endpunkt gibt Antworten auf Aufrufe Ihrer Lambda-Funktionen von Ihrer DB-Instance von RDS für PostgreSQL zurück. Der VPC-Endpunkt verwendet seine eigene private DNS-Auflösung. RDS for PostgreSQL kann den Lambda-VPC-Endpunkt erst verwenden, wenn Sie den Wert von

`rds.custom_dns_resolution` von seinem Standardwert 0 (nicht aktiviert) zu 1 ändern. Gehen Sie hierzu wie folgt vor:

- Erstellen Sie eine benutzerdefinierte DB-Parametergruppe.
- Ändern Sie den Wert des Parameters `rds.custom_dns_resolution` von seinem Standardwert `0` zu 1.
- Ändern Sie Ihre DB-Instance, um Ihre benutzerdefinierte DB-Parametergruppe zu verwenden.
- Starten Sie die DB-Instance neu, damit der bearbeitete Parameter in Kraft tritt.

Ihre VPC kann jetzt auf Netzwerkebene mit der AWS Lambda VPC interagieren. Als Nächstes konfigurieren Sie die Berechtigungen mithilfe von IAM.

Schritt 2: Konfigurieren Sie IAM für Ihren und AWS Lambda

Das Aufrufen von Lambda-Funktionen von Ihrer RDS-for-PostgreSQL-DB-Instance erfordert bestimmte Berechtigungen. Um die erforderlichen Berechtigungen zu konfigurieren, empfehlen wir Ihnen, eine IAM-Richtlinie zu erstellen, die es ermöglicht, Lambda-Funktionen aufzurufen, diese Richtlinie einer Rolle zuzuweisen und die Rolle dann auf Ihre DB-Instance anzuwenden. Dieser Ansatz gewährt der DB-Instance Berechtigungen zum Aufrufen der angegebenen Lambda-Funktion in Ihrem Namen. Nachfolgend wird beschrieben, wie Sie dazu die AWS CLI verwenden können.

IAM-Berechtigungen für die Verwendung Ihrer Amazon-RDS-Instance mit Lambda konfigurieren

1. Verwenden Sie den AWS CLI Befehl create-policy, um eine IAM-Richtlinie zu erstellen, die es Ihrer ermöglicht, die angegebene Lambda-Funktion aufzurufen. (Die Anweisungs-ID (Sid) ist eine optionale Beschreibung für Ihre Richtlinienanweisung und hat keine Auswirkungen auf die Verwendung.) Diese Richtlinie gewährt Ihrer DB-Instance die Mindestberechtigungen zum Aufrufen der angegebenen Lambda-Funktion.

```
aws iam create-policy  --policy-name rds-lambda-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "AllowAccessToExampleFunction",
        "Effect": "Allow",
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:aws-region:444455556666:function:my-function"
        }
    ]
```

```
}'
```

Alternativ können Sie die vordefinierte `AWSLambdaRole`-Richtlinie verwenden, mit der Sie alle Ihre Lambda-Funktionen aufrufen können. Weitere Informationen finden Sie unter Identitätsbasierte IAM-Richtlinien für Lambda.

2. Verwenden Sie den AWS CLI Befehl create-role, um eine IAM-Rolle zu erstellen, die die Richtlinie zur Laufzeit annehmen kann.

```
aws iam create-role  --role-name rds-lambda-role --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Wenden Sie die Richtlinie mithilfe des Befehls auf die Rolle an. attach-role-policy AWS CLI

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::444455556666:policy/rds-lambda-policy \
    --role-name rds-lambda-role --region aws-region
```

4. AWS CLI Dieser letzte Schritt erlaubt den Datenbankbenutzern Ihrer DB-Instances, Lambda-Funktionen aufzurufen.

```
aws rds add-role-to-db-instance \
       --db-instance-identifier my-instance-name \
       --feature-name Lambda \
       --role-arn  arn:aws:iam::444455556666:role/rds-lambda-role   \
       --region aws-region
```

Wenn die VPC- und die IAM-Konfigurationen abgeschlossen sind, können Sie jetzt die `aws_lambda`-Erweiterung installieren. (Beachten Sie, dass Sie die Erweiterung jederzeit installieren können, bis Sie jedoch die richtige VPC-Unterstützung und IAM-Berechtigungen eingerichtet haben, hat die

aws_lambda-Erweiterung keinerlei Auswirkungen auf die Funktionen Ihrer RDS-for-PostgreSQL-DB-Instances.)

Schritt 3: Installieren der **aws_lambda**-Erweiterung für eine RDS-for-PostgreSQL-DB-Instance

Zur Verwendung AWS Lambda mit Ihrem fügen Sie die PostgreSQL-Erweiterung zu Ihrem DB-Instance hinzu. Diese Erweiterung bietet Ihrer RDS-for-PostgreSQL-DB-Instance die Möglichkeit, Lambda-Funktionen von PostgreSQL aus aufzurufen.

aws_lambda-Erweiterung in einer RDS-for-PostgreSQL-DB-Instance installieren

Verwenden Sie die PostgreSQL-psql-Befehlszeile oder das pgAdmin-Tool, um Ihre RDS-for-PostgreSQL-DB-Instance zu verbinden.

1. Verbinden Sie Ihren Ihre RDS-for-PostgreSQL-DB-Instance als Benutzer mit rds_superuser-Berechtigungen. Im Beispiel wird der postgres-Standardbenutzer dargestellt.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Installieren Sie die aws_lambda-Erweiterung. Die aws_commons-Erweiterung ist auch erforderlich. Sie bietet Hilfsfunktionen für aws_lambda und viele andere Aurora-Erweiterungen für PostgreSQL. Wenn Sie noch nicht auf Ihrer RDS-for-PostgreSQL-DB-Instance installiert ist, wird sie mit aws_lambda wie folgt installiert.

   ```
   CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
   NOTICE:  installing required extension "aws_commons"
   CREATE EXTENSION
   ```

Die aws_lambda-Erweiterung wird in Ihrer DB-Instance installiert. Sie können jetzt praktische Strukturen für den Aufruf Ihrer Lambda-Funktionen erstellen.

Schritt 4: Verwenden von Lambda-Hilfsfunktionen mit Ihrer RDS-for-PostgreSQL-DB-Instance (optional)

Sie können die Hilfsfunktionen in der aws_commons-Erweiterung verwenden, um Entitäten vorzubereiten, die sich einfacher über PostgreSQL aufrufen lassen. Hierzu sind die folgenden Informationen zu Ihren Lambda-Funktionen erforderlich:

- Funktionsname – Der Name, der Amazon-Ressourcenname (ARN), die Version oder der Alias der Lambda-Funktion. Die in Schritt 2: Konfigurieren von IAM für Ihre Instance und Lambda erstellte IAM-Richtlinie benötigt den ARN, daher empfehlen wir Ihnen, den ARN Ihrer Funktion zu verwenden.
- AWS Region — (Optional) Die AWS Region, in der sich die Lambda-Funktion befindet, wenn sie sich nicht in derselben Region wie Ihre befindet.

Um die Daten zum Lambda-Funktionsnamen zu speichern, verwenden Sie die aws_commons.create_lambda_function_arn-Funktion. Diese Hilfsfunktion erstellt eine zusammengesetzte `aws_commons._lambda_function_arn_1`-Struktur mit den Details, die von der Aufruffunktion benötigt werden. Im Folgenden finden Sie drei alternative Ansätze zum Einrichten dieser zusammengesetzten Struktur.

```
SELECT aws_commons.create_lambda_function_arn(
   'my-function',
   'aws-region'
) AS aws_lambda_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   '111122223333:function:my-function',
   'aws-region'
) AS lambda_partial_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   'arn:aws:lambda:aws-region:111122223333:function:my-function'
) AS lambda_arn_1 \gset
```

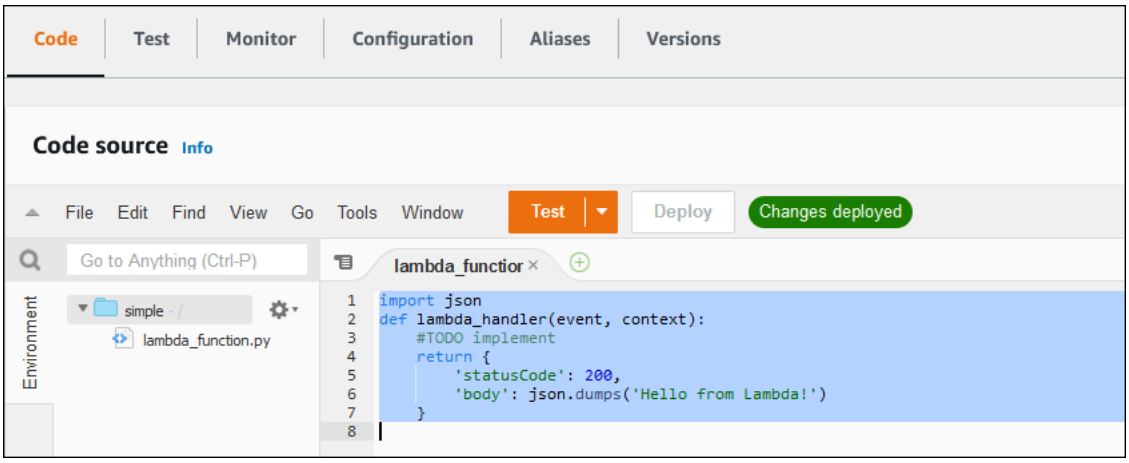
Jeder dieser Werte kann in Aufrufen der aws_lambda.invoke-Funktion verwendet werden. Beispiele finden Sie unter Schritt 5: Aufrufen einer Lambda-Funktion von Ihrem RDS-for-PostgreSQL-DB-Instance.

Schritt 5: Aufrufen einer Lambda-Funktion von Ihrem RDS-for-PostgreSQL-DB-Instance

Die `aws_lambda.invoke`-Funktion verhält sich synchron oder asynchron, je nach `invocation_type`. Die beiden Alternativen für diesen Parameter sind `RequestResponse` (der Standardwert) und `Event`, wie folgt.

- **`RequestResponse`** – Dieser Aufrufstyp ist synchron. Dies ist das Standardverhalten, wenn der Aufruf erfolgt, ohne einen Aufruftyp anzugeben. Die Antwort-Nutzlast beinhaltet die Ergebnisse der `aws_lambda.invoke`-Funktion. Verwenden Sie diesen Aufruftyp, wenn Ihr Workflow Ergebnisse von der Lambda-Funktion erhalten muss, bevor er fortfährt.
- **`Event`** – Dieser Aufrufstyp ist asynchron. Die Antwort umfasst keine Nutzlast, die Ergebnisse enthält. Verwenden Sie diesen Aufruftyp, wenn Ihr Workflow kein Ergebnis der Lambda-Funktion benötigt, um die Verarbeitung fortzusetzen.

Als einfacher Test Ihres Setups können Sie mittels `psql` eine Verbindung mit Ihrer DB-Instance herstellen und eine Beispielfunktion über die Befehlszeile aufrufen. Angenommen, Sie haben eine der Grundfunktionen Ihres Lambda-Services eingerichtet, z. B. die einfache Python-Funktion, die im folgenden Screenshot gezeigt wird.



Beispielfunktion aufrufen

1. Stellen Sie per `psql` oder pgAdmin eine Verbindung zu Ihrer DB-Instance her.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Rufen Sie die Funktion mit ihrem ARN auf.

   ```
   SELECT * from
    aws_lambda.invoke(aws_commons.create_lambda_function_arn('arn:aws:lambda:aws-
   region:444455556666:function:simple', 'us-west-1'), '{"body": "Hello from
    Postgres!"}'::json );
   ```

   Die Antwort sieht wie folgt aus.

```
status_code |                          payload                          |
 executed_version | log_result
-------------+-----------------------------------------------------------
+------------------+------------
         200 | {"statusCode": 200, "body": "\"Hello from Lambda!\""} | $LATEST
     |
(1 row)
```

Schlägt der Aufruf fehl, siehe Fehlermeldungen von Lambda-Funktionen .

Schritt 6: Erteilen der Berechtigung, Lambda-Funktionen aufzurufen, für andere Benutzer

An dieser Stelle in den Prozeduren können nur Sie als `rds_superuser` Ihre Lambda-Funktionen aufrufen. Damit andere Benutzer alle von Ihnen erstellten Funktionen aufrufen können, müssen Sie ihnen die entsprechende Berechtigung erteilen.

Gewähren der Berechtigung zum Aufrufen von Lambda-Funktionen

1. Stellen Sie per `psql` oder pgAdmin eine Verbindung zu Ihrer DB-Instance her.

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. Führen Sie die folgenden SQL-Befehle aus:

```
postgres=>  GRANT USAGE ON SCHEMA aws_lambda TO db_username;
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA aws_lambda TO db_username;
```

Beispiele: Aufrufen von Lambda-Funktionen aus Ihrem SQL

Nachfolgend finden Sie einige Beispiele für den Aufruf der aws_lambda.invoke-Funktion. Die meisten Beispiele verwenden die zusammengesetzte Struktur`aws_lambda_arn_1`, in der Sie erstellen, um die Übergabe der Schritt 4: Verwenden von Lambda-Hilfsfunktionen mit Ihrer RDS-for-PostgreSQL-DB-Instance (optional) Funktionsdetails zu vereinfachen. Ein Beispiel für einen asynchronen Aufruf finden Sie unter Beispiel: Asynchroner Ereignisaufruf (Event) von Lambda-Funktionen. Alle anderen aufgelisteten Beispiele verwenden einen synchronen Aufruf.

Weitere Informationen zu Lambda-Aufruftypen finden Sie unter Aufrufen von Lambda-Funktionen in der AWS Lambda Leitfaden für Entwickler. Mehr über `aws_lambda_arn_1` erfahren Sie unter aws_commons.create_lambda_function_arn.

Liste mit Beispielen

- Beispiel: Synchroner (RequestResponse) -Aufruf von Lambda-Funktionen
- Beispiel: Asynchroner Ereignisaufruf (Event) von Lambda-Funktionen
- Beispiel: Erfassen des Lambda-Ausführungsprotokolls in einer Funktionsantwort
- Beispiel: Einschließen von Client-Kontext in einer Lambda-Funktion
- Beispiel: Aufrufen einer bestimmten Version einer Lambda-Funktion

Beispiel: Synchroner (RequestResponse) -Aufruf von Lambda-Funktionen

Es folgen zwei Beispiele für einen synchronen Lambda-Funktionsaufruf. Die Ergebnisse dieser `aws_lambda.invoke`-Funktionen sind identisch.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json);
```

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse');
```

Die Parameter werden wie folgt beschrieben:

- `:'aws_lambda_arn_1'` – Dieser Parameter identifiziert die zusammengesetzte Struktur, die in Schritt 4: Verwenden von Lambda-Hilfsfunktionen mit Ihrer RDS-for-PostgreSQL-DB-Instance (optional) erstellt wurde, mit der `aws_commons.create_lambda_function_arn`-Hilfsfunktion. Sie können diese Struktur wie folgt auch in Ihren `aws_lambda.invoke`-Aufruf einbinden.

  ```
  SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function',
   'aws-region'),
  '{"body": "Hello from Postgres!"}'::json
  );
  ```

- `'{"body": "Hello from PostgreSQL!"}'::json`— Die JSON Nutzlast, die an die Lambda-Funktion übergeben werden soll.
- `'RequestResponse'` – Der Lambda Aufruftyp.

Beispiel: Asynchroner Ereignisaufruf (Event) von Lambda-Funktionen

Es folgt ein Beispiel für einen asynchronen Lambda-Funktionsaufruf. Der `Event`-Aufruftyp plant den Lambda-Funktionsaufruf mit der angegebenen Eingabe-Nutzlast und sofort Rückgabe. Verwenden Sie den `Event`-Aufruftyp in bestimmten Workflows, die nicht von den Ergebnissen der Lambda-Funktion abhängen.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'Event');
```

Beispiel: Erfassen des Lambda-Ausführungsprotokolls in einer Funktionsantwort

Sie können die letzten 4 KB des Ausführungsprotokolls in die Funktionsantwort aufnehmen, indem Sie den Parameter `log_type` in Ihrem `aws_lambda.invoke`-Funktionsaufruf verwenden. Standardmäßig ist dieser Parameter auf `None` festgelegt, aber Sie können `Tail` angeben, um die Ergebnisse des Lambda-Ausführungsprotokolls in der Antwort zu erfassen, wie nachfolgend gezeigt.

```
SELECT *, select convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Legen Sie den Parameter aws_lambda.invoke der `log_type`-Funktion auf `Tail`, um das Ausführungsprotokoll in die Antwort aufzunehmen. Der Standardwert für diesen `log_type`-Parameter ist `None`.

Das `log_result`, was zurückgegeben wird, ist eine `base64` codierte Zeichenfolge. Sie können den Inhalt mit einer Kombination der Funktionen `decode` und `convert_from` Postgre dekodieren. SQL

Mehr über `log_type` erfahren Sie unter aws_lambda.invoke.

Beispiel: Einschließen von Client-Kontext in einer Lambda-Funktion

Die `aws_lambda.invoke`-Funktion hat einen `context`-Parameter, den Sie verwenden können, um Informationen getrennt von der Nutzlast zu übergeben, wie nachfolgend gezeigt.

```
SELECT *, convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Um den Client-Kontext einzubeziehen, verwenden Sie ein JSON Objekt als Parameter der aws_lambda.invoke `context` Funktion.

Weitere Informationen zum `context`-Parameter finden Sie in der aws_lambda.invoke-Referenz.

Beispiel: Aufrufen einer bestimmten Version einer Lambda-Funktion

Sie können eine bestimmte Version einer Lambda-Funktion angeben, indem Sie den `qualifier`-Parameter mit dem `aws_lambda.invoke`-Aufruf einschließen. Im Folgenden finden Sie ein Beispiel, das diese Funktion mit `'custom_version'` als Alias für die Version erfüllt.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse', 'None', NULL, 'custom_version');
```

Sie können auch einen Lambda-Funktionsqualifizierer wie folgt mit den Daten zum Funktionsnamen angeben.

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-
function:custom_version', 'us-west-2'),
'{"body": "Hello from Postgres!"}'::json);
```

Weitere Informationen zu `qualifier` und anderen Parametern finden Sie in der aws_lambda.invoke-Referenz.

Fehlermeldungen von Lambda-Funktionen

In der folgenden Liste finden Sie Informationen zu Fehlermeldungen sowie mögliche Ursachen und Lösungen.

- VPCProbleme mit der Konfiguration

  VPCKonfigurationsprobleme können beim Verbindungsversuch zu den folgenden Fehlermeldungen führen:

  ```
  ERROR:  invoke API failed
  DETAIL: AWS Lambda client returned 'Unable to connect to endpoint'.
  CONTEXT:  SQL function "invoke" statement 1
  ```

Eine häufige Ursache für diesen Fehler ist eine falsch konfigurierte VPC Sicherheitsgruppe. Stellen Sie sicher, dass Sie eine ausgehende Regel für das TCP Öffnen auf Port 443 Ihrer VPC Sicherheitsgruppe haben, damit Sie eine Verbindung zum VPC Lambda herstellen VPC können.

Wenn Ihre DB-Instance privat ist, überprüfen Sie das private DNS Setup für Ihre. VPC Stellen Sie sicher, dass Sie den `rds.custom_dns_resolution` Parameter auf 1 setzen und einrichten AWS PrivateLink wie unter beschriebenSchritt 1: Konfigurieren Sie Ihren für ausgehende Verbindungen zu AWS Lambda. Weitere Informationen finden Sie unter VPC Schnittstellenendpunkte (AWS PrivateLink).

- Fehlende Berechtigungen, die zum Aufrufen von Lambda-Funktionen erforderlich sind

  Wenn eine der folgenden Fehlermeldungen angezeigt wird, verfügt der Benutzer (Rolle), der die Funktion aufruft, nicht über die entsprechenden Berechtigungen.

  ```
  ERROR:  permission denied for schema aws_lambda
  ```

  ```
  ERROR:  permission denied for function invoke
  ```

  Ein Benutzer (Rolle) muss bestimmte Berechtigungen erhalten, um Lambda-Funktionen aufrufen zu können. Weitere Informationen finden Sie unter Schritt 6: Erteilen der Berechtigung, Lambda-Funktionen aufzurufen, für andere Benutzer.

- Unsachgemäße Handhabung von Fehlern in Ihren Lambda-Funktionen

  Wenn eine Lambda-Funktion während der Anforderungsverarbeitung eine Ausnahme auslöst, `aws_lambda.invoke` schlägt sie mit einem SQL Postgre-Fehler wie dem folgenden fehl.

  ```
  SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
   Postgres!"}'::json);
  ERROR:  lambda invocation failed
  DETAIL:  "arn:aws:lambda:us-west-2:555555555555:function:my-function" returned error
   "Unhandled", details: "<Error details string>".
  ```

  Achten Sie darauf, Fehler in Ihren Lambda-Funktionen oder in Ihrer Postgre-Anwendung zu behandeln. SQL

AWS Lambda -Funktion und Parameterreferenz

Im Folgenden finden Sie die Referenz für die Funktionen und Parameter, die zum Aufrufen von Lambda mit RDS für PostgreSQL verwendet werden sollen.

Funktionen und Parameter

- aws_lambda.invoke
- aws_commons.create_lambda_function_arn
- aws_lambda-Parameter

aws_lambda.invoke

Führt eine Lambda-Funktion für eine für PostgreSQL-DB-Instance aus.

Weitere Informationen zum Aufrufen von Lambda-Funktionen finden Sie unter Invoke (Aufrufen) auch im AWS Lambda-Entwicklerhandbuch.

Syntax

JSON

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSON,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSON,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
```

```
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

JSONB

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSONB,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSONB,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT
)
```

Eingabeparameter

function_name

Der spezifizierte Name der Lambda-Funktion. Der Wert kann der Funktionsname, ein ARN oder ein partieller ARN sein. Eine Auflistung möglicher Formate finden Sie unter Lambda Funktionsnamenformate im AWS Lambda-Entwicklerhandbuch.

Nutzlast

Die Eingabe für die Funktion Lambda. Das Format kann JSON oder JSONB sein. Weitere Informationen finden Sie in der PostgreSQL-Dokumentation zu JSON Types.

region

(Optional) Die Lambda-Region für die Funktion. Standardmäßig verwendet RDS die AWS-Region aus dem vollständigen ARN in der `function_name` oder die RDS for PostgreSQL-DB-Instance-Region. Wenn dieser Region-Wert mit dem im `function_name` ARN angegebenen Wert in Konflikt steht, wird ein Fehler ausgelöst.

invocation_type

Die Aufruftyp der Lambda-Funktion. Bei -Wert ist die Groß- und Kleinschreibung zu beachten. Die folgenden Werte sind möglich:

- `RequestResponse` – Der Standardwert. Diese Art des Aufrufens für eine Lambda-Funktion ist synchron und gibt eine Antwortnutzlast im Ergebnis zurück. Verwenden Sie den `RequestResponse` Aufruftyp, wenn Ihr Workflow vom sofortigen Erhalt des Lambda-Funktionsergebnisses abhängt.
- `Event` – Diese Art des Aufrufs für eine Lambda-Funktion ist asynchron und wird sofort ohne Rückgabe einer Nutzlast zurückgegeben. Verwenden Sie den `Event`-Aufruftyp, wenn Sie keine Ergebnisse der Lambda-Funktion benötigen, bevor Ihr Workflow weitergeht.
- `DryRun` – Diese Art des Aufrufs testet den Zugriff, ohne die Lambda-Funktion auszuführen.

log_typ

Der Typ des Lambda-Protokolls, das im Ausgabeparameter `log_result` ausgegeben werden soll. Bei -Wert ist die Groß- und Kleinschreibung zu beachten. Die folgenden Werte sind möglich:

- Tail – Der zurückgegebene Ausgabeparameter `log_result` enthält die letzten 4 KB des Ausführungsprotokolls.
- Keiner – Es werden keine Lambda-Protokollinformationen zurückgegeben.

context

Client-Kontext im JSON- oder JSONB-Format. Zu verwendende Felder sind dann `custom` und `env`.

Qualifier

Ein Qualifier, der die aufzurufende Version einer Lambda-Funktion spezifiziert. Wenn dieser Wert mit einem im `function_name` ARN angegebenen Wert in Konflikt steht, wird ein Fehler ausgelöst.

Ausgabeparameter

status_code

Ein HTTP-Status-Antwortcode. Weitere Informationen finden Sie unter Lambda Antwortelemente aufrufen im AWS Lambda-Entwicklerhandbuch.

Nutzlast

Die von der ausgeführten Lambda-Funktion zurückgegebenen Daten. Das Format ist in JSON oder JSONB.

executed_version

Die Version der Lambda-Funktion, die ausgeführt wurde.

log_resultat

Die Ausführungsprotokollinformationen werden zurückgegeben, wenn der Wert `log_type` beim Aufruf der Lambda-Funktion `Tail` beträgt. Das Ergebnis enthält die letzten 4 KB des in Base64 codierten Ausführungsprotokolls.

aws_commons.create_lambda_function_arn

Erstellt eine `aws_commons._lambda_function_arn_1`-Struktur für Daten zum Lambda Funktionsnamen. Sie können die Ergebnisse der `aws_commons.create_lambda_function_arn`-Funktion im Parameter `function_name` der aws_lambda.invoke-Funktion aws_lambda.invoke verwenden.

Syntax

```
aws_commons.create_lambda_function_arn(
    function_name TEXT,
    region TEXT DEFAULT NULL
    )
    RETURNS aws_commons._lambda_function_arn_1
```

Eingabeparameter

function_name

Eine erforderliche Textzeichenfolge mit dem Lambda-Funktionsnamen. Der Wert kann ein Funktionsname, ein partieller ARN oder ein vollständiger ARN sein.

region

Eine optionale Textzeichenfolge mit der AWS-Region, in der sich die Lambda-Funktion befindet. Eine Liste der -Regionsnamen und der zugehörigen Werte finden Sie unter Regionen, Availability Zones und Local Zones.

aws_lambda-Parameter

In dieser Tabelle finden Sie Parameter, die der `aws_lambda` Funktion zugeordnet sind.

Parameter	Beschreibung
`aws_lambda.connect_timeout_ms`	Dies ist ein dynamischer Parameter, der die maximale Wartezeit beim Herstellen einer Verbindung mit AWS Lambda festlegt. Die Standardwerte sind `1000`. Zulässige Werte für diesen Parameter sind 1 bis 90 000.
`aws_lambda.request_timeout_ms`	Dies ist ein dynamischer Parameter und er legt die maximale Wartezeit fest, während er auf die Antwort von AWS Lambda wartet. Die Standardwerte sind `3000`. Zulässige Werte für diesen Parameter sind 1 bis 90 000.
`aws_lambda.endpoint_override`	Gibt den Endpunkt an, der für die Verbindung mit AWS Lambda verwendet werden kann. Eine leere Zeichenfolge wählt den StandardAWS-Lambda-Endpunkt für die Region aus. Sie müssen die Datenbank neu starten, damit diese statische Parameteränderung wirksam wird.

Häufige DBA-Aufgaben für Amazon RDS for PostgreSQL

Datenbankadministratoren (DBAs) führen bei der Verwaltung einer Amazon RDS for PostgreSQL PostgreSQL-DB-Instance eine Vielzahl von Aufgaben aus. Wenn Sie als DBA bereits mit PostgreSQL vertraut sind, müssen Sie sich einige wichtige Unterschiede zwischen dem Ausführen von PostgreSQL auf Ihrer Hardware und RDS for PostgreSQL beachten. Da es sich beispielsweise um einen verwalteten Service handelt, lässt Amazon RDS keinen Shell-Zugriff auf Ihre DB-Instances zu. Das bedeutet, dass Sie keinen direkten Zugriff auf `pg_hba.conf` und andere Konfigurationsdateien haben. Bei RDS für PostgreSQL werden Änderungen, die normalerweise an der PostgreSQL-Konfigurationsdatei einer lokalen Instance vorgenommen werden, an einer benutzerdefinierten DB-Parametergruppe vorgenommen, die der DB-Instance von RDS für PostgreSQL zugeordnet ist. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

Darüber hinaus können Sie nicht auf die gleiche Weise auf Protokolldateien zugreifen wie mit einer lokalen PostgreSQL-Instance. Weitere Informationen zur Protokollierung finden Sie unter Datenbank-Protokolldateien von RDS für PostgreSQL.

Sie haben beispielsweise auch keinen Zugriff auf das `superuser`-Konto von PostgreSQL. Bei RDS for PostgreSQL ist die `rds_superuser`-Rolle diejenige mit den meisten Berechtigungen und wird `postgres` zum Zeitpunkt der Einrichtung gewährt. Unabhängig davon, ob Sie mit der On-Premises-Verwendung von PostgreSQL vertraut sind oder Neueinsteiger bei RDS for PostgreSQL sind, empfehlen wir Ihnen, sich mit der `rds_superuser`-Rolle und dem Umgang mit Rollen, Benutzern, Gruppen und Berechtigungen vertraut zu machen. Weitere Informationen finden Sie unter Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen.

Im Folgenden sind einige häufige DBA-Aufgaben für RDS for PostgreSQL aufgeführt.

Themen

- Kollationen, die in RDS unterstützt werden SQL
- Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen
- Behandlung toter Verbindungen in PostgreSQL
-
- Arbeiten mit Protokollierungsmechanismen, die von RDS for PostgreSQL unterstützt werden
- Verwalten temporärer Dateien mit PostgreSQL
- Verwenden von pgBadger für die Protokollanalyse mit PostgreSQL
- PGSnapper Für die Überwachung von PostgreSQL verwenden

- Arbeiten mit Parametern auf der DB-Instance von RDS for PostgreSQL

Kollationen, die in RDS unterstützt werden SQL

Sortierungen sind eine Reihe von Regeln, die bestimmen, wie in der Datenbank gespeicherte Zeichenfolgen sortiert und verglichen werden. Sortierungen spielen eine grundlegende Rolle im Computersystem und sind Teil des Betriebssystems. Sortierungen ändern sich im Laufe der Zeit, wenn neue Zeichen zu Sprachen hinzugefügt werden oder wenn sich die Sortierregeln ändern.

Sortierungsbibliotheken definieren spezifische Regeln und Algorithmen für eine Sortierung. Die beliebtesten Kollationsbibliotheken, die in Postgre verwendet werden, SQL sind GNU C (Glibc) und Internationalisierungskomponenten für Unicode (). ICU Standardmäßig SQL verwendet die Glibc-Kollatierung, die Unicode-Zeichensortierreihenfolgen für Mehrbyte-Zeichenfolgen enthält.

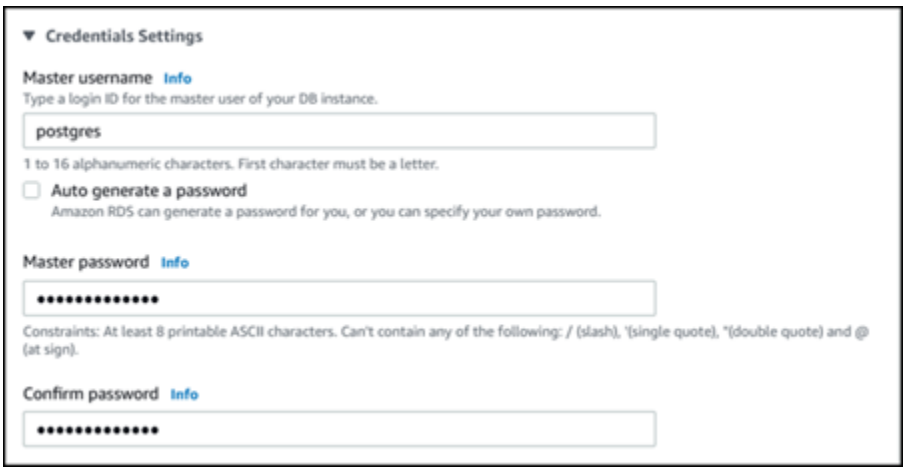
Wenn Sie eine neue für Postgre erstellenSQL, überprüft sie das Betriebssystem auf die verfügbare Sortierung. Die SQL Postgre-Parameter des `CREATE DATABASE` Befehls `LC_COLLATE` und `LC_CTYPE` werden verwendet, um eine Kollation anzugeben, die in dieser Datenbank als Standardkollation gilt. Alternativ können Sie auch den Parameter `LOCALE` in `CREATE DATABASE` verwenden, um diese Parameter festzulegen. Dieser bestimmt die Standardsortierung für Zeichenfolgen in der Datenbank und die Regeln für die Klassifizierung von Zeichen als Buchstaben, Zahlen oder Symbole. Sie können auch eine Sortierung auswählen, die für eine Spalte, einen Index oder eine Abfrage verwendet werden soll.

for Postgre ist für die Kollationsunterstützung von der Glibc-Bibliothek im Betriebssystem SQL abhängig. for SQL Postgre-Instanz wird regelmäßig mit den neuesten Versionen des Betriebssystems aktualisiert. Diese Updates umfassen manchmal eine neuere Version der Glibc-Bibliothek. In seltenen Fällen ändern neuere Versionen von Glibc die Sortierreihenfolge oder Sortierung einiger Zeichen, was dazu führen kann, dass die Daten anders sortiert werden oder ungültige Indexeinträge entstehen. Wenn Sie während eines Updates bei der Sortierung Probleme mit der Sortierreihenfolge feststellen, müssen Sie möglicherweise die Indizes neu erstellen.

Um die möglichen Auswirkungen der Glibc-Updates zu reduzieren, enthält jetzt eine unabhängige Standardkollationsbibliothek. Diese Kollationsbibliothek ist in 14.6, 13.9, 12.13, 11.18, 10.23 und neuere Nebenversionen verfügbar. Sie ist mit Glibc 2.26-59.amzn2 kompatibel und bietet eine stabile Sortierreihenfolge, um falsche Abfrageergebnisse zu verhindern.

Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen

Wenn Sie mit dem eine for PostgreSQL-DB-Instance erstellen AWS Management Console, wird gleichzeitig ein Administratorkonto erstellt. Der Name lautet standardmäßig `postgres`, wie im folgenden Screenshot gezeigt:



Sie können einen anderen Namen auswählen, anstatt den Standard (`postgres`) beizubehalten. In diesem Fall muss der von Ihnen gewählte Name mit einem Buchstaben beginnen und zwischen 1 und 16 alphanumerische Zeichen umfassen. Der Einfachheit halber verwenden wir für das Hauptbenutzerkonto den Standardwert (`postgres`) in diesem Handbuch.

Wenn Sie den `create-db-instance` AWS CLI anstelle von verwenden, erstellen Sie den Namen AWS Management Console, indem Sie ihn zusammen mit dem `master-username` Parameter im Befehl übergeben. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Unabhängig davon AWS Management Console, ob Sie die AWS CLI, oder die Amazon RDS-API verwenden und ob Sie den `postgres` Standardnamen verwenden oder einen anderen Namen wählen, ist dieses erste Datenbankbenutzerkonto Mitglied der `rds_superuser` Gruppe und verfügt über `rds_superuser` Berechtigungen.

Themen

- Die Rolle „rds_superuser“ verstehen
- Steuern des Benutzerzugriffs auf die PostgreSQL−Datenbank
- Delegieren und Steuern der Benutzerpasswortverwaltung
- Verwenden von SCRAM für die PostgreSQL-Passwortverschlüsselung

Die Rolle „rds_superuser“ verstehen

In PostgreSQL kann eine Rolle einen Benutzer, eine Gruppe oder einen Satz bestimmter Berechtigungen definieren, die einer Gruppe oder einem Benutzer für verschiedene Objekte in der Datenbank gewährt werden. PostgreSQL-Befehle für `CREATE USER` und `CREATE GROUP` wurden durch den allgemeineren Befehl `CREATE ROLE` mit bestimmten Eigenschaften zur Unterscheidung von Datenbankbenutzern ersetzt. Einen Datenbankbenutzer kann man sich als Rolle mit der LOGIN-Berechtigung vorstellen.

> **Note**
>
> Die Befehle `CREATE USER` und `CREATE GROUP` können weiterhin verwendet werden. Weitere Informationen dazu finden Sie im Abschnitt Datenbankrollen der PostgreSQL-Dokumentation.

Der `postgres`-Benutzer ist der Datenbankbenutzer mit den meisten Berechtigungen auf Ihrer RDS-for-PostgreSQL-DB-Instance. Er verfügt über die Eigenschaften, die durch die folgende `CREATE ROLE`-Anweisung definiert sind.

```
CREATE ROLE postgres WITH LOGIN NOSUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION
 VALID UNTIL 'infinity'
```

Die Eigenschaften `NOSUPERUSER`, `NOREPLICATION`, `INHERIT` und `VALID UNTIL 'infinity'` sind die Standardoptionen für CREATE ROLE, sofern nicht anders angegeben.

Standardmäßig verfügt `postgres` über Berechtigungen, die der `rds_superuser`-Rolle gewährt wurden, und über Berechtigungen zum Erstellen von Rollen und Datenbanken. Die `rds_superuser`-Rolle erlaubt dem `postgres`-Benutzer, folgende Aktionen auszuführen:

- Erweiterungen für die Verwendung mit Amazon RDS hinzufügen. Weitere Informationen finden Sie unter Arbeiten mit SQL Postgre-Funktionen, die von Amazon RDS für Postgre unterstützt werden SQL
- Rollen für Benutzer erstellen und Benutzern Berechtigungen gewähren. Weitere Informationen dazu finden Sie im Abschnitt CREATE ROLE und GRANT der PostgreSQL-Dokumentation.
- Datenbanken erstellen. Weitere Informationen finden Sie im Abschnitt CREATE DATABASE der PostgreSQL-Dokumentation.

- Gewähren Sie `rds_superuser`-Berechtigungen anderen Benutzerrollen, die nicht über diese Berechtigungen verfügen, und widerrufen Sie diese Berechtigungen bei Bedarf. Es wird empfohlen, diese Rolle nur denjenigen Benutzern zu gewähren, die Superuser-Aufgaben ausführen. Mit anderen Worten, Sie können diese Rolle Datenbankadministratoren (DBAs) oder Systemadministratoren zuweisen.
- Die `rds_replication`-Rolle Datenbankbenutzern gewähren (oder entziehen), die nicht über die `rds_superuser`-Rolle verfügen.
- Die `rds_password`-Rolle Datenbankbenutzern gewähren (oder entziehen), die nicht über die `rds_superuser`-Rolle verfügen.
- Statusinformationen über alle Datenbankverbindungen über die Ansicht `pg_stat_activity` abrufen. Bei Bedarf kann `rds_superuser` alle Verbindungen mit `pg_terminate_backend` oder `pg_cancel_backend` stoppen.

In der `CREATE ROLE postgres...`-Anweisung können Sie sehen, dass die `postgres`-Benutzerrolle PostgreSQL ausdrücklich `superuser`-Berechtigungen verweigert. RDS for PostgreSQL ist ein verwalteter Service, sodass Sie nicht auf das Host-Betriebssystem zugreifen können und keine Verbindung mit dem `superuser`-PostgreSQL-Konto herstellen können. Viele der Aufgaben, die `superuser`-Zugriff auf einem eigenständigem PostgreSQL erfordern, werden von Amazon RDS automatisch verwaltet.

Weitere Informationen zum Gewähren von Berechtigungen finden Sie unter GRANT in der PostgreSQL-Dokumentation.

Die `rds_superuser`-Rolle ist eine von mehreren vordefinierten Rollen in einem . DB-Instance von RDS for PostgreSQL

> **Note**
>
> In PostgreSQL 13 und früheren Versionen werden vordefinierte Rollen als Standardrollen bezeichnet.

In der folgenden Liste finden Sie einige der anderen vordefinierten Rollen, die automatisch für einen neuen erstellt werden. DB-Instance von RDS for PostgreSQL Vordefinierte Rollen und ihre Berechtigungen können nicht geändert werden. Sie können Berechtigungen für diese vordefinierten Rollen nicht löschen, umbenennen oder ändern. Jeder entsprechende Versuch führt zu einem Fehler.

- rds_passwort – Eine Rolle, die Passwörter ändern und Passwortbeschränkungen für Datenbankbenutzer einrichten kann. Die `rds_superuser` Rolle wird standardmäßig zusammen mit dieser Rolle zugewiesen und kann Datenbankbenutzern zugewiesen werden. Weitere Informationen finden Sie unter Steuern des Benutzerzugriffs auf die PostgreSQL−Datenbank.
  - Für RDS for PostgreSQL-Versionen, die älter als 14 sind, kann die `rds_password` Rolle Passwörter ändern und Passwortbeschränkungen für Datenbankbenutzer und Benutzer mit `rds_superuser` Rolle einrichten. Ab RDS für PostgreSQL Version 14 und höher kann die `rds_password` Rolle nur für Datenbankbenutzer Passwörter ändern und Passwortbeschränkungen einrichten. Nur Benutzer mit `rds_superuser` Rolle können diese Aktionen für andere Benutzer mit `rds_superuser` Rolle ausführen.
- rdsadmin – Eine Rolle, die erstellt wurde, um viele der Verwaltungsaufgaben zu erledigen, die der Administrator mit `superuser`-Berechtigungen für eine eigenständige PostgreSQL-Datenbank ausführt. Diese Rolle wird intern von RDS for PostgreSQL für viele Verwaltungsaufgaben verwendet.
- rdstopmgr – Eine Rolle, die intern von Amazon RDS zur Unterstützung von Multi-AZ-Bereitstellungen verwendet wird.
- rds_reserved — Eine Rolle, die intern von Amazon RDS verwendet wird, um Datenbankverbindungen zu reservieren.

Rollen und ihre Rechte anzeigen

Sie können vordefinierte Rollen und ihre Rechte in Ihrer RDS for PostgreSQL-DB-Instance mit unterschiedlichen Befehlen anzeigen, abhängig von Ihrer PostgreSQL-Version. Um alle vordefinierten Rollen zu sehen, können Sie eine Verbindung zu Ihrer RDS for PostgreSQL-DB-Instance herstellen und die folgenden Befehle mit dem ausführen. `psql`

Für **`psql`** 15 und frühere Versionen

Connect zu Ihrer RDS for PostgreSQL-DB-Instance her und verwenden Sie den `\du` Befehl in psql:

```
postgres=> \du
                                                    List of roles
    Role name    |                         Attributes                         |
                 Member of
-----------------+------------------------------------------------------------
+-------------------------------------------------
 postgres        | Create role, Create DB                                     +|
 {rds_superuser}
```

```
                | Password valid until infinity                                  |
 rds_ad         | Cannot login                                                   | {}
 rds_iam        | Cannot login                                                   | {}
 rds_password   | Cannot login                                                   | {}
 rds_replication | Cannot login                                                  | {}
 rds_superuser  | Cannot login                                                   |
 {pg_monitor,pg_signal_backend,rds_password,rds_replication}
 rdsadmin       | Superuser, Create role, Create DB, Replication, Bypass RLS+| {}
                | Password valid until infinity                                  |
```

Für **`psql`** 16 und spätere Versionen

```
postgres=> \drg+
                            List of role grants
   Role name   |          Member of          |       Options       | Grantor
---------------+-----------------------------+---------------------+----------
 postgres      | rds_superuser               | INHERIT, SET        | rdsadmin
 rds_superuser | pg_checkpoint               | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | pg_monitor                  | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | pg_signal_backend           | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | pg_use_reserved_connections | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | rds_password                | ADMIN, INHERIT, SET | rdsadmin
 rds_superuser | rds_replication             | ADMIN, INHERIT, SET | rdsadmin
```

Um die Rollenmitgliedschaft ohne Versionsabhängigkeit zu überprüfen, können Sie die folgende SQL-Abfrage verwenden:

```
SELECT m.rolname AS "Role name", r.rolname AS "Member of"
FROM pg_catalog.pg_roles m
JOIN pg_catalog.pg_auth_members pam ON (pam.member = m.oid)
LEFT JOIN pg_catalog.pg_roles r ON (pam.roleid = r.oid)
LEFT JOIN pg_catalog.pg_roles g ON (pam.grantor = g.oid)
WHERE m.rolname !~ '^pg_'
ORDER BY 1, 2;
```

In der Ausgabe sehen Sie, dass `rds_superuser` keine Datenbankbenutzerrolle ist (sie kann sich nicht anmelden), aber über die Berechtigungen vieler anderer Rollen verfügt. Sie können auch sehen, dass dieser Datenbankbenutzer `postgres` Mitglied der `rds_superuser`-Rolle ist. Wie bereits erwähnt, ist `postgres` der Standardwert auf der Seite Create database (Datenbank erstellen) der Amazon-RDS-Konsole. Wenn Sie einen anderen Namen gewählt haben, wird dieser Name stattdessen in der Rollenliste angezeigt.

Steuern des Benutzerzugriffs auf die PostgreSQL−Datenbank

Neue Datenbanken in PostgreSQL werden immer mit Standardberechtigungen im `public`-Schema der Datenbank erstellt, mit dem alle Datenbankbenutzer und -rollen Objekte erstellen können. Diese Berechtigungen ermöglichen es Datenbankbenutzern, eine Verbindung mit der Datenbank herzustellen und während der Verbindung temporäre Tabellen zu erstellen.

Es wird empfohlen, diese `public`-Standardberechtigungen zu widerrufen, um den Benutzerzugriff auf die Datenbank-Instances, die Sie auf Ihrer DB-Instance von RDS for PostgreSQL erstellen, besser kontrollieren können. Danach erteilen Sie Datenbankbenutzern auf einer detaillierteren Basis spezifische Berechtigungen, wie im Folgenden gezeigt.

So richten Sie Rollen und Berechtigungen für eine neue Datenbank-Instance ein

Angenommen, Sie richten eine Datenbank für eine neu erstellte DB-Instance von RDS for PostgreSQL ein, die von mehreren Forschenden verwendet wird, die alle Lese-/Schreibzugriff auf die Datenbank benötigen.

1. Verwenden Sie `psql` (oder pgAdmin) zum Herstellen einer Verbindung mit Ihrer DB-Instance von RDS for PostgreSQL:

   ```
   psql --host=your-db-instance.666666666666.aws-region.rds.amazonaws.com --port=5432
    --username=postgres --password
   ```

   Geben Sie bei der Aufforderung Ihr Passwort ein. Der `psql`-Client verbindet und zeigt die standardmäßige administrative Verbindungsdatenbank `postgres=>` als Eingabeaufforderung an.

2. Gehen Sie wie folgt vor, um zu verhindern, dass Datenbankbenutzer Objekte im `public`-Schema erstellen:

   ```
   postgres=> REVOKE CREATE ON SCHEMA public FROM PUBLIC;
   REVOKE
   ```

3. Als Nächstes erstellen Sie eine neue Datenbank-Instance:

   ```
   postgres=> CREATE DATABASE lab_db;
   CREATE DATABASE
   ```

4. Widerrufen Sie alle Berechtigungen aus dem `PUBLIC`-Schema in dieser neuen Datenbank.

```
postgres=> REVOKE ALL ON DATABASE lab_db FROM public;
REVOKE
```

5. Erstellen Sie eine Rolle für Datenbankbenutzer.

```
postgres=> CREATE ROLE lab_tech;
CREATE ROLE
```

6. Geben Sie Datenbankbenutzern mit dieser Rolle die Möglichkeit, eine Verbindung mit der Datenbank herzustellen.

```
postgres=> GRANT CONNECT ON DATABASE lab_db TO lab_tech;
GRANT
```

7. Gewähren Sie allen Benutzern mit der `lab_tech`-Rolle alle Berechtigungen für diese Datenbank.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_db TO lab_tech;
GRANT
```

8. Erstellen Sie Datenbankbenutzer wie folgt:

```
postgres=> CREATE ROLE lab_user1 LOGIN PASSWORD 'change_me';
CREATE ROLE
postgres=> CREATE ROLE lab_user2 LOGIN PASSWORD 'change_me';
CREATE ROLE
```

9. Gewähren Sie diesen beiden Benutzern die Berechtigungen, die mit der lab_tech-Rolle verknüpft sind:

```
postgres=> GRANT lab_tech TO lab_user1;
GRANT ROLE
postgres=> GRANT lab_tech TO lab_user2;
GRANT ROLE
```

An dieser Stelle können `lab_user1` und `lab_user2` eine Verbindung mit der `lab_db`-Datenbank herstellen. Dieses Beispiel folgt nicht den bewährten Methoden für den Unternehmensgebrauch, darunter das Erstellen mehrerer Datenbank-Instances, verschiedener Schemas und das Erteilen

eingeschränkter Berechtigungen. Umfassende Informationen und zusätzliche Szenarien finden Sie unter Verwalten von PostgreSQL-Benutzern und -Rollen.

Weitere Informationen zu Berechtigungen in PostgreSQL-Datenbanken finden Sie unter dem Befehl GRANT in der PostgreSQL-Dokumentation.

Delegieren und Steuern der Benutzerpasswortverwaltung

Als DBA sollten Sie ggf. die Verwaltung von Benutzerpasswörtern delegieren. Oder Sie möchten verhindern, dass Datenbankbenutzer ihre Passwörter ändern oder Passwortbeschränkungen wie die Lebensdauer des Passworts neu konfigurieren. Um sicherzustellen, dass nur die von Ihnen ausgewählten Datenbankbenutzer Passworteinstellungen ändern können, können Sie die Funktion zur eingeschränkten Passwortverwaltung aktivieren. Wenn Sie diese Funktion aktivieren, können nur die Datenbankbenutzer, denen die `rds_password`-Rolle gewährt wurde, Passwörter verwalten.

> **Note**
>
> Um die eingeschränkte Passwortverwaltung nutzen zu können, muss Ihre DB-Instance für RDS for PostgreSQL PostgreSQL 10.6 oder höher ausführen.

Standardmäßig lautet diese Funktion `off`, wie im Folgenden gezeigt:

```
postgres=> SHOW rds.restrict_password_commands;
  rds.restrict_password_commands
--------------------------------
 off
(1 row)
```

Zum Aktivieren dieser Funktion verwenden Sie eine benutzerdefinierte Parametergruppe und ändern die Einstellung für `rds.restrict_password_commands` in 1. Stellen Sie sicher, dass Sie Ihre DB-Instance von RDS for PostgreSQL neu starten, damit die Einstellung wirksam wird.

Wenn diese Funktion aktiv ist, werden für die folgenden SQL-Befehle `rds_password`-Berechtigungen benötigt:

```
CREATE ROLE myrole WITH PASSWORD 'mypassword';
CREATE ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
```

```
ALTER ROLE myrole WITH PASSWORD 'mypassword';
ALTER ROLE myrole VALID UNTIL '2023-01-01';
ALTER ROLE myrole RENAME TO myrole2;
```

Das Umbenennen einer Rolle (`ALTER ROLE myrole RENAME TO newname`) ist ebenfalls eingeschränkt, wenn das Passwort den MD5 Hash-Algorithmus verwendet.

Wenn diese Funktion aktiv ist, generiert jeder Versuch, einen dieser SQL-Befehle ohne die `rds_password`-Rollenberechtigungen auszuführen, den folgenden Fehler:

```
ERROR: must be a member of rds_password to alter passwords
```

Wir empfehlen, dass Sie die `rds_password`-Berechtigung nur wenigen Rollen zuweisen, die Sie ausschließlich für die Passwortverwaltung verwenden. Wenn Sie `rds_password`-Berechtigungen für Datenbankbenutzer erteilen, die keine `rds_superuser`-Berechtigungen haben, müssen Sie ihnen auch das `CREATEROLE`-Attribut erteilen.

Stellen Sie sicher, dass Sie die Passwortanforderungen wie Ablaufdatum und erforderliche Komplexität auf Kundenseite überprüfen. Wenn Sie Ihr eigenes clientseitiges Dienstprogramm für passwortbezogene Änderungen verwenden, muss das Dienstprogramm Mitglied von `rds_password` sein und über `CREATE ROLE`-Berechtigungen verfügen.

Verwenden von SCRAM für die PostgreSQL-Passwortverschlüsselung

Der Salted Challenge Response Authentication Mechanism (SCRAM) ist eine Alternative zum standardmäßigen Message Digest (MD5) -Algorithmus von PostgreSQL zur Verschlüsselung von Passwörtern. Der SCRAM-Authentifizierungsmechanismus gilt als sicherer als. MD5 Weitere Informationen zu diesen beiden verschiedenen Ansätzen zur Sicherung von Passwörtern finden Sie unter Passwortauthentifizierung in der PostgreSQL-Dokumentation.

Wir empfehlen, SCRAM statt MD5 als Kennwortverschlüsselungsschema für Ihren zu verwenden. DB-Instance von RDS für PostgreSQL Es ist ein kryptografischer Challenge-Response-Mechanismus, der den scram–sha-256-Algorithmus zur Passwortauthentifizierung und -verschlüsselung nutzt.

Möglicherweise müssen Sie Bibliotheken aktualisieren, damit Ihre Clientanwendungen SCRAM unterstützen können. JDBC-Versionen vor 42.2.0 unterstützen SCRAM beispielsweise nicht. Weitere Informationen finden Sie unter PostgreSQL-JDBC-Treiber in der Dokumentation zu PostgreSQL-

JDBC-Treibern. Für eine Liste anderer PostgreSQL-Treiber und SCRAM-Unterstützung siehe die Treiberliste in der PostgreSQL-Dokumentation.

RDS for PostgreSQL 13.1 und höhere Versionen unterstützen scram-sha-256. Mit diesen Versionen können Sie Ihre DB-Instance auch so konfigurieren, dass SCRAM erforderlich ist, wie in den folgenden Verfahren beschrieben.

Einrichten des DB-Instances von RDS for PostgreSQL, sodass SCRAM erforderlich ist

können Sie verlangen, dass der DB-Instance von RDS for PostgreSQL nur Passwörter akzeptiert, die den scram-sha-256-Algorithmus verwenden.

> **Important**
>
> Wenn Sie bei vorhandenen RDS-Proxys mit PostgreSQL-Datenbanken die Datenbankauthentifizierung so ändern, dass nur `SCRAM` verwendet wird, ist der Proxy für bis zu 60 Sekunden nicht verfügbar. Um das Problem zu vermeiden, führen Sie einen der folgenden Schritte aus:
>
> - Stellen Sie sicher, dass die Datenbank sowohl die `SCRAM`- als auch die `MD5`-Authentifizierung zulässt.
> - Wenn Sie nur die `SCRAM`-Authentifizierung verwenden möchten, erstellen Sie einen neuen Proxy, migrieren Sie Ihren Anwendungsdatenverkehr auf den neuen Proxy und löschen Sie dann den zuvor mit der Datenbank verknüpften Proxy.

Bevor Sie Änderungen an Ihrem System vornehmen, vergewissern Sie sich, dass Sie den folgenden Prozess komplett verstehen:

- Sammeln Sie Informationen über alle Rollen und Passwortverschlüsselung für alle Datenbankbenutzer.
- Überprüfen Sie die Parametereinstellungen für Ihren DB-Instance von RDS for PostgreSQL für die Parameter, die die Passwortverschlüsselung steuern.
- Wenn Ihr DB-Instance von RDS for PostgreSQL eine Standardparametergruppe verwendet, müssen Sie eine benutzerdefinierte DB-Parametergruppe erstellen und sie auf Ihren DB-Instance von RDS for PostgreSQL anwenden, damit Sie bei Bedarf Parameter ändern können. Wenn Ihr DB-Instance von RDS for PostgreSQL eine benutzerdefinierte Parametergruppe verwendet, können Sie die erforderlichen Parameter bei Bedarf später im Prozess ändern.

- Ändern Sie den Parameter `password_encryption` in `scram-sha-256`.
- Informieren Sie alle Datenbankbenutzer, dass sie ihre Passwörter aktualisieren müssen. Wiederholen Sie diesen Schritt für Ihr `postgres`-Konto. Die neuen Passwörter werden mit dem scram-sha-256-Algorithmus verschlüsselt und gespeichert.
- Stellen Sie sicher, dass alle Passwörter mit diesem Verschlüsselungstyp verschlüsselt sind.
- Wenn alle Passwörter scram-sha-256 verwenden, können Sie den `rds.accepted_password_auth_method`-Parameter von `md5+scram` in `scram-sha-256` ändern.

> **Warning**
>
> Nachdem Sie `rds.accepted_password_auth_method` nur in scram-sha-256 geändert haben, können Benutzer (Rollen) mit md5-verschlüsselten Passwörtern keine Verbindung herstellen.

Vorbereiten der SCRAM-Anforderung für Ihre DB-Instance von RDS for PostgreSQL

Bevor Sie Änderungen an Ihrem DB-Instance von RDS for PostgreSQL, vornehmen, überprüfen Sie alle vorhandenen Datenbankbenutzerkonten. Überprüfen Sie auch die Art der Verschlüsselung, die für Passwörter verwendet wird. Sie können für diese Aufgaben die `rds_tools`-Erweiterung verwenden. Informationen darüber, welche PostgreSQL-Versionen unterstützt werden`rds_tools`, finden Sie unter Erweiterungsversionen für Amazon RDS for PostgreSQL.

So erhalten Sie eine Liste der Datenbankbenutzer (Rollen) und Passwortverschlüsselungsmethoden

1. Verwenden Sie `psql` zum Herstellen einer Verbindung mit Ihrer DB-Instance von RDS for PostgreSQL, wie im Folgenden gezeigt.

   ```
   psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Installieren Sie die `rds_tools`-Erweiterung.

   ```
   postgres=> CREATE EXTENSION rds_tools;
   CREATE EXTENSION
   ```

3. Rufen Sie eine Auflistung der Rollen und Verschlüsselungsmethoden ab.

```
postgres=> SELECT * FROM
      rds_tools.role_password_encryption_type();
```

Die Ausgabe entspricht weitgehend der folgenden.

```
       rolname        | encryption_type
----------------------+-----------------
 pg_monitor           |
 pg_read_all_settings |
 pg_read_all_stats    |
 pg_stat_scan_tables  |
 pg_signal_backend    |
 lab_tester           | md5
 user_465             | md5
 postgres             | md5
(8 rows)
```

Erstellen einer benutzerdefinierten DB-Parametergruppe

> **Note**
>
> Wenn Ihre DB-Instance von RDS for PostgreSQL bereits eine benutzerdefinierte Parametergruppe verwendet, müssen Sie keine neue erstellen.

Eine Übersicht über Parametergruppen für Amazon RDS finden Sie unter Arbeiten mit Parametern auf der DB-Instance von RDS for PostgreSQL.

Der für Passwörter verwendete Passwortverschlüsselungstyp wird in einem Parameter, `password_encryption`, festgelegt. Die Verschlüsselung, die der DB-Instance von RDS for PostgreSQL zulässt, wird in einem anderen Parameter, `rds.accepted_password_auth_method`, festgelegt. Wenn Sie den Standardwert eines dieser Parameter ändern, müssen Sie eine benutzerdefinierte DB-Parametergruppe erstellen und auf Ihre Instance anwenden.

Sie können auch die AWS Management Console oder die RDS-API verwenden, um eine benutzerdefinierte zu erstellen. Weitere Informationen finden Sie unter

Sie können jetzt die benutzerdefinierte Parametergruppe Ihrer DB-Instance zuordnen.

So erstellen Sie eine benutzerdefinierte DB-Parametergruppe

1. Verwenden Sie den CLI-Befehl `create-db-parameter-group` zum Erstellen der benutzerdefinierten DB-Parametergruppe. In diesem Beispiel wird `postgres13` als Quelle für diese benutzerdefinierte Parametergruppe verwendet.

   Für LinuxmacOS, oderUnix:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name 'docs-lab-scram-passwords' \
     --db-parameter-group-family postgres13  --description 'Custom parameter group for SCRAM'
   ```

   Windows:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name "docs-lab-scram-passwords" ^
     --db-parameter-group-family postgres13  --description "Custom DB parameter group for SCRAM"
   ```

2. Verwenden Sie den CLI-Befehl `modify-db-instance` zum Anwenden dieser benutzerdefinierten Parametergruppe auf Ihren DB-Cluster von RDS for PostgreSQL.

   Für LinuxmacOS, oderUnix:

   ```
   aws rds modify-db-instance --db-instance-identifier 'your-instance-name' \
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

   Windows:

   ```
   aws rds modify-db-instance --db-instance-identifier "your-instance-name" ^
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

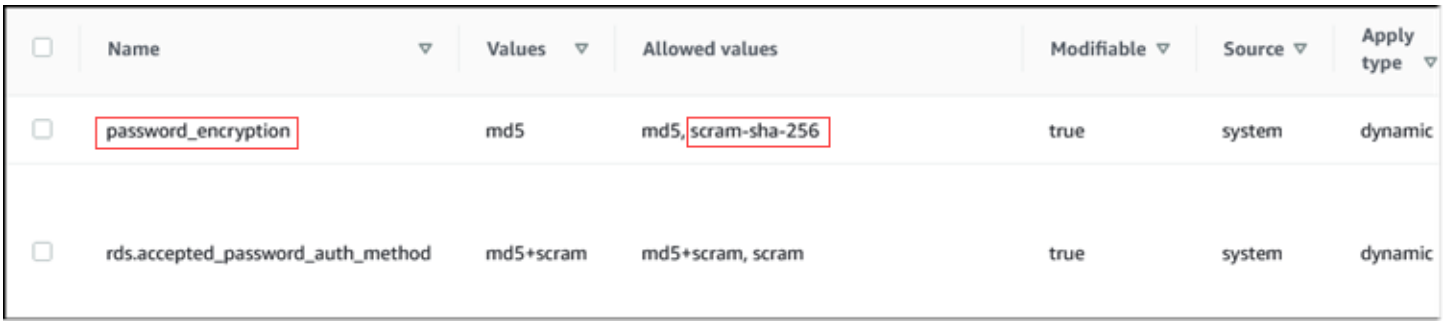
   Zum erneuten Synchronisieren Ihrer DB-Instance von RDS for PostgreSQL mit Ihrer benutzerdefinierten DB-Parametergruppe müssen Sie die primäre und alle anderen Instances des Clusters neu starten. Planen Sie dies während Ihres regulären Wartungsfensters, um die Auswirkungen auf Ihre Benutzer zu minimieren.

Konfigurieren der Passwortverschlüsselung für die Verwendung von SCRAM

Der Passwortverschlüsselungsmechanismus, der von einer DB-Instance von RDS for PostgreSQL verwendet wird, ist in der DB-Parametergruppe auf den Parameter `password_encryption` festgelegt. Zulässige Werte sind keine Angabe, `md5` oder `scram-sha-256`. Der Standardwert hängt von der Version von RDS for PostgreSQL wie folgt ab:

- RDS für PostgreSQL 14 und höher – Der Standardwert ist `scram-sha-256`.
- RDS for PostgreSQL 13 – Der Standardwert ist `md5`.

Mit einer benutzerdefinierten DB-Parametergruppe, die Ihrer DB-Instance von RDS for PostgreSQL angefügt ist, können Sie die Werte für den Passwortverschlüsselungsparameter ändern.



Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	md5	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

So ändern Sie die Passwortverschlüsselungseinstellung in scram-sha-256

- Ändern Sie den Wert der Passwortverschlüsselung in scram-sha-256, wie nachfolgend gezeigt. Die Änderung kann sofort angewendet werden, da der Parameter dynamisch ist. Daher ist kein Neustart erforderlich, damit die Änderung wirksam wird.

  Für LinuxmacOS, oderUnix:

  ```
  aws rds modify-db-parameter-group --db-parameter-group-name \
    'docs-lab-scram-passwords' --parameters
   'ParameterName=password_encryption,ParameterValue=scram-
  sha-256,ApplyMethod=immediate'
  ```

  Windows:

  ```
  aws rds modify-db-parameter-group --db-parameter-group-name ^
  ```

```
  "docs-lab-scram-passwords" --parameters
 "ParameterName=password_encryption,ParameterValue=scram-
sha-256,ApplyMethod=immediate"
```

Migrieren von Passwörtern für Benutzerrollen zu SCRAM

Sie können Passwörter für Benutzerrollen wie folgt zu SCRAM migrieren.

Um Datenbankbenutzer- (Rollen-) Passwörter von MD5 zu SCRAM zu migrieren

1. Melden Sie sich als Administratorbenutzer an (Standardbenutzer `postgres`), wie nachfolgend gezeigt.

   ```
   psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Überprüfen Sie die Einstellung des `password_encryption`-Parameters auf der DB-Instance von RDS for PostgreSQL mithilfe des folgenden Befehls.

   ```
   postgres=> SHOW password_encryption;
    password_encryption
   ---------------------
    md5
    (1 row)
   ```

3. Ändern Sie den Wert dieses Parameters in scram-sha-256. Weitere Informationen finden Sie unter Konfigurieren der Passwortverschlüsselung für die Verwendung von SCRAM.

4. Überprüfen Sie den Wert erneut, um sicherzustellen, dass er jetzt auf `scram-sha-256` festgelegt ist. Gehen Sie dazu wie folgt vor.

   ```
   postgres=> SHOW password_encryption;
    password_encryption
   ---------------------
    scram-sha-256
    (1 row)
   ```

5. Bitten Sie alle Datenbankbenutzer, ihre Passwörter zu ändern. Stellen Sie sicher, dass Sie auch Ihr eigenes Passwort für das `postgres`-Konto ändern (der Datenbankbenutzer mit `rds_superuser`-Berechtigungen).

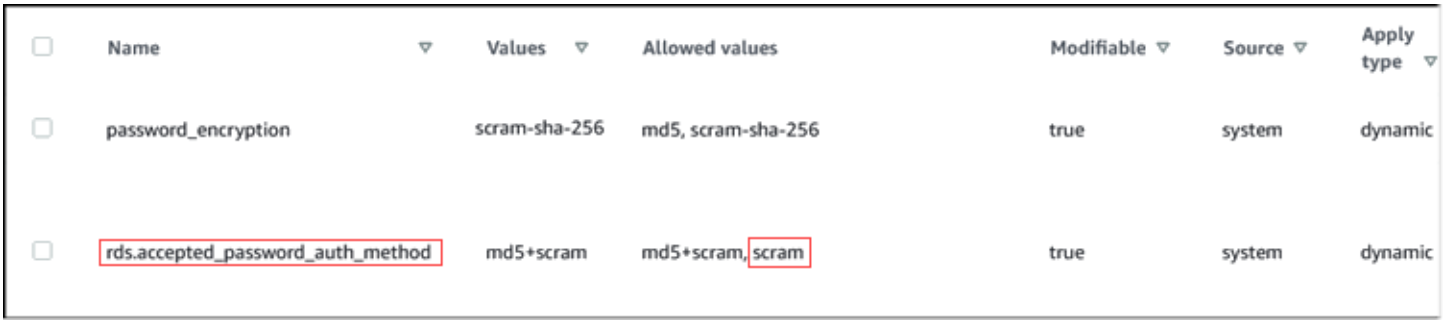
```
labdb=> ALTER ROLE postgres WITH LOGIN PASSWORD 'change_me';
ALTER ROLE
```

6. Wiederholen Sie den Vorgang für alle Datenbanken auf dem DB-Instance von RDS für PostgreSQL

Ändern des Parameters, um SCRAM zu erfordern

Dies ist der letzte Schritt in diesem Prozess. Nachdem Sie die Änderung im folgenden Verfahren vorgenommen haben, können sich Benutzerkonten (Rollen), die nach wie vor die `md5`-Verschlüsselung für Passwörter verwenden, nicht beim anmelden. DB-Instance von RDS für PostgreSQL

Die `rds.accepted_password_auth_method` gibt die Verschlüsselungsmethode an, die die DB-Instance von RDS for PostgreSQL als Benutzerpasswort beim Anmeldevorgang akzeptiert. Der Standardwert ist `md5+scram`, was bedeutet, dass eine der beiden Methoden akzeptiert wird. Im folgenden Bild finden Sie die Standardeinstellung für diesen Parameter.



Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	scram-sha-256	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

Die zulässigen Werte für diesen Parameter sind `md5+scram` oder nur `scram`. Durch Ändern dieses Parameterwerts in `scram` wird er erforderlich.

So ändern Sie den Parameterwert, sodass eine SCRAM-Authentifizierung für Passwörter erforderlich ist

1. Stellen Sie sicher, dass alle Datenbank-Benutzerpasswörter für alle Datenbanken auf Ihrer DB-Instance von RDS for PostgreSQL `scram-sha-256` für die Passwortverschlüsselung verwenden. Fragen Sie hierzu `rds_tools` für die Rolle (Benutzer) und den Verschlüsselungstyp wie folgt ab.

```
postgres=> SELECT * FROM rds_tools.role_password_encryption_type();
  rolname        | encryption_type
```

```
  ----------------------+-----------------
  pg_monitor           |
  pg_read_all_settings |
  pg_read_all_stats    |
  pg_stat_scan_tables  |
  pg_signal_backend    |
  lab_tester           | scram-sha-256
  user_465             | scram-sha-256
  postgres             | scram-sha-256
  ( rows)
```

2. Wiederholen Sie die Abfrage auf allen DB-Instances Ihres DB-Instance von RDS für PostgreSQL

   Wenn alle Passwörter scram-sha-256 verwenden, können Sie fortfahren.

3. Ändern Sie den Wert der akzeptierten Passwortauthentifizierung in scram-sha-256 wie folgt.

   Für LinuxmacOS, oderUnix:

```
aws rds modify-db-parameter-group --db-parameter-group-name 'docs-lab-scram-
passwords' \
  --parameters
 'ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

   Windows:

```
aws rds modify-db-parameter-group --db-parameter-group-name "docs-lab-scram-
passwords" ^
  --parameters
 "ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

Behandlung toter Verbindungen in PostgreSQL

Tote Verbindungen treten auf, wenn eine Datenbanksitzung auf dem Server aktiv bleibt, obwohl die Client-Anwendung abgebrochen oder ungewöhnlich beendet wurde. Diese Situation tritt typischerweise auf, wenn Client-Prozesse abstürzen oder unerwartet beendet werden, ohne dass ihre Datenbankverbindungen ordnungsgemäß geschlossen oder laufende Anfragen abgebrochen werden.

PostgreSQL identifiziert und bereinigt effizient tote Verbindungen, wenn Serverprozesse inaktiv sind oder versuchen, Daten an Clients zu senden. Die Erkennung ist jedoch schwierig für Sitzungen, die inaktiv sind, auf Client-Eingaben warten oder aktiv Abfragen ausführen. Um diese Szenarien

zu handhaben, stellt PostgreSQL Parameter `tcp_keepalives_*tcp_user_timeout`, und `client_connection_check_interval` bereit.

Themen

- TCP-Keepalive verstehen
-
- Anwendungsfälle für TCP-Keepalive-Einstellungen
- Bewährte Methoden

TCP-Keepalive verstehen

TCP Keepalive ist ein Mechanismus auf Protokollebene, der dabei hilft, die Verbindungsintegrität aufrechtzuerhalten und zu überprüfen. Jede TCP-Verbindung verwaltet Einstellungen auf Kernelebene, die das Keepalive-Verhalten steuern. Wenn der Keepalive-Timer abläuft, geht das System wie folgt vor:

- Sendet ein Testpaket ohne Daten und mit gesetztem ACK-Flag.
- Erwartet eine Antwort vom Remote-Endpunkt gemäß den TCP/IP Spezifikationen.
- Verwaltet den Verbindungsstatus auf der Grundlage der Antwort oder des Fehlens einer Antwort.

Parameter	Beschreibung	Standardwerte
`tcp_keepalives_idle`	Gibt die Anzahl der Sekunden der Inaktivität an, bevor eine Keepalive-Nachricht gesendet wird.	300
`tcp_keepalives_interval`	Gibt die Anzahl der Sekunden zwischen erneuten Übertragungen unbestätigter Keepalive-Nachrichten an.	30
`tcp_keepalives_count`	Höchstzahl verloren gegangener Keepalive -Nachrichten, bevor die	2

Parameter	Beschreibung	Standardwerte
	Verbindung für unterbrochen erklärt wird	
`tcp_user_timeout`	Gibt an, wie lange (in Millisekunden) unbestätigte Daten verbleiben können, bevor die Verbindung gewaltsam geschlossen wird.	0
`client_connection_check_interval`	Legt das Intervall (in Millisekunden) für die Überprüfung des Client-Verbindungsstatus bei Abfragen mit langer Laufzeit fest. Dadurch wird eine schnellere Erkennung geschlossener Verbindungen gewährleistet.	0

Anwendungsfälle für TCP-Keepalive-Einstellungen

Inaktive Sitzungen am Leben erhalten

Um zu verhindern, dass inaktive Verbindungen aufgrund von Inaktivität durch Firewalls oder Router beendet werden:

- Konfigurieren Sie `tcp_keepalives_idle` das Senden von Keepalive-Paketen in regelmäßigen Abständen.

Erkennung toter Verbindungen

Um tote Verbindungen umgehend zu erkennen:

- Passen Sie `tcp_keepalives_idletcp_keepalives_interval`, und an`tcp_keepalives_count`. Bei den Standardeinstellungen von Aurora PostgreSQL dauert es beispielsweise etwa eine Minute (2 Sonden × 30 Sekunden), um eine unterbrochene Verbindung zu erkennen. Eine Senkung dieser Werte kann die Erkennung beschleunigen.

- `tcp_user_timeout`Dient zur Angabe der maximalen Wartezeit für eine Bestätigung.

TCP-Keepalive-Einstellungen helfen dem Kernel, tote Verbindungen zu erkennen, aber PostgreSQL reagiert möglicherweise erst, wenn der Socket verwendet wird. Wenn in einer Sitzung eine lange Abfrage ausgeführt wird, werden tote Verbindungen möglicherweise erst nach Abschluss der Abfrage erkannt. `client_connection_check_interval`Kann in PostgreSQL 14 und höheren Versionen die Erkennung unterbrochener Verbindungen beschleunigen, indem der Socket während der Abfrageausführung regelmäßig abgefragt wird.

Bewährte Methoden

- Legen Sie angemessene Keepalive-Intervalle fest: Optimieren Sie`tcp_user_timeout`, und sorgen Sie für ein ausgewogenes Verhältnis zwischen `tcp_keepalives_idle` `tcp_keepalives_count` Erkennungsgeschwindigkeit und `tcp_keepalives_interval` Ressourcennutzung.
- Optimieren Sie für Ihre Umgebung: Passen Sie die Einstellungen an das Netzwerkverhalten, die Firewall-Richtlinien und die Sitzungsanforderungen an.
- Nutzen Sie die PostgreSQL-Funktionen: Verwenden Sie sie `client_connection_check_interval` in PostgreSQL 14 und höheren Versionen für effiziente Verbindungsprüfungen.

Wir empfehlen ausdrücklich die Selbstbereinigungsfunktion zu verwenden, um die Integrität Ihrer PostgreSQL-DB-Instance zu wahren. Die Selbstbereinigung automatisiert den Start der Befehle VACUUM und ANALYZE. Sie prüft auf Tabellen mit einer großen Zahl von eingefügten, aktualisierten oder gelöschten Tupeln. Nach dieser Prüfung wird Speicher durch Entfernen von überflüssigen Daten oder Tupeln aus der PostgreSQL-Datenbank freigegeben.

Standardmäßig ist Autovacuum für die RDS for PostgreSQL Aurora PostgreSQL-DB-Instances erstellen. Andere Konfigurationsparameter, die mit der Selbstbereinigungsfunktion verknüpft sind, sind ebenfalls standardmäßig festgelegt. Da diese Standardwerte generisch sind, können Sie davon profitieren, einige der mit der Selbstbereinigungsfunktion verbundenen Parameter für Ihre spezifische Workload zu optimieren.

Im Folgenden finden Sie weitere Informationen zum Autovacuum und zur Optimierung einiger seiner Parameter auf Ihrer PostgreSQL-DB-Instance von RDS for PostgreSQL. Allgemeine Informationen finden Sie unter. Bewährte Methoden für die Arbeit mit PostgreSQL

Themen

- Zuweisen von Arbeitsspeicher für die Selbstbereinigung
- Verringern der Wahrscheinlichkeit von Transaktions-ID-Wraparounds
- Ermittlung, ob die Tabellen in Ihrer Datenbank bereinigt werden müssen
- Ermittlung, für welche Tabellen derzeit eine Selbstbereinigung nötig ist
- Ermittlung, ob die Selbstbereinigung derzeit ausgeführt wird und wie lange sie dauert
- Ausführen einer manuellen Bereinigungseinfrierung
- Neuindizierung einer Tabelle während der Ausführung einer Selbstbereinigung
- Verwalten der automatischen Bereinigung mit großen Indizes
- Weitere Parameter, die sich auf die Selbstbereinigung auswirken
- Festlegen von Selbstbereinigungsparametern auf Tabellenebene
- Protokollieren von Selbstbereinigung- und Bereinigungsaktivitäten
- Das Verhalten von Autovacuum bei ungültigen Datenbanken verstehen

Zuweisen von Arbeitsspeicher für die Selbstbereinigung

Einer der wichtigsten Parameter, der sich auf die Leistung der Selbstbereinigung auswirkt, ist der Parameter `autovacuum_work_mem`. In RDS for PostgreSQL , Versionen 14 und früher, ist der `autovacuum_work_mem` Parameter auf -1 gesetzt, was bedeutet, dass stattdessen die Einstellung von `maintenance_work_mem` verwendet wird. Wird für alle anderen Versionen durch `autovacuum_work_mem` GREATEST ({DBInstanceClassMemory/32768}, 65536) bestimmt.

Bei manuellen Vakuumvorgängen wird immer die `maintenance_work_mem` Einstellung verwendet, wobei die Standardeinstellung GREATEST ({DBInstanceClassMemory/63963136 *1024}, 65536) ist. Sie kann auch auf Sitzungsebene mithilfe des Befehls für gezieltere manuelle Operationen angepasst werden. `SET VACUUM`

Das `autovacuum_work_mem` legt fest, dass der Speicher für das automatische Vakuumieren die Kennungen von toten Tupeln () zum Absaugen von Indizes enthalten soll. `pg_stat_all_tables.n_dead_tup`

Beachten Sie bei Berechnungen zur Bestimmung des `autovacuum_work_mem` Parameterwerts Folgendes:

- Wenn Sie den Parameter zu niedrig einstellen, muss der Vakuumprozess die Tabelle möglicherweise mehrmals scannen, um die Arbeit abzuschließen. Solche wiederholten Scans können negative Auswirkungen auf die Leistung haben. Bei größeren Instanzen kann die Einstellung von `maintenance_work_mem` oder `autovacuum_work_mem` auf mindestens 1 GB die Leistung beim Bereinigen von Tabellen mit einer hohen Anzahl von toten Tupeln verbessern. In PostgreSQL-Versionen 16 und früheren Versionen ist die Speichernutzung von Vacuum jedoch auf 1 GB begrenzt, was ausreicht, um ungefähr 179 Millionen tote Tupel in einem einzigen Durchgang zu verarbeiten. Wenn eine Tabelle mehr tote Tupel enthält, muss Vacuum mehrere Durchläufe durch die Indizes der Tabelle durchführen, was den Zeitaufwand erheblich erhöht. Ab PostgreSQL Version 17 gibt es kein Limit von 1 GB, und Autovacuum kann mithilfe von Radixbäumen mehr als 179 Millionen Tupel verarbeiten.

  Ein Tupelbezeichner hat eine Größe von 6 Byte. Um den Speicherbedarf für das Löschen eines Tabellenindexes zu schätzen, fragen Sie `pg_stat_all_tables.n_dead_tup` nach der Anzahl der toten Tupel ab und multiplizieren Sie diese Zahl dann mit 6, um den Speicherplatz zu ermitteln, der für das Löschen des Index in einem einzigen Durchgang erforderlich ist. Sie können die folgende Abfrage verwenden:

  ```
  SELECT
      relname AS table_name,
      n_dead_tup,
      pg_size_pretty(n_dead_tup * 6) AS estimated_memory
  FROM
      pg_stat_all_tables
  WHERE
      relname = 'name_of_the_table';
  ```

- Der Parameter `autovacuum_work_mem` funktioniert in Verbindung mit dem Parameter `autovacuum_max_workers`. Jeder Mitarbeiter unter Ihnen `autovacuum_max_workers` kann den von Ihnen zugewiesenen Speicher verwenden. Wenn Sie viele kleine Tabellen haben, müssen Sie mehr `autovacuum_max_workers` und weniger `autovacuum_work_mem` zuteilen. Wenn Sie über große Tabellen (größer als 100 GB) verfügen, sollten Sie mehr Speicher und weniger Arbeitsprozesse zuweisen. Sie müssen genügend Arbeitsspeicher zuweisen, um den Vorgang für Ihre größte Tabelle erfolgreich ausführen zu können. Stellen Sie daher sicher, dass die Kombination aus Arbeitsprozessen und Arbeitsspeicher dem Gesamtspeicher entspricht, den Sie zuweisen möchten.

Verringern der Wahrscheinlichkeit von Transaktions-ID-Wraparounds

In einigen Fällen sind Parametergruppen-Einstellungen, die sich auf die Selbstbereinigung beziehen, möglicherweise nicht aggressiv genug, um Transaktions-ID-Wraparounds zu verhindern. Um dieses Problem zu lösen, bietet RDS for PostgreSQL einen Mechanismus, der die Autovacuum-Parameterwerte automatisch anpasst. Adaptive Autovacuum In der PostgreSQL-Dokumentation finden Sie eine sehr detaillierte Beschreibung von Transaktions-ID-Wraparounds.

Adaptives Autovacuum ist standardmäßig für RDS für PostgreSQL aktiviert, wobei der dynamische Parameter auf ON gesetzt ist. `rds.adaptive_autovacuum` Wir raten dringend dazu, diese Option aktiviert zu lassen. Um die adaptive Optimierung der Selbstbereinigungsparameter zu deaktivieren, stellen Sie den Parameter `rds.adaptive_autovacuum` jedoch auf 0 oder OFF ein.

Der Transaktions-ID-Wraparound ist auch dann noch möglich, wenn Amazon RDS Amazon RDS die Autovacuum-Parameter optimiert. Wir empfehlen Ihnen, einen CloudWatch Amazon-Alarm für den Transaktions-ID-Wraparound zu implementieren. Weitere Informationen finden Sie im Datenbank-Blog im Beitrag Implementieren eines Frühwarnsystems für Transaktions-ID-Wraparound in RDS for PostgreSQL. AWS

Wenn die adaptive Abstimmung der Autovakuum-Parameter aktiviert ist, beginnt Amazon RDS mit der Anpassung der Autovakuum-Parameter, wenn die CloudWatch Metrik den Wert des `autovacuum_freeze_max_age` Parameters oder 500.000.000 `MaximumUsedTransactionIDs` erreicht, je nachdem, welcher Wert größer ist.

Amazon RDS fährt mit dem Anpassen der Parameter für die Selbstbereinigung fort, wenn eine Tabelle weiterhin zu Transaktions-ID-Wraparounds tendiert. Jede dieser Anpassungen stellt weitere Ressourcen für die Selbstbereinigung bereit, um Wraparounds zu vermeiden. Amazon RDS aktualisiert die folgenden Parameter, die sich auf die Selbstbereinigung beziehen:

- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit
- `autovacuum_work_mem`
- autovacuum_naptime

RDS ändert diese Parameter nur, wenn der neue Wert die Selbstbereinigung aggressiver macht. Die Parameter werden im Arbeitsspeicher auf der DB-Instance geändert. Die Werte in der Parametergruppe werden nicht geändert. Um die aktuellen Arbeitsspeichereinstellungen anzuzeigen, verwenden Sie den PostgreSQL-SQL-Befehl SHOW.

Wenn Amazon RDS einen dieser Selbstbereinigungsparameter ändert, wird ein Ereignis für die betroffene DB-Instance erzeugt. Dieses Ereignis ist auf der AWS Management Console und über die Amazon RDS-API sichtbar. Wenn die `MaximumUsedTransactionIDs` CloudWatch Metrik wieder unter den Schwellenwert fällt, setzt Amazon RDS die Autovakuum-bezogenen Parameter im Speicher auf die in der Parametergruppe angegebenen Werte zurück. Es generiert dann ein anderes Ereignis, das dieser Änderung entspricht.

Ermittlung, ob die Tabellen in Ihrer Datenbank bereinigt werden müssen

Sie können die folgende Abfrage verwenden, um die Anzahl der nicht eingefrorenen Transaktionen in einer Datenbank anzuzeigen. Die `datfrozenxid` Spalte einer `pg_database` Datenbankzeile ist eine Untergrenze für die normale Transaktion, die in dieser Datenbank IDs erscheint. Diese Spalte ist der Mindestwert der `relfrozenxid`-Werte pro Tabelle in der Datenbank.

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc
 limit 20;
```

Beispielsweise könnten die Ergebnisse der Ausführung der oben gezeigten Abfrage wie folgt aussehen.

```
datname    | age
mydb       | 1771757888
template0  | 1721757888
template1  | 1721757888
rdsadmin   | 1694008527
postgres   | 1693881061
(5 rows)
```

Wenn das Alter einer Datenbank 2 Milliarden Transaktionen erreicht IDs, erfolgt ein Transaktions-ID (XID) -Wraparound, und die Datenbank wird schreibgeschützt. Sie können diese Abfrage verwenden, um eine Metrik zu erstellen und einige Male am Tag auszuführen. Standardmäßig ist die Selbstbereinigung so festgelegt, dass das Alter der Transaktionen 200 000 000 nicht überschreitet 200,000,000 (`autovacuum_freeze_max_age`).

Eine Überwachungsstrategie kann beispielsweise wie folgt aussehen:

- Stellen Sie den `autovacuum_freeze_max_age`-Wert auf 200 Millionen Transaktionen ein.
- Wenn eine Tabelle 500 Millionen nicht eingefrorene Transaktionen erreicht, löst dies einen Alarm mit niedrigem Schweregrad aus. Dies ist kein unangemessener Wert. Er könnte jedoch zu erkennen geben, dass die Selbstbereinigung nicht Schritt hält.

- Wenn eine Tabelle 1 Milliarde nicht bereinigter Transaktionen aufweist, sollte dies als ein Alarm behandelt werden, für den Maßnahmen zu ergreifen sind. Im Allgemeinen sollte aus Leistungsgründen das Alter möglichst nahe zu `autovacuum_freeze_max_age` liegen. Wir empfehlen, dass Sie eine Untersuchung unter Beachtung der folgenden Empfehlungen durchführen.
- Wenn eine Tabelle 1,5 Milliarden nicht bereinigter Transaktionen erreicht, wird ein Alarm mit hohem Schweregrad ausgelöst. Je nachdem, wie schnell Ihre Datenbank die Transaktion verwendet IDs, kann dieser Alarm darauf hinweisen, dass das System keine Zeit mehr hat, um Autovacuum auszuführen. In diesem Fall empfehlen wir, eine unmittelbare Lösung in Betracht zu ziehen.

Wenn eine Tabelle diese Schwellenwerte konstant überschreitet, ändern Sie die Selbstbereinigungsparameter weiter. Standardmäßig ist die manuelle Verwendung von VACUUM (für den kostenbasierte Verzögerungen deaktiviert sind) aggressiver als die Standardselbstbereinigung. Der Befehl hat jedoch insgesamt auch größere Auswirkungen auf das System.

Wir empfehlen Folgendes:

- Aktivieren Sie einen Überwachungsmechanismus, damit Sie über das Alter der ältesten Transaktionen informiert sind.

  Informationen zum Erstellen eines Prozesses, der Sie vor Transaktions-ID-Wraparound warnt, finden Sie im AWS Datenbank-Blogbeitrag Implementieren Sie ein Frühwarnsystem für Transaktions-ID-Wraparound in Amazon RDS for PostgreSQL.
- Führen Sie für häufiger verwendete Tabellen zusätzlich zur Selbstbereinigung während Wartungsfenstern regelmäßig eine manuelle Bereinigungseinfrierung aus. Informationen zur Ausführung manueller Bereinigungseinfrierungen finden Sie unter Ausführen einer manuellen Bereinigungseinfrierung.

Ermittlung, für welche Tabellen derzeit eine Selbstbereinigung nötig ist

Häufig benötigen eine oder zwei Tabellen eine Bereinigung. Tabellen, deren `relfrozenxid`-Wert größer als die Anzahl von Transaktionen in `autovacuum_freeze_max_age` ist, sind stets Ziel der Selbstbereinigung. Andernfalls, wenn die Anzahl der Tupel, die seit dem letzten überholt wurden, den Vakuumschwellenwert VACUUM überschreitet, wird die Tabelle abgesaugt.

Der Selbstbereinigungsschwellenwert ist definiert als:

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

wo das `vacuum base threshold` ist`autovacuum_vacuum_threshold`, das ist und das `vacuum scale factor` ist`autovacuum_vacuum_scale_factor`. `number of tuples` `pg_class.reltuples`

Führen Sie während der Herstellung der Verbindung mit Ihrer Datenbank die folgende Abfrage aus, um eine Liste der Tabellen anzuzeigen, für die Selbstbereinigungsfunktion eine Bereinigung als notwendig betrachtet.

```
WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold FROM
pg_settings WHERE name = 'autovacuum_vacuum_threshold'),
vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor FROM
pg_settings WHERE name = 'autovacuum_vacuum_scale_factor'),
fma AS (SELECT setting AS autovacuum_freeze_max_age FROM pg_settings WHERE name =
 'autovacuum_freeze_max_age'),
sto AS (select opt_oid, split_part(setting, '=', 1) as param,
split_part(setting, '=', 2) as value from (select oid opt_oid, unnest(reloptions)
 setting from pg_class) opt)
SELECT '"'||ns.nspname||'"."'||c.relname||'"' as relation,
pg_size_pretty(pg_table_size(c.oid)) as table_size,
age(relfrozenxid) as xid_age,
coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age,
(coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples)
AS autovacuum_vacuum_tuples, n_dead_tup as dead_tuples FROM
pg_class c join pg_namespace ns on ns.oid = c.relnamespace
join pg_stat_all_tables stat on stat.relid = c.oid join vbt on (1=1) join vsf on (1=1)
 join fma on (1=1)
left join sto cvbt on cvbt.param = 'autovacuum_vacuum_threshold' and c.oid =
 cvbt.opt_oid
left join sto cvsf on cvsf.param = 'autovacuum_vacuum_scale_factor' and c.oid =
 cvsf.opt_oid
left join sto cfma on cfma.param = 'autovacuum_freeze_max_age' and c.oid = cfma.opt_oid
WHERE c.relkind = 'r' and nspname <> 'pg_catalog'
AND (age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
OR coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup)
ORDER BY age(relfrozenxid) DESC LIMIT 50;
```

Ermittlung, ob die Selbstbereinigung derzeit ausgeführt wird und wie lange sie dauert

Wenn Sie eine Tabelle manuell bereinigen müssen, müssen Sie ermitteln, ob zurzeit eine Selbstbereinigung ausgeführt wird. Wenn dies der Fall ist, müssen Sie möglicherweise die Parameter anpassen, damit es effizienter läuft, oder das automatische Absaugen vorübergehend ausschalten, damit Sie es manuell ausführen VACUUM können.

Verwenden Sie die folgende Abfrage, um zu ermitteln, ob die Selbstbereinigung ausgeführt wird, wie lange diese bereits dauert und ob diese auf eine andere Sitzung wartet.

```
SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query
FROM pg_stat_activity
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;
```

Nach dem Ausführen der Abfrage wird Ihnen eine Ausgabe ähnlich der folgenden angezeigt.

```
 datname | usename  |  pid  | state  | wait_event |      xact_runtime       | query
 --------+----------+-------+--------+------------+-------------------------
+-------------------------------------------------------------------------------------------------
 mydb    | rdsadmin | 16473 | active |            | 33 days 16:32:11.600656 |
 autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
 mydb    | rdsadmin | 22553 | active |            | 14 days 09:15:34.073141 |
 autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
 mydb    | rdsadmin | 41909 | active |            | 3 days 02:43:54.203349  |
 autovacuum: VACUUM ANALYZE public.mytable3
 mydb    | rdsadmin |   618 | active |            | 00:00:00                |
 SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query+
         |          |       |        |            |                         | FROM
 pg_stat_activity
           +
         |          |       |        |            |                         | WHERE
 query like '%VACUUM%'
          +
         |          |       |        |            |                         | ORDER BY
 xact_start;
       +
```

Verschiedene Probleme können dazu führen, dass eine Selbstbereinigungssitzung über eine lange Zeit (mehrere Tage) ausgeführt wird. Das häufigste Problem besteht jedoch darin, dass der Wert Ihres Parameters `maintenance_work_mem` im Verhältnis zur Größe der Tabelle oder zur Häufigkeit der Aktualisierungen zu niedrig festgelegt wurde.

Es wird empfohlen, dass Sie die folgende Formel verwenden, um den Wert des Parameters `maintenance_work_mem` festzulegen.

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

Selbstbereinigungssitzungen, die nur kurze Zeit ausgeführt werden, können ebenfalls auf Probleme hinweisen:

- Es kann bedeuten, dass es für Ihren Workload nicht genügend `autovacuum_max_workers` gibt. In diesem Fall müssen Sie die Anzahl der Worker angeben.
- Es kann bedeuten, dass es eine Indexbeschädigung gibt (die Selbstbereinigung stürzt ab und startet bei derselben Relation neu, es gibt jedoch keine Fortschritte). Führen Sie in diesem Fall ein manuelles `vacuum freeze verbose` *`table`* aus, um die genaue Ursache zu sehen.

Ausführen einer manuellen Bereinigungseinfrierung

Möglicherweise möchten Sie eine manuelle Bereinigung für eine Tabelle ausführen, für die bereits ein Bereinigungsvorgang ausgeführt wird. Dies ist nützlich, wenn Sie eine Tabelle mit einem Alter identifiziert haben, das 2 Milliarden Transaktionen (oder einen Wert oberhalb eines von Ihnen beobachteten Schwellenwerts) erreicht.

Die folgenden Schritte sind Richtlinien mit mehreren Variationen des Prozesses. Angenommen, Sie stellen während der Überprüfung fest, dass der Wert für den Parameter `maintenance_work_mem` zu niedrig festgelegt wurde. Sie müssen sofort Maßnahmen für eine Tabelle einleiten. Sie möchten zum aktuellen Zeitpunkt jedoch keinen Bounce für die Instance auslösen. Ermitteln Sie mittels der Abfragen in vorherigen Abschnitten die Tabelle, die das Problem darstellt, und identifizieren Sie eine über einen langen Zeitraum ausgeführte Selbstbereinigungssitzung. Sie müssen die Einstellung für den Parameter `maintenance_work_mem` ändern. Sie müssen jedoch auch sofortige Maßnahmen einleiten und die betreffende Tabelle bereinigen. Das folgende Verfahren zeigt, was Sie in dieser Situation unternehmen.

So führen Sie manuell eine Bereinigungseinfrierung aus

1. Öffnen Sie zwei Sitzungen für die Datenbank, die Tabelle enthält, die Sie bereinigen möchten. Verwenden Sie für die zweite Sitzung „screen“ oder ein anderes Hilfsprogramm, das die Sitzung beibehält, wenn die Verbindung verloren geht.
2. Rufen Sie in der ersten Sitzung die Prozess-ID (PID) der Selbstbereinigungssitzung ab, die für die Tabelle ausgeführt wird.

   Führen Sie die folgende Abfrage aus, um die PID der Selbstbereinigungssitzung abzurufen.

   ```
   SELECT datname, usename, pid, current_timestamp - xact_start
   AS xact_runtime, query
   FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
   xact_start;
   ```

3. Berechnen Sie in der zweiten Sitzung die Menge des Arbeitsspeichers, den Sie für diese Operation benötigen. In diesem Beispiel ermitteln Sie, dass für diese Operation bis zu 2 GB Arbeitsspeicher verwendet werden können. Daher wird für die aktuelle Sitzung `maintenance_work_mem` auf 2 GB festgelegt.

   ```
   SET maintenance_work_mem='2 GB';
   SET
   ```

4. Geben Sie in der zweiten Sitzung einen `vacuum freeze verbose`-Befehl für die Tabelle aus. Die Verbose-Einstellung ist nützlich, da Sie Aktivität sehen können, auch wenn in PostgreSQL hierfür zurzeit kein Fortschrittsbericht verfügbar ist.

   ```
   \timing on
   Timing is on.
   vacuum freeze verbose pgbench_branches;
   ```

   ```
   INFO:  vacuuming "public.pgbench_branches"
   INFO:  index "pgbench_branches_pkey" now contains 50 row versions in 2 pages
   DETAIL:  0 index row versions were removed.
   0 index pages have been deleted, 0 are currently reusable.
   CPU 0.00s/0.00u sec elapsed 0.00 sec.
   INFO:  index "pgbench_branches_test_index" now contains 50 row versions in 2 pages
   DETAIL:  0 index row versions were removed.
   0 index pages have been deleted, 0 are currently reusable.
   CPU 0.00s/0.00u sec elapsed 0.00 sec.
   ```

```
INFO:  "pgbench_branches": found 0 removable, 50 nonremovable row versions
     in 43 out of 43 pages
DETAIL:  0 dead row versions cannot be removed yet.
There were 9347 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
VACUUM
Time: 2.765 ms
```

5. Wenn in der ersten Sitzung das Autovakuum die Vakuumsitzung blockiert hat, wird `pg_stat_activity` angezeigt, dass auf Ihre Vakuumsitzung T gewartet wird. Beenden Sie in diesem Fall den Autovakuumvorgang wie folgt.

   ```
   SELECT pg_terminate_backend('the_pid');
   ```

   > **Note**
   >
   > Einige niedrigere Versionen von Amazon RDS können einen Autovakuum-Prozess mit dem vorherigen Befehl nicht beenden und schlagen mit dem folgenden Fehler fehl:`ERROR: 42501: must be a superuser to terminate superuser process LOCATION: pg_terminate_backend, signalfuncs.c:227`.

   An diesem Punkt beginnt Ihre Sitzung. Autovacuum wird sofort neu gestartet, da diese Tabelle wahrscheinlich die höchste auf ihrer Arbeitsliste ist.

6. Initiieren Sie Ihren `vacuum freeze verbose`-Befehl in der zweiten Sitzung und beenden Sie dann den Selbstbereinigungsvorgang in der ersten Sitzung.

Neuindizierung einer Tabelle während der Ausführung einer Selbstbereinigung

Wenn ein Index beschädigt wurde, verarbeitet die Selbstbereinigung die Tabelle weiter und schlägt fehl. Wenn Sie in dieser Situation eine manuelle Bereinigung versuchen, werden Sie eine Fehlermeldung ähnlich der folgenden erhalten.

```
postgres=>  vacuum freeze pgbench_branches;
ERROR: index "pgbench_branches_test_index" contains unexpected
   zero page at block 30521
HINT: Please REINDEX it.
```

Wenn der Index beschädigt ist und eine Selbstbereinigung für die Tabelle versucht wird, konkurrieren Sie mit einer Selbstbereinigungssitzung, die bereits ausgeführt wird. Wenn Sie einen REINDEX-Befehl ausgeben, wird eine exklusive Sperre für die Tabelle ausgeführt. Schreiboperationen werden blockiert. Gleiches gilt für Lesevorgänge, die diesen spezifischen Index verwenden.

So führen Sie eine Neuindizierung für eine Tabelle aus, wenn eine Selbstbereinigung für die Tabelle ausgeführt wird

1. Öffnen Sie zwei Sitzungen für die Datenbank, die Tabelle enthält, die Sie bereinigen möchten. Verwenden Sie für die zweite Sitzung „screen“ oder ein anderes Hilfsprogramm, das die Sitzung beibehält, wenn die Verbindung verloren geht.
2. Rufen Sie in der ersten Sitzung die PID der Selbstbereinigungssitzung ab, die für die Tabelle ausgeführt wird.

   Führen Sie die folgende Abfrage aus, um die PID der Selbstbereinigungssitzung abzurufen.

   ```
   SELECT datname, usename, pid, current_timestamp - xact_start
   AS xact_runtime, query
   FROM pg_stat_activity WHERE upper(query) like '%VACUUM%' ORDER BY
   xact_start;
   ```

3. Führen Sie in der zweiten Sitzung den Neuindizierungsbefehl aus.

   ```
   \timing on
   Timing is on.
   reindex index pgbench_branches_test_index;
   REINDEX
     Time: 9.966 ms
   ```

4. Wenn die Selbstbereinigung den Prozess blockiert hat, wird Ihnen in der ersten Sitzung in `pg_stat_activity` angezeigt, dass das Warten für Ihre Bereinigungssitzung "T" ist. In diesem Fall beenden Sie den Selbstbereinigungsprozess.

   ```
   SELECT pg_terminate_backend('the_pid');
   ```

   An diesem Punkt beginnt Ihre Sitzung. Es ist wichtig, zu beachten, dass die Selbstbereinigung sofort neu gestartet wird, da diese Tabelle in der Liste der Aufgaben wahrscheinlich ganz oben steht.

5. Initiieren Sie Ihren Befehl in der zweiten Sitzung und beenden Sie dann den Selbstbereinigungsvorgang in der ersten Sitzung.

Verwalten der automatischen Bereinigung mit großen Indizes

Im Rahmen ihrer Funktion führt die automatische Bereinigung mehrere Bereinigungsphasen aus, während sie für eine Tabelle ausgeführt wird. Bevor die Tabelle bereinigt wird, werden zunächst alle Indizes bereinigt. Wenn mehrere große Indizes entfernt werden, benötigt diese Phase einen großen Zeit- und Ressourcenaufwand. Es hat sich daher bewährt, die Anzahl der Indizes in einer Tabelle zu kontrollieren und ungenutzte Indizes zu entfernen.

Überprüfen Sie für diesen Vorgang zunächst die Gesamtindexgröße. Stellen Sie dann fest, ob es möglicherweise unbenutzte Indizes gibt, die entfernt werden können, wie in den folgenden Beispielen dargestellt.

So überprüfen Sie die Größe der Tabelle und ihrer Indizes

```
postgres=> select pg_size_pretty(pg_relation_size('pgbench_accounts'));
pg_size_pretty
6404 MB
(1 row)
```

```
postgres=> select pg_size_pretty(pg_indexes_size('pgbench_accounts'));
pg_size_pretty
11 GB
(1 row)
```

In diesem Beispiel ist die Größe der Indizes größer als die Tabelle. Dieser Unterschied kann zu Leistungsproblemen führen, da die Indizes überlastet oder ungenutzt sind, was sich sowohl auf die automatische Bereinigung als auch auf Insert-Operationen auswirkt.

So prüfen Sie, ob ungenutzte Indizes vorhanden sind

Mithilfe der Ansicht `pg_stat_user_indexes` können Sie überprüfen, wie oft ein Index für die Spalte `idx_scan` verwendet wird. Im folgenden Beispiel haben die ungenutzten Indizes den `idx_scan`-Wert `0`.

```
postgres=> select * from pg_stat_user_indexes where relname = 'pgbench_accounts' order
 by idx_scan desc;
```

```
relid  | indexrelid | schemaname | relname          | indexrelname          | idx_scan
 | idx_tup_read | idx_tup_fetch
-------+------------+------------+------------------+-----------------------+----------
+--------------+--------------
16433  | 16454      | public     | pgbench_accounts | index_f               | 6
 | 6            | 0
16433  | 16450      | public     | pgbench_accounts | index_b               | 3
 | 199999       | 0
16433  | 16447      | public     | pgbench_accounts | pgbench_accounts_pkey | 0
 | 0            | 0
16433  | 16452      | public     | pgbench_accounts | index_d               | 0
 | 0            | 0
16433  | 16453      | public     | pgbench_accounts | index_e               | 0
 | 0            | 0
16433  | 16451      | public     | pgbench_accounts | index_c               | 0
 | 0            | 0
16433  | 16449      | public     | pgbench_accounts | index_a               | 0
 | 0            | 0
(7 rows)
```

```
postgres=> select schemaname, relname, indexrelname, idx_scan from pg_stat_user_indexes
 where relname = 'pgbench_accounts' order by idx_scan desc;

schemaname  | relname          | indexrelname          | idx_scan
------------+------------------+-----------------------+----------
public      | pgbench_accounts | index_f               | 6
public      | pgbench_accounts | index_b               | 3
public      | pgbench_accounts | pgbench_accounts_pkey | 0
public      | pgbench_accounts | index_d               | 0
public      | pgbench_accounts | index_e               | 0
public      | pgbench_accounts | index_c               | 0
public      | pgbench_accounts | index_a               | 0
(7 rows)
```

Note

Diese Statistiken sind ab dem Zeitpunkt, an dem die Statistiken zurückgesetzt werden, inkrementell. Angenommen, Sie haben einen Index, der nur am Ende eines Geschäftsquartals oder nur für einen bestimmten Bericht verwendet wird. Es ist möglich, dass dieser Index seit dem Zurücksetzen der Statistiken nicht mehr verwendet wurde.

Weitere Informationen finden Sie unter Statistikfunktionen. Indizes, die verwendet werden, um Eindeutigkeit zu erzwingen, werden nicht gescannt und sollten nicht als ungenutzte Indizes identifiziert werden. Um die ungenutzten Indizes zu identifizieren, sollten Sie über fundierte Kenntnisse der Anwendung und ihrer Abfragen verfügen.

Um zu überprüfen, wann die Statistiken für eine Datenbank zuletzt zurückgesetzt wurden, verwenden Sie `pg_stat_database`.

```
postgres=> select datname, stats_reset from pg_stat_database where datname =
 'postgres';

datname   | stats_reset
----------+-------------------------------
postgres  | 2022-11-17 08:58:11.427224+00
(1 row)
```

Möglichst schnelles Bereinigen einer Tabelle

RDS für PostgreSQL 12 und höher

Wenn Sie zu viele Indizes in einer großen Tabelle haben, nähert sich Ihre DB-Instance möglicherweise dem Transaktions-ID-Wraparound (XID), also dem Zeitpunkt, an dem der XID-Zähler auf Null zurückgeht. Wenn diese Option nicht aktiviert ist, kann diese Situation zu Datenverlust führen. Sie können die Tabelle jedoch schnell bereinigen, ohne die Indizes zu bereinigen. In RDS für PostgreSQL 12 und höher können Sie VACUUM mit der Klausel `INDEX_CLEANUP` verwenden.

```
postgres=> VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) pgbench_accounts;

INFO: vacuuming "public.pgbench_accounts"
INFO: table "pgbench_accounts": found 0 removable, 8 nonremovable row versions in 1 out
 of 819673 pages
DETAIL: 0 dead row versions cannot be removed yet, oldest xmin: 7517
Skipped 0 pages due to buffer pins, 0 frozen pages.
CPU: user: 0.01 s, system: 0.00 s, elapsed: 0.01 s.
```

Wenn eine automatische Bereinigungssitzung bereits läuft, müssen Sie sie beenden, um mit dem manuellen VACUUM-Vorgang zu beginnen. Informationen zur Ausführung manueller Bereinigungseinfrierungen finden Sie unter Ausführen einer manuellen Bereinigungseinfrierung.

> **Note**
>
> Wenn die Indexbereinigung regelmäßig übersprungen wird, wird der Index aufgebläht, wodurch die Scanleistung beeinträchtigt wird. Der Index enthält tote Zeilen und die Tabelle enthält tote Zeilenzeiger. Dies hat zur Folge, dass `pg_stat_all_tables.n_dead_tup` die automatische Vakuumierung oder ein manuelles ABSAUGEN mit Indexbereinigung ausgeführt wird. Es hat sich bewährt, dieses Verfahren nur zu verwenden, um einen Umbruch der Transaktions-ID zu verhindern.

RDS für PostgreSQL 11 und niedriger

In RDS für PostgreSQL 11 und niedrigeren Versionen besteht die einzige Möglichkeit, den Bereinigungsvorgang schneller abzuschließen, darin, die Anzahl der Indizes in einer Tabelle zu reduzieren. Das Löschen eines Index kann sich auf Abfragepläne auswirken. Wir empfehlen, zuerst unbenutzte Indizes zu löschen und dann die Indizes löschen, wenn der XID-Wraparound kurz bevorsteht. Nach Abschluss des Bereinigungsvorgangs können Sie diese Indizes neu erstellen.

Weitere Parameter, die sich auf die Selbstbereinigung auswirken

Die folgende Abfrage zeigt die Werte einiger Parameter an, die sich direkt auf die Selbstbereinigung und ihr Verhalten auswirken. Die Selbstbereinigungsparameter werden in der PostgreSQL-Dokumentation vollständig beschrieben.

```
SELECT name, setting, unit, short_desc
FROM pg_settings
WHERE name IN (
'autovacuum_max_workers',
'autovacuum_analyze_scale_factor',
'autovacuum_naptime',
'autovacuum_analyze_threshold',
'autovacuum_analyze_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_cost_delay',
'autovacuum_vacuum_cost_limit',
'vacuum_cost_limit',
'autovacuum_freeze_max_age',
'maintenance_work_mem',
```

```
'vacuum_freeze_min_age');
```

All diese Parameter wirken sich auf die Selbstbereinigung aus. Die wichtigsten unter ihnen sind jedoch:

- maintenance_work_mem
- autovacuum_freeze_max_age
- autovacuum_max_workers
- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit

Festlegen von Selbstbereinigungsparametern auf Tabellenebene

Sie können Speicherparameter, die sich auf die der Selbstbereinigung beziehen, auf Tabellenebene festlegen. Dies kann im Vergleich zur Änderung des Verhaltens der gesamten Datenbank ein bevorzugtes Verfahren sein. Im Fall großer Tabellen müssen möglicherweise aggressive Einstellungen festgelegt werden, und Sie möchten vielleicht nicht, dass sich die Selbstbereinigung für alle Tabellen auf diese Weise verhält.

Die folgende Abfrage zeigt, für welche Tabellen zurzeit Optionen auf Tabellenebene festgelegt wurden.

```
SELECT relname, reloptions
FROM pg_class
WHERE reloptions IS NOT null;
```

Ein Beispiel, in dem dies nützlich sein kann, sind Tabellen, die sehr viel größer als Ihre restlichen Tabellen sind. Angenommen, es gibt eine Tabelle mit 300 GB und 30 weitere Tabellen mit weniger als 1 GB. In diesem Fall würde es sich anbieten, einige spezifische Parameter nur für die große Tabelle festzulegen, um nicht das Verhalten des gesamten Systems zu ändern.

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

Hierdurch wird die kostenbasierte Selbstbereinigungsverzögerung für diese Tabelle auf Kosten einer größeren Ressourcennutzung in Ihrem System deaktiviert. Normalerweise pausiert die Selbstbereinigung für `autovacuum_vacuum_cost_delay` jedes Mal, wenn

`autovacuum_cost_limit` erreicht wird. Weitere Informationen finden Sie in der SQL Postgre-Dokumentation zum kostenbasierten Staubsaugen.

Protokollieren von Selbstbereinigung- und Bereinigungsaktivitäten

Informationen über Bereinigungsaktivitäten werden basierend auf der im`rds.force_autovacuum_logging_level`-Parameter angegebenen Ebene an das `postgresql.log` gesendet. Im Folgenden sind die für diesen Parameter zulässigen Werte und die SQL Postgre-Versionen aufgeführt, für die dieser Wert die Standardeinstellung ist:

- `disabled`(Postgre SQL 10, SQL Postgre 9.6)
- `debug5`, `debug4`, `debug3`, `debug2`, `debug1`
- `info`(Postgramm SQL 12, Postgramm 11) SQL
- `notice`
- `warning`(Postgre SQL 13 und höher)
- `error`, Protokoll, `fatal`, `panic`

Das `rds.force_autovacuum_logging_level` arbeitet mit dem `log_autovacuum_min_duration`-Parameter. Der Wert des `log_autovacuum_min_duration`-Parameters ist der Schwellenwert (in Millisekunden), über dem Selbstbereinigung-Aktionen protokolliert werden. Eine Einstellung von `-1` protokolliert nichts, während eine Einstellung von 0 alle Aktionen protokolliert. Wie bei `rds.force_autovacuum_logging_level`, Standardwerte für `log_autovacuum_min_duration` sind versionsabhängig wie folgt:

- `10000 ms`— Postgre SQL 14, Postgre SQL 13, Postgre SQL 12 und Postgre 11 SQL
- `(empty)`— Kein Standardwert für Postgre 10 und Postgre 9.6 SQL SQL

Wir empfehlen Ihnen, `rds.force_autovacuum_logging_level` auf `WARNING` einzustellen. Wir empfehlen auch, dass Sie `log_autovacuum_min_duration` auf einen Wert von 1000 bis 5000 einstellen. Eine Einstellung von 5000 Protokollaktivitäten, die länger als 5000 Millisekunden dauern. Jede andere Einstellung als –1 protokolliert auch Meldungen, wenn die Selbstbereinigungsaktion aufgrund einer widersprüchlichen Sperre oder gleichzeitig verworfener Beziehungen übersprungen wird. Weitere Informationen finden Sie unter Automatisches Staubsaugen in der Postgre-Dokumentation. SQL

Um Probleme zu beheben, können Sie die `rds.force_autovacuum_logging_level`-Parameter in eine der Debugebenen ändern, von `debug1` bis zu `debug5` für die ausführlichsten Informationen. Wir empfehlen, die Debug-Einstellungen für kurze Zeiträume und nur zur Fehlerbehebung zu verwenden. Weitere Informationen finden Sie in der Postgre-Dokumentation unter Wann müssen Sie sich anmelden?. SQL

> **Note**
>
> Postgre SQL ermöglicht es dem `rds_superuser` Konto, Autovacuum-Sitzungen anzusehen. `pg_stat_activity` Sie können beispielsweise eine Selbstbereinigungssitzung identifizieren und beenden, die die Ausführung eines Befehls blockiert oder langsamer als ein manuell ausgegebener Bereinigungsbefehl ausgeführt wird.

Das Verhalten von Autovacuum bei ungültigen Datenbanken verstehen

Ein neuer Wert `-2` wird in die `datconnlimit` Spalte im `pg_database` Katalog eingefügt, um Datenbanken, die während des DROP DATABASE-Vorgangs unterbrochen wurden, als ungültig kenntlich zu machen.

Dieser neue Wert ist in den folgenden Versionen von verfügbar:

- 15.4 und alle höheren Versionen
- 14.9 und höhere Versionen
- 13.12 und höhere Versionen
- 12.16 und höhere Versionen
- 11.21 und höhere Versionen

Ungültige Datenbanken haben keinen Einfluss auf die Fähigkeit von Autovacuum, gültige Datenbanken einzufrieren. Autovacuum ignoriert ungültige Datenbanken. Folglich funktionieren reguläre Vakuumoperationen weiterhin ordnungsgemäß und effizient für alle gültigen Datenbanken in Ihrer PostgreSQL-Umgebung.

Themen

- Transaktions-ID überwachen
- Anpassung der Überwachungsanfrage
- Das Problem mit einer ungültigen Datenbank wird behoben

Transaktions-ID überwachen

Diese `age(datfrozenxid)` Funktion wird häufig verwendet, um das Alter der Transaktions-ID (XID) von Datenbanken zu überwachen, um zu verhindern, dass die Transaktions-ID umschlüsselt wird.

Da ungültige Datenbanken von der automatischen Datenabsaugung ausgenommen sind, kann ihr Transaktions-ID (XID) -Zähler den Maximalwert von erreichen`2 billion`, bis zu diesem Wert wechseln und diesen Zyklus auf unbestimmte `- 2 billion` Zeit fortsetzen. Eine typische Abfrage zur Überwachung des Transaktions-ID-Wraparounds könnte wie folgt aussehen:

```
SELECT max(age(datfrozenxid)) FROM pg_database;
```

Mit der Einführung des Werts -2 für `datconnlimit` können ungültige Datenbanken jedoch die Ergebnisse dieser Abfrage verfälschen. Da diese Datenbanken nicht gültig sind und nicht Teil regelmäßiger Wartungsprüfungen sein sollten, können sie zu Fehlalarmen führen, sodass Sie glauben, dass der Wert höher `age(datfrozenxid)` ist, als er tatsächlich ist.

Anpassung der Überwachungsanfrage

Um eine genaue Überwachung zu gewährleisten, sollten Sie Ihre Überwachungsanfrage so anpassen, dass ungültige Datenbanken ausgeschlossen werden. Folgen Sie dieser empfohlenen Abfrage:

```
SELECT
    max(age(datfrozenxid))
FROM
    pg_database
WHERE
    datconnlimit <> -2;
```

Diese Abfrage stellt sicher, dass nur gültige Datenbanken bei der `age(datfrozenxid)` Berechnung berücksichtigt werden, sodass das Alter der Transaktions-ID in Ihrer gesamten PostgreSQL-Umgebung genau wiedergegeben wird.

Das Problem mit einer ungültigen Datenbank wird behoben

Wenn Sie versuchen, eine Verbindung zu einer ungültigen Datenbank herzustellen, wird möglicherweise eine Fehlermeldung ähnlich der folgenden angezeigt:

```
postgres=> \c db1
connection to server at "mydb.xxxxxxxxxx.us-west-2.rds.amazonaws.com" (xx.xx.xx.xxx),
 port xxxx failed: FATAL:  cannot connect to invalid database "db1"
HINT:  Use DROP DATABASE to drop invalid databases.
Previous connection kept
```

Wenn der `log_min_messages` Parameter auf `DEBUG2` oder höher gesetzt ist, fallen Ihnen außerdem möglicherweise die folgenden Protokolleinträge auf, die darauf hinweisen, dass der Autovacuum-Prozess die ungültige Datenbank überspringt:

```
2024-07-30 05:59:00 UTC::@:[32000]:DEBUG:  autovacuum: skipping invalid database "db6"
2024-07-30 05:59:00 UTC::@:[32000]:DEBUG:  autovacuum: skipping invalid database "db1"
```

Folgen Sie den `HINT` Anweisungen beim Verbindungsversuch, um das Problem zu beheben. Stellen Sie mit Ihrem RDS-Master-Konto oder einem Datenbankkonto mit der `rds_superuser` Rolle eine Connect zu einer beliebigen gültigen Datenbank her und löschen Sie ungültige Datenbanken.

```
SELECT
    'DROP DATABASE ' || quote_ident(datname) || ';'
FROM
    pg_database
WHERE
    datconnlimit = -2 \gexec
```

In PostgreSQL ist das Vacuuming von entscheidender Bedeutung, um die Integrität der Datenbank sicherzustellen, da dadurch Speicherplatz zurückgewonnen und Transaktions-ID-Wraparound-Probleme vermieden werden. Manchmal kann jedoch verhindert werden, dass das Vacuuming wie gewünscht funktioniert, was zu Leistungseinbußen, zu einer Überlastung des Speichers und sogar zu einer Beeinträchtigung der Verfügbarkeit Ihrer DB-Instance durch den Transaktions-ID-Wraparound führen kann. Daher ist die Identifizierung und Lösung dieser Probleme für eine optimale Datenbankleistung und -verfügbarkeit unerlässlich. Weitere Informationen zu Autovacuum finden Sie

unter Grundlegendes zu Autovacuum-Umgebungen in Amazon RDS for PostgreSQL PostgreSQL-Umgebungen.

Diese `postgres_get_av_diag()` Funktion hilft dabei, Probleme zu identifizieren, die den aggressiven Vakuumprozess entweder verhindern oder verzögern. Es werden Vorschläge bereitgestellt, zu denen Befehle zur Behebung des Problems gehören können, wenn es identifiziert werden kann, oder Anleitungen für weitere Diagnosen, wenn das Problem nicht identifiziert werden kann. Aggressive Vakuumblocker werden gemeldet, wenn das Alter den von RDS festgelegten Schwellenwert für die automatische Vakuumierung von 500 Millionen Transaktionen überschreitet. IDs

Wie alt ist die Transaktions-ID?

Die `age()` Funktion für die Transaktion IDs berechnet die Anzahl der Transaktionen, die seit der ältesten nicht eingefrorenen Transaktions-ID für eine Datenbank (`pg_database.datfrozenxid`) oder Tabelle (`pg_class.relfrozenxid`) stattgefunden haben. Dieser Wert gibt die Datenbankaktivität seit dem letzten aggressiven Vakuumvorgang an und hebt die wahrscheinliche Arbeitslast für bevorstehende VACUUM-Prozesse hervor.

Was ist ein aggressives Vakuum?

Bei einem aggressiven VACUUM-Vorgang werden alle Seiten in einer Tabelle umfassend gescannt, einschließlich der Seiten, die normalerweise bei einem normalen Vorgang übersprungen werden. VACUUMs Dieser gründliche Scan zielt darauf ab, Transaktionen, die IDs sich ihrem Höchstalter nähern, „einzufrieren“ und so eine Situation, die als Transaktions-ID-Wraparound bezeichnet wird, effektiv zu verhindern.

`postgres_get_av_diag()`Um Blocker melden zu können, muss der Blocker mindestens 500 Millionen Transaktionen alt sein.

Themen

-
-
-
-
-
-

Die `postgres_get_av_diag()` Funktion ist derzeit in den folgenden Versionen von RDS für PostgreSQL verfügbar:

- 17.2 und höhere 17-Versionen
- 16.7 und höhere 16 Versionen
- 15.11 und höhere 15-Versionen
- 14.16 und höhere 14-Versionen
- 13.19 und höhere 13 Versionen

Um sie zu verwenden`postgres_get_av_diag()`, müssen Sie die `rds_tools` Erweiterung erstellen.

```
postgres=> CREATE EXTENSION rds_tools ;
CREATE EXTENSION
```

Stellen Sie sicher, dass die Erweiterung installiert ist.

```
postgres=> \dx rds_tools
             List of installed extensions
   Name    | Version |  Schema   |                    Description
 ----------+---------+-----------
+------------------------------------------------------------
 rds_tools |   1.8   | rds_tools | miscellaneous administrative functions for RDS
 PostgreSQL
 1 row
```

Stellen Sie sicher, dass die Funktion erstellt wurde.

```
postgres=> SELECT
    proname function_name,
    pronamespace::regnamespace function_schema,
    proowner::regrole function_owner
FROM
    pg_proc
WHERE
    proname = 'postgres_get_av_diag';
    function_name     | function_schema | function_owner
----------------------+-----------------+----------------
```

```
 postgres_get_av_diag | rds_tools       | rds_superuser
(1 row)
```

Die Abfrage muss in der Datenbank mit der ältesten Transaktions-ID ausgeführt werden, um genaue Ergebnisse zu erhalten. Weitere Hinweise zur Verwendung der Datenbank mit der ältesten Transaktions-ID finden Sie unter Keine Verbindung mit der Datenbank mit dem Alter der ältesten Transaktions-ID

```
SELECT
    blocker,
    DATABASE,
    blocker_identifier,
    wait_event,
    TO_CHAR(autovacuum_lagging_by, 'FM9,999,999,999') AS autovacuum_lagging_by,
    suggestion,
    suggested_action
FROM (
    SELECT
        *
    FROM
        rds_tools.postgres_get_av_diag ()
    ORDER BY
        autovacuum_lagging_by DESC) q;
```

Die `postgres_get_av_diag()` Funktion gibt eine Tabelle mit den folgenden Informationen zurück:

Blocker

Gibt die Kategorie der Datenbankaktivität an, die das Vakuum blockiert.

- Aktive Aussage
- Leerlauf in Transaktion
- Vorbereitete Transaktion
- Steckplatz für logische Replikation
- Replikat mit physischem Replikationssteckplatz lesen
- Replik mit Streaming-Replikation lesen
- Temporäre Tabellen

Datenbank

Gibt den Namen der Datenbank an, sofern zutreffend und unterstützt. Dies ist die Datenbank, in der die Aktivität andauert und die Autovakuierung blockiert oder blockieren wird. Dies ist die Datenbank, zu der Sie eine Verbindung herstellen und Maßnahmen ergreifen müssen.

blocker_identifier

Gibt die Kennung der Aktivität an, die den Autovakuumvorgang blockiert oder blockieren wird. Bei dem Bezeichner kann es sich um eine Prozess-ID zusammen mit einer SQL-Anweisung, einer vorbereiteten Transaktion, einer IP-Adresse einer gelesenen Replik und dem Namen des logischen oder physischen Replikationsslots handeln.

wait_event

Gibt das Wartungsereignis für das der blockierenden Sitzung an und gilt für die folgenden Blocker:

- Aktive Anweisung
- Leerlauf in Transaktion

autovacum_lagging_by

Gibt pro Kategorie die Anzahl der Transaktionen an, bei denen Autovacuum bei seinen Backlog-Arbeiten hinterherhinkt.

Vorschlag

Gibt Vorschläge zur Lösung des Blockers an. Diese Anweisungen enthalten gegebenenfalls den Namen der Datenbank, in der die Aktivität vorhanden ist, gegebenenfalls die Prozess-ID (PID) der Sitzung und die auszuführende Aktion.

suggested_action

Schlägt die Maßnahme vor, die ergriffen werden muss, um den Blocker zu beheben.

Autovacuum führt aggressive Löschvorgänge durch und senkt das Alter der Transaktion auf einen Wert unter den durch den Parameter Ihrer IDs RDS-Instance angegebenen Schwellenwert. `autovacuum_freeze_max_age` Sie können dieses Alter anhand der CloudWatch Amazon-Metrik verfolgen`MaximumUsedTransactionIDs`.

Um die Einstellung von `autovacuum_freeze_max_age` (die standardmäßig 200 Millionen Transaktionen hat IDs) für Ihre Amazon RDS-Instance zu ermitteln, können Sie die folgende Abfrage verwenden:

```
SELECT
    TO_CHAR(setting::bigint, 'FM9,999,999,999') autovacuum_freeze_max_age
FROM
    pg_settings
WHERE
    name = 'autovacuum_freeze_max_age';
```

Beachten Sie, dass `postgres_get_av_diag()` nur dann nach aggressiven Vakuumblockern gesucht wird, wenn das Alter den adaptiven Autovakuum-Schwellenwert von Amazon RDS von 500 Millionen Transaktionen überschreitet. IDs `postgres_get_av_diag()`Um Blocker zu erkennen, muss der Blocker mindestens 500 Millionen Transaktionen alt sein.

Die `postgres_get_av_diag()` Funktion identifiziert die folgenden Arten von Blockern:

Themen

- Aktive Aussage
- Leerlauf in Transaktion
- Vorbereitete Transaktion
- Steckplatz für logische Replikation
- Read Replicas
- Temporäre Tabellen

Aktive Aussage

In PostgreSQL ist eine aktive Anweisung eine SQL-Anweisung, die gerade von der Datenbank ausgeführt wird. Dazu gehören Abfragen, Transaktionen oder alle laufenden Operationen. Bei der Überwachung über `pg_stat_activity` gibt die Statusspalte an, dass der Prozess mit der entsprechenden PID aktiv ist.

Die `postgres_get_av_diag()` Funktion zeigt eine Ausgabe ähnlich der folgenden an, wenn sie eine Anweisung identifiziert, bei der es sich um eine aktive Anweisung handelt.

```
blocker               | Active statement
database              | my_database
blocker_identifier    | SELECT pg_sleep(20000);
wait_event            | Timeout:PgSleep
autovacuum_lagging_by | 568,600,871
```

```
suggestion            | Connect to database "my_database", review carefully and you
 may consider terminating the process using suggested_action. For more information, see
 Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"SELECT pg_terminate_backend (29621);"}
```

Vorgeschlagene Maßnahme

Gemäß den Anweisungen in der `suggestion` Spalte kann der Benutzer eine Verbindung zu der Datenbank herstellen, in der sich die aktive Anweisung befindet. Wie in der `suggested_action` Spalte angegeben, ist es ratsam, die Option zum Beenden der Sitzung sorgfältig zu prüfen. Wenn die Beendigung sicher ist, können Sie die `pg_terminate_backend()` Funktion verwenden, um die Sitzung zu beenden. Diese Aktion kann von einem Administrator (z. B. dem RDS-Hauptkonto) oder einem Benutzer mit den erforderlichen Rechten ausgeführt `pg_terminate_backend()` werden.

> **Warning**
>
> Eine beendete Sitzung macht die von ihr vorgenommenen Änderungen rückgängig (`ROLLBACK`). Je nach Ihren Anforderungen möchten Sie die Anweisung möglicherweise erneut ausführen. Es wird jedoch empfohlen, dies erst zu tun, nachdem der Autovakuumprozess seinen aggressiven Vakuumvorgang abgeschlossen hat.

Leerlauf in Transaktion

Eine Idle in Transaction-Anweisung bezieht sich auf jede Sitzung, die eine explizite Transaktion geöffnet hat (z. B. durch Ausgabe einer `BEGIN` Anweisung), einige Arbeiten ausgeführt hat und nun darauf wartet, dass der Client entweder weitere Aufgaben weitergibt oder das Ende der Transaktion durch Ausgabe eines`COMMIT`,`ROLLBACK`, oder signalisiert `END` (was zu einem impliziten `COMMIT` Ergebnis führen würde).

Die `postgres_get_av_diag()` Funktion zeigt eine Ausgabe an, die der folgenden ähnelt, wenn sie eine `idle in transaction` Anweisung als Blocker identifiziert.

```
blocker               | idle in transaction
database              | my_database
blocker_identifier    | INSERT INTO tt SELECT * FROM tt;
wait_event            | Client:ClientRead
autovacuum_lagging_by | 1,237,201,759
suggestion            | Connect to database "my_database", review carefully and you
 may consider terminating the process using suggested_action. For more information, see
 Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
```

```
suggested_action      | {"SELECT pg_terminate_backend (28438);"}
```

Vorgeschlagene Maßnahme

Wie in der `suggestion` Spalte angegeben, können Sie eine Verbindung zu der Datenbank herstellen, in der sich die inaktive Transaktionssitzung befindet, und die Sitzung mithilfe der `pg_terminate_backend()` Funktion beenden. Der Benutzer kann Ihr Admin-Benutzer (RDS-Hauptkonto) oder ein Benutzer mit der entsprechenden `pg_terminate_backend()` Berechtigung sein.

> **Warning**
>
> Eine beendete Sitzung macht die von ihr vorgenommenen Änderungen rückgängig (`ROLLBACK`). Je nach Ihren Anforderungen möchten Sie die Anweisung möglicherweise erneut ausführen. Es wird jedoch empfohlen, dies erst zu tun, nachdem der Autovakuumprozess seinen aggressiven Vakuumvorgang abgeschlossen hat.

Vorbereitete Transaktion

PostgreSQL ermöglicht Transaktionen, die Teil einer zweiphasigen Commit-Strategie sind, die als vorbereitete Transaktionen bezeichnet werden. Diese werden aktiviert, indem der `max_prepared_transactions` Parameter auf einen Wert ungleich Null gesetzt wird. Vorbereitete Transaktionen sollen sicherstellen, dass eine Transaktion dauerhaft ist und auch nach Datenbankabstürzen, Neustarts oder Verbindungsabbrüchen des Clients verfügbar bleibt. Wie bei normalen Transaktionen wird ihnen eine Transaktions-ID zugewiesen, was sich auf das Autovakuum auswirken kann. In einem vorbereiteten Zustand kann Autovacuum das Einfrieren nicht durchführen und es kann zu einem Umbruch der Transaktions-ID kommen.

Wenn Transaktionen auf unbestimmte Zeit vorbereitet bleiben, ohne von einem Transaktionsmanager bearbeitet zu werden, werden sie zu verwaisten vorbereiteten Transaktionen. Die einzige Möglichkeit, dies zu beheben, besteht darin, die Transaktion mit den `ROLLBACK PREPARED` Befehlen oder festzuschreiben `COMMIT PREPARED` oder rückgängig zu machen.

> **Note**
>
> Beachten Sie, dass ein Backup, das während einer vorbereiteten Transaktion erstellt wurde, diese Transaktion auch nach der Wiederherstellung enthält. In den folgenden Informationen erfahren Sie, wie Sie solche Transaktionen finden und abschließen können.

Die `postgres_get_av_diag()` Funktion zeigt die folgende Ausgabe an, wenn sie einen Blocker identifiziert, bei dem es sich um eine vorbereitete Transaktion handelt.

```
blocker               | Prepared transaction
database              | my_database
blocker_identifier    | myptx
wait_event            | Not applicable
autovacuum_lagging_by | 1,805,802,632
suggestion            | Connect to database "my_database" and consider either COMMIT
 or ROLLBACK the prepared transaction using suggested_action. For more information, see
 Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"COMMIT PREPARED 'myptx';",[OR],"ROLLBACK PREPARED 'myptx';"}
```

Vorgeschlagene Maßnahme

Stellen Sie, wie in der Spalte mit den Vorschlägen erwähnt, eine Verbindung zu der Datenbank her, in der sich die vorbereitete Transaktion befindet. Prüfen Sie anhand der `suggested_action` Spalte sorgfältig, ob Sie entweder `COMMIT` oder ausführen `ROLLBACK` möchten und welche Aktion dann angemessen ist.

Um vorbereitete Transaktionen generell zu überwachen, bietet PostgreSQL eine Katalogansicht namens. `pg_prepared_xacts` Sie können die folgende Abfrage verwenden, um vorbereitete Transaktionen zu finden.

```
SELECT
    gid,
    prepared,
    owner,
    database,
    transaction AS oldest_xmin
FROM
    pg_prepared_xacts
ORDER BY
```

```
    age(transaction) DESC;
```

Steckplatz für logische Replikation

Der Zweck eines Replikationsslots besteht darin, nicht verbrauchte Änderungen so lange aufzubewahren, bis sie auf einen Zielserver repliziert wurden. Weitere Informationen finden Sie unter Logische Replikation von PostgreSQL.

Es gibt zwei Arten von Steckplätzen für logische Replikation.

Inaktive Steckplätze für logische Replikation

Wenn die Replikation beendet wird, können nicht verbrauchte Transaktionsprotokolle nicht entfernt werden, und der Replikationssteckplatz wird inaktiv. Obwohl ein inaktiver logischer Replikationssteckplatz derzeit nicht von einem Abonnenten verwendet wird, verbleibt er auf dem Server, was zur Aufbewahrung von WAL-Dateien führt und verhindert, dass alte Transaktionsprotokolle entfernt werden. Dies kann die Festplattennutzung erhöhen und insbesondere verhindern, dass Autovacuum interne Katalogtabellen bereinigt, da das System verhindern muss, dass LSN-Informationen überschrieben werden. Wenn dies nicht behoben wird, kann dies zu einer Überlastung des Katalogs, zu Leistungseinbußen und zu einem erhöhten Risiko eines Rundumschutzes führen, was möglicherweise zu Transaktionsausfällen führen kann.

Aktive, aber langsame Steckplätze für logische Replikation

Manchmal verzögert sich das Entfernen toter Katalogtupel aufgrund von Leistungseinbußen bei der logischen Replikation. Diese Verzögerung bei der Replikation verlangsamt die Aktualisierung des Systems `catalog_xmin` und kann zu einer Überlastung des Katalogs und zu einem vollständigen Vakuum führen.

Die `postgres_get_av_diag()` Funktion zeigt eine Ausgabe ähnlich der folgenden an, wenn sie einen logischen Replikationsslot als Blocker findet.

```
blocker               | Logical replication slot
database              | my_database
blocker_identifier    | slot1
wait_event            | Not applicable
autovacuum_lagging_by | 1,940,103,068
suggestion            | Ensure replication is active and resolve any lag for the slot
 if active. If inactive, consider dropping it using the command in suggested_action.
 For more information, see Working with PostgreSQL autovacuum in the Amazon RDS User
 Guide.
```

```
suggested_action      | {"SELECT pg_drop_replication_slot('slot1') FROM
 pg_replication_slots WHERE active = 'f';"}
```

Vorgeschlagene Maßnahme

Um dieses Problem zu beheben, überprüfen Sie die Replikationskonfiguration auf Probleme mit dem Zielschema oder den Daten, die den Anwenden-Prozess möglicherweise beenden könnten. Die häufigsten Gründe sind die folgenden:

- Fehlende Spalten
- Inkompatibler Datentyp
- Daten stimmen nicht überein
- Fehlende Tabelle

Wenn das Problem mit Infrastrukturproblemen zusammenhängt:

- Netzwerkprobleme — Wie löse ich Probleme mit einer Amazon RDS-Datenbank in einem inkompatiblen Netzwerkstatus? .
- Die Datenbank oder DB-Instance ist aus den folgenden Gründen nicht verfügbar:
  - Die Replikat-Instance hat keinen Speicherplatz mehr — Informationen zum Hinzufügen von Speicherplatz finden Sie unter Amazon RDS-DB-Instances ist nicht mehr genügend Speicherplatz verfügbar.
  - Inkompatible Parameter — Testbericht Wie kann ich eine Amazon RDS-DB-Instance reparieren, die im Status „inkompatible Parameter“ hängengeblieben ist? für weitere Informationen darüber, wie Sie das Problem lösen können.

Wenn sich Ihre Instance außerhalb des AWS Netzwerks oder in Betrieb befindet AWS EC2, wenden Sie sich an Ihren Administrator, um zu erfahren, wie Sie die Verfügbarkeits- oder Infrastrukturprobleme lösen können.

Den inaktiven Slot löschen

> ⚠ Warning
>
> Vorsicht: Bevor Sie einen Replikationssteckplatz löschen, stellen Sie sorgfältig sicher, dass er nicht repliziert wird, dass er inaktiv ist und sich in einem Zustand befindet, der

nicht wiederhergestellt werden kann. Ein vorzeitiges Löschen eines Steckplatzes kann die Replikation unterbrechen oder zu Datenverlust führen.

Nachdem Sie sich vergewissert haben, dass der Replikationssteckplatz nicht mehr benötigt wird, löschen Sie ihn, damit die automatische Bereinigung fortgesetzt werden kann. Diese Bedingung `active = 'f'` stellt sicher, dass nur ein inaktiver Steckplatz gelöscht wird.

```
SELECT pg_drop_replication_slot('slot1') WHERE active ='f'
```

Read Replicas

Wenn die `hot_standby_feedback` Einstellung für Amazon RDS-Read Replicas aktiviert ist, verhindert sie, dass Autovacuum in der Primärdatenbank tote Zeilen entfernt, die möglicherweise noch von Abfragen benötigt werden, die auf der Read Replica ausgeführt werden. Dies betrifft alle Arten von physischen Read Replicas, einschließlich solcher, die mit oder ohne Replikationssteckplätze verwaltet werden. Dieses Verhalten ist notwendig, da Abfragen, die auf dem Standby-Replikat ausgeführt werden, erfordern, dass diese Zeilen auf dem primären Replikat verfügbar bleiben, wodurch Abfragekonflikte und Abbruchvorgänge vermieden werden.

Read Replica mit physischem Replikationssteckplatz

Diese Steckplätze stellen sicher, dass die Primärdatenbank wichtige Write-Ahead-Protokolldateien aufbewahrt, bis das Replikat sie verarbeitet, wodurch die Datenkonsistenz auch bei Netzwerkunterbrechungen gewahrt bleibt.

Beginnend mit RDS für PostgreSQL Version 14 verwenden alle Replikate Replikationsslots. In früheren Versionen verwendeten nur regionsübergreifende Replikate Replikationsslots.

Die `postgres_get_av_diag()` Funktion zeigt eine Ausgabe ähnlich der folgenden an, wenn sie eine Lesereplik mit einem physischen Replikationssteckplatz als Blocker findet.

```
blocker               | Read replica with physical replication slot
database              |
blocker_identifier    | rds_us_west_2_db_xxxxxxxxxxxxxxxxxxxx
wait_event            | Not applicable
autovacuum_lagging_by | 554,080,689
suggestion            | Run the following query on the replica
 "rds_us_west_2_db_xxxxxxxxxxxxxxxxxxx" to find the long running query:
```

```
                      | SELECT * FROM pg_catalog.pg_stat_activity WHERE
 backend_xmin::text::bigint = 757989377;

                      | Review carefully and you may consdier terminating the query on
 read replica using suggested_action. For more information, see Working with PostgreSQL
 autovacuum in the Amazon RDS User Guide.                                    +
          |
suggested_action      | {"SELECT pg_terminate_backend(pid) FROM
 pg_catalog.pg_stat_activity WHERE backend_xmin::text::bigint = 757989377;","
                                                                             +
                      | [OR]

                                                +
                      | ","Disable hot_standby_feedback","

                                                +
                      | [OR]

                                                +
                      | ","Delete the read replica if not needed"}
```

Read Replica mit Streaming-Replikation

Amazon RDS ermöglicht die Einrichtung von Read Replicas ohne physischen Replikationssteckplatz in älteren Versionen bis Version 13. Dieser Ansatz reduziert den Overhead, da der Primärserver WAL-Dateien aggressiver recyceln kann, was in Umgebungen mit begrenztem Festplattenspeicher vorteilhaft ist und gelegentlich toleriert werden kann. ReplicaLag Ohne Steckplatz muss der Standby-Modus jedoch synchron bleiben, um zu verhindern, dass WAL-Dateien verloren gehen. Amazon RDS verwendet archivierte WAL-Dateien, damit das Replikat aufholen kann, wenn es ins Hintertreffen gerät. Dieser Vorgang erfordert jedoch eine sorgfältige Überwachung und kann langsam sein.

Die `postgres_get_av_diag()` Funktion zeigt eine Ausgabe ähnlich der folgenden an, wenn sie eine Streaming-Read-Replik als Blocker findet.

```
blocker               | Read replica with streaming replication slot
database              | Not applicable
blocker_identifier    | xx.x.x.xxx/xx
wait_event            | Not applicable
autovacuum_lagging_by | 610,146,760
suggestion            | Run the following query on the replica "xx.x.x.xxx" to find the
 long running query:
                                                                              +
```

```
                      | SELECT * FROM pg_catalog.pg_stat_activity WHERE
 backend_xmin::text::bigint = 348319343;

               +
                      | Review carefully and you may consdier terminating the query on
 read replica using suggested_action. For more information, see Working with PostgreSQL
 autovacuum in the Amazon RDS User Guide.                                          +
                      |
suggested_action      | {"SELECT pg_terminate_backend(pid) FROM
 pg_catalog.pg_stat_activity WHERE backend_xmin::text::bigint = 348319343;","

                       +
                      | [OR]

                                                                                   +
                      | ","Disable hot_standby_feedback","

                                                                                   +
                      | [OR]

                                                                                   +
                      | ","Delete the read replica if not needed"}
```

Vorgeschlagene Maßnahme

Wie in der `suggested_action` Spalte empfohlen, sollten Sie diese Optionen sorgfältig prüfen, um die automatische Vakuumierung zu entsperren.

- Die Abfrage beenden — Folgen Sie den Anweisungen in der Spalte mit den Vorschlägen, um eine Verbindung zur Read Replica herzustellen, wie in der Spalte suggested_action angegeben. Es empfiehlt sich, die Option zum Beenden der Sitzung sorgfältig zu prüfen. Wenn die Beendigung als sicher erachtet wird, können Sie die `pg_terminate_backend()` Funktion verwenden, um die Sitzung zu beenden. Diese Aktion kann von einem Administrator (wie dem RDS-Hauptkonto) oder einem Benutzer mit den erforderlichen pg_terminate_backend () -Privilegien ausgeführt werden.

  Sie können den folgenden SQL-Befehl auf der Read Replica ausführen, um die Abfrage zu beenden, die verhindert, dass das Vakuum auf der Primärseite alte Zeilen bereinigt. Der Wert von `backend_xmin` wird in der Ausgabe der Funktion angegeben:

  ```
  SELECT
      pg_terminate_backend(pid)
  FROM
  ```

```
    pg_catalog.pg_stat_activity
WHERE
    backend_xmin::text::bigint = backend_xmin;
```

- Feedback im Hot-Standby-Modus deaktivieren — Erwägen Sie, den `hot_standby_feedback` Parameter zu deaktivieren, wenn er zu erheblichen Vakuumverzögerungen führt.

  Der `hot_standby_feedback` Parameter ermöglicht es einem Read Replica, den Primärserver über seine Abfrageaktivität zu informieren. Dadurch wird verhindert, dass der Primärserver Tabellen oder Zeilen, die im Standby-Modus verwendet werden, entfernt. Dadurch wird zwar die Stabilität der Abfrage im Standby-Modus gewährleistet, das Löschen auf dem Primärserver kann jedoch erheblich verzögert werden. Wenn Sie diese Funktion deaktivieren, kann der Primärcomputer mit dem Staubsaugen fortfahren, ohne darauf warten zu müssen, dass der Standby-Modus aufholt. Dies kann jedoch dazu führen, dass Abfragen im Standby-Modus abgebrochen oder fehlschlagen, wenn dieser versucht, auf Zeilen zuzugreifen, die vom Primärserver gelöscht wurden.
- Löschen Sie die Read Replica, falls sie nicht mehr benötigt wird — Wenn die Read Replica nicht mehr benötigt wird, können Sie sie löschen. Dadurch entfällt der damit verbundene Replikationsaufwand und der Primärserver kann Transaktionsprotokolle recyceln, ohne vom Replikat zurückgehalten zu werden.

Temporäre Tabellen

Temporäre Tabellen, die mit dem `TEMPORARY` Schlüsselwort erstellt wurden, befinden sich im temporären Schema, z. B. pg_temp_xxx, und sind nur für die Sitzung zugänglich, in der sie erstellt wurden. Temporäre Tabellen werden gelöscht, wenn die Sitzung endet. Diese Tabellen sind jedoch für den Autovacuum-Prozess von PostgreSQL unsichtbar und müssen von der Sitzung, in der sie erstellt wurden, manuell gelöscht werden. Der Versuch, die temporäre Tabelle aus einer anderen Sitzung zu löschen, hat keine Wirkung.

Unter ungewöhnlichen Umständen ist eine temporäre Tabelle vorhanden, ohne dass sie einer aktiven Sitzung gehört. Wenn die Eigentümersitzung aufgrund eines fatalen Absturzes, eines Netzwerkproblems oder eines ähnlichen Ereignisses unerwartet beendet wird, wird die temporäre Tabelle möglicherweise nicht bereinigt, sodass sie als „verwaiste“ Tabelle zurückbleibt. Wenn der PostgreSQL-Autovacuum-Prozess eine verwaiste temporäre Tabelle erkennt, protokolliert er die folgende Meldung:

```
LOG: autovacuum: found orphan temp table \"%s\".\"%s\" in database \"%s\"
```

Die `postgres_get_av_diag()` Funktion zeigt eine Ausgabe ähnlich der folgenden an, wenn sie eine temporäre Tabelle als Blocker identifiziert. Damit die Funktion die Ausgabe in Bezug auf temporäre Tabellen korrekt anzeigt, muss sie in derselben Datenbank ausgeführt werden, in der diese Tabellen existieren.

```
blocker               | Temporary table
database              | my_database
blocker_identifier    | pg_temp_14.ttemp
wait_event            | Not applicable
autovacuum_lagging_by | 1,805,802,632
suggestion            | Connect to database "my_database". Review carefully, you
 may consider dropping temporary table using command in suggested_action. For more
 information, see Working with PostgreSQL autovacuum in the Amazon RDS User Guide.
suggested_action      | {"DROP TABLE ttemp;"}
```

Vorgeschlagene Maßnahme

Folgen Sie den Anweisungen in der `suggestion` Spalte der Ausgabe, um die temporäre Tabelle zu identifizieren und zu entfernen, die das Ausführen von Autovacuum verhindert. Verwenden Sie den folgenden Befehl, um die temporäre Tabelle zu löschen, die von `postgres_get_av_diag()` gemeldet wurde. Ersetzen Sie den Tabellennamen auf der Grundlage der von der `postgres_get_av_diag()` Funktion bereitgestellten Ausgabe.

```
DROP TABLE my_temp_schema.my_temp_table;
```

Die folgende Abfrage kann verwendet werden, um temporäre Tabellen zu identifizieren:

```
SELECT
    oid,
    relname,
    relnamespace::regnamespace,
    age(relfrozenxid)
FROM
    pg_class
WHERE
relpersistence = 't'
ORDER BY
    age(relfrozenxid) DESC;
```

In diesem Abschnitt werden weitere Gründe untersucht, die verhindern können, dass das Staubsaugen voranschreitet. Diese Probleme sind derzeit anhand der Funktion nicht direkt identifizierbar. `postgres_get_av_diag()`

Themen

- Ungültige Seiten
- Inkonsistenz im Index
- Außergewöhnlich hohe Transaktionsrate

Ungültige Seiten

Ein ungültiger Seitenfehler tritt auf, wenn PostgreSQL beim Zugriff auf diese Seite eine Nichtübereinstimmung in der Prüfsumme einer Seite feststellt. Der Inhalt ist nicht lesbar, wodurch Autovacuum verhindert, dass Tupel eingefroren werden. Dadurch wird der Säuberungsvorgang effektiv gestoppt. Der folgende Fehler wird in das Protokoll von PostgreSQL geschrieben:

```
WARNING:  page verification failed, calculated checksum YYYYY but expected XXXX
ERROR:  invalid page in block ZZZZZ of relation base/XXXXX/XXXXX
CONTEXT:  automatic vacuum of table myschema.mytable
```

Ermitteln Sie den Objekttyp

```
ERROR: invalid page in block 4305910 of relation base/16403/186752608
WARNING: page verification failed, calculated checksum 50065 but expected 60033
```

Aus der Fehlermeldung geht hervor, dass der Pfad die folgenden Informationen `base/16403/186752608` enthält:

- „base“ ist der Verzeichnisname unter dem PostgreSQL-Datenverzeichnis.
- „16403" ist die Datenbank-OID, die Sie im Systemkatalog nachschlagen können. `pg_database`
- „186752608“ ist die`relfilenode`, mit der Sie das Schema und den Objektnamen im Systemkatalog nachschlagen können. `pg_class`

Indem Sie die Ausgabe der folgenden Abfrage in der betroffenen Datenbank überprüfen, können Sie den Objekttyp ermitteln. Die folgende Abfrage ruft Objektinformationen für oid ab: 186752608. Ersetzen Sie die OID durch die OID, die für den aufgetretenen Fehler relevant ist.

```
SELECT
    relname AS object_name,
    relkind AS object_type,
    nspname AS schema_name
FROM
    pg_class c
    JOIN pg_namespace n ON c.relnamespace = n.oid
WHERE
    c.oid = 186752608;
```

Weitere Informationen finden Sie in der PostgreSQL-Dokumentation pg_classfür alle unterstützten Objekttypen, die in der `relkind` Spalte unter aufgeführt sind. `pg_class`

Anleitung

Die effektivste Lösung für dieses Problem hängt von der Konfiguration Ihrer spezifischen Amazon RDS-Instance und der Art der Daten ab, die von der inkonsistenten Seite betroffen sind.

Wenn der Objekttyp ein Index ist:

Es wird empfohlen, den Index neu zu erstellen.

- Verwenden der **`CONCURRENTLY`** Option — Vor PostgreSQL Version 12 erforderte die Neuerstellung eines Indexes eine exklusive Tabellensperre, die den Zugriff auf die Tabelle einschränkte. In PostgreSQL Version 12 und späteren Versionen ermöglicht die `CONCURRENTLY` Option das Sperren auf Zeilenebene, wodurch die Verfügbarkeit der Tabelle erheblich verbessert wird. Es folgt der Befehl:

  ```
  REINDEX INDEX ix_name CONCURRENTLY;
  ```

  Er `CONCURRENTLY` ist zwar weniger störend, kann aber bei stark frequentierten Tabellen langsamer sein. Erwägen Sie, den Index nach Möglichkeit in Zeiten mit geringem Datenverkehr zu erstellen.

  Weitere Informationen finden Sie in der PostgreSQL REINDEX-Dokumentation.

- Verwenden Sie die **`INDEX_CLEANUP FALSE`** Option — Wenn die Indizes umfangreich sind und schätzungsweise viel Zeit benötigen, bis sie fertig sind, können Sie die automatische Bereinigung aufheben, indem Sie ein manuelles Verfahren ausführen und dabei Indizes ausschließen. `VACUUM FREEZE` Diese Funktionalität ist in PostgreSQL Version 12 und späteren Versionen verfügbar.

Durch das Umgehen von Indizes können Sie den Vakuumprozess des inkonsistenten Indexes überspringen und das Wraparound-Problem verringern. Das zugrundeliegende Problem mit ungültigen Seiten wird dadurch jedoch nicht gelöst. Um das Problem mit der ungültigen Seite vollständig zu beheben und zu lösen, müssen Sie den Index dennoch neu erstellen.

Wenn es sich bei dem Objekttyp um eine materialisierte Ansicht handelt:

Wenn in einer materialisierten Ansicht ein ungültiger Seitenfehler auftritt, melden Sie sich bei der betroffenen Datenbank an und aktualisieren Sie sie, um die ungültige Seite zu beheben:

Aktualisieren Sie die materialisierte Ansicht:

```
REFRESH MATERIALIZED VIEW schema_name.materialized_view_name;
```

Wenn die Aktualisierung fehlschlägt, versuchen Sie, Folgendes neu zu erstellen:

```
DROP MATERIALIZED VIEW schema_name.materialized_view_name;
CREATE MATERIALIZED VIEW schema_name.materialized_view_name AS query;
```

Durch das Aktualisieren oder Neuerstellen der materialisierten Ansicht wird sie wiederhergestellt, ohne dass sich dies auf die zugrunde liegenden Tabellendaten auswirkt.

Für alle anderen Objekttypen:

Für alle anderen Objekttypen wenden Sie sich an den AWS Support.

Inkonsistenz im Index

Ein logisch inkonsistenter Index kann verhindern, dass das Autovakuumverfahren voranschreitet. Die folgenden oder ähnliche Fehler werden entweder während der Vakuumphase des Indexes oder beim Zugriff auf den Index durch SQL-Anweisungen protokolliert.

```
ERROR: right sibling's left-link doesn't match:block 5 links to 10 instead of expected
 2 in index ix_name
```

```
ERROR: failed to re-find parent key in index "XXXXXXXXXX" for deletion target page XXX
CONTEXT:  while vacuuming index index_name of relation schema.table
```

Anleitung

Erstellen Sie den Index manuell `VACUUM FREEZE` neu oder überspringen Sie `INDEX_CLEANUP` Indizes. Informationen zur Neuerstellung des Index finden Sie unter Wenn der Objekttyp ein Index ist.

- Verwenden der Option CONCURRENTLY — Vor PostgreSQL Version 12 erforderte die Neuerstellung eines Indexes eine exklusive Tabellensperre, die den Zugriff auf die Tabelle einschränkte. Bei PostgreSQL Version 12 und späteren Versionen ermöglicht die Option CONCURRENTLY das Sperren auf Zeilenebene, wodurch die Verfügbarkeit der Tabelle erheblich verbessert wird. Es folgt der Befehl:

  ```
  REINDEX INDEX ix_name CONCURRENTLY;
  ```

  CONCURRENTLY ist zwar weniger störend, kann aber bei stark frequentierten Tabellen langsamer sein. Erwägen Sie, den Index nach Möglichkeit in Zeiten mit geringem Datenverkehr zu erstellen. Weitere Informationen finden Sie unter REINDEX in der PostgreSQL-Dokumentation.
- Verwendung der Option INDEX_CLEANUP FALSE — Wenn die Indizes umfangreich sind und schätzungsweise viel Zeit bis zur Fertigstellung in Anspruch nehmen, können Sie die automatische Bereinigung aufheben, indem Sie einen manuellen Befehl mit dem Befehl VACUUM FREEZE ausführen und dabei Indizes ausschließen. Diese Funktionalität ist in PostgreSQL Version 12 und späteren Versionen verfügbar.

  Durch das Umgehen von Indizes können Sie den Vakuumprozess des inkonsistenten Indexes überspringen und das Wraparound-Problem verringern. Das zugrundeliegende Problem mit ungültigen Seiten wird dadurch jedoch nicht gelöst. Um das Problem mit der ungültigen Seite vollständig zu beheben und zu lösen, müssen Sie den Index dennoch neu erstellen.

Außergewöhnlich hohe Transaktionsrate

In PostgreSQL können hohe Transaktionsraten die Leistung von Autovacuum erheblich beeinträchtigen, was zu einer langsameren Bereinigung toter Tupel und einem erhöhten Risiko eines Transaktions-ID-Wraparounds führt. Sie können die Transaktionsrate überwachen, indem Sie den Unterschied `max(age(datfrozenxid))` zwischen zwei Zeiträumen messen, typischerweise pro Sekunde. Darüber hinaus können Sie die folgenden Zählermetriken von RDS Performance Insights verwenden, um die Transaktionsrate (die Summe aus xact_commit und xact_rollback) zu messen, die die Gesamtzahl der Transaktionen darstellt.

Zähler	Typ	Einheit	Metrik
xact_commit	Transaktionen	Commits pro Sekunde	db.Transactions.xact_commit
xact_rollback	Transaktionen	Rollbacks pro Sekunde	db.Transactions.xact_rollback

Ein schneller Anstieg deutet auf eine hohe Transaktionslast hin, die Autovacuum überfordern und zu Blähungen, Sperrkonflikten und potenziellen Leistungsproblemen führen kann. Dies kann sich auf verschiedene Weise negativ auf den Autovakuumprozess auswirken:

- Tabellenaktivität: Bei der spezifischen Tabelle, die gelöscht wird, kann es zu einem hohen Transaktionsvolumen kommen, was zu Verzögerungen führen kann.
- Systemressourcen Das gesamte System ist möglicherweise überlastet, sodass es für Autovacuum schwierig ist, auf die Ressourcen zuzugreifen, die für einen effizienten Betrieb erforderlich sind.

Ziehen Sie die folgenden Strategien in Betracht, damit das Autovakuumgerät effektiver arbeiten und seine Aufgaben erfüllen kann:

1. Reduzieren Sie nach Möglichkeit die Transaktionsrate. Erwägen Sie, ähnliche Transaktionen nach Möglichkeit zu stapeln oder zu gruppieren.
2. Nehmen Sie häufig aktualisierte Tabellen ins Visier und verwenden Sie die manuelle `VACUUM FREEZE` Bedienung jede Nacht, Woche oder alle zwei Wochen außerhalb der Spitzenzeiten.
3. Erwägen Sie, Ihre Instance-Klasse zu skalieren, um mehr Systemressourcen für das hohe Transaktionsvolumen und die automatische Bereinigung bereitzustellen.

In diesem Abschnitt werden Faktoren erörtert, die häufig zu einer langsameren Vakuumleistung beitragen, und es wird erläutert, wie diese Probleme behoben werden können.

Themen

- Saugen Sie große Indizes
- Zu viele Tabellen oder Datenbanken zum Löschen
- Es läuft ein aggressives Vakuum (um eine Rundum-Ummantelung zu verhindern)

Saugen Sie große Indizes

VACUUM durchläuft aufeinanderfolgende Phasen: Initialisierung, Heap-Scanning, Index- und Heap-Bereinigung, Indexbereinigung, Heap-Kürzung und abschließende Bereinigung. Während des Heap-Scans werden Seiten gelöscht, defragmentiert und eingefroren. Nach Abschluss des Heap-Scans bereinigt VACUUM die Indizes, gibt leere Seiten an das Betriebssystem zurück und führt letzte Bereinigungsaufgaben wie das Löschen der freien Speicherzuweisung und das Aktualisieren von Statistiken durch.

Das Löschen von Indizes kann mehrere Durchläufe erfordern, wenn `maintenance_work_mem` (oder) nicht ausreicht, um den Index zu verarbeiten. `autovacuum_work_mem` In PostgreSQL 16 und früher erzwang ein Speicherlimit von 1 GB für das Speichern toter Tupel IDs häufig mehrere Durchläufe großer Indizes. PostgreSQL 17 führt ein`TidStore`, das Speicher dynamisch zuweist, anstatt ein Array mit einer einzigen Zuweisung zu verwenden. Dadurch wird die Beschränkung auf 1 GB aufgehoben, der Arbeitsspeicher wird effizienter genutzt und der Bedarf an mehreren Indexscans pro Index reduziert.

Große Indizes können in PostgreSQL 17 immer noch mehrere Durchläufe erfordern, wenn der verfügbare Speicher nicht die gesamte Indexverarbeitung auf einmal aufnehmen kann. In der Regel enthalten größere Indizes mehr tote Tupel, für die mehrere Durchläufe erforderlich sind.

Erkennung langsamer Vakuumvorgänge

Die `postgres_get_av_diag()` Funktion kann erkennen, wenn der Vakuumbetrieb aufgrund unzureichenden Speichers langsam abläuft. Weitere Informationen zu dieser Funktion finden Sie unter.

Die `postgres_get_av_diag()` Funktion gibt die folgenden Hinweise aus, wenn der verfügbare Speicher nicht ausreicht, um die Indexbereinigung in einem einzigen Durchgang abzuschließen.

`rds_tools`1.8

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound and
 it might be slow.
```

```
NOTICE: The current setting of autovacuum_work_mem is "XXX" and might not be
 sufficient. Consider increasing the setting, and if necessary, scaling up the Amazon
 RDS instance class for more memory.
        Additionally, review the possibility of manual vacuum with exclusion of indexes
 using (VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) table_name;).
```

rds_tools1,9

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound and
 it might be slow.
```

```
NOTICE: The current setting of autovacuum_work_mem is XX might not be sufficient.
 Consider increasing the setting to XXX, and if necessary, scaling up the RDS instance
 class for more
       memory. The suggested value is an estimate based on the current number of dead
 tuples for the table being vacuumed, which might not fully reflect the latest state.
 Additionally, review the possibility of manual
       vacuum with exclusion of indexes using (VACUUM (INDEX_CLEANUP FALSE, VERBOSE
 TRUE) table_name;). For more information, see
       Working with PostgreSQL autovacuum in the Amazon Amazon RDS User Guide
       .
```

> **Note**
>
> Die `postgres_get_av_diag()` Funktion stützt sich auf die Schätzung der Speichermenge, die `pg_stat_all_tables.n_dead_tup` für das Absaugen von Indizes benötigt wird.

Wenn die `postgres_get_av_diag()` Funktion einen langsamen Vakuumvorgang identifiziert, für den mehrere Indexscans erforderlich sind, weil er nicht ausreicht`autovacuum_work_mem`, wird die folgende Meldung generiert:

```
NOTICE: Your vacuum is performing multiple index scans due to insufficient
 autovacuum_work_mem:XXX for index vacuuming.
       For more information, see Working with PostgreSQL autovacuum in the Amazon
 Amazon RDS User Guide.
```

Anleitung

Sie können die folgenden Problemumgehungen manuell anwenden, `VACUUM FREEZE` um das Einfrieren der Tabelle zu beschleunigen.

Erhöhen Sie den Speicher für das Staubsaugen

Wie von der `postgres_get_av_diag()` Funktion vorgeschlagen, ist es ratsam, den `autovacuum_work_mem` Parameter zu erhöhen, um möglichen Speicherbeschränkungen auf

Instanzebene zu begegnen. Obwohl es `autovacuum_work_mem` sich um einen dynamischen Parameter handelt, ist es wichtig zu beachten, dass der Autovacuum-Daemon seine Worker neu starten muss, damit die neue Speichereinstellung wirksam wird. Um dies zu erreichen:

1. Vergewissern Sie sich, dass die neue Einstellung aktiviert ist.
2. Beenden Sie die Prozesse, die derzeit Autovacuum ausführen.

Dieser Ansatz stellt sicher, dass die angepasste Speicherzuweisung auf neue Autovacuum-Operationen angewendet wird.

Um unmittelbarere Ergebnisse zu erzielen, sollten Sie erwägen, einen `VACUUM FREEZE` Vorgang mit einer erhöhten `maintenance_work_mem` Einstellung innerhalb Ihrer Sitzung manuell durchzuführen:

```
SET maintenance_work_mem TO '1GB';
VACUUM FREEZE VERBOSE table_name;
```

Wenn Sie Amazon RDS verwenden und feststellen, dass Sie zusätzlichen Speicher benötigen, um höhere Werte für `maintenance_work_mem` oder zu unterstützen`autovacuum_work_mem`, sollten Sie ein Upgrade auf eine Instance-Klasse mit mehr Speicher in Betracht ziehen. Dadurch können die erforderlichen Ressourcen bereitgestellt werden, um sowohl die manuellen als auch die automatischen Vakuumvorgänge zu verbessern, was zu einer insgesamt verbesserten Vakuum- und Datenbankleistung führt.

Deaktivieren Sie INDEX_CLEANUP

Manuell `VACUUM` in PostgreSQL Version 12 und höher ermöglicht das Überspringen der Indexbereinigungsphase, während das Notfall-Autovacuum in PostgreSQL Version 14 und höher dies automatisch auf der Grundlage des Parameters erledigt. `vacuum_failsafe_age`

> **Warning**
>
> Das Überspringen der Indexbereinigung kann zu einer Aufblähung des Index führen und sich negativ auf die Abfrageleistung auswirken. Um dem entgegenzuwirken, sollten Sie erwägen, die betroffenen Indizes während eines Wartungsfensters neu zu indizieren oder zu löschen.

Weitere Hinweise zum Umgang mit großen Indizes finden Sie in der Dokumentation zu. Verwalten der automatischen Bereinigung mit großen Indizes

Paralleles Absaugen von Indizes

Ab PostgreSQL 13 können Indizes standardmäßig manuell parallel gesaugt und bereinigt werden`VACUUM`, wobei jedem Index ein Vacuum Worker-Prozess zugewiesen wird. Damit PostgreSQL jedoch feststellen kann, ob eine Vakuumoperation für die parallel Ausführung in Frage kommt, müssen bestimmte Kriterien erfüllt sein:

- Es müssen mindestens zwei Indizes vorhanden sein.
- Der `max_parallel_maintenance_workers` Parameter sollte auf mindestens 2 gesetzt sein.
- Die Indexgröße muss den `min_parallel_index_scan_size` Grenzwert überschreiten, der standardmäßig 512 KB beträgt.

Sie können die `max_parallel_maintenance_workers` Einstellung auf der Grundlage der Anzahl der auf Ihrer Amazon RDS-Instance CPUs verfügbaren v und der Anzahl der Indizes in der Tabelle anpassen, um die Bearbeitungszeit für das Vacuuming zu optimieren.

Weitere Informationen finden Sie unter Parallele Vakuumierung in Amazon RDS for PostgreSQL und Amazon Aurora PostgreSQL.

Zu viele Tabellen oder Datenbanken zum Löschen

Wie in der Dokumentation The Autovacuum-Daemon von PostgreSQL erwähnt, arbeitet der Autovacuum-Daemon mit mehreren Prozessen. Dazu gehört auch ein persistenter Autovacuum-Launcher, der für das Starten von Autovacuum-Worker-Prozessen für jede Datenbank innerhalb des Systems verantwortlich ist. Der Launcher plant, dass diese Worker ungefähr alle `autovacuum_naptime` Sekunden pro Datenbank initiiert werden.

Bei 'N' Datenbanken beginnt ungefähr alle [`autovacuum_naptime`/N Sekunden] ein neuer Worker. Die Gesamtzahl der gleichzeitigen Mitarbeiter ist jedoch durch die `autovacuum_max_workers` Einstellung begrenzt. Wenn die Anzahl der Datenbanken oder Tabellen, die gelöscht werden müssen, diesen Grenzwert überschreitet, wird die nächste Datenbank oder Tabelle verarbeitet, sobald ein Worker verfügbar ist.

Wenn viele große Tabellen oder Datenbanken gleichzeitig gelöscht werden müssen, kann es sein, dass alle verfügbaren Autovakuum-Mitarbeiter für einen längeren Zeitraum ausgelastet sind, was die Wartung anderer Tabellen und Datenbanken verzögert. In Umgebungen mit hohen Transaktionsraten kann dieser Engpass schnell eskalieren und möglicherweise zu umfassenden Vakuumproblemen innerhalb Ihrer Amazon RDS-Instance führen.

Wenn eine große Anzahl von Tabellen oder Datenbanken `postgres_get_av_diag()` erkannt wird, gibt es die folgende Empfehlung:

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound and
 it might be slow.
```

```
NOTICE: The current setting of autovacuum_max_workers:3 might not be sufficient.
 Consider increasing the setting and, if necessary, consider scaling up the Amazon RDS
 instance class for more workers.
```

Anleitung

Erhöhen Sie autovacuum_max_workers

Um das Absaugen zu beschleunigen, empfehlen wir, den Parameter so anzupassen, dass mehr Mitarbeiter gleichzeitig mit dem Autovakuieren arbeiten können. `autovacuum_max_workers` Wenn weiterhin Leistungsengpässe bestehen, sollten Sie erwägen, Ihre Amazon RDS-Instance auf eine Klasse mit mehr v zu skalierenCPUs, wodurch die Parallelverarbeitungsmöglichkeiten weiter verbessert werden können.

Es läuft ein aggressives Vakuum (um eine Rundum-Ummantelung zu verhindern)

Das Alter der Datenbank (MaximumUsedTransactionIDs) in PostgreSQL nimmt nur ab, wenn ein aggressives Vakuum (um Wraparound zu verhindern) erfolgreich abgeschlossen wird. Bis dieses Vakuum beendet ist, wird das Alter je nach Transaktionsrate weiter zunehmen.

Die `postgres_get_av_diag()` Funktion generiert Folgendes`NOTICE`, wenn sie ein aggressives Vakuum erkennt. Sie löst diesen Ausgang jedoch erst aus, wenn das Vakuum mindestens zwei Minuten lang aktiv war.

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound,
 monitor autovacuum performance.
```

Weitere Informationen zu aggressivem Vakuum finden Sie unter Wenn bereits ein aggressives Vakuum läuft.

Mit der folgenden Abfrage können Sie überprüfen, ob ein aggressives Vakuumieren im Gange ist:

```
SELECT
    a.xact_start AS start_time,
```

```
    v.datname "database",
    a.query,
    a.wait_event,
    v.pid,
    v.phase,
    v.relid::regclass,
    pg_size_pretty(pg_relation_size(v.relid)) AS heap_size,
    (
        SELECT
            string_agg(pg_size_pretty(pg_relation_size(i.indexrelid)) || ':' ||
 i.indexrelid::regclass || chr(10), ', ')
        FROM
            pg_index i
        WHERE
            i.indrelid = v.relid
    ) AS index_sizes,
    trunc(v.heap_blks_scanned * 100 / NULLIF(v.heap_blks_total, 0)) AS step1_scan_pct,
    v.index_vacuum_count || '/' || (
        SELECT
            count(*)
        FROM
            pg_index i
        WHERE
            i.indrelid = v.relid
    ) AS step2_vacuum_indexes,
    trunc(v.heap_blks_vacuumed * 100 / NULLIF(v.heap_blks_total, 0)) AS
 step3_vacuum_pct,
    age(CURRENT_TIMESTAMP, a.xact_start) AS total_time_spent_sofar
FROM
    pg_stat_activity a
    INNER JOIN pg_stat_progress_vacuum v ON v.pid = a.pid;
```

Sie können anhand der Abfragespalte in der Ausgabe feststellen, ob es sich um ein aggressives Vakuum handelt (um einen Rundum-Vorgang zu verhindern). Der Ausdruck „zur Verhinderung von Rundumbildung“ weist darauf hin, dass es sich um ein aggressives Vakuum handelt.

```
query                 | autovacuum: VACUUM public.t3 (to prevent wraparound)
```

Nehmen wir zum Beispiel an, Sie haben einen Blocker im Transaktionsalter von 1 Milliarde und eine Tabelle, die ein aggressives Vakuum erfordert, um Wraparound im gleichen Transaktionsalter zu verhindern. Darüber hinaus gibt es einen weiteren Blocker im Transaktionsalter von 750 Millionen. Nach der Aufhebung des Blockers im Transaktionsalter von 1 Milliarde wird das Transaktionsalter

nicht sofort auf 750 Millionen sinken. Es wird so lange hoch bleiben, bis der Tisch, der das aggressive Vakuum benötigt, oder eine Transaktion mit einem Alter von über 750 Millionen abgeschlossen ist. Während dieses Zeitraums wird das Transaktionsalter Ihres PostgreSQL-Clusters weiter steigen. Sobald der Vakuumvorgang abgeschlossen ist, wird das Transaktionsalter auf 750 Millionen sinken, aber es wird wieder steigen, bis das weitere Vakuumieren abgeschlossen ist. Dieser Zyklus wird so lange fortgesetzt, wie diese Bedingungen bestehen, bis das Transaktionsalter irgendwann auf das für Ihre Amazon RDS-Instance konfigurierte Niveau fällt, das von `autovacuum_freeze_max_age` angegeben ist.

Die `postgres_get_av_diag()` Funktion stellt die folgenden NOTICE-Meldungen bereit:

Wenn das Alter die Überwachungsschwelle noch nicht erreicht hat

Die Überwachungsschwelle `postgres_get_av_diag()` zur Identifizierung von Blockern liegt standardmäßig bei 500 Millionen Transaktionen. Wenn der folgende HINWEIS `postgres_get_av_diag()` generiert wird, bedeutet dies, dass das Transaktionsalter diesen Schwellenwert noch nicht erreicht hat.

```
NOTICE: postgres_get_av_diag() checks for blockers that prevent aggressive vacuums
 only, it does so only after exceeding dvb_threshold which is 500,000,000 and age of
 this PostgreSQL cluster is currently at 2.
```

Es wurde keine Verbindung zu der Datenbank mit dem Alter der ältesten Transaktions-ID hergestellt

Die `postgres_get_av_diag()` Funktion liefert die genaueste Ausgabe, wenn sie mit der Datenbank mit der ältesten Transaktions-ID verbunden ist. Die Datenbank mit dem ältesten von gemeldeten Transaktions-ID-Alter unterscheidet `postgres_get_av_diag()` sich in Ihrem Fall von „my_database“. Wenn Sie nicht mit der richtigen Datenbank verbunden sind, wird der folgende HINWEIS generiert:

```
NOTICE: You are not connected to the database with the age of oldest transaction
 ID. Connect to my_database database and run postgres_get_av_diag() for accurate
 reporting.
```

Aus den folgenden Gründen ist es wichtig, eine Verbindung zu der Datenbank mit dem ältesten Transaktionsalter herzustellen:

- Identifizierung temporärer Tabellenblocker: Da die Metadaten für temporäre Tabellen für jede Datenbank spezifisch sind, befinden sie sich normalerweise in der Datenbank, in der

sie erstellt wurden. Wenn jedoch eine temporäre Tabelle der häufigste Blocker ist und sich in der Datenbank mit der ältesten Transaktion befindet, könnte dies irreführend sein. Durch die Verbindung mit der richtigen Datenbank wird die genaue Identifizierung des Blockers für temporäre Tabellen gewährleistet.

- Diagnose von Slow-Vacuum: Die Index-Metadaten und die Informationen zur Anzahl der Tabellen sind datenbankspezifisch und für die Diagnose von Problemen mit Slow Vacuum erforderlich.

Die Datenbank mit der ältesten Transaktion nach Alter befindet sich in einer rdsadmin- oder template0-Datenbank

In bestimmten Fällen können die `template0` Datenbanken `rdsadmin` oder als die Datenbank mit der ältesten Transaktions-ID identifiziert werden. In diesem Fall `postgres_get_av_diag()` wird der folgende HINWEIS ausgegeben:

```
NOTICE: The database with the age of oldest transaction ID is rdsadmin or template0,
 reach out to support if the reported blocker is in rdsadmin or template0.
```

Stellen Sie sicher, dass der aufgelistete Blocker nicht aus einer dieser beiden Datenbanken stammt. Wenn gemeldet wird, dass der Blocker in einer `rdsadmin` oder vorhanden ist`template0`, wenden Sie sich an den Support, da Benutzer nicht auf diese Datenbanken zugreifen können und ein Eingreifen erforderlich ist.

Es ist sehr unwahrscheinlich, dass die `template0` Datenbank `rdsadmin` oder einen Top-Blocker enthält.

Wenn bereits ein aggressives Vakuum läuft

Die `postgres_get_av_diag()` Funktion ist so konzipiert, dass sie meldet, wenn ein aggressiver Vakuumprozess läuft, löst diese Ausgabe jedoch erst aus, wenn das Vakuum mindestens 1 Minute lang aktiv war. Diese absichtliche Verzögerung trägt dazu bei, die Wahrscheinlichkeit falsch positiver Ergebnisse zu verringern. Durch das Warten stellt die Funktion sicher, dass nur effektive, signifikante Vakuumwerte gemeldet werden, was zu einer genaueren und zuverlässigeren Überwachung der Vakuumaktivität führt.

Die `postgres_get_av_diag()` Funktion generiert den folgenden HINWEIS, wenn sie feststellt, dass ein oder mehrere aggressive Absaugungen im Gange sind.

```
NOTICE: Your database is currently running aggressive vacuum to prevent wraparound,
 monitor autovacuum performance.
```

Wie in der MITTEILUNG angegeben, sollten Sie die Leistung des Staubsaugers weiterhin überwachen. Weitere Informationen über aggressives Vakuum finden Sie unter Es läuft ein aggressives Vakuum (um eine Rundum-Ummantelung zu verhindern)

Wenn die automatische Vakuumierung ausgeschaltet ist

Die `postgres_get_av_diag()` Funktion generiert den folgenden HINWEIS, wenn Autovacuum auf Ihrer Datenbankinstanz deaktiviert ist:

```
NOTICE: Autovacuum is OFF, we strongly recommend to enable it, no restart is
 necessary.
```

Autovacuum ist eine wichtige Funktion Ihrer RDS for PostgreSQL Aurora PostgreSQL-DB-Instance Datenbankbetrieb gewährleistet. Es entfernt automatisch alte Zeilenversionen, gibt Speicherplatz zurück und verhindert ein Aufblähen von Tabellen und trägt so dazu bei, dass Tabellen und Indizes effizient bleiben und eine optimale Leistung erzielt wird. Darüber hinaus schützt es vor Transaktions-ID-Wraparound, wodurch Transaktionen auf Ihrer Amazon RDS-Instance gestoppt werden können. Die Deaktivierung von Autovacuum kann zu langfristigen Einbußen der Datenbankleistung und -stabilität führen. Wir empfehlen Ihnen, es ständig eingeschaltet zu lassen. Weitere Informationen finden Sie unter Grundlegendes zum Autovacuum in RDS für PostgreSQL PostgreSQL-Umgebungen.

> **Note**
>
> Durch das Ausschalten der automatischen Vakuumierung werden aggressive Sauger nicht gestoppt. Diese treten weiterhin auf, sobald Ihre Tabellen den Schwellenwert erreichen. `autovacuum_freeze_max_age`

Die Anzahl der verbleibenden Transaktionen ist äußerst gering

Die `postgres_get_av_diag()` Funktion generiert den folgenden HINWEIS, wenn ein Rundumvakuum unmittelbar bevorsteht. Dieser HINWEIS wird ausgegeben, wenn Ihre Amazon RDS-Instance 100 Millionen Transaktionen davon entfernt ist, neue Transaktionen potenziell abzulehnen.

```
WARNING: Number of transactions remaining is critically low, resolve issues with
 autovacuum or perform manual VACUUM FREEZE before your instance stops accepting
 transactions.
```

Ihr sofortiges Handeln ist erforderlich, um Ausfallzeiten der Datenbank zu vermeiden. Sie sollten Ihre Löschvorgänge genau überwachen und erwägen, manuell einen `VACUUM FREEZE` Vorgang in der betroffenen Datenbank zu initiieren, um Transaktionsfehler zu vermeiden.

Arbeiten mit Protokollierungsmechanismen, die von RDS for PostgreSQL unterstützt werden

Es gibt mehrere Parameter, Erweiterungen und andere konfigurierbare Elemente, die Sie festlegen können, um Aktivitäten zu protokollieren, die auf Ihrer PostgreSQL-DB-Instance auftreten. Diese umfassen u. a. folgende:

- Der Parameter `log_statement` kann verwendet werden, um in Ihrer PostgreSQL-Datenbank Benutzeraktivitäten zu protokollieren. Weitere Informationen über RDS for PostgreSQL-Protokollierung und zur Überwachung der Protokolle finden Sie unter Datenbank-Protokolldateien von RDS für PostgreSQL.
- Der Parameter `rds.force_admin_logging_level` protokolliert Aktionen durch den internen Amazon-RDS-Benutzer (rdsadmin) in den Datenbanken in der DB-Instance. Die Ausgabe wird in das PostgreSQL-Fehlerprotokoll geschrieben. Zulässige Werte sind `disabled`, `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, Protokoll, `fatal` und `panic`. Der Standardwert ist `disabled`.
- Der `rds.force_autovacuum_logging_level`-Parameter kann festgelegt werden, um verschiedene Selbstbereinigungsoperationen im PostgreSQL-Fehlerprotokoll zu erfassen. Weitere Informationen finden Sie unter Protokollieren von Selbstbereinigung- und Bereinigungsaktivitäten.
- Die PostgreSQL Audit (pgAudit)-Erweiterung kann installiert und konfiguriert werden, um Aktivitäten auf Sitzungsebene oder auf Objektebene zu erfassen. Weitere Informationen finden Sie unter Wird verwendet pgAudit , um Datenbankaktivitäten zu protokollieren.
- Die `log_fdw`-Erweiterung ermöglicht es Ihnen, über SQL auf das Datenbank-Engine-Protokoll zuzugreifen. Weitere Informationen finden Sie unter Verwenden der Erweiterung log_fdw für den Zugriff auf das DB-Protokoll mit SQL.
- Die `pg_stat_statements`-Bibliothek ist als Standardwert für den `shared_preload_libraries`-Parameter in RDS for PostgreSQL Version 10 und höher angegeben. Mit dieser Bibliothek können Sie laufende Abfragen analysieren. Stellen Sie sicher, dass `pg_stat_statements` in Ihrer DB-Parametergruppe festgelegt ist. Weitere Informationen zur Überwachung Ihrer DB-Instance von RDS for PostgreSQL unter Verwendung

der Informationen, die diese Bibliothek bereitstellt, finden Sie unter SQL-Statistiken für RDS PostgreSQL.

- Der `log_hostname`-Parameter erfasst den Hostnamen jeder Client-Verbindung im Protokoll. Für RDS für PostgreSQL Version 12 und höhere Versionen ist dieser Parameter standardmäßig auf `off` festgelegt. Wenn Sie ihn aktivieren, achten Sie darauf, die Verbindungszeiten der Sitzung zu überwachen. Wenn diese Option aktiviert ist, verwendet der Dienst die Reverse-Lookup-Anfrage des Domain Name Systems (DNS), um den Hostnamen des Clients, der die Verbindung herstellt, abzurufen und dem PostgreSQL-Protokoll hinzuzufügen. Dies hat spürbare Auswirkungen während der Sitzungsverbindung. Wir empfehlen, diesen Parameter nur für Fehlerbehebungszwecke zu aktivieren.

Im Allgemeinen ist der Zweck der Protokollierung, dass der DBA die Leistung überwachen, optimieren und Fehler beheben kann. Viele der Protokolle werden automatisch auf Amazon CloudWatch oder Performance Insights hochgeladen. Hier werden sie sortiert und gruppiert, um vollständige Metriken für die DB-Instance bereitzustellen. Weitere Informationen zur Amazon-RDS-Überwachung und -Metriken finden Sie unter Metriken in einer RDS Amazon-Instance überwachen.

Verwalten temporärer Dateien mit PostgreSQL

In PostgreSQL kann eine komplexe Abfrage mehrere Sortier- oder Hashoperationen gleichzeitig ausführen, wobei jede Operation Instanzspeicher verwendet, um Ergebnisse bis zu dem `work_mem`im Parameter angegebenen Wert zu speichern. Wenn der Instance-Speicher nicht ausreicht, werden temporäre Dateien erstellt, um die Ergebnisse zu speichern. Diese werden auf die Festplatte geschrieben, um die Abfrageausführung abzuschließen. Später werden diese Dateien automatisch entfernt, nachdem die Abfrage abgeschlossen ist. In RDS für PostgreSQL werden diese Dateien in Amazon EBS auf dem Datenvolume gespeichert. Weitere Informationen finden Sie unter Amazon-RDS-DB-Instance Speicher. Sie können die in veröffentlichte `FreeStorageSpace` Metrik überwachen CloudWatch , um sicherzustellen, dass Ihre DB-Instance über ausreichend freien Speicherplatz verfügt. Weitere Informationen finden Sie unter `FreeStorageSpace` .

Wir empfehlen die Verwendung von Amazon RDS Optimized Read für Workloads mit mehreren gleichzeitigen Abfragen, die die Nutzung temporärer Dateien erhöhen. Diese verwenden lokalen, auf Non-Volatile Memory Express (NVMe) basierenden Solid-State-Drive-Speicher (SSD) auf Blockebene, um die temporären Dateien zu platzieren. Weitere Informationen finden Sie unter Verbesserung der Abfrageleistung für RDS für PostgreSQL mit Amazon-RDS-optimierten Lesevorgängen.

Sie können die folgenden Parameter und Funktionen verwenden, um die temporären Dateien in Ihrer Instance zu verwalten.

- **temp_file_limit** – Dieser Parameter bricht jede Abfrage ab, die die Größe von temp_files in KB überschreitet. Dieses Limit verhindert, dass Abfragen endlos ausgeführt werden und Speicherplatz mit temporären Dateien belegen. Sie können den Wert anhand der Ergebnisse des Parameters `log_temp_files` schätzen. Es hat sich bewährt, das Workload-Verhalten zu untersuchen und das Limit der Schätzung entsprechend festzulegen. Das folgende Beispiel zeigt, wie eine Abfrage abgebrochen wird, wenn sie das Limit überschreitet.

```
postgres=>select * from pgbench_accounts, pg_class, big_table;
```

```
ERROR: temporary file size exceeds temp_file_limit (64kB)
```

- **log_temp_files** – Dieser Parameter sendet Nachrichten an die Datei postgresql.log, wenn die temporären Dateien einer Sitzung entfernt werden. Dieser Parameter erstellt Protokolle, nachdem eine Abfrage erfolgreich abgeschlossen wurde. Daher ist er bei der Fehlerbehebung aktiver, lang andauernder Abfragen möglicherweise nicht hilfreich.

  Das folgende Beispiel zeigt, dass nach erfolgreichem Abschluss der Abfrage die Einträge in der Datei postgresql.log protokolliert werden, während die temporären Dateien bereinigt werden.

```
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.5", size 140353536
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.4", size 180428800
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
```

- **pg_ls_tmpdir** – Diese Funktion, die von RDS für PostgreSQL 13 und höher verfügbar ist, bietet Einblick in die aktuelle Nutzung temporärer Dateien. Die abgeschlossene Abfrage erscheint nicht

in den Ergebnissen der Funktion. Im folgenden Beispiel können Sie sich die Ergebnisse dieser Funktion ansehen.

```
postgres=>select * from pg_ls_tmpdir();
```

```
      name       |    size    |      modification
-----------------+------------+------------------------
 pgsql_tmp8355.1 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.0 | 1072250880 | 2023-02-06 22:54:43+00
 pgsql_tmp8327.0 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.1 |  703168512 | 2023-02-06 22:54:56+00
 pgsql_tmp8355.0 | 1072250880 | 2023-02-06 22:54:00+00
 pgsql_tmp8328.1 |  835031040 | 2023-02-06 22:54:56+00
 pgsql_tmp8328.0 | 1072250880 | 2023-02-06 22:54:40+00
(7 rows)
```

```
postgres=>select query from pg_stat_activity where pid = 8355;

query
-------------------------------------------------------------------------------------------
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid
(1 row)
```

Der Dateiname enthält die Verarbeitungs-ID (PID) der Sitzung, die die temporäre Datei generiert hat. Eine komplexere Abfrage, wie im folgenden Beispiel, führt eine Summenberechnung der temporären Dateien für jede PID durch.

```
postgres=>select replace(left(name, strpos(name, '.')-1),'pgsql_tmp','') as pid,
 count(*), sum(size) from pg_ls_tmpdir() group by pid;
```

```
 pid  | count |   sum
------+-------+------------
 8355 |     2 | 2144501760
 8351 |     2 | 2090770432
 8327 |     1 | 1072250880
 8328 |     2 | 2144501760
```

```
(4 rows)
```

- **pg_stat_statements** – Wenn Sie den Parameter pg_stat_statements aktivieren, können Sie die durchschnittliche Nutzung temporärer Dateien pro Aufruf einsehen. Sie können die query_id der Abfrage identifizieren und verwenden, um die Nutzung temporärer Dateien zu untersuchen, wie im folgenden Beispiel gezeigt.

```
postgres=>select queryid from pg_stat_statements where query like 'select a.aid from
 pgbench%';
```

```
       queryid
----------------------
 -7170349228837045701
(1 row)
```

```
postgres=>select queryid, substr(query,1,25), calls, temp_blks_read/calls
 temp_blks_read_per_call, temp_blks_written/calls temp_blks_written_per_call from
 pg_stat_statements where queryid = -7170349228837045701;
```

```
       queryid        |          substr           | calls | temp_blks_read_per_call |
 temp_blks_written_per_call
----------------------+---------------------------+-------+-------------------------
+---------------------------
 -7170349228837045701 | select a.aid from pgbench |    50 |                  239226 |
                     388678
(1 row)
```

- **Performance Insights** – Im Performance-Insights-Dashboard können Sie die Nutzung temporärer Dateien einsehen, indem Sie die Metriken temp_bytes und temp_files aktivieren. Anschließend können Sie den Durchschnitt dieser beiden Metriken sehen und feststellen, wie sie dem Abfrage-Workload entsprechen. In der Ansicht in Performance Insights werden nicht speziell die Abfragen angezeigt, die die temporären Dateien generieren. Wenn Sie jedoch Performance Insights mit der für `pg_ls_tmpdir` angezeigten Abfrage kombinieren, können Sie Fehler in Ihrem Abfrage-Workload beheben, analysieren und die Änderungen ermitteln.

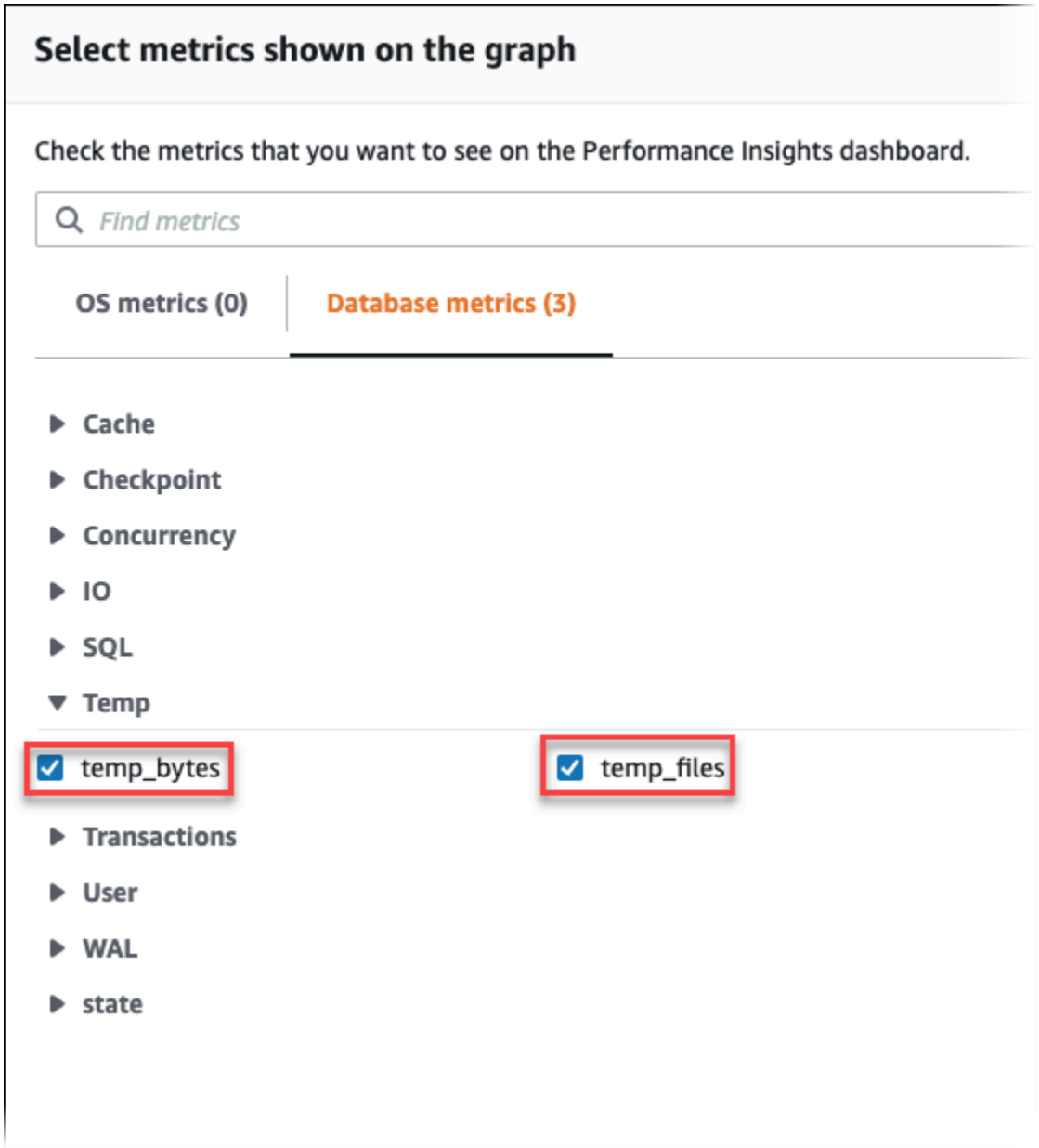
Weitere Informationen zur Analyse von Metriken und Abfragen mit Performance Insights finden Sie unterAnalyse der Metriken mit dem Performance Insights-Dashboard.

Ein Beispiel für die Anzeige der Nutzung temporärer Dateien mit Performance Insights finden Sie unter Verwendung temporärer Dateien mit Performance Insights anzeigen

Verwendung temporärer Dateien mit Performance Insights anzeigen

Sie können Performance Insights verwenden, um die Nutzung temporärer Dateien anzuzeigen, indem Sie die Metriken temp_bytes und temp_files aktivieren. In der Ansicht in Performance Insights werden die spezifischen Abfragen, die temporäre Dateien generieren, nicht angezeigt. Wenn Sie jedoch Performance Insights mit der für angezeigten Abfrage kombinierenpg_ls_tmpdir, können Sie Fehler beheben, analysieren und die Änderungen an Ihrer Abfrage-Workload ermitteln.

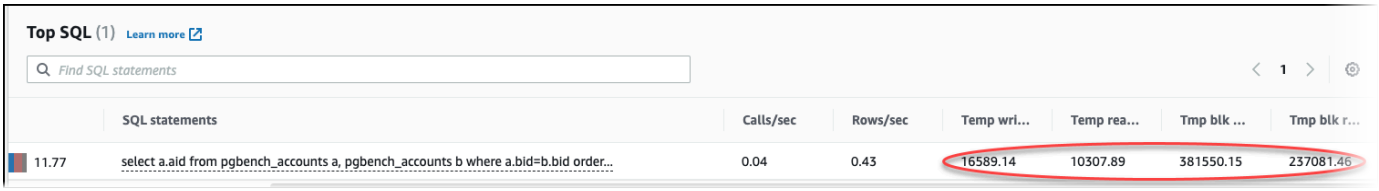
1. Wählen Sie im Performance-Insights-Dashboard Metriken verwalten aus.
2. Wählen Sie Datenbankmetriken und die Metriken temp_bytes und temp_files aus, wie im folgenden Screenshot gezeigt.



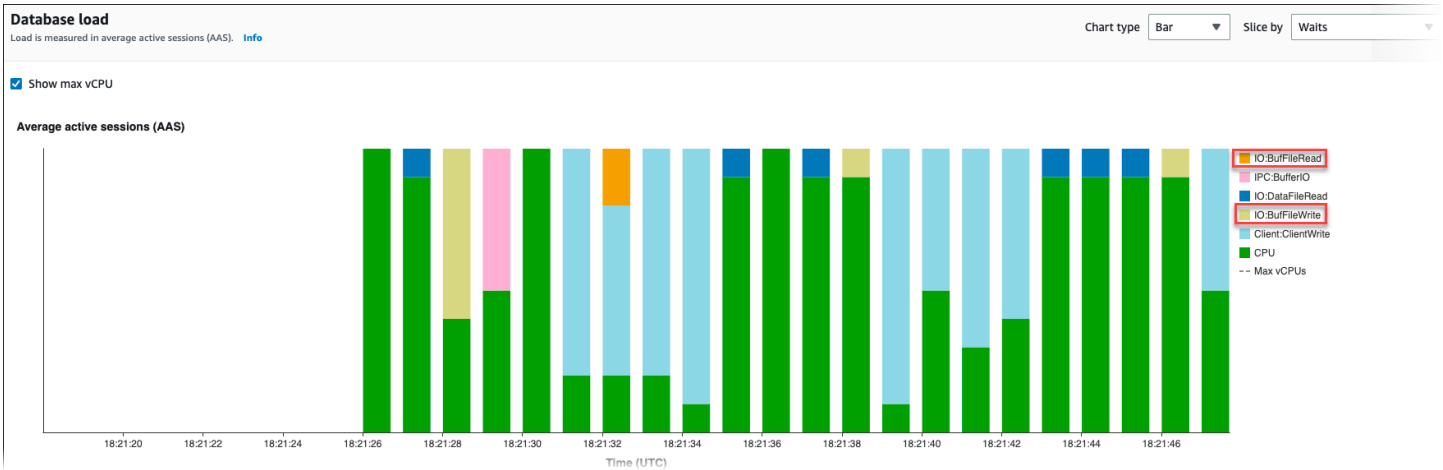
3. Wählen Sie auf der oberen SQL Registerkarte das Symbol Einstellungen aus.
4. Aktivieren Sie im Fenster „Einstellungen“ die folgenden Statistiken, sodass sie auf der SQL Registerkarte „Oben“ angezeigt werden, und wählen Sie „Weiter“.

   - Temporäre Schreibvorgänge pro Sekunde
   - Temporäre Lesevorgänge pro Sekunde
   - Temporäre Massenschreibvorgänge/Aufruf
   - Temporäre Massenlesevorgänge/Aufruf

5. Die temporäre Datei wird aufgegliedert, wenn sie mit der für `pg_ls_tmpdir` gezeigten Abfrage kombiniert wird, wie im folgenden Beispiel gezeigt.



Die `IO:BufFileRead`- und `IO:BufFileWrite`-Ereignisse treten auf, wenn die häufigsten Abfragen in Ihrem Workload häufig temporäre Dateien erstellen. Sie können Performance Insights verwenden, um die häufigsten Abfragen zu identifizieren, auf die gewartet wird, `IO:BufFileRead` und `IO:BufFileWrite` indem Sie die durchschnittliche aktive Sitzung (AAS) in den SQL Abschnitten Database Load und Top überprüfen.



Weitere Informationen zur Analyse von Top-Abfragen und Last nach Warteereignis mit Performance Insights finden Sie unter Überblick über den SQL Tab „Top“. Sie sollten die Abfragen identifizieren und optimieren, die zu einer erhöhten Nutzung temporärer Dateien und damit verbundenen Warteereignissen führen. Weitere Informationen zu diesen Warteereignissen und deren Behebung finden Sie unter IO: BufFileRead und IO: BufFileWrite .

> **Note**
>
> Der Parameter `work_mem` steuert, wann der Sortiervorgang nicht mehr genügend Speicherplatz hat und die Ergebnisse in temporäre Dateien geschrieben werden. Wir empfehlen, die Einstellung dieses Parameters nicht höher als auf den Standardwert festzulegen, da dadurch jede Datenbanksitzung mehr Speicher belegen würde. Außerdem kann eine einzelne Sitzung, die komplexe Verknüpfungen und Sortierungen durchführt, parallele Operationen ausführen, bei denen jeder Vorgang Speicherplatz belegt.

Wenn Sie einen umfangreichen Bericht mit mehreren Verknüpfungen und Sortierungen haben, empfiehlt es sich, diesen Parameter mit dem Befehl `SET work_mem` auf Sitzungsebene festzulegen. Dann wird die Änderung nur auf die aktuelle Sitzung angewendet und der Wert nicht global geändert.

Verwenden von pgBadger für die Protokollanalyse mit PostgreSQL

Sie können einen Protokollanalysierer wie pgBadger verwenden, um PostgreSQL-Protokolle zu analysieren. Die pgBadger-Dokumentation besagt, dass das Muster %l (Die Protokollzeile für die Sitzung oder den Prozess) ein Teil des Präfixes sein muss. Wenn Sie jedoch das aktuelle RDS `log_line_prefix` als Parameter für pgBadger angeben, wird dennoch ein Bericht erstellt.

Beispielsweise formatiert der folgende Befehl eine Amazon-RDS-for-PostgreSQL-Protokolldatei mit dem Datum 2014-02-04 korrekt unter Verwendung von pgBadger.

```
./pgbadger -f stderr -p '%t:%r:%u@%d:[%p]:' postgresql.log.2014-02-04-00
```

PGSnapper Für die Überwachung von PostgreSQL verwenden

Sie können PGSnapper es als Unterstützung bei der regelmäßigen Erfassung leistungsbezogener Statistiken und Metriken von Amazon RDS for PostgreSQL verwenden. Weitere Informationen finden Sie unter Überwachen der Leistung von Amazon RDS for PostgreSQL mithilfe von. PGSnapper

Arbeiten mit Parametern auf der DB-Instance von RDS for PostgreSQL

In einigen Fällen können Sie eine DB-Instance für RDS for PostgreSQL erstellen, ohne eine benutzerdefinierte Parametergruppe anzugeben. In diesem Fall wird Ihre DB-Instance mit der Standardparametergruppe für die von Ihnen gewählte Version von PostgreSQL erstellt. Angenommen, Sie erstellen eine DB-Instance für RDS for PostgreSQL mit PostgreSQL 13.3. In diesem Fall wird die DB-Instance unter Verwendung der Werte in der Parametergruppe für PostgreSQL-13-Versionen, `default.postgres13`, erstellt.

Sie können auch eine eigene benutzerdefinierte DB-Parametergruppe erstellen. Sie müssen dies tun, wenn Sie die Einstellungen der Standardwerte für die DB-Instance von RDS for PostgreSQL ändern möchten. Um zu erfahren wie dies geht, vgl. Parametergruppen für Amazon RDS.

Sie können die Einstellungen auf der DB-Instance von RDS for PostgreSQL auf verschiedene Arten verfolgen. Sie können die AWS Management Console AWS CLI, oder die Amazon RDS-API

verwenden. Sie können die Werte auch aus der PostgreSQL `pg_settings`-Tabelle Ihrer Instance abfragen, wie im Folgenden dargestellt.

```
SELECT name, setting, boot_val, reset_val, unit
 FROM pg_settings
 ORDER BY name;
```

Weitere Informationen zu von dieser Abfrage zurückgegebenen Werten finden Sie unter `pg_settings` in der PostgreSQL-Dokumentation.

Seien Sie besonders vorsichtig, wenn Sie die Einstellungen für `max_connections` und `shared_buffers` auf Ihrer DB-Instance von RDS for PostgreSQL ändern. Nehmen Sie zum Beispiel an, dass Sie die Einstellungen für `max_connections` oder `shared_buffers` ändern und Werte verwenden, die für den tatsächlichen Workload zu hoch sind. In diesem Fall wird die DB-Instance von RDS for PostgreSQL nicht gestartet. In diesem Fall wird Ihnen in `postgres.log` ein Fehler ähnlich dem folgenden angezeigt:

```
2018-09-18 21:13:15 UTC::@:[8097]:FATAL:  could not map anonymous shared memory: Cannot
 allocate memory
2018-09-18 21:13:15 UTC::@:[8097]:HINT:  This error usually means that PostgreSQL's
 request for a shared memory segment
exceeded available memory or swap space. To reduce the request size (currently
 3514134274048 bytes), reduce
PostgreSQL's shared memory usage, perhaps by reducing shared_buffers or
 max_connections.
```

Sie können jedoch keine Werte der Einstellungen ändern, die in Standard-RDS-for-PostgreSQL-DB-Parametergruppen enthalten sind. Um die Einstellungen für Parameter zu ändern, erstellen Sie zuerst eine benutzerdefinierte DB-Parametergruppe. Ändern Sie dann die Einstellungen in dieser benutzerdefinierten Gruppe und wenden Sie dann die benutzerdefinierte Parametergruppe auf die DB-Instance von RDS for PostgreSQL an. Weitere Informationen hierzu finden Sie unter Parametergruppen für Amazon RDS.

In RDS für PostgreSQL gibt es zwei Arten von Parametern.

- Statische Parameter – Statische Parameter erfordern den Neustart der DB-Instance von RDS for PostgreSQL nach einer Änderung, damit der neue Wert angewendet wird.
- Dynamische Parameter – Dynamische Parameter erfordern keinen Neustart, nachdem ihre Einstellungen geändert wurden.

> **Note**
>
> Wenn Ihre DB-Instance von RDS for PostgreSQL Ihre eigene benutzerdefinierte DB-Parametergruppe verwendet, können Sie die Werte dynamischer Parameter auf der laufenden DB-Instance ändern. Sie können dies tun, indem Sie die AWS Management Console AWS CLI, oder die Amazon RDS-API verwenden.

Wenn Sie die nötigen Berechtigungen dafür besitzen, können Sie Parameterwerte auch ändern, indem Sie die Befehle `ALTER DATABASE`, `ALTER ROLE` und `SET` verwenden.

DB-Instance-Parameterliste von RDS for PostgreSQL

In der folgenden Tabelle sind einige (nicht alle) Parameter aufgeführt, die in einer DB-Instance von RDS für PostgreSQL verfügbar sind. Um alle verfügbaren Parameter anzuzeigen, verwenden Sie den describe-db-parameters AWS CLI Befehl. Wenn Sie beispielsweise die Liste aller Parameter abrufen möchten, die in der Standardparametergruppe für Version 13 von RDS für PostgreSQL verfügbar sind, führen Sie den folgenden Befehl aus.

```
aws rds describe-db-parameters --db-parameter-group-name default.postgres13
```

Sie können auch die Konsole verwenden. Wählen Sie Parameter groups (Parametergruppen) aus dem Amazon-RDS-Menü und dann die Parametergruppe aus den in Ihrer AWS-Region verfügbaren Gruppen aus.

Parametername	Apply_Type	Beschreibung
application_name	Dynamisch	Legt den Namen der Anwendung fest, der in Statistiken und Protokollen verwendet werden soll.
archive_command	Dynamisch	Legt den Shell-Befehl fest, der zum Archivieren einer WAL-Datei aufgerufen wird.
array_nulls	Dynamisch	Ermöglicht die Eingabe von NULL-Elementen in Arrays.
authentication_timeout	Dynamisch	Legt die Zeit fest, die maximal zulässig ist, um die Client-Authentifizierung durchzuführen.
autovacuum	Dynamisch	Startet den Untervorgang der Selbstbereinigung.
autovacuum_analyze_scale_factor	Dynamisch	Anzahl von Tupel-Einfügungen, -Aktualisierungen oder -Löschungen vor der Analyse als Bruchteil von Reltupeln.
autovacuum_analyze_threshold	Dynamisch	Mindestanzahl von Tupel-Einfügungen, -Aktualisierungen oder -Löschungen vor der Analyse.

Parametername	Apply_Type	Beschreibung
autovacuum_freeze_max_age	Statisch	Alter, bei dem eine Selbstbereinigung für eine Tabelle ausgeführt werden soll, um einen Transaktions-ID-Wraparound zu verhindern.
autovacuum_naptime	Dynamisch	Inaktivitätszeit zwischen Selbstbereinigungen.
autovacuum_max_workers	Statisch	Legt die maximale Anzahl gleichzeitig ausgeführter Worker-Vorgänge für die Selbstbereinigung fest.
autovacuum_vacuum_cost_delay	Dynamisch	Bereinigungskostenverzögerung (in Millisekunden) für die Selbstbereinigung.
autovacuum_vacuum_cost_limit	Dynamisch	Bereinigungskostenbetrag für die Selbstbereinigung, der vor der Inaktivität verfügbar ist.
autovacuum_vacuum_scale_factor	Dynamisch	Anzahl von Tupel-Aktualisierungen oder -Löschungen vor der Bereinigung als Bruchteil von Reltupeln.
autovacuum_vacuum_threshold	Dynamisch	Mindestanzahl von Tupel-Aktualisierungen oder -Löschungen vor der Bereinigung.
backslash_quote	Dynamisch	Legt fest, ob in Zeichenfolgeliteralen ein Backslash (\) zulässig ist.
bgwriter_delay	Dynamisch	Inaktivitätszeit des Hintergrundschreibers zwischen Runden.
bgwriter_lru_maxpages	Dynamisch	Maximale Anzahl von LRU-Seiten eines Hintergrundschreibers, für die pro Runde ein Flush ausgeführt werden kann.
bgwriter_lru_multiplier	Dynamisch	Mehrfaches der durchschnittlichen Puffernutzung, die pro Runde freigegeben werden soll.
bytea_output	Dynamisch	Legt das Ausgabeformat für Bytes fest.

Parametername	Apply_Type	Beschreibung
check_function_bodies	Dynamisch	Überprüft die Funktionstexte während CREATE FUNCTION.
checkpoint_completion_target	Dynamisch	Zeit für den Flush ungültiger Puffer während des Prüfpunkts als Bruchteil des Prüfpunktintervalls.
checkpoint_segments	Dynamisch	Legt die maximale Entfernung in Protokoll segmenten zwischen automatischen Write-Ahead Log (WAL)-Prüfpunkten fest.
checkpoint_timeout	Dynamisch	Legt die maximale Zeit zwischen automatischen WAL-Prüfpunkten fest.
checkpoint_warning	Dynamisch	Ermöglicht Warnungen, wenn Prüfpunktsegmente häufiger als hierdurch angegeben gefüllt werden.
client_connection_check_interval	Dynamisch	Legt das Zeitintervall zwischen Prüfungen auf Verbindungsabbrüche während der Ausführung von Abfragen fest.
client_encoding	Dynamisch	Legt die Zeichensatzkodierung des Client fest.
client_min_messages	Dynamisch	Legt die Nachrichtenebenen fest, die an den Client gesendet werden.
commit_delay	Dynamisch	Legt die Verzögerung (in Mikrosekunden) zwischen dem Transaktions-Commit und dem Flush von WAL zum Datenträger fest.
commit_siblings	Dynamisch	Legt die Mindestzahl gleichzeitiger offener Transaktionen fest, bevor eine Commit-Verzögerung ausgeführt wird.
constraint_exclusion	Dynamisch	Ermöglicht dem Planer die Verwendung von Einschränkungen, um Abfragen zu optimieren.

Parametername	Apply_Type	Beschreibung
cpu_index_tuple_cost	Dynamisch	Legt die Schätzung des Planers für die Kosten der Verarbeitung der einzelnen Indexeinträge während einer Indexprüfung fest.
cpu_operator_cost	Dynamisch	Legt die Schätzung des Planers für die Kosten der Verarbeitung der einzelnen Operator- oder Funktionsaufrufe fest.
cpu_tuple_cost	Dynamisch	Legt die Schätzung des Planers für die Kosten der Verarbeitung der einzelnen Tupeln (Zeilen) fest.
cursor_tuple_fraction	Dynamisch	Legt die Schätzung des Planers für den Bruchteil der Zeilen eines Cursors fest, die abgerufen werden.
datestyle	Dynamisch	Legt das Anzeigeformat für Datum- und Uhrzeitwerte fest.
deadlock_timeout	Dynamisch	Legt die Zeit fest, die während einer Sperre gewartet wird, bevor auf einen Deadlock geprüft wird.
debug_pretty_print	Dynamisch	Erstellt Einschübe für Analyse- und Planstruk turanzeigen.
debug_print_parse	Dynamisch	Protokolliert die Analysestruktur der einzelnen Abfragen.
debug_print_plan	Dynamisch	Protokolliert den Ausführungsplan der einzelnen Abfragen.
debug_print_rewritten	Dynamisch	Protokolliert die neu geschriebene Analysestruktur der einzelnen Abfragen.
default_statistics_target	Dynamisch	Legt das Standardstatistikziel fest.

Parametername	Apply_Type	Beschreibung
default_tablespace	Dynamisch	Legt den Standardtabellenraum fest, in dem Tabellen und Indexe erstellt werden.
default_transaction_deferrable	Dynamisch	Legt den Standardaufschiebbarkeitsstatus neuer Transaktionen fest.
default_transaction_isolation	Dynamisch	Legt die Transaktionsisolierungsstufe jeder neuen Transaktion fest.
default_transaction_read_only	Dynamisch	Legt den Standardschreibschutzstatus neuer Transaktionen fest.
default_with_oids	Dynamisch	Erstellt standardmäßig neue Tabellen mit object IDs (OIDs).
effective_cache_size	Dynamisch	Legt die Annahme des Planers hinsichtlich der Größe des Datenträger-Caches fest.
effective_io_concurrency	Dynamisch	Die Anzahl der gleichzeitigen Anfragen, die durch das Datenträgersubsystem effizient bearbeitet werden können.
enable_bitmapscan	Dynamisch	Ermöglicht die Verwendung von Bitmap-Prüfungsplänen durch den Planer.
enable_hashagg	Dynamisch	Ermöglicht die Verwendung von Hash-Aggregationsplänen durch den Planer.
enable_hashjoin	Dynamisch	Ermöglicht die Verwendung von Hash-Join-Plänen durch den Planer.
enable_indexscan	Dynamisch	Ermöglicht die Verwendung von Indexprüfungsplänen durch den Planer.
enable_material	Dynamisch	Ermöglicht die Verwendung von Materialisierung durch den Planer.

Parametername	Apply_Type	Beschreibung
enable_mergejoin	Dynamisch	Ermöglicht die Verwendung von Zusammenführungs-Join-Plänen durch den Planer.
enable_nestloop	Dynamisch	Ermöglicht die Verwendung von Join-Plänen mit verschachtelten Schleifen durch den Planer.
enable_seqscan	Dynamisch	Ermöglicht die Verwendung von sequenziellen Prüfungsplänen durch den Planer.
enable_sort	Dynamisch	Ermöglicht die Verwendung von expliziten Sortierschritten durch den Planer.
enable_tidscan	Dynamisch	Ermöglicht die Verwendung von TID-Prüfungsplänen durch den Planer.
escape_string_warning	Dynamisch	Gibt Warnungen zu Escape-Notierungen mit Backslash (\) in gewöhnlichen Zeichenfolgeliteralen aus.
extra_float_digits	Dynamisch	Legt die Anzahl der Stellen fest, die für Gleitkommawerte angezeigt werden.
from_collapse_limit	Dynamisch	Legt die Größe der FROM-Liste fest, jenseits der Unterabfragen nicht ausgeblendet werden.
fsync	Dynamisch	Erzwingt die Synchronisierung von Aktualisierungen zum Datenträger.
full_page_writes	Dynamisch	Schreibt bei der ersten Änderung nach einem Prüfpunkt vollständige Seiten zu WAL.
geqo	Dynamisch	Ermöglicht die genetische Abfrageoptimierung.
geqo_effort	Dynamisch	GEQO: Der Aufwand, der verwendet wird, um den Standard für andere GEQO-Parameter festzulegen.

Parametername	Apply_Type	Beschreibung
geqo_generations	Dynamisch	GEQO: Die Zahl der Iterationen des Algorithmus.
geqo_pool_size	Dynamisch	GEQO: Die Anzahl der Personen in der Population.
geqo_seed	Dynamisch	GEQO: Der Seed für die zufällige Pfadauswahl.
geqo_selection_bias	Dynamisch	GEQO: Selektiver Druck innerhalb der Population.
geqo_threshold	Dynamisch	Legt den Schwellenwert für FROM-Elemente fest, jenseits derer GEQO verwendet wird.
gin_fuzzy_search_limit	Dynamisch	Legt das maximal zulässige Ergebnis für die exakte Suche durch GIN fest.
hot_standby_feedback	Dynamisch	Legt fest, ob ein Hot Standby Rückmeldungen an den primären oder Upstream Standby sendet.
intervalstyle	Dynamisch	Legt das Anzeigeformat für Intervallwerte fest.
join_collapse_limit	Dynamisch	Legt die Größe der FROM-Liste fest, jenseits der JOIN-Konstrukte nicht auf eine Ebene gebracht werden.
lc_messages	Dynamisch	Legt die Sprache fest, in der Nachrichten angezeigt werden.
lc_monetary	Dynamisch	Legt das Gebietsschema für die Formatierung von monetären Beträgen fest.
lc_numeric	Dynamisch	Legt das Gebietsschema für die Formatierung von Zahlen fest.
lc_time	Dynamisch	Legt das Gebietsschema für die Formatierung von Datum- und Uhrzeitwerten fest.

Parametername	Apply_Type	Beschreibung
log_autovacuum_min_duration	Dynamisch	Legt die Mindestausführungszeit fest, ab der Aktionen für die Selbstbereinigung protokolliert werden.
log_checkpoints	Dynamisch	Protokolliert jeden Prüfpunkt.
log_connections	Dynamisch	Protokolliert jede erfolgreiche Verbindung.
log_disconnections	Dynamisch	Protokolliert das Ende einer Sitzung einschließlich der Dauer.
log_duration	Dynamisch	Protokolliert die Dauer jeder abgeschlossenen SQL-Anweisung.
log_error_verbosity	Dynamisch	Legt die Ausführlichkeit protokollierter Nachrichten fest.
log_executor_stats	Dynamisch	Schreibt die Leistungsstatistik des Executors in das Serverprotokoll.
log_filename	Dynamisch	Legt das Dateinamenmuster für Protokolldateien fest.
log_file_mode	Dynamisch	Legt Dateiberechtigungen für Protokolldateien fest. Der Standardwert ist 0644.
log_hostname	Dynamisch	Protokolliert den Hostnamen in den Verbindungsprotokollen. Ab PostgreSQL 12 und späteren Versionen ist dieser Parameter standardmäßig „off“. Wenn diese Option aktiviert ist, verwendet die Verbindung DNS-Reverse-Lookup, um den Hostnamen abzurufen, der in den Verbindungsprotokollen erfasst wird. Wenn Sie diesen Parameter aktivieren, sollten Sie überwachen, welche Auswirkungen er auf die Zeit hat, die für den Verbindungsaufbau benötigt wird.

Parametername	Apply_Type	Beschreibung
log_line_prefix	Dynamisch	Steuert Informationen, die jeder Protokollzeile vorangestellt sind.
log_lock_waits	Dynamisch	Protokolliert lange Sperrenwartezeiten.
log_min_duration_statement	Dynamisch	Legt die Mindestausführungszeit fest, ab der Anweisungen protokolliert werden.
log_min_error_statement	Dynamisch	Veranlasst, dass alle Anweisungen, die einen Fehler auf oder jenseits dieser Stufe generieren, protokolliert werden.
log_min_messages	Dynamisch	Legt die Nachrichtenebenen fest, die protokolliert werden.
log_parser_stats	Dynamisch	Schreibt die Leistungsstatistik des Parsers in das Serverprotokoll.
log_planner_stats	Dynamisch	Schreibt die Leistungsstatistik des Planers in das Serverprotokoll.
log_rotation_age	Dynamisch	Die automatische Protokolldateirotation wird nach N Minuten ausgeführt.
log_rotation_size	Dynamisch	Die automatische Protokolldateirotation wird nach N Kilobytes ausgeführt.
log_statement	Dynamisch	Legt den Typ der protokollierten Anweisungen fest.
log_statement_stats	Dynamisch	Schreibt kumulative Leistungsstatistiken in das Serverprotokoll.
log_temp_files	Dynamisch	Protokolliert die Verwendung temporärer Dateien, die größer als diese Zahl von Kilobytes sind.
log_timezone	Dynamisch	Legt die Zeitzone fest, die in Protokollmeldungen verwendet werden soll.

Parametername	Apply_Type	Beschreibung
log_truncate_on_rotation	Dynamisch	Kürzt vorhandene Protokolldateien mit demselben Namen während der Protokollrotation.
logging_collector	Statisch	Startet einen Unterprozess, um die stderr-Ausgabe und/oder csvlogs in Protokolldateien zu erfassen.
maintenance_work_mem	Dynamisch	Legt den maximalen Arbeitsspeicher fest, der für Wartungsoperationen verwendet werden darf.
max_connections	Statisch	Legt die maximale Anzahl gleichzeitiger Verbindungen fest.
max_files_per_process	Statisch	Legt die maximale Anzahl gleichzeitig geöffneter Dateien für die einzelnen Serverprozesse fest.
max_locks_per_transaction	Statisch	Legt die maximale Anzahl von Sperren pro Transaktion fest.
max_pred_locks_per_transaction	Statisch	Legt die maximale Anzahl von Prädikatssperren pro Transaktion fest.
max_prepared_transactions	Statisch	Legt die maximale Anzahl gleichzeitig vorbereiteter Transaktionen fest.
max_stack_depth	Dynamisch	Legt die maximale Stack-Tiefe in Kilobytes fest.
max_standby_archive_delay	Dynamisch	Legt die maximale Verzögerung fest, bevor Abfragen storniert werden, wenn ein Hot Standby-Server archivierte WAL-Daten verarbeitet.
max_standby_streaming_delay	Dynamisch	Legt die maximale Verzögerung fest, bevor Abfragen storniert werden, wenn ein Hot Standby-Server gestreamte WAL-Daten verarbeitet.

Parametername	Apply_Type	Beschreibung

`max_wal_size`	Dynamisch	Legt die WAL-Größe (MB) fest, die den Prüfpunkt auslöst.
		• Für RDS für PostgreSQL 15 und frühere Versionen sind es 2 GB.
		• Für RDS for PostgreSQL 16 und spätere Versionen konfiguriert RDS for PostgreSQL automatisch auf der Grundlage der zugewiesenen Speichergröße Ihrer Instance:
		  • 6 GB für Instances mit zugewiesenem Speicher von 100 GB und mehr.
		  • 2 GB für Instances mit einem zugewiesenen Speicher von weniger als 100 GB.
		
		Verwenden Sie den folgenden Befehl auf Ihrer Amazon RDS for PostgreSQL PostgreSQL-DB-Instance, um ihren aktuellen Wert zu sehen:
		`SHOW max_wal_size;`

`min_wal_size`	Dynamisch	Legt die Mindestgröße fest, auf die das WAL verkleinert werden soll. Für PostgreSQL-Version 9.6 und früher liegt `min_wal_size` in Einheiten von 16 MB vor. Für PostgreSQ-Version 10 und höher liegt `min_wal_size` in Einheiten von 1 MB vor.
`quote_all_identifiers`	Dynamisch	Fügt beim Generieren von SQL-Fragmenten allen Bezeichnern Anführungszeichen (") hinzu.

Parametername	Apply_Type	Beschreibung
random_page_cost	Dynamisch	Legt die Schätzung des Planers für die Kosten einer nicht sequenziell abgerufenen Datenträgerseite fest. Dieser Parameter hat keinen Wert, es sei denn, die Abfrageplanverwaltung (QPM) ist aktiviert. Wenn QPM aktiviert ist, lautet der Standardwert für diesen Parameter 4.
rds.adaptive_autovacuum	Dynamisch	Optimiert die Selbstbereinigungsparameter automatisch, wenn die Transaktions-ID-Schwellenwerte überschritten werden.
rds.force_ssl	Dynamisch	Erfordert die Verwendung von SSL-Verbindungen. Der Standardwert ist für RDS für PostgreSQL Version 15 auf 1 (ein) festgelegt. Bei allen anderen Hauptversionen von RDS für PostgreSQL bis 14 ist der Standardwert auf 0 (aus) festgelegt.
rds.local_volume_spill_enabled	Statisch	Ermöglicht das Schreiben logischer Spill-Dateien auf das lokale Volume.
rds.log_retention_period	Dynamisch	Legt die Protokollaufbewahrung so fest, dass Amazon RDS PostgreSQL-Protokolle löscht, die älter als n Minuten sind.
rds.rds_superuser_reserved_connections	Statisch	Legt die Anzahl der Verbindungs-Slots fest, die für rds_superusers reserviert sind. Dieser Parameter ist nur in den Versionen 15 und früher verfügbar. Weitere Informationen finden Sie in der PostgreSQL-Dokumentation reserved_connections.

Parametername	Apply_Type	Beschreibung
`rds.replica_identity_full`	Dynamisch	Wenn Sie diesen Parameter auf setzenon, überschreibt er die Einstellung der Replikatidentität auf für alle Datenbanktabellen. `FULL` Das bedeutet, dass alle Spaltenwerte unabhängig von Ihren `REPLICA IDENTITY FULL` Einstellungen in das Write-Ahead-Protokoll (WAL) geschrieben werden. **Note** Das Aktivieren dieses Parameters kann die IOPS Ihrer Datenbankinstanz aufgrund der zusätzlichen WAL-Protokollierung erhöhen.
`rds.restrict_password_commands`	Statisch	Schränkt ein, wer Passwörter für Benutzer mit der Rolle `rds_password` verwalten darf. Setzen Sie diesen Parameter auf 1, um die Passwortbeschränkung zu aktivieren. Der Standardwert ist 0.
`search_path`	Dynamisch	Legt die Schemasuchreihenfolge für Namen fest, die nicht schemaqualifiziert sind.
`seq_page_cost`	Dynamisch	Legt die Schätzung des Planers für die Kosten einer sequenziell abgerufenen Datenträgerseite fest.
`session_replication_role`	Dynamisch	Legt das Sitzungsverhalten für Auslöser und Neuschreibungsregeln fest.
`shared_buffers`	Statisch	Legt die maximale Anzahl freigegebener Arbeitsspeicherpuffer fest, die vom Server verwendet werden.

Parametername	Apply_Type	Beschreibung
shared_preload_libraries	Statisch	Listet die freigegebenen Bibliotheken auf, die in die DB-Instance von RDS for PostgreSQL vorab geladen werden sollen. Zu den unterstützten Werten gehören: auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_tle, pg_transport, plprofiler und plrust.
ssl	Dynamisch	Ermöglicht SSL-Verbindungen.
sql_inheritance	Dynamisch	Veranlasst den standardmäßigen Einschluss von Untertabellen in verschiedene Befehle.
ssl_renegotiation_limit	Dynamisch	Legt die Menge des Datenverkehrs fest, der gesendet und empfangen werden soll, bevor die Verschlüsselungsschlüssel neu verhandelt werden.
standard_conforming_strings	Dynamisch	Veranlasst Zeichenfolgen „...“, Backslashes als Zeichen zu behandeln.
statement_timeout	Dynamisch	Legt die maximal zulässige Dauer von Anweisungen fest.
synchronize_seqscans	Dynamisch	Ermöglicht synchronisierte sequenzielle Prüfungen.
synchronous_commit	Dynamisch	Legt die Synchronisierungsstufe aktueller Transaktionen fest.
tcp_keepalives_count	Dynamisch	Maximale Anzahl von TCP-Keepalive-Neuübertragungen.
tcp_keepalives_idle	Dynamisch	Zeit zwischen der Ausgabe von TCP-Keepalives.
tcp_keepalives_interval	Dynamisch	Zeit zwischen der Ausgabe von TCP-Keepalive-Neuübertragungen.

Parametername	Apply_Type	Beschreibung
temp_buffers	Dynamisch	Legt die maximale Anzahl der temporären Puffer fest, die von den einzelnen Sitzungen verwendet werden.
temp_file_limit	Dynamisch	Legt die maximale Größe in KB fest, temporären Dateien maximal annehmen können.
temp_tablespaces	Dynamisch	Legt die Tabellenräume fest, die für temporäre Tabellen und Sortierdateien verwendet werden sollen.
timezone	Dynamisch	Legt die Zeitzone für die Anzeige und Interpretation von Zeitstempeln fest. Die Internet Assigned Numbers Authority (IANA) veröffentlicht mehrmals im Jahr neue Zeitzonen unter https://www.iana.org/time-zones. Jedes Mal, wenn RDS eine neue Wartungsnebenversion von PostgreSQL veröffentlicht, wird diese mit den neuesten Zeitzonendaten zum Zeitpunkt der Veröffentlichung ausgeliefert. Wenn Sie die neuesten Versionen von RDS für PostgreSQL verwenden, verfügen Sie über aktuelle Zeitzonendaten von RDS. Wenn Sie sichergehen möchten, dass Ihre DB-Instance über aktuelle Zeitzonendaten verfügt, empfehlen wir ein Upgrade auf eine höhere DB-Engine-Version. Sie können die Zeitzonen-Tabellen in PostgreSQL-DB-Instances nicht ändern. Die Zeitzonendaten laufender DB-Instances werden von RDS nicht geändert oder zurückgesetzt. Neue Zeitzonendaten werden nur installiert, wenn Sie ein Upgrade der Datenbank-Engine-Version durchführen.

Parametername	Apply_Type	Beschreibung
track_activities	Dynamisch	Sammelt Informationen zu Befehlen, die ausgeführt werden.
track_activity_query_size	Statisch	Legt die für pg_stat_activity.current_query reservierte Größe in Bytes fest.
track_counts	Dynamisch	Sammelt Statistiken zur Datenbankaktivität.
track_functions	Dynamisch	Sammelt Statistiken auf Funktionsebene zur Datenbankaktivität.
track_io_timing	Dynamisch	Sammelt Zeitpunktstatistiken zur Datenbank-I/O-Aktivität.
transaction_deferrable	Dynamisch	Gibt an, ob eine schreibgeschützte serialisierbare Transaktion aufgeschoben werden kann, bis sie ohne mögliche Serialisierungsfehler gestartet werden kann.
transaction_isolation	Dynamisch	Legt die Isolierungsstufe aktueller Transaktionen fest.
transaction_read_only	Dynamisch	Legt den Schreibschutzstatus aktueller Transaktionen fest.
transform_null_equals	Dynamisch	Behandelt expr=NULL als expr IS NULL.
update_process_title	Dynamisch	Aktualisiert den Titel des Vorgangs, um den aktiven SQL-Befehl anzuzeigen.
vacuum_cost_delay	Dynamisch	Bereinigungskostenverzögerung (in Millisekunden).
vacuum_cost_limit	Dynamisch	Bereinigungskostenbetrag, der vor der Inaktivität verfügbar ist.

Parametername	Apply_Type	Beschreibung
vacuum_cost_page_dirty	Dynamisch	Bereinigungskosten für eine Seite, die durch eine Bereinigung ungültig wurde.
vacuum_cost_page_hit	Dynamisch	Bereinigungskosten für eine Seite, die im Puffer-Cache gefunden wurde.
vacuum_cost_page_miss	Dynamisch	Bereinigungskosten für eine Seite, die nicht im Puffer-Cache gefunden wurde.
vacuum_defer_cleanup_age	Dynamisch	Anzahl der Transaktionen, um die Bereinigung und Hot Cleanup aufgeschoben werden sollen, wenn vorhanden.
vacuum_freeze_min_age	Dynamisch	Mindestalter, bei dem die Bereinigung eine Tabellenzeile eingefroren werden sollte.
vacuum_freeze_table_age	Dynamisch	Alter, bei dem die Bereinigung eine Tabelle vollständig scannen sollte, um Tupel einzufrieren.
wal_buffers	Statisch	Legt die Anzahl von Datenträgerseitenpuffern im freigegebenen Arbeitsspeicher für WAL fest.
wal_writer_delay	Dynamisch	Inaktivitätszeit des WAL-Schreibers zwischen WAL-Flushes.
work_mem	Dynamisch	Legt den maximalen Arbeitsspeicher fest, der für Abfrage-Workspaces verwendet werden darf.
xmlbinary	Dynamisch	Legt die Kodierung von Binärwerten in XML fest.
xmloption	Dynamisch	Legt fest, ob XML-Daten in impliziten Parsing- und Serialisierungsoperationen als Dokumente oder Inhaltsfragmente betrachtet werden sollen.

Amazon RDS verwendet die PostgreSQL-Standardeinheiten für alle Parameter. Die folgende Tabelle zeigt die PostgreSQL-Standardeinheit für die einzelnen Parameter.

Parametername	Einheit
archive_timeout	S
authentication_timeout	S
autovacuum_naptime	S
autovacuum_vacuum_cost_delay	ms
bgwriter_delay	ms
checkpoint_timeout	S
checkpoint_warning	S
deadlock_timeout	ms
effective_cache_size	8 KB
lock_timeout	ms
log_autovacuum_min_duration	ms
log_min_duration_statement	ms
log_rotation_age	Minuten
log_rotation_size	KB
log_temp_files	KB
maintenance_work_mem	KB
max_stack_depth	KB
max_standby_archive_delay	ms
max_standby_streaming_delay	ms
post_auth_delay	S

Parametername	Einheit
pre_auth_delay	S
segment_size	8 KB
shared_buffers	8 KB
statement_timeout	ms
ssl_renegotiation_limit	KB
tcp_keepalives_idle	S
tcp_keepalives_interval	S
temp_file_limit	KB
work_mem	KB
temp_buffers	8 KB
vacuum_cost_delay	ms
wal_buffers	8 KB
wal_receiver_timeout	ms
wal_segment_size	B
wal_sender_timeout	ms
wal_writer_delay	ms
wal_receiver_status_interval	S

Optimierung mit Warteereignissen für RDS für PostgreSQL

Warteereignisse sind ein wichtiges Optimierungs-Tool für RDS für PostgreSQL. Wenn Sie herausfinden können, warum Sitzungen auf Ressourcen warten und was sie tun, können Sie Engpässe besser reduzieren. Anhand der Informationen in diesem Abschnitt können Sie mögliche Ursachen und Abhilfemaßnahmen ermitteln. In diesem Abschnitt werden außerdem grundlegende PostgreSQL-Optimierungskonzepte erörtert.

Die Warteereignisse in diesem Abschnitt gelten speziell für RDS für PostgreSQL.

Themen

- Grundlegende Konzepte für die Optimierung von RDS für PostgreSQL
- Warteereignisse von RDS für PostgreSQL
- Kunde: ClientRead
- Kunde: ClientWrite
- CPU
- io:BuffileRead und io:BuffileWrite
- E/A:DataFileRead
- IO: WALWrite
- IPC:Parallele Warteereignisse
- Lock:advisory
- Lock:extend
- Lock:Relation
- Lock:transactionid
- Lock:tuple
- LWLock:BufferMapping (LWLock:buffer_mapping)
- LWLock:BufferIO (IPC:BufferIO)
- LWLock:buffer_content (BufferContent)
- LWLock:lock_manager (:lockmanager) LWLock
- LWLock:pg_stat_statements
- Timeout:PgSleep
- Timeout:VacuumDelay

Grundlegende Konzepte für die Optimierung von RDS für PostgreSQL

Bevor Sie Ihre Datenbank von RDS für PostgreSQL optimieren, sollten Sie wissen, was Warteereignisse sind und warum sie auftreten. Sehen Sie sich auch die grundlegende Speicher- und Festplattenarchitektur von RDS für PostgreSQL an. Ein hilfreiches Architekturdiagramm finden Sie im PostgreSQL-Wikibook.

Themen

- Warteereignisse von RDS für PostgreSQL
- Speicher von RDS für PostgreSQL
- Prozesse von RDS für PostgreSQL

Warteereignisse von RDS für PostgreSQL

Ein Warteereignis ist ein Hinweis darauf, dass die Sitzung auf eine Ressource wartet. Das Warteereignis `Client:ClientRead` tritt beispielsweise auf, wenn RDS für PostgreSQL darauf wartet, Daten vom Client zu empfangen. Sitzungen warten in der Regel auf Ressourcen wie die folgenden.

- Singlethread-Zugriff auf einen Puffer, beispielsweise wenn eine Sitzung versucht, einen Puffer zu ändern
- Eine Zeile, die derzeit von einer anderen Sitzung gesperrt ist
- Eine gelesene Datendatei
- Eine geschriebene Protokolldatei

Um beispielsweise eine Abfrage zu erfüllen, kann die Sitzung einen vollständigen Tabellenscan durchführen. Wenn sich die Daten noch nicht im Arbeitsspeicher befinden, wartet die Sitzung, I/O bis die Festplatte abgeschlossen ist. Wenn die Puffer in den Speicher gelesen werden, muss die Sitzung möglicherweise warten, da andere Sitzungen auf dieselben Puffer zugreifen. Die Datenbank zeichnet die Wartezeiten unter Verwendung eines vordefinierten Warteereignisses auf. Diese Ereignisse sind in Kategorien eingeteilt.

Ein einzelnes Warteereignis ist kein Anzeichen für ein Leistungsproblem. Wenn sich beispielsweise die angeforderten Daten nicht im Speicher befinden, müssen die Daten von der Festplatte gelesen werden. Wenn eine Sitzung eine Zeile für eine Aktualisierung sperrt, wartet eine andere Sitzung darauf, dass die Zeile entsperrt wird, damit sie sie aktualisieren kann. Bei einem Commit muss

gewartet werden, bis der Schreibvorgang in eine Protokolldatei abgeschlossen ist. Wartezeiten sind ein wesentlicher Bestandteil des normalen Funktionierens einer Datenbank.

Eine große Anzahl von Warteereignissen hingegen weist normalerweise auf ein Leistungsproblem hin. In solchen Fällen können Sie Warteereignisdaten verwenden, um zu bestimmen, wo die Sitzungen Zeit verbringen. Wenn beispielsweise ein Bericht, der normalerweise in Minuten ausgeführt wird, jetzt Stunden dauert, können Sie die Warteereignisse identifizieren, die am meisten zur Gesamtwartezeit beitragen. Wenn Sie die Ursachen für die häufigsten Warteereignisse ermitteln können, können Sie manchmal Änderungen vornehmen, die die Leistung verbessern. Wenn Ihre Sitzung beispielsweise auf eine Zeile wartet, die von einer anderen Sitzung gesperrt wurde, können Sie die Sperrsitzung beenden.

Speicher von RDS für PostgreSQL

Der Arbeitsspeicher von RDS für PostgreSQL ist in gemeinsam genutztem und lokalem Speicher unterteilt.

Themen

- Gemeinsam genutzter Arbeitsspeicher von RDS für PostgreSQL
- Lokaler Arbeitsspeicher in RDS für PostgreSQL

Gemeinsam genutzter Arbeitsspeicher von RDS für PostgreSQL

RDS für PostgreSQL weist beim Start der Instance gemeinsam genutzten Speicher zu. Shared Memory ist in mehrere Teilbereiche unterteilt. In den folgenden Abschnitten werden die wichtigsten beschrieben.

Themen

- Freigegebene Puffer
- Write-Ahead-Protokoll (WAL)-Puffer

Freigegebene Puffer

Der freigegebene Pufferpool ist ein Speicherbereich von RDS für PostgreSQL, der alle Seiten enthält, die von Anwendungsverbindungen verwendet werden oder wurden. Eine Seite ist die Speicherversion eines Plattenblocks. Der gemeinsam genutzte Pufferpool zwischenspeichert die von der Platte gelesenen Datenblöcke. Der Pool reduziert die Notwendigkeit, Daten erneut von der Festplatte zu lesen, wodurch die Datenbank effizienter arbeitet.

Jede Tabelle und jeder Index wird als Array von Seiten einer festen Größe gespeichert. Jeder Block enthält mehrere Tupel, die Zeilen entsprechen. Ein Tupel kann auf jeglicher Seite gespeichert werden.

Der gemeinsam genutzte Pufferpool hat endlichen Speicher. Wenn eine neue Anforderung eine Seite erfordert, die sich nicht im Speicher befindet und kein Speicher mehr vorhanden ist, entfernt RDS für PostgreSQL eine weniger häufig verwendete Seite, um die Anforderung zu erfüllen. Die Räumungsrichtlinie wird durch einen Takt-Sweep-Algorithmus implementiert.

Der Parameter `shared_buffers` bestimmt, wie viel Speicher der Server für das Caching von Daten bereitstellt. Der Standardwert ist auf `{DBInstanceClassMemory/32768}` Byte festgelegt, basierend auf dem verfügbaren Speicher für die DB-Instance.

Write-Ahead-Protokoll (WAL)-Puffer

Ein Write-Ahead-Protokoll-Puffer (WAL) enthält Transaktionsdaten, die RDS für PostgreSQL später in den persistenten Speicher schreibt. Mithilfe des WAL-Mechanismus kann RDS für PostgreSQL folgende Vorgänge ausführen:

- Daten nach einem Fehler wiederherstellen
- Reduzieren Sie die I/O Festplattenkapazität, indem Sie häufige Schreibvorgänge auf die Festplatte vermeiden

Wenn ein Client Daten ändert, schreibt RDS für PostgreSQL die Änderungen in den WAL-Puffer. Wenn der Client einen `COMMIT` ausgibt, schreibt der WAL-Writerprozess Transaktionsdaten in die WAL-Datei.

Der `wal_level` Parameter bestimmt, wie viele Informationen in die WAL geschrieben werden, mit möglichen Werten wie `minimal`, `replica`, und`logical`.

Lokaler Arbeitsspeicher in RDS für PostgreSQL

Jeder Backend-Prozess weist lokalen Speicher für die Abfrageverarbeitung zu.

Themen

- Arbeitsspeicherbereich
- Wartungs-Arbeitsspeicherbereich
- Temporärer Pufferbereich

Arbeitsspeicherbereich

Der Arbeitsspeicherbereich enthält temporäre Daten für Abfragen, die Sortierungen und Hashes durchführen. Beispielsweise führt eine Abfrage mit einer `ORDER BY`-Klausel eine Sortierung durch. Abfragen verwenden Hash-Tabellen in Hash-Joins und Aggregationen.

Der `work_mem` Parameter gibt die Speichermenge an, die von internen Sortieroperationen und Hashtabellen vor dem Schreiben in temporäre Festplattendateien verwendet werden soll, gemessen in Megabyte. Der Standardwert lautet 4 MB. Mehrere Sitzungen können gleichzeitig ausgeführt werden, und jede Sitzung kann Wartungsvorgänge parallel ausführen. Aus diesem Grund kann der gesamte verwendete Arbeitsspeicher ein Vielfaches der Einstellung `work_mem` betragen.

Wartungs-Arbeitsspeicherbereich

Der Wartungsarbeitsspeicherbereich speichert Daten für Wartungsvorgänge zwischen. Zu diesen Vorgängen gehören das Vakuumieren, das Erstellen eines Index und das Hinzufügen von Fremdschlüsseln.

Der `maintenance_work_mem` Parameter gibt die maximale Speichermenge an, die für Wartungsvorgänge verwendet werden soll, gemessen in Megabyte. Der Standardwert lautet 64 MB. In einer Datenbanksitzung kann jeweils nur ein Wartungsvorgang ausgeführt werden.

Temporärer Pufferbereich

Der temporäre Pufferbereich speichert temporäre Tabellen für jede Datenbanksitzung zwischen.

Jede Sitzung weist temporäre Puffer nach Bedarf bis zu dem von Ihnen angegebenen Limit zu. Wenn die Sitzung endet, löscht der Server die Puffer.

Der `temp_buffers` Parameter legt die maximale Anzahl temporärer Puffer fest, die von jeder Sitzung verwendet werden, gemessen in Megabyte. Der Standardwert ist 8 MB. Vor der ersten Verwendung temporärer Tabellen innerhalb einer Sitzung können Sie den `temp_buffers`-Wert ändern.

Prozesse von RDS für PostgreSQL

RDS für PostgreSQL verwendet mehrere Prozesse.

Themen

- Postmaster-Prozess
- Backend-Prozesse

- Hintergrundprozesse

Postmaster-Prozess

Der Postmaster-Prozess ist der erste Prozess, der beim Öffnen von RDS für PostgreSQL gestartet wird. Der Postmaster-Prozess hat die folgenden Hauptaufgaben:

- Hintergrundprozesse teilen und überwachen
- Empfangen Sie Authentifizierungsanfragen von Clientprozessen und authentifizieren Sie sie, bevor Sie der Datenbank erlauben, Anfragen zu bearbeiten

Backend-Prozesse

Wenn der Postmaster eine Client-Anfrage authentifiziert, forkisiert der Postmaster einen neuen Backend-Prozess, auch Postgres-Prozess genannt. Ein Client-Prozess verbindet sich mit genau einem Backend-Prozess. Der Client-Prozess und der Backend-Prozess kommunizieren direkt ohne Eingriff des Postmaster-Prozesses.

Hintergrundprozesse

Der Postmaster-Prozess teilt mehrere Prozesse, die unterschiedliche Backend-Aufgaben ausführen. Einige der wichtigeren sind die folgenden:

- WAL-Writer

  Aurora PostgreSQL schreibt Daten im WAL-Puffer (Write Ahead Logging) in die Protokolldateien. Das Prinzip der Write-Ahead-Protokollierung besteht darin, dass die Datenbank keine Änderungen in die Datendateien schreiben kann, bis die Datenbank Protokolldatensätze geschrieben hat, die diese Änderungen auf die Festplatte beschreiben. Der WAL-Mechanismus reduziert Festplatten-I/O und ermöglicht RDS für PostgreSQL, die Protokolle zu verwenden, um die Datenbank nach einem Fehler wiederherzustellen.
- Hintergrund-Autor

  Dieser Prozess schreibt regelmäßig schmutzige (modifizierte) Seiten aus den Speicherpuffern in die Datendateien. Eine Seite wird schmutzig, wenn ein Backend-Prozess sie im Speicher ändert.
- Autovacuum-Daemon

  Der Daemon besteht aus Folgendem:

- Der Autovacuum-Launcher
- Die Autovacuum-Worker-Prozesse

Wenn Autovacuum aktiviert ist, sucht es nach Tabellen mit einer großen Anzahl eingefügter, aktualisierter oder gelöschter Tupel. Der Daemon hat folgende Aufgaben:

- Wiederherstellen oder Wiederverwenden von Speicherplatz, der von aktualisierten oder gelöschten Zeilen belegt ist
- Vom Planer verwendete Statistiken aktualisieren
- Schutz vor Verlust alter Daten durch Transaktions-ID-Wraparound

Die Autovacuum-Funktion automatisiert die Ausführung von `VACUUM`- und `ANALYZE`-Befehlen. `VACUUM` hat folgende Varianten: Standard und Voll. Standardvakuum läuft parallel zu anderen Datenbankvorgängen. `VACUUM FULL` erfordert eine exklusive Sperre für die Tabelle, an der es arbeitet. Daher kann es nicht parallel zu Vorgängen ausgeführt werden, die auf dieselbe Tabelle zugreifen. `VACUUM`erzeugt eine beträchtliche Menge an I/O Datenverkehr, was zu Leistungseinbußen bei anderen aktiven Sitzungen führen kann.

Warteereignisse von RDS für PostgreSQL

Die folgende Tabelle listet die Warteereignisse für RDS für PostgreSQL auf, die am häufigsten auf Leistungsprobleme hinweisen, und fasst die häufigsten Ursachen und Korrekturmaßnahmen zusammen.

Warteereignis	Definition
Kunde: ClientRead	Dieses Ereignis tritt auf, wenn RDS für PostgreSQL darauf wartet, Daten vom Client zu empfangen.
Kunde: ClientWrite	Dieses Ereignis tritt auf, wenn RDS für PostgreSQL darauf wartet, Daten an den Client zu schreiben.
CPU	Dieses Ereignis tritt auf, wenn ein Thread in der CPU aktiv ist oder auf die CPU wartet.
io:BuffileRead und io:BuffileWrite	Diese Ereignisse treten auf, wenn RDS für PostgreSQL temporäre Dateien erstellt.

Warteereignis	Definition
E/A:DataFileRead	Dieses Ereignis tritt auf, wenn eine Verbindung darauf wartet, dass ein Backend-Prozess eine erforderliche Seite aus dem Speicher liest, da die Seite nicht im gemeinsam genutzten Speicher verfügbar ist.
IO: WALWrite	Dieses Ereignis tritt auf, wenn RDS für PostgreSQL darauf wartet, dass die Write-Ahead-Protokoll-Puffe r (WAL) in eine WAL-Datei geschrieben werden.
Lock:advisory	Dieses Ereignis tritt auf, wenn eine PostgreSQL-Anwendung eine Sperre verwendet, um Aktivitäten über mehrere Sitzungen hinweg zu koordinieren.
Lock:extend	Dieses Ereignis tritt ein, wenn ein Backend-Prozess darauf wartet, eine Beziehung zu sperren, um sie zu erweitern, während ein anderer Prozess diese Beziehung für denselben Zweck gesperrt hat.
Lock:Relation	Dieses Ereignis tritt ein, wenn eine Abfrage darauf wartet, eine Sperre für eine Tabelle oder Ansicht zu erhalten, die derzeit von einer anderen Transaktion gesperrt ist.
Lock:transactionid	Dieses Ereignis tritt ein, wenn eine Transaktion auf eine Sperre auf Zeilenebene wartet.
Lock:tuple	Dieses Ereignis tritt ein, wenn ein Backend-P rozess darauf wartet, eine Sperre für ein Tupel zu erlangen.
LWLock:BufferMapping (LWLock:b uffer_mapping)	Dieses Ereignis tritt ein, wenn eine Sitzung darauf wartet, einen Datenblock einem Puffer im gemeinsam genutzten Pufferpool zuzuordnen.

Warteereignis	Definition
LWLock:BufferIO (IPC:BufferIO)	Dieses Ereignis tritt auf, wenn RDS for PostgreSQL darauf wartet, dass andere Prozesse ihre input/out put (I/O-) Operationen beenden, wenn gleichzeitig versucht wird, auf eine Seite zuzugreifen.
LWLock:buffer_content (BufferContent)	Dieses Ereignis tritt ein, wenn eine Sitzung darauf wartet, eine Datenseite im Speicher zu lesen oder zu schreiben, während eine andere Sitzung diese Seite zum Schreiben gesperrt hat.
LWLock:lock_manager (:lockmanager) LWLock	Dieses Ereignis tritt auf, wenn die Engine von RDS für PostgreSQL den Speicherbereich der gemeinsam genutzten Sperre verwaltet, um eine Sperre zuzuweisen, zu überprüfen und aufzuheben, wenn eine Fast-Path-Sperre nicht möglich ist.
Timeout:PgSleep	Dieses Ereignis tritt ein, wenn ein Serverprozess die Funktion `pg_sleep` aufgerufen hat und darauf wartet, dass das Sleep-Timeout abläuft.
Timeout:VacuumDelay	Dieses Ereignis weist darauf hin, dass der Bereinigungsprozess inaktiv ist, da die geschätzte Kostengrenze erreicht wurde.

Kunde: ClientRead

Das `Client:ClientRead`-Ereignis tritt auf, wenn RDS für PostgreSQL darauf wartet, Daten vom Client zu empfangen.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für RDS für PostgreSQL Version 10 und höher unterstützt.

Kontext

Eine DB-Instance von RDS für PostgreSQL wartet darauf, Daten vom Client zu empfangen. Die DB-Instance von RDS für PostgreSQL muss die Daten vom Client empfangen, bevor sie weitere Daten an den Client senden kann. Die Zeit, die die Instance wartet, bevor sie Daten vom Client empfängt, ist ein `Client:ClientRead`-Ereignis.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Häufige Gründe dafür, dass das `Client:ClientRead`-Ereignis in den Top-Wartezeiten angezeigt wird, sind die folgenden:

Erhöhte Netzwerklatenz

Es kann zu einer erhöhten Netzwerklatenz zwischen dem DB-Cluster von RDS für PostgreSQL und dem Client kommen. Eine höhere Netzwerklatenz erhöht die Zeit, die die DB-Instance benötigt, um Daten vom Client zu empfangen.

Erhöhte Belastung des Clients

Auf dem Client kann es zu CPU-Druck oder Netzwerksättigung kommen. Eine Zunahme der Last auf dem Client kann die Übertragung von Daten vom Client zur DB-Instance von RDS für PostgreSQL verzögern.

Übermäßige Netzwerkrundfahrten

Eine große Anzahl von Netzwerk-Roundtrips zwischen der DB-Instance von RDS für PostgreSQL und dem Client kann die Datenübertragung vom Client zur DB-Instance von RDS für PostgreSQL verzögern.

Großer Kopiervorgang

Während eines Kopiervorgangs werden die Daten vom Dateisystem des Clients an die DB-Instance von RDS für PostgreSQL übertragen. Das Senden einer großen Datenmenge an die DB-Instance kann die Übertragung von Daten vom Client zur DB-Instance verzögern.

Verbindung von Clients im Leerlauf

Wenn sich ein Client in einem `idle in transaction`-Zustand mit der DB-Instance von RDS für PostgreSQL verbindet, wartet die DB-Instance möglicherweise darauf, dass der Client weitere

Daten sendet oder einen Befehl ausgibt. Eine Verbindung in diesem Zustand kann zu einer Zunahme von `Client:ClientRead`-Ereignissen führen.

PgBouncer wird für das Verbindungspooling verwendet

PgBouncer hat eine Netzwerkkonfigurationseinstellung auf niedriger Ebene aufgerufen`pkt_buf`, die standardmäßig auf 4.096 gesetzt ist. Wenn der Workload Abfragepakete mit mehr als 4.096 Byte durchsendet, empfehlen wir PgBouncer, die Einstellung auf 8.192 zu erhöhen. `pkt_buf` Wenn die neue Einstellung die Anzahl der `Client:ClientRead`-Ereignisse nicht verringert, empfehlen wir, die `pkt_buf`-Einstellung auf höhere Werte zu erhöhen, z. B. 16.384 oder 32.768. Wenn der Abfragetext groß ist, kann die größere Einstellung besonders hilfreich sein.

Aktionen

Abhängig von den Ursachen Ihres Warteereignisses empfehlen wir verschiedene Aktionen.

Themen

- Platzieren Sie die Clients in derselben Availability Zone und im gleichen VPC-Subnetz wie die Instance
- Skalieren Sie Ihren -C
- Instances der aktuellen Generation verwenden
- Erhöhung der Netzwerkbandbreite
- Überwachen Sie Maximen für die Netzwerkleistung
- Überwachen Sie auf Transaktionen im Status „Leerlauf in Transaktion“

Platzieren Sie die Clients in derselben Availability Zone und im gleichen VPC-Subnetz wie die Instance

Um die Netzwerklatenz zu reduzieren und den Netzwerkdurchsatz zu erhöhen, platzieren Sie Clients in dieselbe Availability Zone und das gleiche Virtual Private Cloud (VPC)-Subnetz wie die DB-Instance von RDS für PostgreSQL. Stellen Sie sicher, dass sich die Clients geografisch so nah wie möglich an der DB-Instance befinden.

Skalieren Sie Ihren -C

Ermitteln Sie anhand von Amazon CloudWatch oder anderen Host-Metriken, ob Ihr Client derzeit durch CPU- oder Netzwerkbandbreite oder beides eingeschränkt ist. Wenn der Kunde eingeschränkt ist, skalieren Sie Ihren Kunden entsprechend.

Instances der aktuellen Generation verwenden

In einigen Fällen verwenden Sie möglicherweise keine DB-Instance-Klasse, die Jumbo-Frames unterstützt. Wenn Sie Ihre Anwendung auf Amazon EC2 ausführen, sollten Sie eine Instance der aktuellen Generation für den Client verwenden. Konfigurieren Sie außerdem die maximale Übertragungseinheit (MTU) im Kundenvorgangssystem Diese Technik könnte die Anzahl der Netzläufe reduzieren und den Netzwerkdurchsatz erhöhen. Weitere Informationen finden Sie unter Jumbo Frames (9001 MTU) im Amazon EC2 EC2-Benutzerhandbuch.

Weitere Informationen zu DB-Instance-Klassen finden Sie unter . Um die DB-Instance-Klasse zu bestimmen, die einem Amazon EC2-Instance-Typ entspricht, platzieren Sie `db.` vor dem Namen des Amazon EC2-Instance-Typs. Beispielsweise entspricht die `r5.8xlarge`-Amazon EC2-Instance der `db.r5.8xlarge`-DB-Instance-Klasse.

Erhöhung der Netzwerkbandbreite

Verwenden Sie `NetworkReceiveThroughput` und `NetworkTransmitThroughput` CloudWatch Amazon-Metriken, um den eingehenden und ausgehenden Netzwerkverkehr auf der DB-Instance zu überwachen. Diese Metriken können Ihnen helfen festzustellen, ob die Netzwerkbandbreite für Ihre Workload ausreicht.

Wenn Ihre Netzwerkbandbreite nicht ausreicht, erhöhen Sie sie. Wenn der AWS Client oder Ihre DB-Instance die Grenzwerte für die Netzwerkbandbreite erreicht, besteht die einzige Möglichkeit, die Bandbreite zu erhöhen, darin, Ihre DB-Instance-Größe zu erhöhen. Weitere Informationen finden Sie unter DB-Instance-Klassenarten.

Weitere Informationen zu CloudWatch Metriken finden Sie unter CloudWatch Amazon-Metriken für Amazon RDS.

Überwachen Sie Maximen für die Netzwerkleistung

Wenn Sie Amazon EC2-Clients verwenden, bietet Amazon EC2 Maximum für Netzwerkleistungsmetriken, einschließlich der aggregierten eingehenden und ausgehenden Netzwerkbandbreite. Es bietet auch Verbindungsverfolgung, um sicherzustellen, dass Pakete wie erwartet zurückgegeben werden, und Zugriff auf Link-lokale Dienste für Dienste wie das Domain Name System (DNS). Um diese Maximen zu überwachen, verwenden Sie einen aktuell erweiterten Netzwerktreiber und überwachen Sie die Netzwerkleistung für Ihren Client.

Weitere Informationen finden Sie unter Überwachen der Netzwerkleistung für Ihre Amazon EC2 EC2-Instance im Amazon EC2 EC2-Benutzerhandbuch und Überwachen der Netzwerkleistung für Ihre Amazon EC2 EC2-Instance im Amazon EC2 EC2-Benutzerhandbuch.

Überwachen Sie auf Transaktionen im Status „Leerlauf in Transaktion“

Überprüfen Sie, ob Sie eine steigende Anzahl von `idle in transaction`-Verbindungen haben. Beobachten Sie dazu die `state`-Spalte in der `pg_stat_activity`-Tabelle. Möglicherweise können Sie die Verbindungsquelle identifizieren, indem Sie eine Abfrage ähnlich der folgenden ausführen.

```
select client_addr, state, count(1) from pg_stat_activity
where state like 'idle in transaction%'
group by 1,2
order by 3 desc
```

Kunde: ClientWrite

Das `Client:ClientWrite`-Ereignis tritt auf, wenn RDS für PostgreSQL darauf wartet, Daten an den Client zu schreiben.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für RDS für PostgreSQL Version 10 und höher unterstützt.

Kontext

Ein Clientprozess muss alle Daten lesen, die von einem DB-Cluster von RDS für PostgreSQL empfangen wurden, bevor der Cluster weitere Daten senden kann. Die Zeit, die der Cluster wartet, bevor weitere Daten an den Client gesendet werden, ist ein `Client:ClientWrite`-Ereignis.

Ein reduzierter Netzwerkdurchsatz zwischen dem DB-Cluster von RDS für PostgreSQL und dem Client kann dieses Ereignis verursachen. CPU-Druck und Netzwerksättigung auf dem Client können dieses Ereignis ebenfalls verursachen. CPU-Druck liegt vor, wenn die CPU voll ausgelastet ist und Aufgaben auf CPU-Zeit warten. Eine Netzwerksättigung liegt vor, wenn das Netzwerk zwischen Datenbank und Client mehr Daten überträgt, als es verarbeiten kann.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Häufige Gründe dafür, dass das `Client:ClientWrite`-Ereignis in den Top-Wartezeiten angezeigt wird, sind die folgenden:

Erhöhte Netzwerklatenz

Es kann zu einer erhöhten Netzwerklatenz zwischen dem DB-Cluster von RDS für PostgreSQL und dem Client kommen. Eine höhere Netzwerklatenz erhöht die Zeit, die der Client benötigt, um die Daten zu empfangen.

Erhöhte Belastung des Clients

Auf dem Client kann es zu CPU-Druck oder Netzwerksättigung kommen. Eine Erhöhung der Last des Clients verzögert den Empfang von Daten aus der DB-Instance von RDS für PostgreSQL.

Große Datenmenge, die an den Kunden gesendet werden

Die DB-Instance von RDS für PostgreSQL sendet möglicherweise eine große Datenmenge an den Client. Ein Client kann die Daten möglicherweise nicht so schnell empfangen, wie der Cluster sie sendet. Aktivitäten wie das Kopieren einer großen Tabelle können zu einer Zunahme von `Client:ClientWrite`-Ereignissen führen.

Aktionen

Abhängig von den Ursachen Ihres Warteereignisses empfehlen wir verschiedene Aktionen.

Themen

- Platzieren Sie die Clients im selben Availability Zone- und VPC-Subnetz wie der Cluster
- Instances der aktuellen Generation verwenden
- Reduzieren Sie die an den Kunden gesendeten Daten
- Skalieren Sie Ihren -C

Platzieren Sie die Clients im selben Availability Zone- und VPC-Subnetz wie der Cluster

Um die Netzwerklatenz zu reduzieren und den Netzwerkdurchsatz zu erhöhen, platzieren Sie Clients in dieselbe Availability Zone und das gleiche Virtual Private Cloud (VPC)-Subnetz wie die DB-Instance von RDS für PostgreSQL.

Instances der aktuellen Generation verwenden

In einigen Fällen verwenden Sie möglicherweise keine DB-Instance-Klasse, die Jumbo-Frames unterstützt. Wenn Sie Ihre Anwendung auf Amazon EC2 ausführen, sollten Sie eine Instance der aktuellen Generation für den Client verwenden. Konfigurieren Sie außerdem die maximale Übertragungseinheit (MTU) im Kundenvorgangssystem Diese Technik könnte die Anzahl der Netzläufe reduzieren und den Netzwerkdurchsatz erhöhen. Weitere Informationen finden Sie unter Jumbo Frames (9001 MTU) im Amazon EC2 EC2-Benutzerhandbuch.

Weitere Informationen zu DB-Instance-Klassen finden Sie unter . Um die DB-Instance-Klasse zu bestimmen, die einem Amazon EC2-Instance-Typ entspricht, platzieren Sie `db.` vor dem Namen des Amazon EC2-Instance-Typs. Beispielsweise entspricht die `r5.8xlarge`-Amazon EC2-Instance der `db.r5.8xlarge`-DB-Instance-Klasse.

Reduzieren Sie die an den Kunden gesendeten Daten

Passen Sie Ihre Anwendung nach Möglichkeit an, um die Datenmenge zu reduzieren, die die DB-Instance von RDS für PostgreSQL Cluster an den Client sendet. Solche Anpassungen entlasten die CPU- und Netzwerkkonflikte auf dem Client.

Skalieren Sie Ihren -C

Ermitteln Sie anhand von Amazon CloudWatch oder anderen Host-Metriken, ob Ihr Client derzeit durch CPU- oder Netzwerkbandbreite oder beides eingeschränkt ist. Wenn der Kunde eingeschränkt ist, skalieren Sie Ihren Kunden entsprechend.

CPU

Dieses Ereignis tritt auf, wenn ein Thread in der CPU aktiv ist oder auf die CPU wartet.

Themen

- Unterstützte Engine-Versionen
- Context
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen sind für alle Versionen von RDS für PostgreSQL relevant.

Context

Die Zentrale Verarbeitungseinheit (CPU) ist die Komponente eines Computers, die Anweisungen ausführt. Beispielsweise führen CPU-Anweisungen arithmetische Vorgänge aus und tauschen Daten im Speicher aus. Wenn eine Abfrage die Anzahl der Anweisungen erhöht, die sie über das Datenbank-Engine ausführt, erhöht sich der Zeitaufwand für die Ausführung der Abfrage. CPU-Scheduling gibt einem Prozess CPU-Zeit. Die Planung wird vom Kernel des Betriebssystems orchestriert.

Themen

- Wie kann man sagen, wann diese Wartezeit stattfindet
- DbloadCPU-Metrik
- Metriken für Os.cPuUtilization
- Wahrscheinliche Ursache für CPU-Planung

Wie kann man sagen, wann diese Wartezeit stattfindet

Dieses CPU-Warteereignis zeigt an, dass ein Backend-Prozess in der CPU aktiv ist oder auf die CPU wartet. Sie wissen, dass es passiert, wenn eine Abfrage die folgenden Informationen anzeigt:

- Die Spalte `pg_stat_activity.state` hat den Wert `active`.
- Die Spalten `wait_event_type` und `wait_event` in `pg_stat_activity` sind beide `null`.

Führen Sie die folgende Abfrage aus, um die Backend-Prozesse anzuzeigen, die auf der CPU verwenden oder auf dieser warten.

```
SELECT *
FROM   pg_stat_activity
WHERE  state = 'active'
AND    wait_event_type IS NULL
AND    wait_event IS NULL;
```

DbloadCPU-Metrik

Die Performance Insights-Metrik für CPU ist `DBLoadCPU`. Der Wert für `DBLoadCPU` kann vom Wert für die Amazon CloudWatch-Metrik `CPUUtilization` abweichen. Die letztere Metrik wird vom HyperVisor für eine Datenbank-Instance gesammelt.

Metriken für Os.cPuUtilization

Performance Insights Betriebssystem-Metriken liefern detaillierte Informationen zur CPU-Auslastung. Sie können beispielsweise die folgenden Metriken anzeigen:

- `os.cpuUtilization.nice.avg`
- `os.cpuUtilization.total.avg`
- `os.cpuUtilization.wait.avg`
- `os.cpuUtilization.idle.avg`

Performance Insights meldet die CPU-Auslastung durch die Datenbank-Engine als `os.cpuUtilization.nice.avg`.

Wahrscheinliche Ursache für CPU-Planung

Der Betriebssystem-Kernel übernimmt die Planung für die CPU. Wenn die CPU aktiv ist, muss ein Prozess möglicherweise warten, bis er geplant wird. Die CPU ist aktiv, während sie Berechnungen durchführt. Sie ist außerdem aktiv, während sie einen inaktiven Thread vorliegen hat, der nicht läuft, d. h. einen inaktiven Thread, der auf Speicher-I/O wartet. Diese Art von I/O dominiert die typische Datenbank-Workload.

Wenn die folgenden Bedingungen erfüllt sind, werden Prozesse wahrscheinlich warten, bis die folgenden Bedingungen erfüllt sind:

- Die CloudWatch `CPUUtilization`-Metrik liegt nahe bei 100 Prozent.
- Die durchschnittliche Belastung ist größer als die Anzahl der vCPUs, was auf eine hohe Last hinweist. Sie finden die `loadAverageMinute`-Metrik im Abschnitt Betriebssystemmetriken in Performance Insights.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn das CPU-Warteereignis mehr als normal auftritt und möglicherweise auf ein Leistungsproblem hinweist, sind typische Ursachen die folgenden.

Themen

- Wahrscheinliche Ursachen für plötzliche Stacheln
- Wahrscheinliche Ursachen für langfristige Hochfrequenz
- Corner Cases

Wahrscheinliche Ursachen für plötzliche Stacheln

Die wahrscheinlichsten Ursachen für plötzliche Stacheln sind wie folgt:

- Ihre Anwendung hat zu viele gleichzeitige Verbindungen zur Datenbank geöffnet. Dieses Szenario wird als „Verbindungssturm“ bezeichnet.
- Ihre Anwendungs-Workload hat sich auf eine der folgenden Weisen geändert:
  - Neue Anfragen
  - Eine Zunahme der Größe Ihres Datensatzes
  - Indexpflege oder -erstellung
  - Neue Funktionen
  - Neue Betreiber
  - Eine Zunahme der parallelen Abfrageausführung
- Ihre Abfrageausführungspläne haben sich geändert. In einigen Fällen kann eine Änderung zu einem Anstieg der Puffer führen. Beispielsweise verwendet die Abfrage jetzt einen sequentiellen Scan, als sie zuvor einen Index verwendet hat. In diesem Fall benötigen die Abfragen mehr CPU, um dasselbe Ziel zu erreichen.

Wahrscheinliche Ursachen für langfristige Hochfrequenz

Die wahrscheinlichsten Ursachen für Ereignisse, die über einen langen Zeitraum auftreten:

- Zu viele Backend-Prozesse laufen gleichzeitig auf der CPU. Diese Prozesse können parallele Arbeiter sein.
- Abfragen funktionieren suboptimal, da sie eine große Anzahl von Puffern benötigen.

Corner Cases

Wenn sich herausstellt, dass keine der wahrscheinlichen Ursachen tatsächliche Ursachen ist, können folgende Situationen auftreten:

- Die CPU tauscht Prozesse ein- und aus.
- Die CPU verwaltet möglicherweise Seitentabelleneinträge, wenn die Funktion Huge Pages deaktiviert wurde. Die Speicherverwaltungsfunktion ist standardmäßig für alle DB-Instance-Klassen aktiviert, außer Micro, Small und Medium. Weitere Informationen finden Sie unter Riesige Seiten RDS für Postgre SQL .

Aktionen

Wenn das CPU-Wait-Ereignis die Datenbankaktivität dominiert, weist dies nicht unbedingt auf ein Leistungsproblem hin. Reagieren Sie auf dieses Ereignis nur, wenn sich die Leistung verschlechtert.

Themen

- Untersuchen Sie, ob die Datenbank den CPU-Anstieg verursacht
- Bestimmen Sie, ob die Anzahl der Verbindungen gestiegen ist
- Reagieren auf Workload-Änderungen

Untersuchen Sie, ob die Datenbank den CPU-Anstieg verursacht

Untersuchen Sie die `os.cpuUtilization.nice.avg`-Metrik in Performance Insights. Wenn dieser Wert weit unter der CPU-Auslastung liegt, tragen Nicht-Datenbankprozesse den Hauptbeitrag zur CPU bei.

Bestimmen Sie, ob die Anzahl der Verbindungen gestiegen ist

Untersuchen Sie die Metrik `DatabaseConnections` in Amazon CloudWatch. Ihre Aktion hängt davon ab, ob die Zahl während des Zeitraums erhöhter CPU-Warteereignisse erhöht oder gesunken ist.

Die Verbindungen nahmen zu

Wenn die Anzahl der Verbindungen gestiegen ist, vergleichen Sie die Anzahl der Backend-Prozesse, die CPU verbrauchen, mit der Anzahl der vCPUs. Die folgenden Szenarien sind möglich:

- Die Anzahl der Backend-Prozesse, die CPU verbrauchen, ist geringer als die Anzahl der vCPUs.

  In diesem Fall ist die Anzahl der Verbindungen kein Problem. Möglicherweise versuchen Sie jedoch weiterhin, die CPU-Auslastung zu reduzieren.
- Die Anzahl der Backend-Prozesse, die CPU verbrauchen, ist größer als die Anzahl der vCPUs.

  Ziehen Sie in diesem Fall die folgenden Optionen in Betracht:
  - Verringern Sie die Anzahl der Backend-Prozesse, die mit Ihrer Datenbank verbunden sind. Implementieren Sie beispielsweise eine Verbindungs-Pooling-Lösung wie RDS Proxy. Weitere Informationen hierzu finden Sie unter Amazon RDS-Proxy .
  - Aktualisieren Sie Ihre Instance-Größe, um eine höhere Anzahl von vCPUs zu erhalten.
  - Leiten Sie ggf. einige schreibgeschützte Workloads auf Reader-Knoten um.

Die Verbindungen haben nicht zugenommen

Untersuchen Sie die `blks_hit`-Metriken in Performance Insights. Suchen Sie nach einer Korrelation zwischen einem Anstieg von `blks_hit` und der CPU-Auslastung. Die folgenden Szenarien sind möglich:

- CPU-Auslastung und `blks_hit` sind korreliert.

  Suchen Sie in diesem Fall die wichtigsten SQL-Anweisungen, die mit der CPU-Auslastung verknüpft sind, und suchen Sie nach Planänderungen. Sie können eine der folgenden Techniken verwenden:

  - Erklären Sie die Pläne manuell und vergleichen Sie sie mit dem erwarteten Ausführungsplan.
  - Achten Sie auf eine Zunahme der Blocktreffer pro Sekunde und lokalen Blocktreffern pro Sekunde. Wählen Sie im Abschnitt Top-SQL des Performance Insights-Dashboards Einstellungen aus.
- CPU-Auslastung und `blks_hit` sind nicht korreliert.

  Stellen Sie in diesem Fall fest, ob einer der folgenden Fälle auftritt:

  - Die Anwendung stellt schnell eine Verbindung zur Datenbank her und trennt sie von dieser.

    Diagnostizieren Sie dieses Verhalten, indem Sie `log_connections` und `log_disconnections` aktivieren und dann die PostgreSQL-Protokolle analysieren. Ziehen Sie in Erwägung, den `pgbadger`-Protokoll-Analyzer zu verwenden. Weitere Informationen finden Sie unter https://github.com/darold/pgbadger.
  - Das Betriebssystem ist überlastet.

    In diesem Fall zeigt Performance Insights, dass Backend-Prozesse länger CPU verbrauchen als gewöhnlich. Suchen Sie in den Metriken von Performance Insights `os.cpuUtilization` oder der Metrik CloudWatch `CPUUtilization` nach Beweisen. Wenn das Betriebssystem überlastet ist, sehen Sie sich Enhanced Monitoring-Metriken an, um eine weitere Diagnose zu erhalten. Schauen Sie sich insbesondere die Prozessliste und den Prozentsatz der CPU an, die von jedem Prozess verbraucht wird.
  - Top SQL-Anweisungen verbrauchen zu viel CPU.

    Untersuchen Sie Anweisungen, die mit der CPU-Auslastung verknüpft sind, um festzustellen, ob sie weniger CPU verbrauchen können. Führen Sie einen `EXPLAIN`-Befehl aus und konzentrieren Sie sich auf die Planknoten, die die größte Auswirkung haben. Erwägen Sie, einen Visualizer

für PostgreSQL-Ausführungspläne zu verwenden. Um dieses Tool auszuprobieren, siehe http://explain.dalibo.com/.

Reagieren auf Workload-Änderungen

Wenn sich Ihre Workload geändert hat, suchen Sie nach folgenden Arten von Änderungen:

Neue Anfragen

Überprüfen Sie, ob die neuen Abfragen erwartet werden. Stellen Sie in diesem Fall sicher, dass ihre Ausführungspläne und die Anzahl der Ausführungen pro Sekunde erwartet werden.

Eine Zunahme der Größe des Datensatzes

Bestimmen Sie, ob die Partitionierung, falls sie noch nicht implementiert ist, helfen könnte. Diese Strategie könnte die Anzahl der Seiten verringern, die eine Abfrage abrufen muss.

Indexpflege oder -erstellung

Überprüfen Sie, ob der Zeitplan für die Wartung erwartet wird. Eine bewährte Methode besteht darin, Wartungsarbeiten außerhalb der Hauptverkehrszeiten zu planen.

Neue Funktionen

Überprüfen Sie, ob diese Funktionen während des Tests erwartungsgemäß funktionieren. Überprüfen Sie insbesondere, ob die Anzahl der Ausführungen pro Sekunde erwartet wird.

Neue Betreiber

Überprüfen Sie, ob sie während des Tests erwartungsgemäß funktionieren.

Eine Zunahme der laufenden parallelen Abfragen

Stellen Sie fest, ob eine der folgenden Situationen aufgetreten ist:

- Die beteiligten Relationen oder Indizes sind plötzlich so groß geworden, dass sie sich deutlich von `min_parallel_table_scan_size` oder `min_parallel_index_scan_size` unterscheiden.
- Die letzten Änderungen wurden an `parallel_setup_cost` oder `parallel_tuple_cost` vorgenommen.
- Die letzten Änderungen wurden an `max_parallel_workers` oder `max_parallel_workers_per_gather` vorgenommen.

io:BuffileRead und io:BuffileWrite

Die Ereignisse `IO:BufFileRead` und `IO:BufFileWrite` treten auf, wenn RDS für PostgreSQL temporäre Dateien erstellt. Wenn Vorgänge mehr Arbeitsspeicher benötigen, als die derzeit definierten Arbeitsspeicherparameter definieren, schreiben sie temporäre Daten in persistenten Speicher. Dieser Vorgang wird manchmal als „Verschütten auf die Festplatte“ bezeichnet.

Themen

- Unterstützte Engine-Versionen
- Context
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Context

`IO:BufFileRead` und `IO:BufFileWrite` beziehen sich auf den Arbeitsspeicherbereich und den Wartungsarbeitsspeicherbereich. Weitere Informationen zu diesen lokalen Speicherbereichen finden Sie unter Ressourcennutzung in der PostgreSQL-Dokumentation.

Der Standardwert für `work_mem` ist 4 MB. Wenn eine Sitzung parallel Vorgänge ausführt, verwendet jeder Worker, der die Parallelität bearbeitet, 4 MB Speicher. Stellen Sie `work_mem` daher sorgfältig ein. Wenn Sie den Wert zu stark erhöhen, verbraucht eine Datenbank mit vielen Sitzungen möglicherweise zu viel Speicher. Wenn Sie den Wert zu niedrig festlegen, erstellt RDS für PostgreSQL temporäre Dateien im lokalen Speicher. Die Festplatten-I/O für diese temporären Dateien kann die Leistung verringern.

Wenn Sie die folgende Ereignisfolge beobachten, generiert Ihre Datenbank möglicherweise temporäre Dateien:

1. Plötzlicher und starker Rückgang der Verfügbarkeit
2. Schnelle Erholung für den freien Speicherplatz

Möglicherweise sehen Sie auch ein „Kettensäge“ -Muster. Dieses Muster kann darauf hinweisen, dass Ihre Datenbank ständig kleine Dateien erstellt.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Im Allgemeinen werden diese Warteereignisse durch Vorgänge verursacht, die mehr Speicher verbrauchen, als die Parameter `work_mem` oder `maintenance_work_mem` zuweisen. Um dies zu kompensieren, schreiben die Vorgänge in temporäre Dateien. Häufige Ursachen für die `IO:BufFileRead`- und `IO:BufFileWrite`-Ereignisse sind die folgenden:

Abfragen, die mehr Speicher benötigen als im Arbeitsspeicherbereich vorhanden

Abfragen mit den folgenden Merkmalen verwenden den Arbeitsspeicherbereich:

- Hash-Verknüpfungen
- `ORDER BY`-Klausel
- `GROUP BY`-Klausel
- `DISTINCT`
- Fensterfunktionen
- `CREATE TABLE AS SELECT`
- Aktualisierung der materialisierten Ansicht

Anweisungen, die mehr Speicher benötigen als im Arbeitsspeicherbereich für Wartungsarbeiten vorhanden

Die folgenden Anweisungen verwenden den Arbeitsspeicherbereich für Wartungsarbeiten:

- `CREATE INDEX`
- `CLUSTER`

Aktionen

Abhängig von den Ursachen Ihres Wait-Ereignisses empfehlen wir verschiedene Aktionen.

Themen

- Identifizieren Sie das Problem
- Untersuchen Sie Ihre Join-Anfragen
- Überprüfen Sie Ihre ORDER BY- und GROUP BY Anfragen
- Verwenden Sie den DISTINCT-Vorgang nicht

- Erwägen Sie, Fensterfunktionen anstelle von GROUP-BY-Funktionen zu verwenden
- Untersuchen Sie materialisierte Ansichten und CTAS-Aussagen
- Verwenden von pg_repack beim Neuerstellen von Indizes
- Erhöhen Sie maintenance work_mem, wenn Sie Tabellen clustern
- Optimieren Sie den Speicher, um io:BuffileRead und io:BuffileWrite zu verhindern

Identifizieren Sie das Problem

Nehmen Sie an, dass Performance Insights nicht aktiviert ist und Sie vermuten, dass `IO:BufFileRead` und `IO:BufFileWrite` häufiger als normal auftreten. Um die Ursache des Problems zu ermitteln, können Sie den `log_temp_files`-Parameter so festlegen, dass alle Abfragen protokolliert werden, die mehr als den angegebenen Schwellenwert an temporären Dateien in KB generieren. `log_temp_files` ist standardmäßig auf `-1` festgelegt, wodurch diese Protokollierungsfunktion deaktiviert wird. Wenn Sie diesen Parameter auf `0` einstellen, protokolliert RDS für PostgreSQL alle temporären Dateien. Wenn der Wert `1024` ist, protokolliert RDS für PostgreSQL alle Abfragen, die temporäre Dateien erzeugen, die größer als 1 MB sind. Weitere Informationen zu `log_temp_files` finden Sie unter Fehlerberichte und -protokollierung in der PostgreSQL-Dokumentation.

Untersuchen Sie Ihre Join-Anfragen

Es ist wahrscheinlich, dass Ihre Abfrage Joins verwendet. Die folgende Abfrage verbindet beispielsweise vier Tabellen.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = order.customer_id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

Eine mögliche Ursache für Spitzen bei der Verwendung temporärer Dateien ist ein Problem in der Abfrage selbst. Beispielsweise filtert eine defekte Klausel die Joins möglicherweise nicht richtig. Betrachten Sie den zweiten inneren Join im folgenden Beispiel.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = customer.id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

Die obige Abfrage verknüpft fälschlicherweise `customer.id` mit `customer.id`, wodurch zwischen jedem Kunden und jeder Bestellung ein kartesisches Produkt generiert wird. Diese Art des versehentlichen Joins generiert große temporäre Dateien. Abhängig von der Größe der Tabellen kann eine kartesische Abfrage sogar Speicher füllen. Wenn die folgenden Bedingungen erfüllt sind, hat Ihre Anwendung möglicherweise kartesische Joins:

- Sie sehen einen großen, starken Rückgang der Speicherverfügbarkeit, gefolgt von einer schnellen Wiederherstellung.
- Es werden keine Indizes erstellt.
- Es werden keine `CREATE TABLE FROM SELECT`-Anweisungen ausgegeben.
- Es werden keine materialisierten Ansichten aktualisiert.

Um festzustellen, ob die Tabellen mit den richtigen Schlüsseln verbunden werden, überprüfen Sie Ihre Abfrage- und objektrelationalen Mapping-Anweisungen. Denken Sie daran, dass bestimmte Abfragen Ihrer Anwendung nicht ständig aufgerufen werden und einige Abfragen dynamisch generiert werden.

Überprüfen Sie Ihre ORDER BY- und GROUP BY Anfragen

In einigen Fällen kann eine `ORDER BY`-Klausel zu übermäßig vielen temporären Dateien führen. Berücksichtigen Sie die folgenden Hinweise:

- Schließen Sie Spalten nur dann in eine `ORDER BY`-Klausel ein, wenn sie sortiert werden müssen. Diese Richtlinie ist besonders wichtig für Abfragen, die Tausende von Zeilen zurückgeben und viele Spalten in der `ORDER BY`-Klausel angeben.
- Ziehen Sie in Betracht, Indizes zu erstellen, um `ORDER BY`-Klauseln zu beschleunigen, wenn sie mit Spalten übereinstimmen, die dieselbe aufsteigende oder absteigende Reihenfolge aufweisen.

Partielle Indizes sind vorzuziehen, da sie kleiner sind. Kleinere Indizes werden schneller gelesen und durchquert.

- Wenn Sie Indizes für Spalten erstellen, die Nullwerte akzeptieren können, überlegen Sie, ob die Nullwerte am Ende oder am Anfang der Indizes gespeichert werden sollen.

  Reduzieren Sie nach Möglichkeit die Anzahl der Zeilen, die sortiert werden müssen, indem Sie die Ergebnismenge filtern. Wenn Sie `WITH`-Klausel-Anweisungen oder Unterabfragen verwenden, denken Sie daran, dass eine innere Abfrage eine Ergebnismenge generiert und an die äußere Abfrage übergibt. Je mehr Zeilen eine Abfrage herausfiltern kann, desto weniger muss die Abfrage erledigen.
- Wenn Sie nicht die vollständige Ergebnismenge abrufen müssen, verwenden Sie die `LIMIT`-Klausel. Wenn Sie beispielsweise nur die obersten fünf Zeilen benötigen, generiert eine Abfrage mit der `LIMIT`-Klausel keine Ergebnisse. Auf diese Weise benötigt die Abfrage weniger Speicher und temporäre Dateien.

Eine Abfrage, die eine `GROUP BY`-Klausel verwendet, kann auch temporäre Dateien erfordern. `GROUP BY`-Abfragen fassen Werte mithilfe von Funktionen wie den folgenden zusammen:

- `COUNT`
- `AVG`
- `MIN`
- `MAX`
- `SUM`
- `STDDEV`

Befolgen Sie zum Optimieren von `GROUP BY`-Abfragen die Empfehlungen für `ORDER BY`-Abfragen.

Verwenden Sie den DISTINCT-Vorgang nicht

Vermeiden Sie nach Möglichkeit die Verwendung des `DISTINCT`-Vorgangs, um doppelte Zeilen zu entfernen. Je mehr unnötige und doppelte Zeilen Ihre Abfrage zurückgibt, desto teurer wird der `DISTINCT`-Vorgang. Fügen Sie nach Möglichkeit Filter in der `WHERE`-Klausel hinzu, auch wenn Sie dieselben Filter für verschiedene Tabellen verwenden. Das Filtern der Abfrage und der korrekte Beitritt verbessern Ihre Leistung und reduziert den Ressourcennutzung. Es verhindert auch falsche Berichte und Ergebnisse.

Wenn Sie DISTINCT für mehrere Zeilen derselben Tabelle verwenden müssen, sollten Sie einen zusammengesetzten Index erstellen. Das Gruppieren mehrerer Spalten in einem Index kann die Zeit zum Auswerten verschiedener Zeilen verbessern. Wenn Sie Version 10 von RDS für PostgreSQL oder höher verwenden, können Sie außerdem mithilfe des CREATE STATISTICS-Befehls Statistiken zwischen mehreren Spalten korrelieren.

Erwägen Sie, Fensterfunktionen anstelle von GROUP-BY-Funktionen zu verwenden

Mit GROUP BY ändern Sie die Ergebnismenge und rufen dann das aggregierte Ergebnis ab. Mithilfe von Fensterfunktionen aggregieren Sie Daten, ohne die Ergebnismenge zu ändern. Eine Fensterfunktion verwendet die OVER-Klausel, um Berechnungen über die von der Abfrage definierten Mengen durchzuführen und eine Zeile mit einer anderen zu korrelieren. Sie können alle GROUP BY-Funktionen in Fensterfunktionen verwenden, aber auch Funktionen wie die folgenden verwenden:

- RANK
- ARRAY_AGG
- ROW_NUMBER
- LAG
- LEAD

Um die Anzahl der temporären Dateien zu minimieren, die von einer Fensterfunktion generiert werden, entfernen Sie Duplikationen für dieselbe Ergebnismenge, wenn Sie zwei verschiedene Aggregationen benötigen. Betrachten Sie folgende Abfrage.

```
SELECT sum(salary) OVER (PARTITION BY dept ORDER BY salary DESC) as sum_salary
     , avg(salary) OVER (PARTITION BY dept ORDER BY salary ASC) as avg_salary
  FROM empsalary;
```

Sie können die Abfrage mit der WINDOW-Klausel wie folgt umschreiben.

```
SELECT sum(salary) OVER w as sum_salary
         , avg(salary) OVER w as_avg_salary
    FROM empsalary
  WINDOW w AS (PARTITION BY dept ORDER BY salary DESC);
```

Standardmäßig konsolidiert der Ausführungsplaner von RDS für PostgreSQL ähnliche Knoten, sodass keine Vorgänge dupliziert werden. Durch die Verwendung einer expliziten Deklaration für

den Fensterblock können Sie die Abfrage jedoch einfacher pflegen. Sie können die Leistung auch verbessern, indem Sie Doppelarbeit verhindern.

Untersuchen Sie materialisierte Ansichten und CTAS-Aussagen

Wenn eine materialisierte Ansicht aktualisiert wird, wird eine Abfrage ausgeführt. Diese Abfrage kann einen Vorgang wie `GROUP BY`, `ORDER BY` oder `DISTINCT` enthalten. Während einer Aktualisierung können Sie eine große Anzahl temporärer Dateien und die Warteereignisse `IO:BufFileWrite` und `IO:BufFileRead` beobachten. Wenn Sie eine Tabelle basierend auf einer `SELECT`-Anweisung erstellen, führt die `CREATE TABLE`-Anweisung eine Abfrage aus. Um die benötigten temporären Dateien zu reduzieren, optimieren Sie die Abfrage.

Verwenden von pg_repack beim Neuerstellen von Indizes

Wenn Sie einen Index erstellen, ordnet die Engine die Ergebnismenge an. Wenn Tabellen größer werden und die Werte in der indizierten Spalte vielfältiger werden, benötigen die temporären Dateien mehr Speicherplatz. In den meisten Fällen können Sie die Erstellung temporärer Dateien für große Tabellen nicht verhindern, ohne den Speicherbereich für Wartungsarbeiten zu ändern. Weitere Informationen zu `maintenance_work_mem` finden Sie unter https://www.postgresql.org/docs/current/runtime-config-resource.html in der PostgreSQL-Dokumentation.

Eine mögliche Problemumgehung beim Neuerstellen eines großen Index besteht darin, die pg_repack-Erweiterung zu verwenden. Weitere Informationen finden Sie unter Reorganisieren von Tabellen in PostgreSQL-Datenbanken mit minimalen Sperren in der pg_repack-Dokumentation. Informationen zum Einrichten der Erweiterung in Ihrer DB-Instance von RDS for PostgreSQL finden Sie unter Reduzieren von überflüssigen Daten in Tabellen und Indizes mit der Erweiterung pg_repack.

Erhöhen Sie maintenance work_mem, wenn Sie Tabellen clustern

Der Befehl `CLUSTER` gruppiert die durch table_name angegebene Tabelle basierend auf einem vorhandenen Index, der durch index_name angegeben wird. RDS für PostgreSQL erstellt die Tabelle physisch so neu, dass sie der Reihenfolge eines bestimmten Indexes entspricht.

Als magnetische Speicherung vorherrschend war, war Clustering üblich, da der Speicherdurchsatz begrenzt war. Jetzt, da SSD-basierter Speicher üblich ist, ist Clustering weniger beliebt. Wenn Sie jedoch Tabellen clustern, können Sie die Leistung je nach Tabellengröße, Index, Abfrage usw. immer noch geringfügig steigern.

Wenn Sie den Befehl `CLUSTER` ausführen und die Warteereignisse `IO:BufFileWrite` und `IO:BufFileRead` beobachten, stimmen Sie `maintenance_work_mem` ab. Erhöhen Sie die

Speichergröße auf einen ziemlich großen Betrag. Ein hoher Wert bedeutet, dass die Engine mehr Speicher für den Clustering-Vorgang verwenden kann.

Optimieren Sie den Speicher, um io:BuffileRead und io:BuffileWrite zu verhindern

In einigen Situationen müssen Sie den Speicher optimieren. Ihr Ziel ist es, mithilfe der entsprechenden Parameter den Arbeitsspeicher in den folgenden Verbrauchsbereichen wie folgt auszugleichen.

- Der `work_mem`-Wert
- Der Speicher, der nach Abzug des Werts `shared_buffers` verbleibt
- Die maximale Anzahl geöffneter und verwendeter Verbindungen, die durch `max_connections` begrenzt ist

Weitere Informationen zum Optimieren des Arbeitsspeichers finden Sie unter Ressourcennutzung in der PostgreSQL-Dokumentation.

Erhöhen Sie die Größe des Arbeitsspeicherbereichs

In einigen Situationen besteht Ihre einzige Möglichkeit darin, den von Ihrer Sitzung verwendeten Speicher zu erhöhen. Wenn Ihre Abfragen richtig geschrieben sind und die richtigen Schlüssel für Joins verwenden, sollten Sie den `work_mem`-Wert erhöhen.

Um herauszufinden, wie viele temporäre Dateien eine Abfrage generiert, setzen Sie `log_temp_files` auf `0`. Wenn Sie den `work_mem`-Wert auf den in den Protokollen angegebenen Höchstwert erhöhen, verhindern Sie, dass die Abfrage temporäre Dateien generiert. `work_mem` legt jedoch das Maximum pro Planknoten für jede Verbindung oder jeden parallelen Worker fest. Wenn die Datenbank über 5.000 Verbindungen verfügt und jede 256 MiB-Speicher verwendet, benötigt die Engine 1,2 TiB RAM. Daher kann es sein, dass Ihrer Instance der Speicher knapp wird.

Reservieren Sie ausreichend Speicher für den freigegebenen Pufferpool

Ihre Datenbank verwendet Speicherbereiche wie den freigegebenen Pufferpool, nicht nur den Arbeitsspeicherbereich. Berücksichtigen Sie die Anforderungen dieser zusätzlichen Speicherbereiche, bevor Sie `work_mem` erhöhen.

Angenommen, Ihre Instance-Klasse von RDS für PostgreSQL ist db.r5.2xlarge. Diese Klasse hat 64 GiB Speicher. Standardmäßig sind 25 Prozent des Arbeitsspeichers für den freigegebenen Pufferpool

reserviert. Nachdem Sie den dem Shared Memory-Bereich zugewiesenen Betrag abgezogen haben, bleiben 16.384 MB übrig. Weisen Sie den verbleibenden Speicher nicht ausschließlich dem Arbeitsspeicherbereich zu, da das Betriebssystem und die Engine ebenfalls Speicher benötigen.

Der Speicher, den Sie `work_mem` zuordnen können, hängt von der Instance-Klasse ab. Wenn Sie eine größere Instance-Klasse verwenden, ist mehr Speicher verfügbar. Im vorhergehenden Beispiel können Sie jedoch nicht mehr als 16 GiB verwenden. Andernfalls ist Ihre Instance nicht verfügbar, wenn ihr der Speicher ausgeht. Um die Instance aus dem nicht verfügbaren Status wiederherzustellen, werden die Automatisierungsdienste von RDS für PostgreSQL automatisch neu gestartet.

Verwalten der Anzahl der Verbindungen

Angenommen, Ihre Datenbank-Instance hat 5.000 gleichzeitige Verbindungen. Jede Verbindung verwendet mindestens 4 MB `work_mem`. Der hohe Speicherverbrauch der Verbindungen dürfte die Leistung beeinträchtigen. Als Reaktion darauf haben Sie die folgenden Optionen:

- Aktualisieren Sie auf eine größere Instance-Klasse.
- Verringern Sie die Anzahl gleichzeitiger Datenbankverbindungen mit einem Verbindungsproxy oder Pooler.

Berücksichtigen Sie bei Proxys Amazon RDS Proxy, PGBouncer oder einen auf Ihrer Anwendung basierenden Verbindungspooler. Diese Lösung lindert die CPU-Last. Es reduziert auch das Risiko, wenn alle Verbindungen den Arbeitsspeicherbereich benötigen. Wenn weniger Datenbankverbindungen vorhanden sind, können Sie den Wert von `work_mem` erhöhen. Auf diese Weise reduzieren Sie das Auftreten der `IO:BufFileRead`- und `IO:BufFileWrite`-Warteereignisse. Auch die Abfragen, die auf den Arbeitsspeicherbereich warten, beschleunigen sich erheblich.

E/A:DataFileRead

Das `IO:DataFileRead`-Ereignis tritt auf, wenn eine Verbindung darauf wartet, dass ein Backend-Prozess eine erforderliche Seite aus dem Speicher liest, da die Seite nicht im gemeinsam genutzten Speicher verfügbar ist.

Themen

- Unterstützte Engine-Versionen
- Kontext

- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Kontext

Alle Abfragen und Datenmanipulationsvorgänge (DML) greifen auf Seiten im Pufferpool zu. Zu den Anweisungen, die Lesevorgänge auslösen können, gehören `SELECT`, `UPDATE` und `DELETE`. Ein `UPDATE` kann beispielsweise Seiten aus Tabellen oder Indizes lesen. Wenn sich die angeforderte oder aktualisierte Seite nicht im gemeinsam genutzten Pufferpool befindet, kann dieser Lesevorgang zum `IO:DataFileRead`-Ereignis führen.

Da der gemeinsame Pufferpool endlich ist, kann er sich füllen. In diesem Fall zwingen Anfragen nach Seiten, die sich nicht im Speicher befinden, die Datenbank dazu, Blöcke von der Festplatte zu lesen. Wenn das `IO:DataFileRead`-Ereignis häufig auftritt, ist Ihr gemeinsam genutzter Pufferpool möglicherweise zu klein für Ihre Workload. Dieses Problem ist bei `SELECT`-Abfragen akut, die eine große Anzahl von Zeilen lesen, die nicht in den Pufferpool passen. Weitere Informationen zum Pufferpool finden Sie unter Ressourcenbnutzung in der PostgreSQL-Dokumentation.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Häufige Ursachen für das Ereignis `IO:DataFileRead` sind die folgenden:

Verbindungsspitzen

Möglicherweise finden Sie mehrere Verbindungen, die die gleiche Anzahl von IO:DataFileRead wait-Ereignissen generieren. In diesem Fall kann eine Spitze (plötzlicher und starker Anstieg) bei `IO:DataFileRead`-Ereignissen auftreten.

SELECT- und DML-Anweisungen, die sequentielle Scans durchführen

Ihre Anwendung führt möglicherweise einen neuen Vorgang aus. Oder ein vorhandener Vorgang könnte sich aufgrund eines neuen Ausführungsplans ändern. Suchen Sie in solchen Fällen nach Tabellen (insbesondere großen Tabellen), die einen größeren `seq_scan`-Wert haben. Finden Sie sie, indem Sie `pg_stat_user_tables` abfragen. Verwenden Sie die Erweiterung `pg_stat_statements`, um Abfragen zu verfolgen, die mehr Lesevorgänge generieren.

CTAS und CREATE INDEX für große Datensätze

Ein CTAS ist eine `CREATE TABLE AS SELECT`-Anweisung. Wenn Sie ein CTAS mit einem großen Datensatz als Quelle ausführen oder einen Index für eine große Tabelle erstellen, kann das `IO:DataFileRead`-Ereignis auftreten. Wenn Sie einen Index erstellen, muss die Datenbank möglicherweise das gesamte Objekt mithilfe eines sequentiellen Scans lesen. Ein CTAS generiert `IO:DataFile`-Reads, wenn Seiten nicht im Speicher sind.

Mehrere Vakuumarbeiter laufen gleichzeitig

Vakuumarbeiter können manuell oder automatisch ausgelöst werden. Wir empfehlen, eine aggressive Vakuumstrategie zu verabschieden. Wenn eine Tabelle jedoch viele aktualisierte oder gelöschte Zeilen enthält, erhöhen sich die `IO:DataFileRead`-Wartezeiten. Nachdem der Raum zurückgewonnen wurde, nimmt die für `IO:DataFileRead` aufgewendete Vakuumzeit ab.

Aufnahme großer Datenmengen

Wenn Ihre Anwendung große Datenmengen aufnimmt, können `ANALYZE`-Vorgänge häufiger auftreten. Der `ANALYZE`-Prozess kann durch einen Autovacuum Launcher ausgelöst oder manuell aufgerufen werden.

Der `ANALYZE`-Vorgang liest eine Teilmenge der Tabelle. Die Anzahl der zu scannenden Seiten wird berechnet, indem 30 mit dem `default_statistics_target`-Wert multipliziert wird. Weitere Informationen finden Sie in der PostgreSQL-Dokumentation. Der Parameter `default_statistics_target` akzeptiert Werte zwischen 1 und 10.000, wobei der Standardwert 100 ist.

Hungertod

Wenn Netzwerkbandbreite oder CPU der Instance verbraucht werden, kann das `IO:DataFileRead`-Ereignis häufiger auftreten.

Aktionen

Abhängig von den Ursachen Ihres Warteereignisses empfehlen wir verschiedene Aktionen.

Themen

- Überprüfen Sie Prädikatfilter auf Abfragen, die Wartezeiten generieren
- Minimieren Sie die Auswirkungen von Wartungsvorgängen
- Reagieren Sie auf eine hohe Anzahl von Verbindungen

Überprüfen Sie Prädikatfilter auf Abfragen, die Wartezeiten generieren

Angenommen, Sie identifizieren bestimmte Abfragen, die `IO:DataFileRead`-Warteereignisse generieren. Sie können sie mit den folgenden Techniken identifizieren:

- Performance Insights
- Katalogansichten wie die der Erweiterung `pg_stat_statements`
- Die Katalogansicht `pg_stat_all_tables`, wenn sie periodisch eine erhöhte Anzahl von physischen Lesevorgängen anzeigt
- Die `pg_statio_all_tables`-Ansicht, wenn sie zeigt, dass `_read`-Zähler steigen

Wir empfehlen Ihnen, zu bestimmen, welche Filter im Prädikat (`WHERE`-Klausel) dieser Abfragen verwendet werden. Befolgen Sie diese Richtlinien:

- Führen Sie den Befehl `EXPLAIN` aus. Geben Sie in der Ausgabe an, welche Arten von Scans verwendet werden. Ein sequentieller Scan weist nicht zwangsläufig ein Problem an. Abfragen, die sequenzielle Scans verwenden, erzeugen im Vergleich zu Abfragen, die Filter verwenden, natürlich mehr `IO:DataFileRead`-Ereignisse.

  Finden Sie heraus, ob die in der `WHERE`-Klausel aufgeführte Spalte indiziert ist. Wenn nicht, erwägen Sie, einen Index für diese Spalte zu erstellen. Dieser Ansatz vermeidet die sequentiellen Scans und reduziert die `IO:DataFileRead`-Ereignisse. Wenn eine Abfrage restriktive Filter enthält und immer noch sequenzielle Scans erzeugt, bewerten Sie, ob die richtigen Indizes verwendet werden.
- Finden Sie heraus, ob die Abfrage auf eine sehr große Tabelle zugreift. In einigen Fällen kann das Partitionieren einer Tabelle die Leistung verbessern, sodass die Abfrage nur notwendige Partitionen lesen kann.
- Untersuchen Sie die Kardinalität (Gesamtzahl der Zeilen) Ihrer Join-Vorgänge. Beachten Sie, wie restriktiv die Werte sind, die Sie den Filtern für Ihre `WHERE`-Klausel übergeben. Wenn möglich, stimmen Sie Ihre Abfrage ein, um die Anzahl der Zeilen zu reduzieren, die in jedem Schritt des Plans übergeben werden.

Minimieren Sie die Auswirkungen von Wartungsvorgängen

Wartungsvorgänge wie `VACUUM` und `ANALYZE` sind wichtig. Wir empfehlen, sie nicht zu deaktivieren, da Sie `IO:DataFileRead`-Warteereignisse im Zusammenhang mit diesen Wartungsvorgängen finden. Die folgenden Ansätze können die Auswirkungen dieser Vorgänge minimieren:

- Führen Sie Wartungsvorgänge während außerhalb der Hauptverkehrszeiten manuell aus. Diese Technik verhindert, dass die Datenbank den Schwellenwert für automatische Vorgänge erreicht.
- Erwägen Sie bei sehr großen Tabellen, die Tabelle zu partitionieren. Diese Technik reduziert den Overhead von Wartungsvorgängen. Die Datenbank greift nur auf die Partitionen zu, die gewartet werden müssen.
- Wenn Sie große Datenmengen aufnehmen, sollten Sie die Funktion zur automatischen Analyse deaktivieren.

Die Auto-Vakuum-Funktion wird automatisch für eine Tabelle ausgelöst, wenn die folgende Formel zutrifft.

```
pg_stat_user_tables.n_dead_tup > (pg_class.reltuples x autovacuum_vacuum_scale_factor)
 + autovacuum_vacuum_threshold
```

Die Ansicht `pg_stat_user_tables` und der Katalog `pg_class` haben mehrere Zeilen. Eine Zeile kann einer Zeile in Ihrer Tabelle entsprechen. Diese Formel geht davon aus, dass die `reltuples` für eine bestimmte Tabelle stehen. Die Parameter `autovacuum_vacuum_scale_factor` (standardmäßig 0.20) und `autovacuum_vacuum_threshold` (standardmäßig 50 Tupel) werden normalerweise global für die gesamte Instance gesetzt. Sie können jedoch verschiedene Werte für eine bestimmte Tabelle festlegen.

Themen

- Suchen nach Tabellen, die unnötigerweise Speicherplatz belegen
- Suchen nach Indizes, die unnötigerweise Speicherplatz belegen
- Finden Sie Tabellen, die für die automatische Vakuumierung berechtigt sind

Suchen nach Tabellen, die unnötigerweise Speicherplatz belegen

Um Tabellen zu finden, die unnötig Speicherplatz beanspruchen, können Sie Funktionen der `pgstattuple`-Erweiterung von PostgreSQL verwenden. Diese Erweiterung (Modul) ist standardmäßig auf allen DB-Instances von RDS für PostgreSQL verfügbar und kann mit dem folgenden Befehl auf der Instance instanziiert werden.

```
CREATE EXTENSION pgstattuple;
```

Weitere Informationen zu dieser Erweiterung finden Sie unter pgstattuple in der PostgreSQL-Dokumentation.

Sie können in Ihrer Anwendung nach einer Überlastung von Tabellen und Indizes suchen. Weitere Informationen finden Sie unter Diagnostizieren einer Überlastung von Tabellen und Indizes.

Suchen nach Indizes, die unnötigerweise Speicherplatz belegen

Um aufgeblähte Indizes zu finden und abzuschätzen, wie viel Speicherplatz in den Tabellen, für die Sie Leserechte haben, unnötig beansprucht wird, können Sie die folgende Abfrage ausführen.

```
-- WARNING: rows with is_na = 't' are known to have bad statistics ("name" type is not
 supported).
-- This query is compatible with PostgreSQL 8.2 and later.

SELECT current_database(), nspname AS schemaname, tblname, idxname,
 bs*(relpages)::bigint AS real_size,
  bs*(relpages-est_pages)::bigint AS extra_size,
  100 * (relpages-est_pages)::float / relpages AS extra_ratio,
  fillfactor, bs*(relpages-est_pages_ff) AS bloat_size,
  100 * (relpages-est_pages_ff)::float / relpages AS bloat_ratio,
  is_na
  -- , 100-(sub.pst).avg_leaf_density, est_pages, index_tuple_hdr_bm,
  -- maxalign, pagehdr, nulldatawidth, nulldatahdrwidth, sub.reltuples, sub.relpages
  -- (DEBUG INFO)
FROM (
  SELECT coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)/(4+nulldatahdrwidth)::float)), 0
       -- ItemIdData size + computed avg size of a tuple (nulldatahdrwidth)
    ) AS est_pages,
    coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)*fillfactor/
(100*(4+nulldatahdrwidth)::float))), 0
    ) AS est_pages_ff,
    bs, nspname, table_oid, tblname, idxname, relpages, fillfactor, is_na
    -- , stattuple.pgstatindex(quote_ident(nspname)||'.'||quote_ident(idxname)) AS
 pst,
    -- index_tuple_hdr_bm, maxalign, pagehdr, nulldatawidth, nulldatahdrwidth,
 reltuples
    -- (DEBUG INFO)
  FROM (
    SELECT maxalign, bs, nspname, tblname, idxname, reltuples, relpages, relam,
 table_oid, fillfactor,
```

```
     ( index_tuple_hdr_bm +
         maxalign - CASE -- Add padding to the index tuple header to align on MAXALIGN
           WHEN index_tuple_hdr_bm%maxalign = 0 THEN maxalign
           ELSE index_tuple_hdr_bm%maxalign
         END
       + nulldatawidth + maxalign - CASE -- Add padding to the data to align on
MAXALIGN
           WHEN nulldatawidth = 0 THEN 0
           WHEN nulldatawidth::integer%maxalign = 0 THEN maxalign
           ELSE nulldatawidth::integer%maxalign
         END
     )::numeric AS nulldatahdrwidth, pagehdr, pageopqdata, is_na
     -- , index_tuple_hdr_bm, nulldatawidth -- (DEBUG INFO)
   FROM (
     SELECT
       i.nspname, i.tblname, i.idxname, i.reltuples, i.relpages, i.relam, a.attrelid
AS table_oid,
       current_setting('block_size')::numeric AS bs, fillfactor,
       CASE -- MAXALIGN: 4 on 32bits, 8 on 64bits (and mingw32 ?)
         WHEN version() ~ 'mingw32' OR version() ~ '64-bit|x86_64|ppc64|ia64|amd64'
THEN 8
         ELSE 4
       END AS maxalign,
       /* per page header, fixed size: 20 for 7.X, 24 for others */
       24 AS pagehdr,
       /* per page btree opaque data */
       16 AS pageopqdata,
       /* per tuple header: add IndexAttributeBitMapData if some cols are null-able */
       CASE WHEN max(coalesce(s.null_frac,0)) = 0
         THEN 2 -- IndexTupleData size
         ELSE 2 + (( 32 + 8 - 1 ) / 8)
         -- IndexTupleData size + IndexAttributeBitMapData size ( max num filed per
index + 8 - 1 /8)
       END AS index_tuple_hdr_bm,
       /* data len: we remove null values save space using it fractionnal part from
stats */
       sum( (1-coalesce(s.null_frac, 0)) * coalesce(s.avg_width, 1024)) AS
nulldatawidth,
       max( CASE WHEN a.atttypid = 'pg_catalog.name'::regtype THEN 1 ELSE 0 END ) > 0
AS is_na
     FROM pg_attribute AS a
       JOIN (
         SELECT nspname, tbl.relname AS tblname, idx.relname AS idxname,
           idx.reltuples, idx.relpages, idx.relam,
```

```
            indrelid, indexrelid, indkey::smallint[] AS attnum,
            coalesce(substring(
              array_to_string(idx.reloptions, ' ')
               from 'fillfactor=([0-9]+)')::smallint, 90) AS fillfactor
          FROM pg_index
            JOIN pg_class idx ON idx.oid=pg_index.indexrelid
            JOIN pg_class tbl ON tbl.oid=pg_index.indrelid
            JOIN pg_namespace ON pg_namespace.oid = idx.relnamespace
          WHERE pg_index.indisvalid AND tbl.relkind = 'r' AND idx.relpages > 0
        ) AS i ON a.attrelid = i.indexrelid
        JOIN pg_stats AS s ON s.schemaname = i.nspname
          AND ((s.tablename = i.tblname AND s.attname =
 pg_catalog.pg_get_indexdef(a.attrelid, a.attnum, TRUE))
          -- stats from tbl
          OR  (s.tablename = i.idxname AND s.attname = a.attname))
          -- stats from functional cols
        JOIN pg_type AS t ON a.atttypid = t.oid
      WHERE a.attnum > 0
      GROUP BY 1, 2, 3, 4, 5, 6, 7, 8, 9
    ) AS s1
  ) AS s2
    JOIN pg_am am ON s2.relam = am.oid WHERE am.amname = 'btree'
) AS sub
-- WHERE NOT is_na
ORDER BY 2,3,4;
```

Finden Sie Tabellen, die für die automatische Vakuumierung berechtigt sind

Führen Sie die folgende Abfrage aus, um Tabellen zu finden, die für die automatische Vakuumierung berechtigt sind.

```
--This query shows tables that need vacuuming and are eligible candidates.
--The following query lists all tables that are due to be processed by autovacuum.
-- During normal operation, this query should return very little.
WITH  vbt AS (SELECT setting AS autovacuum_vacuum_threshold
              FROM pg_settings WHERE name = 'autovacuum_vacuum_threshold')
    , vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor
              FROM pg_settings WHERE name = 'autovacuum_vacuum_scale_factor')
    , fma AS (SELECT setting AS autovacuum_freeze_max_age
              FROM pg_settings WHERE name = 'autovacuum_freeze_max_age')
    , sto AS (SELECT opt_oid, split_part(setting, '=', 1) as param,
                split_part(setting, '=', 2) as value
              FROM (SELECT oid opt_oid, unnest(reloptions) setting FROM pg_class) opt)
```

```
SELECT
    '"'||ns.nspname||'"."'||c.relname||'"' as relation
    , pg_size_pretty(pg_table_size(c.oid)) as table_size
    , age(relfrozenxid) as xid_age
    , coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age
    , (coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
         coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
 c.reltuples)
         as autovacuum_vacuum_tuples
    , n_dead_tup as dead_tuples
FROM pg_class c
JOIN pg_namespace ns ON ns.oid = c.relnamespace
JOIN pg_stat_all_tables stat ON stat.relid = c.oid
JOIN vbt on (1=1)
JOIN vsf ON (1=1)
JOIN fma on (1=1)
LEFT JOIN sto cvbt ON cvbt.param = 'autovacuum_vacuum_threshold' AND c.oid =
 cvbt.opt_oid
LEFT JOIN sto cvsf ON cvsf.param = 'autovacuum_vacuum_scale_factor' AND c.oid =
 cvsf.opt_oid
LEFT JOIN sto cfma ON cfma.param = 'autovacuum_freeze_max_age' AND c.oid = cfma.opt_oid
WHERE c.relkind = 'r'
AND nspname <> 'pg_catalog'
AND (
    age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
    or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
      coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples
 <= n_dead_tup
    -- or 1 = 1
)
ORDER BY age(relfrozenxid) DESC;
```

Reagieren Sie auf eine hohe Anzahl von Verbindungen

Wenn Sie Amazon überwachen CloudWatch, stellen Sie möglicherweise fest, dass die `DatabaseConnections` Metrik Spitzen aufweist. Dieser Anstieg deutet auf eine erhöhte Anzahl von Verbindungen zu Ihrer Datenbank hin. Wir empfehlen folgende Vorgehensweise:

- Beschränken Sie die Anzahl der Verbindungen, die die Anwendung mit jeder Instance öffnen kann. Wenn Ihre Anwendung über eine eingebettete Verbindungspool-Funktion verfügt, legen Sie eine

angemessene Anzahl von Verbindungen fest. Basieren Sie die Zahl darauf, was die vCPUs in Ihrer Instance effektiv parallelisieren können.

Wenn Ihre Anwendung keine Verbindungspool-Funktion verwendet, sollten Sie den Amazon RDS Proxy oder eine Alternative in Betracht ziehen. Mit diesem Ansatz können Ihre Anwendung mehrere Verbindungen mit dem Load Balancer öffnen. Der Balancer kann dann eine begrenzte Anzahl von Verbindungen mit der Datenbank öffnen. Da weniger Verbindungen parallel laufen, führt Ihre DB-Instance weniger Kontextwechsel im Kernel durch. Abfragen sollten schneller voranschreiten und zu weniger Warteereignissen führen. Weitere Informationen finden Sie unter Amazon RDS-Proxy .

- Nutzen Sie nach Möglichkeit die Lesereplikate für RDS für PostgreSQL. Wenn Ihre Anwendung eine schreibgeschützten Operation ausführt, senden Sie diese Anfragen an das bzw. die Lesereplikate. Mit dieser Methode wird der I/O-Druck auf den primären (Writer-)Knoten) reduziert.
- Ziehen Sie, Ihre DB-Instance zu skalieren. Eine Instance-Klasse mit höherer Kapazität bietet mehr Speicher, was RDS für PostgreSQL einen größeren freigegebenen Pufferpool zum Speichern von Seiten gibt. Die größere Größe gibt der DB-Instance auch mehr vCPUs für die Handhabung von Verbindungen. Mehr vCPUs sind besonders hilfreich, wenn die Vorgänge, die `IO:DataFileRead`-Warteereignisse generieren, Schreibvorgänge sind.

IO: WALWrite

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL 10 und höher unterstützt.

Kontext

Die Aktivität in der Datenbank, die Write-Ahead-Protokolldaten generiert, füllt zuerst die WAL-Puffer und schreibt dann asynchron auf die Festplatte. Das Warteereignis `IO:WALWrite` wird generiert, wenn die SQL-Sitzung darauf wartet, dass die WAL-Daten das Schreiben auf die Festplatte abgeschlossen haben, damit sie den COMMIT-Aufruf der Transaktion freigeben kann.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn dieses Warteereignis häufig auftritt, sollten Sie Ihre Workload und die Art der Aktualisierungen, die Ihre Workload durchführt, sowie deren Häufigkeit überprüfen. Suchen Sie insbesondere nach folgender Art von Aktivität.

Starke DML-Aktivität

Das Ändern von Daten in Datenbanktabellen erfolgt nicht sofort. Beim Einfügen in eine Tabelle muss möglicherweise auf eine Einfügung oder Aktualisierung derselben Tabelle von einem anderen Client gewartet werden. Die Data Manipulation Language (DML)-Anweisungen zum Ändern von Datenwerten (INSERT, UPDATE, DELETE, COMMIT, ROLLBACK TRANSACTION) können Konflikte verursachen, die dazu führen, dass die Write-Ahead-Protokolldatei darauf wartet, dass die Puffer geleert werden. Diese Situation wird in den folgenden Metriken für Erkenntnisse zur Amazon-RDS-Leistung erfasst, die auf eine starke DML-Aktivität hinweisen.

- `tup_inserted`
- `tup_updated`
- `tup_deleted`
- `xact_rollback`
- `xact_commit`

Weitere Informationen zu diesen Metriken finden Sie unter Performance Insights-Zähler für Amazon RDS for PostgreSQL.

Häufige Checkpoint-Aktivitäten

Häufige Checkpoints tragen zu einer höheren Anzahl von WAL-Dateien bei. In RDS für PostgreSQL sind Schreibvorgänge für vollständige Seiten immer aktiviert. Schreibvorgänge für vollständige Seiten schützen vor Datenverlust. Wenn Checkpoints jedoch zu häufig auftreten, kann es zu Leistungseinbußen des Systems kommen. Dies gilt insbesondere für Systeme mit

starker DML-Aktivität. In einigen Fällen finden Sie möglicherweise Fehlermeldungen in Ihrem `postgresql.log`, dass „Checkpoints zu häufig vorkommen“.

Wir empfehlen, bei der Optimierung von Checkpoints die Leistung sorgfältig gegen den erwarteten Zeitbedarf für die Wiederherstellung im Falle eines abnormalen Shutdowns abzuwägen.

Aktionen

Wir empfehlen die folgenden Aktionen, um die Vorkommen dieses Warteereignis zu reduzieren.

Themen

- Reduzieren Sie die Anzahl der Commits
- Überwachen Ihrer Checkpoints
- Hochskalieren von I/O
- Dediziertes Protokollvolumen (DLV)

Reduzieren Sie die Anzahl der Commits

Um die Anzahl der Commits zu reduzieren, können Sie Anweisungen in Transaktionsblöcke kombinieren. Verwenden Sie Erkenntnisse zur Amazon-RDS-Leistung, um die Art der ausgeführten Abfragen zu untersuchen. Sie können große Wartungsvorgänge auch auf Zeiten außerhalb der Spitzenzeiten verlegen. Erstellen Sie beispielsweise Indizes oder verwenden Sie `pg_repack`-Operationen außerhalb der Produktionszeiten.

Überwachen Ihrer Checkpoints

Es gibt zwei Parameter, die Sie überwachen können, um zu sehen, wie oft Ihre DB-Instance von RDS für PostgreSQL wegen Checkpoints in die WAL-Datei schreibt.

- `log_checkpoints` – Dieser Parameter ist standardmäßig aktiviert. Dadurch wird für jeden Checkpoint eine Nachricht an das PostgreSQL-Protokoll gesendet. Diese Protokollnachrichten enthalten die Anzahl der geschriebenen Puffer, die für das Schreiben aufgewendete Zeit und die Anzahl der für den angegebenen Checkpoint hinzugefügten, entfernten oder recycelten WAL-Dateien.

  Weitere Informationen zu diesem Parameter finden Sie unter Fehlerberichte und -protokollierung in der PostgreSQL-Dokumentation.

- `checkpoint_warning` – Dieser Parameter legt einen Schwellenwert (in Sekunden) für die Checkpoint-Häufigkeit fest, bei dessen Überschreitung eine Warnung generiert wird. Standardmäßig ist dieser Parameter in RDS für PostgreSQL nicht festgelegt. Sie können den Wert dieses Parameters festlegen, um eine Warnung zu erhalten, wenn die Datenbankänderungen in Ihrer RDS-für-PostgreSQL-DB-Instance mit einer Geschwindigkeit geschrieben werden, für die die WAL-Dateien nicht dimensioniert sind. Angenommen, Sie haben diesen Parameter auf 30 festgelegt. Wenn Ihre RDS-für-PostgreSQL-Instance Änderungen öfter als alle 30 Sekunden schreiben muss, wird die Warnung bezüglich zu häufig vorkommender Checkpoints an das PostgreSQL-Protokoll gesendet. Dies kann darauf hindeuten, dass Ihr `max_wal_size`-Wert erhöht werden sollte.

  Weitere Informationen finden Sie unter Write-Ahead-Protokoll in der PostgreSQL-Dokumentation.

Hochskalieren von I/O

Diese Art von input/output (IO) wait event can remediated by scaling the input/output Operationen pro Sekunde (IOPs) sorgt für eine schnellere I/O. Die Skalierung von I/O ist der Skalierung der CPU vorzuziehen, da die Skalierung der CPU zu noch mehr I/O-Konflikten führen kann. Der Grund hierfür ist, dass die erhöhte CPU mehr Arbeit bewältigen kann und somit den I/O-Engpass noch verschlimmert. Im Allgemeinen empfehlen wir, die Workload zu optimieren, bevor Sie Skalierungsvorgänge durchführen.

Dediziertes Protokollvolumen (DLV)

Sie können ein dediziertes Protokoll-Volume (DLV) für eine DB-Instance, die Provisioned IOPS (PIOPS)-Speicher verwendet, mithilfe der Amazon-RDS-Konsole, AWS CLI oder der Amazon-RDS-API verwenden. Ein DLV verschiebt die Transaktionsprotokolle der PostgreSQL-Datenbank auf ein Speichervolume, das von dem Volume getrennt ist, das die Datenbanktabellen enthält. Weitere Informationen finden Sie unter Dediziertes Protokollvolumen (DLV).

IPC:Parallele Warteereignisse

Die folgenden `IPC:parallel wait events` Hinweise deuten darauf hin, dass eine Sitzung auf die Kommunikation zwischen Prozessen im Zusammenhang mit parallel Abfrageausführungen wartet.

- `IPC:BgWorkerStartup`- Ein Prozess wartet darauf, dass ein parallel Worker-Prozess seine Startsequenz abschließt. Dies passiert, wenn Worker für die parallel Abfrageausführung initialisiert werden.

- `IPC:BgWorkerShutdown`- Ein Prozess wartet darauf, dass ein parallel Worker-Prozess seine Shutdown-Sequenz abschließt. Dies geschieht während der Bereinigungsphase der parallel Abfrageausführung.
- `IPC:ExecuteGather`- Ein Prozess wartet darauf, während der Abfrageausführung Daten von parallel Worker-Prozessen zu empfangen. Dies ist der Fall, wenn der Leader-Prozess Ergebnisse von seinen Workern sammeln muss.
- `IPC:ParallelFinish`- Ein Prozess wartet darauf, dass parallel Mitarbeiter ihre Ausführung beenden und ihre Endergebnisse melden. Dies geschieht während der Abschlussphase der parallel Abfrageausführung.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von Aurora PostgreSQL unterstützt.

Kontext

Bei der parallelen Abfrageausführung in PostgreSQL arbeiten mehrere Prozesse zusammen, um eine einzelne Abfrage zu verarbeiten. Wenn festgestellt wird, dass eine Abfrage für die Parallelisierung geeignet ist, koordiniert sich ein führender Prozess auf der Grundlage der `max_parallel_workers_per_gather` Parametereinstellung mit einem oder mehreren parallel Worker-Prozessen. Der Leader-Prozess teilt die Arbeit auf die einzelnen Mitarbeiter auf, jeder Mitarbeiter verarbeitet seinen Teil der Daten unabhängig, und die Ergebnisse werden an den Leader-Prozess zurückgemeldet.

> **Note**
>
> Jeder parallel Worker arbeitet als separater Prozess mit Ressourcenanforderungen, die einer vollständigen Benutzersitzung ähneln. Das bedeutet, dass eine parallel Abfrage mit 4 Workern bis zu fünfmal so viele Ressourcen (CPU, Arbeitsspeicher, I/O Bandbreite)

verbrauchen kann wie eine nicht parallele Abfrage, da sowohl der Leader-Prozess als auch jeder Worker-Prozess ihre eigenen Ressourcenzuweisungen beibehalten. Beispielsweise `work_mem` werden Einstellungen wie individuell auf jeden Worker angewendet, wodurch sich die gesamte Speicherbelegung aller Prozesse potenziell vervielfacht.

Die parallel Abfragearchitektur besteht aus drei Hauptkomponenten:

- Leader-Prozess: Der Hauptprozess, der den Parallelbetrieb einleitet, die Arbeitslast aufteilt und sich mit den Arbeitsprozessen koordiniert.
- Arbeitsprozesse: Hintergrundprozesse, die Teile der Abfrage parallel ausführen.
- Zusammenführung von Sammeln/Sammeln: Operationen, bei denen Ergebnisse aus mehreren Arbeitsprozessen bis zum Leader zusammengefasst werden

Während der parallel Ausführung müssen Prozesse über IPC-Mechanismen (Inter-Process Communication) miteinander kommunizieren. Diese IPC-Warteereignisse treten in verschiedenen Phasen auf:

- Worker-Startup: Wenn parallel Worker initialisiert werden
- Datenaustausch: Wenn Mitarbeiter Daten verarbeiten und Ergebnisse an den Leiter senden
- Worker-Shutdown: Wenn die parallel Ausführung abgeschlossen ist und Worker beendet werden
- Synchronisierungspunkte: Wenn Prozesse koordiniert werden müssen oder darauf warten müssen, dass andere Prozesse ihre Aufgaben abgeschlossen haben

Das Verständnis dieser Warteereignisse ist entscheidend für die Diagnose von Leistungsproblemen im Zusammenhang mit der parallel Abfrageausführung, insbesondere in Umgebungen mit hoher Parallelität, in denen mehrere parallel Abfragen gleichzeitig ausgeführt werden können.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Verschiedene Faktoren können zu einer Zunahme parallelbedingter IPC-Warteereignisse beitragen:

Hohe Gleichzeitigkeit parallel Abfragen

Wenn viele parallel Abfragen gleichzeitig ausgeführt werden, kann dies zu Ressourcenkonflikten und längeren Wartezeiten für IPC-Operationen führen. Dies ist besonders häufig bei Systemen mit hohem Transaktionsvolumen oder analytischen Workloads der Fall.

Suboptimale Pläne für parallel Abfragen

Wenn der Abfrageplaner ineffiziente Parallelpläne auswählt, kann dies zu unnötiger Parallelisierung oder schlechter Arbeitsverteilung unter den Mitarbeitern führen. Dies kann zu erhöhten IPC-Wartezeiten führen, insbesondere bei Ereignissen und Ereignissen. `IPC:ExecuteGather` `IPC:ParallelFinish` Diese Planungsprobleme sind häufig auf veraltete Statistiken und table/index unübersichtliche Zahlen zurückzuführen.

Häufiges Starten und Herunterfahren von Parallelarbeitern

Kurzlebige Abfragen, die häufig parallel Worker initiieren und beenden, können zu einer Zunahme `IPC:BgWorkerStartup` von `IPC:BgWorkerShutdown` AND-Ereignissen führen. Dies tritt häufig bei OLTP-Workloads mit vielen kleinen, parallelisierbaren Abfragen auf.

Einschränkungen für Ressourcen

Eine begrenzte CPU, ein begrenzter Arbeitsspeicher oder eine begrenzte I/O Kapazität können zu Engpässen bei der parallel Ausführung führen, was zu längeren Wartezeiten bei allen IPC-Ereignissen führt. Wenn die CPU beispielsweise ausgelastet ist, kann es länger dauern, bis Arbeitsprozesse gestartet werden oder ihren Teil der Arbeit verarbeiten.

Komplexe Abfragestrukturen

Abfragen mit mehreren Parallelitätsebenen (z. B. parallel Verknüpfungen, gefolgt von parallel Aggregationen) können zu komplexeren IPC-Mustern und potenziell längeren Wartezeiten führen, insbesondere bei Ereignissen. `IPC:ExecuteGather`

Große Ergebnismengen

Abfragen, die große Ergebnismengen liefern, können zu längeren `IPC:ExecuteGather` Wartezeiten führen, da der führende Prozess mehr Zeit damit verbringt, Ergebnisse aus Arbeitsprozessen zu sammeln und zu verarbeiten.

Das Verständnis dieser Faktoren kann bei der Diagnose und Behebung von Leistungsproblemen im Zusammenhang mit der parallel Abfrageausführung in Aurora PostgreSQL hilfreich sein.

Aktionen

Wenn Sie Wartezeiten im Zusammenhang mit parallel Abfragen sehen, bedeutet dies in der Regel, dass ein Backend-Prozess parallel Worker-Prozesse koordiniert oder darauf wartet. Diese Wartezeiten treten häufig bei der Ausführung parallel Pläne auf. Sie können die Auswirkungen dieser Wartezeiten untersuchen und abmildern, indem Sie die parallel Worker-Nutzung überwachen, die

Parametereinstellungen überprüfen und die Abfrageausführung und die Ressourcenzuweisung optimieren.

Themen

- Analysieren Sie Abfragepläne auf ineffiziente Parallelität
- Überwachen Sie die Verwendung parallel Abfragen
- Einstellungen für parallel Abfragen überprüfen und anpassen
- Optimieren Sie die Ressourcenzuweisung
- Untersuchen Sie das Verbindungsmanagement
- Überprüfen und optimieren Sie die Wartungsarbeiten

Analysieren Sie Abfragepläne auf ineffiziente Parallelität

Die parallele Ausführung von Abfragen kann häufig zu Systeminstabilität, CPU-Spitzen und unvorhersehbaren Leistungsschwankungen bei Abfragen führen. Es ist wichtig, gründlich zu analysieren, ob Parallelität Ihre spezifische Arbeitslast tatsächlich verbessert. Verwenden Sie EXPLAIN ANALYZE, um Pläne parallel Ausführung von Abfragen zu überprüfen.

Deaktivieren Sie vorübergehend die Parallelität auf Sitzungsebene, um die Effizienz des Plans zu vergleichen:

```
SET max_parallel_workers_per_gather = 0;
EXPLAIN ANALYZE <your_query>;
```

Aktivieren Sie die Parallelität erneut und vergleichen Sie:

```
RESET max_parallel_workers_per_gather;
EXPLAIN ANALYZE <your_query>;
```

Wenn die Deaktivierung der Parallelität zu besseren oder konsistenteren Ergebnissen führt, sollten Sie erwägen, sie für bestimmte Abfragen auf Sitzungsebene mithilfe von SET-Befehlen zu deaktivieren. Um eine umfassendere Wirkung zu erzielen, sollten Sie die Parallelität auf Instance-Ebene deaktivieren, indem Sie die entsprechenden Parameter in Ihrer DB-Parametergruppe

anpassen. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Überwachen Sie die Verwendung parallel Abfragen

Verwenden Sie die folgenden Abfragen, um Einblick in die Aktivität und Kapazität parallel Abfragen zu erhalten:

Überprüfen Sie die aktiven parallel Worker-Prozesse:

```
SELECT
    COUNT(*)
FROM
    pg_stat_activity
WHERE
    backend_type = 'parallel worker';
```

Diese Abfrage zeigt die Anzahl der aktiven parallel Worker-Prozesse. Ein hoher Wert kann darauf hinweisen, dass Ihr `max_parallel_workers` mit einem hohen Wert konfiguriert ist, und Sie sollten erwägen, diesen zu reduzieren.

Überprüfen Sie gleichzeitige parallel Abfragen:

```
SELECT
    COUNT(DISTINCT leader_pid)
FROM
    pg_stat_activity
WHERE
    leader_pid IS NOT NULL;
```

Diese Abfrage gibt die Anzahl der unterschiedlichen Leader-Prozesse zurück, die parallel Abfragen gestartet haben. Eine hohe Zahl weist hier darauf hin, dass mehrere Sitzungen parallel Abfragen gleichzeitig ausführen, was die CPU- und Speicherbelastung erhöhen kann.

Einstellungen für parallel Abfragen überprüfen und anpassen

Überprüfen Sie die folgenden Parameter, um sicherzustellen, dass sie mit Ihrer Arbeitslast übereinstimmen:

- `max_parallel_workers`: Gesamtzahl der parallel Mitarbeiter in allen Sitzungen.
- `max_parallel_workers_per_gather`: Max. Anzahl an Arbeitern pro Abfrage.

Bei OLAP-Workloads kann eine Erhöhung dieser Werte die Leistung verbessern. Für OLTP-Workloads werden im Allgemeinen niedrigere Werte bevorzugt.

```
SHOW max_parallel_workers;
SHOW max_parallel_workers_per_gather;
```

Optimieren Sie die Ressourcenzuweisung

Überwachen Sie die CPU-Auslastung und erwägen Sie, die Anzahl von v CPUs anzupassen, wenn sie konstant hoch ist und Ihre Anwendung von parallel Abfragen profitiert. Stellen Sie sicher, dass ausreichend Speicher für parallel Operationen verfügbar ist.

- Ermitteln Sie anhand von Performance Insights Insights-Metriken, ob das System CPU-gebunden ist.
- Jeder parallel Worker verwendet seinen eigenen`work_mem`. Stellen Sie sicher, dass die Gesamtspeichernutzung innerhalb der Instanzgrenzen liegt.

Die parallel Abfragen können erheblich mehr Ressourcen verbrauchen als nicht parallele Abfragen, da jeder Arbeitsprozess ein völlig separater Prozess ist, der ungefähr die gleichen Auswirkungen auf das System hat wie eine zusätzliche Benutzersitzung. Dies sollte bei der Auswahl eines Werts für diese Einstellung sowie bei der Konfiguration anderer Einstellungen zur Steuerung der Ressourcennutzung berücksichtigt werden, wie `work_mem` z. Weitere Informationen finden Sie in der PostgreSQL-Dokumentation. Ressourcenbegrenzungen, wie `work_mem` sie für jeden Worker individuell gelten, bedeutet, dass die Gesamtauslastung über alle Prozesse hinweg viel höher sein kann, als dies normalerweise für einen einzelnen Prozess der Fall wäre.

Erwägen Sie, v zu erhöhen CPUs oder die Speicherparameter zu optimieren, wenn Ihre Arbeitslast stark parallelisiert ist.

Untersuchen Sie das Verbindungsmanagement

Wenn die Verbindung erschöpft ist, überprüfen Sie die Strategien für das Verbindungspooling der Anwendung. Erwägen Sie die Implementierung von Verbindungspooling auf Anwendungsebene, sofern es nicht bereits verwendet wird.

Überprüfen und optimieren Sie die Wartungsarbeiten

Koordinieren Sie die Indexerstellung und andere Wartungsaufgaben, um Ressourcenkonflikte zu vermeiden. Erwägen Sie, diese Operationen außerhalb der Spitzenzeiten zu planen. Vermeiden Sie es, umfangreiche Wartungsarbeiten (z. B. parallel Indexerstellungen) in Zeiten hoher Benutzerabfragen einzuplanen. Diese Operationen können parallel Worker beanspruchen und die Leistung bei regulären Abfragen beeinträchtigen.

Lock:advisory

Das `Lock:advisory`-Ereignis tritt auf, wenn eine PostgreSQL-Anwendung eine Sperre verwendet, um Aktivitäten über mehrere Sitzungen hinweg zu koordinieren.

Themen

- Relevante Engine-Versionen
- Context
- Ursachen
- Aktionen

Relevante Engine-Versionen

Diese Warteereignisinformationen sind für die RDS-für-PostgreSQL-Versionen 9.6 und höher relevant.

Context

PostgreSQL-Beratungssperren sind Anwendungsebene, kooperative Sperren werden explizit durch den Anwendungscode des Benutzers gesperrt und freigeschaltet. Eine Anwendung kann PostgreSQL-Beratungssperren verwenden, um Aktivitäten über mehrere Sitzungen hinweg zu koordinieren. Im Gegensatz zu normalen Sperren auf Objekt- oder Zeilenebene hat die Anwendung die volle Kontrolle über die Lebensdauer des Schlosses. Weitere Informationen finden Sie unter Empfohlene Sperren in der PostgreSQL-Dokumentation.

Beratungssperren können vor dem Ende einer Transaktion freigegeben werden oder von einer Sitzung über Transaktionen hinweg gehalten werden. Dies gilt nicht für implizite, vom System erzwungene Sperren, wie z. B. eine zugriffsexklusive Sperre für eine Tabelle, die von einer `CREATE INDEX`-Anweisung abgerufen wird.

Eine Beschreibung der Funktionen zum Erlangen (Sperren) und Freigeben (Entsperren) von Advisory Locks finden Sie unter Advisory Lock Functions in der PostgreSQL-Dokumentation.

Advisory Locks werden zusätzlich zum regulären PostgreSQL-Locking-System implementiert und sind in der `pg_locks`-Systemansicht sichtbar.

Ursachen

Dieser Schlosstyp wird ausschließlich von einer Anwendung gesteuert, die ihn explizit verwendet. Beratungssperren, die für jede Zeile als Teil einer Abfrage erworben werden, können zu einem Anstieg der Sperren oder zu einem langfristigen Aufbau führen.

Diese Effekte treten auf, wenn die Abfrage so ausgeführt wird, dass Sperren für mehr Zeilen erwirbt, als von der Abfrage zurückgegeben werden. Die Anwendung muss schließlich jede Sperre freigeben, aber wenn Sperren für Zeilen erworben werden, die nicht zurückgegeben werden, kann die Anwendung nicht alle Sperren finden.

Das folgende Beispiel stammt aus Empfohlene Sperren in der PostgreSQL-Dokumentation.

```
SELECT pg_advisory_lock(id) FROM foo WHERE id > 12345 LIMIT 100;
```

In diesem Beispiel kann die `LIMIT`-Klausel die Ausgabe der Abfrage nur stoppen, nachdem die Zeilen bereits intern ausgewählt und ihre ID-Werte gesperrt wurden. Dies kann plötzlich passieren, wenn ein wachsendes Datenvolumen dazu führt, dass der Planer einen anderen Ausführungsplan auswählt, der während der Entwicklung nicht getestet wurde. Der Aufbau erfolgt in diesem Fall, weil die Anwendung `pg_advisory_unlock` explizit für jeden gesperrten ID-Wert aufruft. In diesem Fall kann es jedoch nicht die Sperren finden, die für Zeilen erworben wurden, die nicht zurückgegeben wurden. Da die Sperren auf Sitzungsebene erworben werden, werden sie am Ende der Transaktion nicht automatisch freigegeben.

Eine weitere mögliche Ursache für Spikes bei blockierten Sperrversuchen sind unbeabsichtigte Konflikte. In diesen Konflikten teilen sich nicht verwandte Teile der Anwendung versehentlich denselben Sperren-ID-Raum.

Aktionen

Überprüfen Sie die Anwendungsnutzung von Beratungssperren und Details, wo und wann im Anwendungsablauf jede Art von Beratungssperre erworben und freigegeben wird.

Stellen Sie fest, ob eine Sitzung zu viele Sperren erwirbt oder eine lang andauernde Sitzung keine Sperren früh genug freigibt, was zu einem langsamen Aufbau von Sperren führt. Sie können einen langsamen Aufbau von Sperren auf Sitzungsebene korrigieren, indem Sie die Sitzung mit `pg_terminate_backend(pid)` beenden.

Ein Client, der auf eine Beratungssperre wartet, erscheint in `pg_stat_activity` mit `wait_event_type=Lock` und `wait_event=advisory`. Sie können bestimmte Sperrwerte erhalten, indem Sie die `pg_locks`-Systemansicht nach demselben `pid` abfragen und nach `locktype=advisory` und `granted=f` suchen.

Sie können dann die blockierende Sitzung identifizieren, indem Sie `pg_locks` nach derselben beratenden Sperre mit `granted=t` abfragen, wie im folgenden Beispiel gezeigt.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
         concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
 blocked_wait_event,
         concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
 blocking_wait_event,
         blocked_activity.state AS blocked_state,
         blocking_activity.state AS blocking_state,
         blocked_locks.locktype AS blocked_locktype,
         blocking_locks.locktype AS blocking_locktype,
         blocked_activity.query AS blocked_statement,
         blocking_activity.query AS blocking_statement
    FROM pg_catalog.pg_locks blocked_locks
    JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
 blocked_locks.pid
    JOIN pg_catalog.pg_locks blocking_locks
        ON blocking_locks.locktype = blocked_locks.locktype
        AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
        AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
        AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
```

```
        AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
        AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
        AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
        AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
        AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
        AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
        AND blocking_locks.pid != blocked_locks.pid
    JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
    WHERE NOT blocked_locks.GRANTED;
```

Alle API-Funktionen für beratende Sperren haben zwei Sätze von Argumenten, entweder ein `bigint`-Argument oder zwei `integer`-Argumente:

- Bei den API-Funktionen mit einem `bigint`-Argument befinden sich die oberen 32 Bit in `pg_locks.classid` und die unteren 32 Bit in `pg_locks.objid`.
- Bei den API-Funktionen mit zwei `integer`-Argumenten ist das erste Argument `pg_locks.classid` und das zweite Argument ist `pg_locks.objid`.

Der `pg_locks.objsubid`-Wert gibt an, welches API-Formular verwendet wurde: 1 bedeutet ein `bigint`-Argument; 2 bedeutet zwei `integer`-Argumente.

Lock:extend

Das `Lock:extend`-Ereignis tritt ein, wenn ein Backend-Prozess darauf wartet, eine Beziehung zu sperren, um sie zu erweitern, während ein anderer Prozess diese Beziehung für denselben Zweck gesperrt hat.

Themen

- Unterstützte Engine-Versionen
- Context
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Context

Das Ereignis `Lock:extend` zeigt an, dass ein Backend-Prozess darauf wartet, eine Beziehung zu erweitern, für die ein anderer Backend-Prozess eine Sperre hält, während er diese Beziehung erweitert. Da jeweils nur ein Prozess eine Beziehung erweitern kann, generiert das System ein `Lock:extend`-Warteereignis. `INSERT`-, `COPY`- und `UPDATE`-Vorgänge können dieses Ereignis erzeugen.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn das `Lock:extend`-Ereignis mehr als normal auftritt, was möglicherweise auf ein Leistungsproblem hinweist, sind die folgenden typischen Ursachen:

Anstieg der gleichzeitigen Einfügungen oder Aktualisierungen derselben Tabelle

Es kann zu einer Zunahme der Anzahl gleichzeitiger Sitzungen mit Abfragen kommen, die in dieselbe Tabelle einfügen oder aktualisieren.

Unzureichende Netzwerkbandbreite

Die Netzwerkbandbreite auf der DB-Instance reicht möglicherweise nicht aus, um die Speicherkommunikationsanforderungen der aktuellen Workload zu gewährleisten. Dies kann zu einer Speicherlatenz führen, die zu einem Anstieg der `Lock:extend`-Ereignisse führt.

Aktionen

Abhängig von den Ursachen Ihres Wait-Ereignisses empfehlen wir verschiedene Aktionen.

Themen

- Reduzieren Sie gleichzeitige Einfügungen und Aktualisierungen auf dieselbe Beziehung
- Erhöhung der Netzwerkbandbreite

Reduzieren Sie gleichzeitige Einfügungen und Aktualisierungen auf dieselbe Beziehung

Stellen Sie zunächst fest, ob die `tup_inserted`- und `tup_updated`-Metriken und damit auch dieses Warteereignis gestiegen ist. Überprüfen Sie in diesem Fall, welche Beziehungen für Einfüge- und Aktualisierungsvorgänge in hohem Streit stehen. Um dies zu ermitteln, fragen Sie die `pg_stat_all_tables`-Ansicht nach den Werten in den `n_tup_ins`- und `n_tup_upd`-Feldern

ab. Informationen zur Ansicht `pg_stat_all_tables` finden Sie unter pg_stat_all_tables in der PostgreSQL-Dokumentation.

Um weitere Informationen über das Blockieren und blockierte Abfragen zu erhalten, fragen Sie `pg_stat_activity` wie im folgenden Beispiel ab:

```
SELECT
    blocked.pid,
    blocked.usename,
    blocked.query,
    blocking.pid AS blocking_id,
    blocking.query AS blocking_query,
    blocking.wait_event AS blocking_wait_event,
    blocking.wait_event_type AS blocking_wait_event_type
FROM pg_stat_activity AS blocked
JOIN pg_stat_activity AS blocking ON blocking.pid = ANY(pg_blocking_pids(blocked.pid))
where
blocked.wait_event = 'extend'
and blocked.wait_event_type = 'Lock';

   pid  | usename  |            query             | blocking_id |
            blocking_query                          | blocking_wait_event |
 blocking_wait_event_type
  ------+----------+------------------------------+-------------
+-------------------------------------------------------------------
+---------------------+--------------------------
   7143 |  myuser  | insert into tab1 values (1); |        4600 | INSERT INTO tab1 (a)
 SELECT s FROM generate_series(1,1000000) s; | DataFileExtend      | IO
```

Nachdem Sie Beziehungen identifiziert haben, die zur Erhöhung von `Lock:extend`-Ereignissen beitragen, verwenden Sie die folgenden Techniken, um die Konflikte zu reduzieren:

- Finden Sie heraus, ob Sie Partitionierung verwenden können, um die Konflikte für dieselbe Tabelle zu reduzieren. Das Trennen eingefügter oder aktualisierter Tupel in verschiedene Partitionen kann die Konflikte verringern. Weitere Informationen zur Partitionierung finden Sie unter Verwalten von PostgreSQL-Partitionen mit der Erweiterung pg_partman.
- Wenn das Warteereignis hauptsächlich auf Aktualisierungsaktivitäten zurückzuführen ist, sollten Sie erwägen, den Füllfaktor-Wert der Beziehung zu reduzieren. Dies kann Anfragen nach neuen Blöcken während des Updates reduzieren. Der Füllfaktor ist ein Speicherparameter für eine Tabelle, der den maximalen Speicherplatz zum Packen einer Tabellenseite bestimmt. Es wird als

Prozentsatz des gesamten Speicherplatzes für eine Seite ausgedrückt. Weitere Informationen zum Parameter fillfactor finden Sie unter CREATE TABLE in der PostgreSQL-Dokumentation.

> **Important**
>
> Wir empfehlen dringend, Ihr System zu testen, wenn Sie den Füllfaktor ändern, da sich die Änderung dieses Wertes je nach Workload negativ auf die Leistung auswirken kann.

Erhöhung der Netzwerkbandbreite

Um zu sehen, ob die Schreiblatenz zunimmt, überprüfen Sie die `WriteLatency`-Metrik in CloudWatch. Wenn dies der Fall ist, verwenden Sie die Amazon-CloudWatch-Metriken `WriteThroughput` und `ReadThroughput`, um den speicherbezogenen Datenverkehr auf der DB-Instance zu überwachen. Diese Metriken können Ihnen helfen festzustellen, ob die Netzwerkbandbreite für die Speicheraktivität Ihrer Workload ausreicht.

Wenn Ihre Netzwerkbandbreite nicht ausreicht, erhöhen Sie sie. Wenn Ihre DB-Instance die Grenzen der Netzwerkbandbreite erreicht, besteht die einzige Möglichkeit, die Bandbreite zu erhöhen, darin, die Größe Ihrer DB-Instance zu erhöhen.

Weitere Informationen zu CloudWatch-Metriken finden Sie unter Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS. Informationen zur Netzwerkleistung für jede DB-Instance-Klasse finden Sie unter Metriken CloudWatch auf Amazon-Instanzebene für Amazon RDS.

Lock:Relation

Das `Lock:Relation`-Ereignis tritt ein, wenn eine Abfrage darauf wartet, eine Sperre für eine Tabelle oder Sicht (Relation) zu erhalten, die derzeit von einer anderen Transaktion gesperrt ist.

Themen

- Unterstützte Engine-Versionen
- Context
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Context

Die meisten PostgreSQL-Befehle verwenden implizit Sperren, um den gleichzeitigen Zugriff auf Daten in Tabellen zu steuern. Sie können diese Sperren auch explizit mit dem `LOCK`-Befehl in Ihrem Anwendungscode verwenden. Viele Sperrmodi sind nicht miteinander kompatibel und können Transaktionen blockieren, wenn sie versuchen, auf dasselbe Objekt zuzugreifen. In diesem Fall generiert RDS für PostgreSQL ein `Lock:Relation`-Ereignis. Einige gängige Beispiele sind die folgenden:

- Exklusive Sperren wie `ACCESS EXCLUSIVE` können alle gleichzeitigen Zugriffe blockieren. Vorgänge in der Datendefinitionssprache (DDL) wie `DROP TABLE`, `TRUNCATE`, `VACUUM FULL` und `CLUSTER` erwerben implizit `ACCESS EXCLUSIVE`-Sperren. `ACCESS EXCLUSIVE` ist auch der Standardsperrmodus für `LOCK TABLE`-Anweisungen, die keinen Modus explizit angeben.
- Die Verwendung von `CREATE INDEX (without CONCURRENT)` für eine Tabelle steht in Konflikt mit den DML-Anweisungen `UPDATE`, `DELETE` und `INSERT`, die `ROW EXCLUSIVE`-Sperren anfordern.

Weitere Informationen zu Sperren auf Tabellenebene und widersprüchlichen Sperrmodi finden Sie unter Explizite Sperren in der PostgreSQL-Dokumentation.

Blockieren von Abfragen und Transaktionen entsperren in der Regel auf eine der folgenden Arten:

- Blockierende Abfrage — Die Anwendung kann die Abfrage abbrechen oder der Benutzer kann den Prozess beenden. Die Engine kann die Abfrage auch aufgrund des Statement-Timeouts einer Sitzung oder eines Deadlock-Erkennungsmechanismus zum Ende zwingen.
- Blockieren einer Transaktion – Eine Transaktion stoppt die Blockierung, wenn sie eine `ROLLBACK`- oder `COMMIT`-Anweisung ausführt. Rollbacks erfolgen auch automatisch, wenn Sitzungen von einem Client oder durch Netzwerkprobleme getrennt oder beendet werden. Sitzungen können beendet werden, wenn das Datenbank-Engine heruntergefahren wird, wenn das System keinen Arbeitsspeicher mehr hat usw.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn das `Lock:Relation`-Ereignis häufiger als normal auftritt, kann dies auf ein Leistungsproblem hinweisen. Zu den typischen Ursachen zählen auch die Folgenden:

Gleichzeitige Sitzungen mit widersprüchlichen Tabellensperren erhöht

Es kann zu einer Zunahme der Anzahl gleichzeitiger Sitzungen mit Abfragen kommen, die dieselbe Tabelle mit widersprüchlichen Sperrmodi sperren.

Wartungsvorgänge

Zustandswartungsvorgänge wie `VACUUM` und `ANALYZE` können die Anzahl widersprüchlicher Sperren erheblich erhöhen. `VACUUM FULL` erhält eine `ACCESS EXCLUSIVE`-Sperre und `ANALYSE` erhält eine `SHARE UPDATE EXCLUSIVE`-Sperre. Beide Arten von Sperren können ein `Lock:Relation`-Wait-Ereignis verursachen. Wartungsvorgänge für Anwendungsdaten wie das Aktualisieren einer materialisierten Ansicht können auch blockierte Abfragen und Transaktionen erhöhen.

Sperrt bei Reader-In

Es könnte ein Konflikt zwischen den Beziehungssperren bestehen, die vom Writer und den Readern gehalten werden. Derzeit werden nur `ACCESS EXCLUSIVE`-Beziehungssperren auf Reader-Instances repliziert. Allerdings gerät die `ACCESS EXCLUSIVE`-Beziehungssperre in Konflikt mit jeder `ACCESS SHARE`-Beziehungssperre, die vom Reader gehalten wird. Dies kann zu einer Erhöhung der Warteereignisse für die Sperrbeziehung des Readers führen.

Aktionen

Abhängig von den Ursachen Ihres Wait-Ereignisses empfehlen wir verschiedene Aktionen.

Themen

- Reduzieren Sie die Auswirkungen der Blockierung von SQL-Anweisungen
- Minimieren Sie die Auswirkungen von Wartungsvorgängen

Reduzieren Sie die Auswirkungen der Blockierung von SQL-Anweisungen

Um die Auswirkungen des Blockierens von SQL-Anweisungen zu reduzieren, ändern Sie Ihren Anwendungscode nach Möglichkeit. Es folgen zwei gängige Techniken zum Reduzieren von Blöcken:

- Verwenden Sie die Option `NOWAIT` – Einige SQL-Befehle, wie z. B. `SELECT`- und `LOCK`-Anweisungen, unterstützen diese Option. Die `NOWAIT`-Direktive bricht die Sperre anfordernde Abfrage ab, wenn die Sperre nicht sofort erworben werden kann. Diese Technik kann dazu beitragen, zu verhindern, dass eine Blockiersitzung eine Anhäufung blockierter Sitzungen dahinter verursacht.

  Beispiel: Angenommen, Transaktion A wartet auf eine Sperre, die von Transaktion B gehalten wird. Wenn B nun eine Sperre für eine Tabelle anfordert, die durch Transaktion C gesperrt ist, könnte Transaktion A blockiert werden, bis die Transaktion C abgeschlossen ist. Wenn Transaktion B jedoch ein `NOWAIT` verwendet, wenn sie die Sperre für C anfordert, kann sie schnell fehlschlagen und sicherstellen, dass Transaktion A nicht unbegrenzt warten muss.
- Verwenden Sie `SET lock_timeout` – Legen Sie einen `lock_timeout`-Wert fest, um die Zeit zu begrenzen, die eine SQL-Anweisung wartet, um eine Sperre für eine Beziehung zu erhalten. Wenn die Sperre nicht innerhalb des angegebenen Timeouts erworben wird, wird die Transaktion, die die Sperre anfordert, abgebrochen. Stellen Sie diesen Wert auf Sitzungsebene ein.

Minimieren Sie die Auswirkungen von Wartungsvorgängen

Wartungsvorgänge wie `VACUUM` und `ANALYZE` sind wichtig. Wir empfehlen, sie nicht zu deaktivieren, da Sie `Lock:Relation`-Warteereignisse im Zusammenhang mit diesen Wartungsvorgängen finden. Die folgenden Ansätze können die Auswirkungen dieser Vorgänge minimieren:

- Führen Sie Wartungsvorgänge während außerhalb der Hauptverkehrszeiten manuell aus.
- Um `Lock:Relation`-Wartezeiten zu reduzieren, die durch Autovacuum-Aufgaben verursacht werden, führen Sie alle erforderlichen Autovacuum-Optimierungen durch. Informationen zum Optimieren von Autovacuum finden Sie unter Arbeiten mit PostgreSQL Autovacuum auf Amazon RDS im Amazon RDS-Benutzerhandbuch.

Lock:transactionid

Das `Lock:transactionid`-Ereignis tritt ein, wenn eine Transaktion auf eine Sperre auf Zeilenebene wartet.

Themen

- Unterstützte Engine-Versionen
- Context

- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Context

Das Ereignis `Lock:transactionid` tritt ein, wenn eine Transaktion versucht, eine Sperre auf Zeilenebene zu erlangen, die bereits einer gleichzeitig laufenden Transaktion gewährt wurde. Die Sitzung, die das `Lock:transactionid`-Wait-Ereignis anzeigt, ist aufgrund dieser Sperre blockiert. Nachdem die blockierende Transaktion entweder in einer `COMMIT`- oder `ROLLBACK`-Anweisung endet, kann die blockierte Transaktion fortgesetzt werden.

Die Semantik der Multiversions-Parallelität von RDS für PostgreSQL garantiert, dass Leser keine Autoren blockieren und Autoren Leser nicht blockieren. Damit Konflikte auf Zeilenebene auftreten können, müssen blockierende und blockierte Transaktionen widersprüchliche Anweisungen der folgenden Typen ausgeben:

- `UPDATE`
- `SELECT … FOR UPDATE`
- `SELECT … FOR KEY SHARE`

Die Anweisung `SELECT … FOR KEY SHARE` ist ein Sonderfall. Die Datenbank verwendet die Klausel `FOR KEY SHARE`, um die Leistung der referenziellen Integrität zu optimieren. Eine Sperre auf Zeilenebene für eine Zeile kann `INSERT`-, `UPDATE`- und `DELETE`-Befehle für andere Tabellen blockieren, die auf die Zeile verweisen.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn dieses Ereignis mehr als normal auftritt, sind die Ursache normalerweise `UPDATE`-, `SELECT … FOR UPDATE`-, `SELECT … FOR KEY SHARE`-Anweisungen in Kombination mit den folgenden Bedingungen.

Themen

- Hohe Gleichzeitigkeit

- Leerlauf in Transaktion
- Lang laufende Transaktionen

Hohe Gleichzeitigkeit

RDS für PostgreSQL kann eine körnige Sperrsemantik auf Zeilenebene verwenden. Die Wahrscheinlichkeit von Konflikten auf Zeilenebene steigt, wenn die folgenden Bedingungen erfüllt sind:

- Eine sehr gleichzeitige Workload beansprucht dieselben Zeilen.
- Parallelbetrieb steigt.

Leerlauf in Transaktion

Manchmal zeigt die Spalte `pg_stat_activity.state` den Wert `idle in transaction` an. Dieser Wert wird für Sitzungen angezeigt, die eine Transaktion gestartet, aber noch kein `COMMIT` oder `ROLLBACK` ausgegeben haben. Wenn der `pg_stat_activity.state`-Wert nicht `active` ist, ist die in `pg_stat_activity` angezeigte Abfrage die letzte, die ausgeführt wurde. Die blockierende Sitzung verarbeitet eine Abfrage nicht aktiv, da eine offene Transaktion eine Sperre hält.

Wenn eine Leerlauf-Transaktion eine Sperre auf Zeilenebene erworben hat, kann dies verhindern, dass andere Sitzungen sie erwerben. Diese Bedingung führt zu einem häufigen Auftreten des Warteereignisses `Lock:transactionid`. Um das Problem zu diagnostizieren, überprüfen Sie die Ausgabe von `pg_stat_activity` und `pg_locks`.

Lang laufende Transaktionen

Transaktionen, die lange laufen, erhalten lange Zeit Sperren. Diese langjährigen Sperren können die Ausführung anderer Transaktionen verhindern.

Aktionen

Zeilensperre ist ein Konflikt zwischen `UPDATE`-, `SELECT … FOR UPDATE`- oder `SELECT … FOR KEY SHARE`-Anweisungen. Bevor Sie eine Lösung versuchen, sollten Sie herausfinden, wann diese Anweisungen in derselben Zeile ausgeführt werden. Wählen Sie mit diesen Informationen eine in den folgenden Abschnitten beschriebene Strategie aus.

Themen

- Reagieren auf hohe Parallelbetrieb

- Reagieren Sie auf ungenutzte Transaktionen
- Reagieren Sie auf lang andauernde Transaktionen

Reagieren auf hohe Parallelbetrieb

Wenn Parallelität das Problem darstellt, versuchen Sie eine der folgenden Techniken:

- Senken Sie die Parallelität in der Anwendung. Verringern Sie beispielsweise die Anzahl der aktiven Sitzungen.
- Implementieren Sie einen Verbindungspool. Informationen zum Poolen von Verbindungen mit RDS-Proxy finden Sie unter Amazon RDS-Proxy .
- Entwerfen Sie die Anwendung oder das Datenmodell, um konkurrierende `UPDATE`- und `SELECT … FOR UPDATE`-Anweisungen zu vermeiden. Sie können auch die Anzahl der Fremdschlüssel verringern, auf die von `SELECT … FOR KEY SHARE`-Anweisungen zugegriffen wird.

Reagieren Sie auf ungenutzte Transaktionen

Wenn `pg_stat_activity.state idle in transaction` anzeigt, verwenden Sie die folgenden Strategien:

- Schalten Sie nach Möglichkeit Autocommit ein. Dieser Ansatz verhindert, dass Transaktionen andere Transaktionen blockieren, während sie auf ein `COMMIT` oder `ROLLBACK` warten.
- Suchen Sie nach Codepfaden, denen `COMMIT`, `ROLLBACK` oder `END` fehlt.
- Stellen Sie sicher, dass die Ausnahmebehandlungslogik in Ihrer Anwendung immer einen Pfad zu einem gültigen `end of transaction` hat.
- Stellen Sie sicher, dass Ihre Anwendung Abfrageergebnisse verarbeitet, nachdem die Transaktion mit `COMMIT` oder `ROLLBACK` beendet wurde.

Reagieren Sie auf lang andauernde Transaktionen

Wenn Transaktionen mit langer Laufzeit das häufige Auftreten von `Lock:transactionid` verursachen, versuchen Sie die folgenden Strategien:

- Halten Sie Zeilensperren von lang andauernden Transaktionen fern.
- Beschränken Sie die Länge von Abfragen, indem Sie nach Möglichkeit Autocommit implementieren.

Lock:tuple

Das `Lock:tuple`-Ereignis tritt ein, wenn ein Backend-Prozess darauf wartet, eine Sperre für ein Tupel zu erlangen.

Themen

- Unterstützte Engine-Versionen
- Context
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Context

Das Ereignis `Lock:tuple` zeigt an, dass ein Back-End darauf wartet, eine Sperre für ein Tupel zu erlangen, während ein anderes Back-End eine widersprüchliche Sperre für dasselbe Tupel hält. Die folgende Tabelle veranschaulicht ein Szenario, in dem Sitzungen das `Lock:tuple`-Ereignis generieren.

Zeit	1. Sitzung	2. Sitzung	3. Sitzung
t1	Startet eine Transaktion.		
t2	Aktualisiert Zeile 1.		
t3		Aktualisiert Zeile 1. Die Sitzung erwirbt eine exklusive Sperre für das Tupel und wartet dann darauf, dass Sitzung 1 die Sperre durch Commit oder Rollback freigibt.	
t4			Aktualisiert Zeile 1. Die Sitzung wartet darauf, dass Sitzung 2

Zeit	1. Sitzung	2. Sitzung	3. Sitzung
			die exklusive Sperre für das Tupel freigibt.

Oder Sie können dieses Warteereignis simulieren, indem Sie das Benchmarking-Tool `pgbench` verwenden. Konfigurieren Sie eine hohe Anzahl gleichzeitiger Sitzungen, um dieselbe Zeile in einer Tabelle mit einer benutzerdefinierten SQL-Datei zu aktualisieren.

Weitere Informationen zu widersprüchlichen Sperrmodi finden Sie unter Explizite Sperren in der PostgreSQL-Dokumentation. Weitere Informationen zu `pgbench` finden Sie unter pgbench in der PostgreSQL-Dokumentation.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn dieses Ereignis mehr als normal auftritt und möglicherweise auf ein Leistungsproblem hinweist, sind typische Ursachen:

- Eine große Anzahl gleichzeitiger Sitzungen versucht, eine widersprüchliche Sperre für dasselbe Tupel zu erlangen, indem sie `UPDATE`- oder `DELETE`-Anweisungen ausführen.
- In hochgradig gleichzeitigen Sitzungen wird eine `SELECT`-Anweisung ausgeführt, die den `FOR UPDATE`- oder `FOR NO KEY UPDATE`-Sperrmodus verwendet.
- Verschiedene Faktoren veranlassen Anwendungs- oder Verbindungspools dazu, weitere Sitzungen zu öffnen, um dieselben Vorgänge auszuführen. Wenn neue Sitzungen versuchen, dieselben Zeilen zu ändern, kann die DB-Last stark ansteigen und `Lock:tuple` kann erscheinen.

Weitere Informationen finden Sie unter Sperren auf Zeilenebene in der PostgreSQL-Dokumentation.

Aktionen

Abhängig von den Ursachen Ihres Wait-Ereignisses empfehlen wir verschiedene Aktionen.

Themen

- Untersuchen Sie Ihre Anwendungslogik
- Finde die Blocker-Sitzung
- Reduzieren Sie Parallelität, wenn es hoch ist

- Beheben von Engpässen

Untersuchen Sie Ihre Anwendungslogik

Finden Sie heraus, ob sich eine Blocker-Sitzung schon lange im `idle in transaction`-Zustand befindet. Wenn ja, erwägen Sie, die Blocker-Sitzung als kurzfristige Lösung zu beenden. Sie können die Funktion `pg_terminate_backend` verwenden. Weitere Informationen zu dieser Funktion finden Sie unter Server-Signalisierungsfunktionen in der PostgreSQL-Dokumentation.

Gehen Sie für eine langfristige Lösung wie folgt vor:

- Passen Sie die Anwendungslogik an.
- Verwenden Sie den Parameter `idle_in_transaction_session_timeout`. Dieser Parameter beendet jede Sitzung mit einer offenen Transaktion, die länger als die angegebene Zeitspanne im Leerlauf ist. Weitere Informationen finden Sie unter Standardeinstellungen für Clientverbindungen in der PostgreSQL-Dokumentation.
- Verwenden Sie Autocommit so weit wie möglich. Weitere Informationen finden Sie unter SET AUTOCOMMIT in der PostgreSQL-Dokumentation.

Finde die Blocker-Sitzung

Identifizieren Sie während des `Lock:tuple`-Wait-Ereignisses den Blocker und die blockierte Sitzung, indem Sie herausfinden, welche Sperren voneinander abhängen. Weitere Informationen finden Sie unter Informationen zur Sperrabhängigkeit im PostgreSQL-Wiki.

Im folgenden Beispiel werden alle Sitzungen abgefragt, nach `tuple` gefiltert und nach `wait_time` sortiert.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
         concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
 blocked_wait_event,
         concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
 blocking_wait_event,
         blocked_activity.state AS blocked_state,
         blocking_activity.state AS blocking_state,
```

```
         blocked_locks.locktype AS blocked_locktype,
         blocking_locks.locktype AS blocking_locktype,
         blocked_activity.query AS blocked_statement,
         blocking_activity.query AS blocking_statement
   FROM pg_catalog.pg_locks blocked_locks
   JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
blocked_locks.pid
   JOIN pg_catalog.pg_locks blocking_locks
       ON blocking_locks.locktype = blocked_locks.locktype
       AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
       AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
       AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
       AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
       AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
       AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
       AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
       AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
       AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
       AND blocking_locks.pid != blocked_locks.pid
   JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
   WHERE NOT blocked_locks.GRANTED;
```

Reduzieren Sie Parallelität, wenn es hoch ist

Das `Lock:tuple`-Ereignis kann ständig auftreten, insbesondere in einer arbeitsreichen Zeit. Erwägen Sie in dieser Situation, die hohe Parallelität für sehr belegte Reihen zu reduzieren. Oft steuern nur wenige Zeilen eine Warteschlange oder die boolesche Logik, was diese Zeilen sehr ausgelastet macht.

Sie können die Parallelität reduzieren, indem Sie verschiedene Ansätze verwenden, die auf der Geschäftsanforderung, der Anwendungslogik und dem Workload-Typ basieren. Sie können z. B. Folgendes tun:

- Gestalten Sie Ihre Tabellen- und Datenlogik neu, um hohe Parallelität zu reduzieren.
- Ändern Sie die Anwendungslogik, um die hohe Parallelität auf Zeilenebene zu reduzieren.
- Nutzen und gestalten Sie Abfragen mit Sperren auf Zeilenebene.
- Verwenden Sie die `NOWAIT`-Klausel mit Wiederholungsvorgänge.
- Erwägen Sie, optimistische und hybridsperrende Logik-Parallelitätssteuerung zu nutzen.
- Überlegen Sie, die Isolationsstufe der Datenbank zu ändern.

Beheben von Engpässen

Das `Lock:tuple` kann bei Engpässen wie CPU-Aushungerungen oder maximaler Nutzung der Amazon EBS-Bandbreite auftreten. Um Engpässe zu verringern, sollten Sie die folgenden Ansätze berücksichtigen:

- Skalieren Sie Ihren Instance-Klassentyp hoch.
- Optimieren Sie ressourcenintensive Abfragen.
- Ändern Sie die Anwendungslogik.
- Archivieren Sie Daten, die selten zugegriffen wird.

LWLock:BufferMapping (LWLock:buffer_mapping)

Dieses Ereignis tritt ein, wenn eine Sitzung darauf wartet, einen Datenblock einem Puffer im gemeinsam genutzten Pufferpool zuzuordnen.

> **Note**
>
> Dieses Ereignis trägt in RDS für PostgreSQL Version 13 und höheren Versionen den Namen `LWLock:BufferMapping`. In PostgreSQL Version 12 und älteren Versionen lautet die Bezeichnung dieses Ereignisses `LWLock:buffer_mapping`.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Ursachen
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen sind für RDS für PostgreSQL Version 9.6 und höher relevant.

Kontext

Der freigegebene Pufferpool ist ein PostgreSQL-Speicherbereich, der alle Seiten enthält, die von Prozessen verwendet werden oder wurden. Wenn ein Prozess eine Seite benötigt, liest er die Seite

in den freigegebenen Pufferpool. Der Parameter `shared_buffers` legt die Größe des gemeinsam genutzten Puffers fest und reserviert einen Speicherbereich zum Speichern der Tabellen- und Indexseiten. Wenn Sie diesen Parameter ändern, stellen Sie sicher, dass Sie die Datenbank neu starten.

Das `LWLock:buffer_mapping`-Wait-Ereignis tritt in den folgenden Szenarien auf:

- Ein Prozess durchsucht die Puffertabelle nach einer Seite und erwirbt eine freigegebene Puffer-Mapping-Sperre.
- Ein Prozess lädt eine Seite in den Pufferpool und erwirbt eine exklusive Puffer-Mapping-Sperre.
- Ein Prozess entfernt eine Seite aus dem Pool und erwirbt eine exklusive Puffer-Mapping-Sperre.

Ursachen

Wenn dieses Ereignis mehr als normal auftritt, was möglicherweise auf ein Leistungsproblem hinweist, greift die Datenbank in und aus dem freigegebenen Pufferpool aus. Zu den typischen Ursachen zählen auch die Folgenden:

- Große Abfragen
- Aufgeblähte Indizes und Tabellen
- Vollständige Tabellenscans
- Eine gemeinsame Poolgröße, die kleiner als der Arbeitssatz ist

Aktionen

Abhängig von den Ursachen Ihres Warteereignisses empfehlen wir verschiedene Aktionen.

Themen

- Überwachen Sie pufferbezogene Metriken
- Bewerten Sie Ihre Indexierungsstrategie
- Reduzieren Sie die Anzahl der Puffer, die schnell zugewiesen werden müssen

Überwachen Sie pufferbezogene Metriken

Wenn `LWLock:buffer_mapping` auf Spitze wartet, untersuchen Sie die Puffertrefferquote. Sie können diese Metriken verwenden, um ein besseres Verständnis dafür zu erhalten, was im Puffer-Cache passiert. Untersuchen Sie die folgenden Metriken:

`blks_hit`

Diese Zählermetrik für Performance Insights gibt die Anzahl der Blöcke an, die aus dem freigegebenen Pufferpool abgerufen wurden. Nachdem das Wait-Ereignis `LWLock:buffer_mapping` aufgetreten ist, können Sie eine Spitze in `blks_hit` beobachten.

`blks_read`

Diese Zählermetrik für Performance Insights gibt die Anzahl der Blöcke an, für die I/O in den freigegebenen Pufferpool eingelesen werden mussten. Sie können im Vorfeld des `LWLock:buffer_mapping`-Warteereignisses eine Spitze in `blks_read` beobachten.

Bewerten Sie Ihre Indexierungsstrategie

Überprüfen Sie Folgendes, um zu bestätigen, dass Ihre Indexierungsstrategie die Leistung nicht beeinträchtigt:

Indexblähung

Stellen Sie sicher, dass Index und Tabellenaufblähungen nicht dazu führen, dass unnötige Seiten in den freigegebenen Puffer gelesen werden. Wenn Ihre Tabellen nicht verwendete Zeilen enthalten, sollten Sie die Daten archivieren und die Zeilen aus den Tabellen entfernen. Sie können dann die Indizes für die skalierten Tabellen neu erstellen.

Indizes für häufig verwendete Abfragen

Um festzustellen, ob Sie über die optimalen Indizes verfügen, überwachen Sie die Metriken der DB-Engine in Performance Insights. Die `tup_returned`-Metrik zeigt die Anzahl der gelesenen Zeilen an. Die `tup_fetched`-Metrik zeigt die Anzahl der an den Client zurückgegebenen Zeilen. Wenn `tup_returned` deutlich größer als `tup_fetched` ist, werden die Daten möglicherweise nicht richtig indiziert. Außerdem sind Ihre Tabellenstatistiken möglicherweise nicht aktuell.

Reduzieren Sie die Anzahl der Puffer, die schnell zugewiesen werden müssen

Um die `LWLock:buffer_mapping`-Warteereignisse zu reduzieren, versuchen Sie, die Anzahl der Puffer zu reduzieren, die schnell zugewiesen werden müssen. Eine Strategie besteht darin, kleinere Batch-Vorgänge durchzuführen. Möglicherweise können Sie kleinere Batches erreichen, indem Sie Ihre Tabellen partitionieren.

LWLock:BufferIO (IPC:BufferIO)

Das `LWLock:BufferIO` Ereignis tritt ein, wenn RDS for PostgreSQL darauf wartet, dass andere Prozesse ihre input/output (I/O-) Operationen beenden, wenn gleichzeitig versucht wird, auf eine Seite zuzugreifen. Sein Zweck besteht darin, dass dieselbe Seite in den freigegebenen Puffer eingelesen wird.

Themen

- Relevante Engine-Versionen
- Kontext
- Ursachen
- Aktionen

Relevante Engine-Versionen

Diese Warteereignisinformationen sind für alle Versionen von RDS für PostgreSQL relevant. Für Aurora PostgreSQL 12 und frühere Versionen wird dieses Warteereignis als lwlock:buffer_io bezeichnet, während es in der Version RDS für PostgreSQL 13 den Namen lwlock:bufferio trägt. Aus der Version RDS für PostgreSQL 14 wurde das BufferIO-Warteereignis von `LWLock` zum Warteereignistyp `IPC` (IPC:BufferIO) verschoben.

Kontext

Jeder gemeinsam genutzte Puffer hat eine I/O Sperre, die mit dem `LWLock:BufferIO` Wait-Ereignis verknüpft ist, und zwar jedes Mal, wenn ein Block (oder eine Seite) außerhalb des gemeinsam genutzten Pufferpools abgerufen werden muss.

Diese Sperre wird verwendet, um mehrere Sitzungen zu behandeln, die alle Zugriff auf denselben Block benötigen. Dieser Block muss von außerhalb des gemeinsam genutzten Pufferpools gelesen werden, der durch den `shared_buffers`-Parameter definiert wird.

Sobald die Seite innerhalb des Shared Buffer Pool gelesen wird, wird die `LWLock:BufferIO`-Sperre freigegeben.

> **Note**
>
> Das `LWLock:BufferIO`-Wait-Ereignis geht dem E/A:DataFileRead-Warteereignis voraus. Das `IO:DataFileRead`-Wait-Ereignis tritt auf, während Daten aus dem Speicher gelesen werden.

Weitere Informationen zu leichten Sperren finden Sie unter Übersicht über Sperren.

Ursachen

Häufige Gründe dafür, dass das `LWLock:BufferIO`-Ereignis in den Top-Wartezeiten angezeigt wird, sind die folgenden:

- Mehrere Backends oder Verbindungen versuchen, auf dieselbe Seite zuzugreifen, für die ebenfalls ein I/O Vorgang aussteht
- Das Verhältnis zwischen der Größe des gemeinsam genutzten Pufferpools (definiert durch den `shared_buffers`-Parameter) und der Anzahl der Puffer, die von der aktuellen Workload benötigt werden
- Die Größe des freigegebenen Pufferpools ist nicht gut mit der Anzahl der Seiten, die durch andere Vorgänge geräumt werden
- Große oder aufgeblähte Indizes, bei denen die Engine mehr Seiten als nötig in den freigegebenen Pufferpool lesen muss
- Mangel an Indizes, die die DB-Engine dazu zwingen, mehr Seiten aus den Tabellen als nötig zu lesen
- Checkpoints, die zu häufig auftreten oder zu viele geänderte Seiten leeren müssen
- Plötzliche Spitzen für Datenbankverbindungen, die versuchen, Vorgänge auf derselben Seite auszuführen

Aktionen

Abhängig von den Ursachen Ihres Wait-Ereignisses empfehlen wir verschiedene Aktionen:

- Stimmen Sie `max_wal_size` und `checkpoint_timeout` basierend auf der Spitzenzeit Ihrer Workload ab, wenn Sie sehen, dass `LWLock:BufferIO` mit Einbrüchen der Metrik `BufferCacheHitRatio` zusammenfällt. Identifizieren Sie dann, welche Abfrage sie möglicherweise verursachen könnte.
- Überprüfen Sie, ob Sie nicht verwendete Indizes haben, und entfernen Sie sie dann.
- Verwenden Sie partitionierte Tabellen (die auch partitionierte Indizes haben). Dies hilft, die Neuordnung des Index niedrig zu halten und ihre Auswirkungen zu reduzieren.
- Vermeiden Sie, Spalten unnötig zu indizieren.
- Verhindern Sie plötzliche Spitzen der Datenbankverbindung, indem Sie einen Verbindungspool verwenden.
- Beschränken Sie die maximale Anzahl von Verbindungen zur Datenbank als bewährte Methode

LWLock:buffer_content (BufferContent)

Das Ereignis `LWLock:buffer_content` tritt ein, wenn eine Sitzung darauf wartet, eine Datenseite im Speicher zu lesen oder zu schreiben, während eine andere Sitzung diese Seite zum Schreiben gesperrt hat. In RDS für PostgreSQL 13 und höher heißt dieses Warteereignis `BufferContent`.

Themen

- Unterstützte Engine-Versionen
- Context
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Context

Um Daten zu lesen oder zu manipulieren, greift PostgreSQL über Shared Memory Puffer darauf zu. Um aus dem Puffer zu lesen, erhält ein Prozess eine leichte Sperre (LWLock) für den Pufferinhalt im freigegebenen Modus. Um in den Puffer zu schreiben, wird diese Sperre im exklusiven Modus angezeigt. Gemeinsame Sperren ermöglichen es anderen Prozessen, gleichzeitig gemeinsame

Sperren für diesen Inhalt zu erwerben. Exklusive Sperren verhindern, dass andere Prozesse irgendeine Art von Sperre erhalten.

Die`LWLock:buffer_content`(`BufferContent`)-Ereignis zeigt an, dass mehrere Prozesse versuchen, den Inhalt eines bestimmten Puffers zu sperren.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn das `LWLock:buffer_content`(`BufferContent`)-Ereignis mehr als normal auftritt, was möglicherweise auf ein Leistungsproblem hinweist, sind die folgenden typischen Ursachen:

Die gleichzeitigen Aktualisierungen der gleichen Daten wurden erhöht

Es kann zu einer Zunahme der Anzahl gleichzeitiger Sitzungen mit Abfragen kommen, die denselben Pufferinhalt aktualisieren. Diese Behauptung kann bei Tabellen mit vielen Indizes ausgeprägter sein.

Workload-Daten befinden sich nicht im Speicher

Wenn sich Daten, die die aktive Workload verarbeitet, nicht im Speicher befinden, können diese Warteereignisse zunehmen. Dieser Effekt liegt daran, dass Prozesse, die Sperren halten, sie länger halten können, während sie Festplatten-I/O-Vorgänge ausführen.

Übermäßiger Einsatz von Fremdschlüsselbeschränkungen

Fremdschlüsseleinschränkungen können die Zeit erhöhen, die ein Prozess an einer Pufferinhaltssperre hält. Dieser Effekt liegt daran, dass Lesevorgänge eine gemeinsame Pufferinhaltssperre für den referenzierten Schlüssel erfordern, während dieser Schlüssel aktualisiert wird.

Aktionen

Abhängig von den Ursachen Ihres Wait-Ereignisses empfehlen wir verschiedene Aktionen. Sie können `LWLock:buffer_content`(`BufferContent`)-Ereignisse identifizieren, indem Sie Amazon RDS Performance Insights verwenden oder die Ansicht `pg_stat_activity` abfragen.

Themen

- Verbessern Sie die Effizienz im Speicher
- Reduzieren Sie die Verwendung von Fremdschlüsselbeschränkungen
- Entferne nicht verwendete Indizes

- Erhöhen der Cachegröße bei Verwendung von Sequenzen

Verbessern Sie die Effizienz im Speicher

Um die Wahrscheinlichkeit zu erhöhen, dass sich aktive Workload-Daten im Speicher befinden, partitionieren Sie Tabellen oder skalieren Sie Ihre Instance-Klasse hoch. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Reduzieren Sie die Verwendung von Fremdschlüsselbeschränkungen

Untersuchen Sie Workloads, bei denen eine hohe Anzahl von `LWLock:buffer_content`(`BufferContent`)-Wait-Ereignissen auf die Verwendung von Fremdschlüsseleinschränkungen auftritt. Entfernen Sie unnötige Fremdschlüsselbeschränkungen.

Entferne nicht verwendete Indizes

Identifizieren Sie bei Workloads mit einer hohen Anzahl von `LWLock:buffer_content`(`BufferContent`)-Wait-Ereignissen nicht verwendete Indizes und entfernen Sie sie.

Erhöhen der Cachegröße bei Verwendung von Sequenzen

Wenn Ihre Tabellen Sequenzen verwenden, erhöhen Sie die Cachegröße, um Konflikte auf Sequenzseiten und Indexseiten zu vermeiden. Jede Sequenz ist eine einzelne Seite im gemeinsam genutzten Arbeitsspeicher. Der vordefinierte Cache gilt pro Verbindung. Dies reicht möglicherweise nicht aus, um die Workload zu bewältigen, wenn viele gleichzeitige Sitzungen einen Sequenzwert erhalten.

LWLock:lock_manager (:lockmanager) LWLock

Dieses Ereignis tritt auf, wenn die Engine von RDS für PostgreSQL den Speicherbereich der gemeinsam genutzten Sperre verwaltet, um eine Sperre zuzuweisen, zu überprüfen und aufzuheben, wenn eine Fast-Path-Sperre nicht möglich ist.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen sind für RDS für PostgreSQL Version 9.6 und höher relevant. Für ältere Version von RDS für PostgreSQL als Version 13 lautet der Name dieses Warteereignisses `LWLock:lock_manager`. Für RDS für PostgreSQL Version 13 und höher lautet der Name dieses Warteereignisses `LWLock:lockmanager`.

Kontext

Wenn Sie eine SQL-Anweisung ausgeben, zeichnet RDS für PostgreSQL Sperren auf, um die Struktur, Daten und Integrität Ihrer Datenbank während gleichzeitiger Vorgänge zu schützen. Der Motor kann dieses Ziel mit einer schnellen Pfadsperre oder einer nicht schnellen Pfadsperre erreichen. Eine Pfadsperre, die nicht schnell ist, ist teurer und erzeugt mehr Overhead als eine schnelle Pfadsperre.

Schnelle Pfadsperre

Um den Overhead von Sperren zu reduzieren, die häufig genommen und freigegeben werden, aber selten in Konflikt geraten, können Backend-Prozesse eine schnelle Pfadsperrung verwenden. Die Datenbank verwendet diesen Mechanismus für Sperren, die die folgenden Kriterien erfüllen:

- Sie verwenden die STANDARD–Sperrmethode.
- Sie stellen eine Sperre für eine Datenbankbeziehung statt einer gemeinsamen Beziehung dar.
- Sie sind schwache Sperren, die wahrscheinlich nicht in Konflikt stehen.
- Die Engine kann schnell überprüfen, dass keine widersprüchlichen Sperren existieren können.

Die Engine kann keine schnelle Pfadsperre verwenden, wenn eine der folgenden Bedingungen erfüllt ist:

- Die Sperre erfüllt nicht die vorhergehenden Kriterien.
- Für den Backend-Prozess sind keine Slots mehr verfügbar.

Um Ihre Abfragen für Fast-Path-Sperren zu optimieren, können Sie die folgende Abfrage verwenden.

```
SELECT count(*), pid, mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
```

```
 GROUP BY 4,3,2
 ORDER BY pid, mode;
 count | pid  |      mode       | fastpath
-------+------+-----------------+----------
16 | 9185 | AccessShareLock | t
336 | 9185 | AccessShareLock | f
1 | 9185 | ExclusiveLock   | t
```

Die folgende Abfrage zeigt nur die Gesamtsumme in der gesamten Datenbank.

```
SELECT count(*), mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 3,2
 ORDER BY mode,1;
count |      mode       | fastpath
-------+-----------------+----------
16 | AccessShareLock | t
337 | AccessShareLock | f
1 | ExclusiveLock   | t
(3 rows)
```

Weitere Informationen zum Sperren von Fast Path finden Sie unter Fast Path in der README-Datei des PostgreSQL-Sperrmanagers und unter pg-locks in der PostgreSQL-Dokumentation.

Beispiel für ein Skalierungsproblem für den Sperrmanager

In diesem Beispiel speichert eine Tabelle mit dem Namen `purchases` Daten aus fünf Jahren, aufgeteilt nach Tagen. Jede Partition hat zwei Indizes. Die folgende Abfolge von Ereignissen tritt auf:

1. Sie fragen Daten für viele Tage ab, wodurch die Datenbank viele Partitionen lesen muss.
2. Die Datenbank erstellt einen Sperreintrag für jede Partition. Wenn Partitionsindizes Teil des Optimizer-Zugriffspfads sind, erstellt die Datenbank auch für sie einen Sperreintrag.
3. Wenn die Anzahl der angeforderten Sperreneinträge für denselben Backend-Prozess höher als 16 ist, was dem Wert von `FP_LOCK_SLOTS_PER_BACKEND` entspricht, verwendet der Sperrenmanager die Sperrmethode ohne Fast Path.

Moderne Anwendungen haben möglicherweise Hunderte von Sitzungen. Wenn gleichzeitige Sitzungen das übergeordnete Element ohne ordnungsgemäßen Schnitt von Partitionen abfragen, erstellt die Datenbank möglicherweise Hunderte oder sogar Tausende von nicht schnellen

Pfadsperren. In der Regel tritt das `LWLock:lock_manager` Wait-Ereignis auf, wenn diese Parallelität höher als die Anzahl von v CPUs ist.

> **Note**
>
> Das Wait-Ereignis `LWLock:lock_manager` hat nichts mit der Anzahl der Partitionen oder Indizes in einem Datenbankschema zu tun. Stattdessen hängt es mit der Anzahl der nicht schnellen Pfadsperren zusammen, die die Datenbank steuern muss.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Wenn das `LWLock:lock_manager` häufiger als normal auftritt, was möglicherweise auf ein Leistungsproblem hinweist, sind die wahrscheinlichsten Ursachen für plötzliche Spitzen wie folgt:

- Gleichzeitige aktive Sitzungen führen Abfragen aus, die keine schnellen Pfadsperren verwenden. Diese Sitzungen überschreiten auch die maximale vCPU.
- Eine große Anzahl gleichzeitiger aktiver Sitzungen greift auf eine stark partitionierte Tabelle zu. Jede Partition hat mehrere Indizes.
- Die Datenbank erlebt einen Verbindungssturm. Standardmäßig erzeugen einige Anwendungen und Connection Pool-Software mehr Verbindungen, wenn die Datenbank langsam ist. Diese Praxis verschlimmert das Problem. Optimieren Sie Ihre Connection Pool-Software so, dass keine Verbindungsstürme auftreten.
- Eine große Anzahl von Sitzungen fragt eine übergeordnete Tabelle ab, ohne Partitionen zu beschneiden.
- Eine Datendefinitionssprache (DDL), Datenmanipulationssprache (DML) oder ein Wartungsbefehl sperrt ausschließlich eine Beleg-Beziehung oder Tupel, auf die häufig zugegriffen oder geändert werden.

Aktionen

Wenn das `CPU`-Wait-Ereignis auftritt, weist dies nicht unbedingt auf ein Leistungsproblem hin. Reagieren Sie auf dieses Ereignis nur, wenn sich die Leistung verschlechtert und dieses Wait-Ereignis die DB-Last dominiert.

Themen

- Verwenden Sie das Beschneiden von Partitionen

- Entfernen unnötiger Indizes
- Optimieren Sie Ihre Abfragen für schnelles Pfadsperren
- Tune auf andere Warteereignisse
- Reduzieren von Hardware-Engpässen
- Verwenden eines Verbindungs-Poolers
- Durchführen eines Upgrades Ihrer Version von RDS für PostgreSQL

Verwenden Sie das Beschneiden von Partitionen

Die Partitionsbereinigung ist eine Strategie zur Abfrageoptimierung für deklarativ partitionierte Tabellen, die nicht benötigte Partitionen von Tabellenscans ausschließt und dadurch die Leistung verbessert. Das Beschneiden der Partition ist standardmäßig aktiviert. Wenn es ausgeschaltet ist, schalten Sie es wie folgt ein.

```
SET enable_partition_pruning = on;
```

Abfragen können die Partitionsbereinigung nutzen, wenn ihre `WHERE`-Klausel die für die Partitionierung verwendete Spalte enthält. Weitere Informationen finden Sie unter Partitionsbereinigung in der PostgreSQL-Dokumentation.

Entfernen unnötiger Indizes

Ihre Datenbank enthält möglicherweise nicht verwendete oder selten verwendete Indizes. Wenn ja, erwägen Sie, sie zu löschen. Führen Sie eine der folgenden Aufgaben aus:

- Erfahren Sie, wie Sie unnötige Indizes finden, indem Sie Ungenutzte Indizes im PostgreSQL-Wiki lesen.
- Führen Sie PG Collector aus. Dieses SQL-Skript sammelt Datenbankinformationen und präsentiert sie in einem konsolidierten HTML-Bericht. Überprüfen Sie den Abschnitt „Unbenutzte Indizes“. Weitere Informationen finden Sie unter pg-collector im Labs-Repository. AWS GitHub

Optimieren Sie Ihre Abfragen für schnelles Pfadsperren

Um herauszufinden, ob Ihre Abfragen Fast Path Locking verwenden, fragen Sie die `fastpath`-Spalte in der `pg_locks`-Tabelle ab. Wenn Ihre Abfragen keine schnelle Pfadsperre verwenden, versuchen Sie, die Anzahl der Beziehungen pro Abfrage auf weniger als 16 zu reduzieren.

Tune auf andere Warteereignisse

Wenn `LWLock:lock_manager` in der Liste der Top-Waits an erster oder zweiter Stelle steht, überprüfen Sie, ob die folgenden Wait-Ereignisse auch in der Liste erscheinen:

- `Lock:Relation`
- `Lock:transactionid`
- `Lock:tuple`

Wenn die vorhergehenden Ereignisse in der Liste hoch erscheinen, sollten Sie zuerst diese Warteereignisse optimieren. Diese Ereignisse können ein Treiber für `LWLock:lock_manager`.

Reduzieren von Hardware-Engpässen

Möglicherweise haben Sie einen Hardware-Engpass wie CPU-Hunger oder maximale Auslastung Ihrer Amazon EBS-Bandbreite. Ziehen Sie in diesen Fällen die Verringerung der Hardware-Engpässe in Betracht. Berücksichtigen Sie die folgenden Aktionen:

- Skalieren Sie Ihre Instance-Klasse hoch.
- Optimieren Sie Abfragen, die große Mengen an CPU und Speicher verbrauchen.
- Ändern Sie Ihre Anwendungslogik.
- Archiviere deine Daten.

Weitere Informationen zu CPU, Arbeitsspeicher und EBS-Netzwerkbandbreite finden Sie unter Amazon RDS-Instance-Typen.

Verwenden eines Verbindungs-Poolers

Wenn Ihre Gesamtzahl aktiver Verbindungen die maximale vCPU überschreitet, benötigen mehr Betriebssystemprozesse CPU, als Ihr Instance-Typ unterstützen kann. Ziehen Sie in diesem Fall die Verwendung oder Abstimmung eines Verbindungspool in Betracht. Weitere Informationen über das v CPUs für Ihren Instance-Typ finden Sie unter Amazon RDS-Instance-Typen.

Weitere Informationen zum Verbindungspooling finden Sie in den folgenden Ressourcen:

- Amazon RDS-Proxy
- pgbouncer
- Verbindungspools und Datenquellen in der PostgreSQL-Dokumentation

Durchführen eines Upgrades Ihrer Version von RDS für PostgreSQL

Wenn Ihre aktuelle Version von RDS für PostgreSQL niedriger als 12 ist, aktualisieren Sie auf Version 12 oder höher. Die PostgreSQL-Versionen 12 und höher haben einen verbesserten Partitionsmechanismus. Weitere Informationen zu Version 12 finden Sie in den Versionshinweisen zu PostgreSQL 12.0. Weitere Informationen zum Aktualisieren von RDS für PostgreSQL finden Sie unter Upgrades der RDS for PostgreSQL-DB-Engine.

LWLock:pg_stat_statements

Das LWLock Warteereignis:PG_STAT_STATEMENTS tritt ein, wenn die Erweiterung die Hashtabelle, die SQL-Anweisungen verfolgt, exklusiv sperrt. `pg_stat_statements` Dies passiert in den folgenden Szenarien:

- Wenn die Anzahl der verfolgten Anweisungen den konfigurierten `pg_stat_statements.max` Parameterwert erreicht und Platz für weitere Einträge geschaffen werden muss, sortiert die Erweiterung die Anzahl der Aufrufe, entfernt die 5% der am wenigsten ausgeführten Anweisungen und füllt den Hash erneut mit den verbleibenden Einträgen auf.
- When `pg_stat_statements` führt eine `garbage collection` Operation an der `pgss_query_texts.stat` Datei auf der Festplatte durch und schreibt die Datei neu.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL PostgreSQL unterstützt.

Kontext

Grundlegendes zur Erweiterung pg_stat_statements — Die Erweiterung pg_stat_statements verfolgt Statistiken zur Ausführung von SQL-Anweisungen in einer Hashtabelle. Die

Erweiterung verfolgt SQL-Anweisungen bis zu dem durch den Parameter definierten Grenzwert. `pg_stat_statements.max` Dieser Parameter bestimmt die maximale Anzahl von Anweisungen, die verfolgt werden können, was der maximalen Anzahl von Zeilen in der Ansicht pg_stat_statements entspricht.

Persistenz von Anweisungsstatistiken — Die Erweiterung behält die Anweisungsstatistiken bei jedem Neustart der Instanz bei, indem sie:

- Daten in eine Datei namens pg_stat_statements.stat schreiben
- Verwendung des Parameters pg_stat_statements.save zur Steuerung des Persistenzverhaltens

Wenn pg_stat_statements.save auf Folgendes gesetzt ist:

- on (Standard): Statistiken werden beim Herunterfahren gespeichert und beim Serverstart erneut geladen
- aus: Statistiken werden weder beim Herunterfahren gespeichert noch beim Serverstart neu geladen

Textspeicher für Abfragen — Die Erweiterung speichert den Text nachverfolgter Abfragen in einer Datei mit dem Namen`pgss_query_texts.stat`. Diese Datei kann auf das Doppelte der durchschnittlichen Größe aller verfolgten SQL-Anweisungen anwachsen, bevor die Speicherbereinigung erfolgt. Die Erweiterung erfordert eine exklusive Sperre für die Hashtabelle während der Bereinigungsvorgänge und beim Umschreiben `pgss_query_texts.stat` der Datei.

Prozess zur Freigabe von Kontoauszügen — Wenn die Anzahl nachverfolgter Kontoauszüge das `pg_stat_statements.max` Limit erreicht und neue Kontoauszüge nachverfolgt werden müssen, erfolgt die Erweiterung wie folgt:

- Erhält eine exklusive Sperre (:pg_stat_statementsLWLock) für die Hashtabelle.
- Lädt vorhandene Daten in den lokalen Speicher.
- Führt eine Schnellsortierung auf der Grundlage der Anzahl der Aufrufe durch.
- Entfernt die am wenigsten aufgerufenen Anweisungen (unterste 5%).
- Füllt die Hashtabelle erneut mit den verbleibenden Einträgen auf.

Überwachung der Freigabe von Anweisungen — In PostgreSQL 14 und höher können Sie die Freigabe von Anweisungen mithilfe der Ansicht pg_stat_statements_info überwachen. Diese

Ansicht enthält eine Dealloc-Spalte, in der angezeigt wird, wie oft die Zuweisung von Anweisungen aufgehoben wurde, um Platz für neue Anweisungen zu schaffen

Wenn die Freigabe von Anweisungen häufig erfolgt, führt dies zu einer häufigeren Garbage-Collection der Datei auf der `pgss_query_texts.stat` Festplatte.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Zu den typischen Ursachen für erhöhte `LWLock:pg_stat_statements` Wartezeiten gehören:

- Eine Zunahme der Anzahl von eindeutigen Abfragen, die von der Anwendung verwendet werden.
- Der `pg_stat_statements.max` Parameterwert ist klein im Vergleich zur Anzahl der verwendeten eindeutigen Abfragen.

Aktionen

Abhängig von den Ursachen Ihres Warteereignisses empfehlen wir verschiedene Aktionen. Sie können `LWLock:pg_stat_statements` Ereignisse mithilfe von Amazon RDS Performance Insights oder durch Abfragen der Ansicht `pg_stat_activity` identifizieren.

Passen Sie die folgenden `pg_stat_statements` Parameter an, um das Tracking-Verhalten zu steuern und zu LWLock reduzieren:pg_stat_-Anweisungen warten auf Ereignisse.

Themen

- Deaktivieren Sie den Parameter pg_stat_statements.track
- Erhöhen Sie den Parameter pg_stat_statements.max
- Deaktivieren Sie den Parameter pg_stat_statements.track_utility

Deaktivieren Sie den Parameter pg_stat_statements.track

Wenn sich das LWLock Wait-Event:pg_stat_statements negativ auf die Datenbankleistung auswirkt und eine schnelle Lösung erforderlich ist, bevor die Ansicht weiter analysiert wird, um die Ursache `pg_stat_statements` zu ermitteln, kann der Parameter deaktiviert werden, indem er auf gesetzt wird. `pg_stat_statements.track` `none` Dadurch wird die Erfassung von Kontoauszugsstatistiken deaktiviert.

Erhöhen Sie den Parameter pg_stat_statements.max

Erhöhen Sie den Wert des Parameters, um die Deallokation zu reduzieren und die Garbage-Collection der `pgss_query_texts.stat` Datei auf der Festplatte zu minimieren. `pg_stat_statements.max` Der Standardwert ist `5,000`.

> **Note**
>
> Der `pg_stat_statements.max` Parameter ist statisch. Sie müssen Ihre DB-Instance neu starten, um Änderungen an diesem Parameter zu übernehmen.

Deaktivieren Sie den Parameter pg_stat_statements.track_utility

Sie können die Ansicht pg_stat_statements analysieren, um festzustellen, welche Dienstprogrammbefehle die meisten Ressourcen verbrauchen, die von erfasst werden. `pg_stat_statements`

Der `pg_stat_statements.track_utility` Parameter steuert, ob das Modul Dienstprogrammbefehle verfolgt, zu denen alle Befehle außer SELECT, INSERT, UPDATE, DELETE und MERGE gehören. Dieser Parameter ist standardmäßig auf on festgelegt.

Wenn Ihre Anwendung beispielsweise viele Savepoint-Abfragen verwendet, die von Natur aus einzigartig sind, kann dies die Deallocation von Anweisungen erhöhen. Um dieses Problem zu beheben, können Sie den `pg_stat_statements.track_utility` Parameter deaktivieren, sodass Savepoint-Abfragen nicht mehr verfolgt `pg_stat_statements` werden.

> **Note**
>
> Der `pg_stat_statements.track_utility` Parameter ist ein dynamischer Parameter. Sie können seinen Wert ändern, ohne Ihre Datenbankinstanz neu zu starten.

Example Beispiel für eindeutige Speicherpunktabfragen in pg_stat_statements

```
                    query                         |       queryid
--------------------------------------------------+---------------------
 SAVEPOINT JDBC_SAVEPOINT_495701                  | -7249565344517699703
 SAVEPOINT JDBC_SAVEPOINT_1320                    | -1572997038849006629
```

```
SAVEPOINT JDBC_SAVEPOINT_26739                  |  54791337410474486
SAVEPOINT JDBC_SAVEPOINT_1294466                |  8170064357463507593
ROLLBACK TO SAVEPOINT JDBC_SAVEPOINT_65016      | -33608214779996400
SAVEPOINT JDBC_SAVEPOINT_14185                  | -2175035613806809562
SAVEPOINT JDBC_SAVEPOINT_45837                  | -6201592986750645383
SAVEPOINT JDBC_SAVEPOINT_1324                   |  6388797791882029332
```

PostgreSQL 17 führt mehrere Verbesserungen für die Befehlsverfolgung von Dienstprogrammen ein:

- Savepoint-Namen werden jetzt als Konstanten angezeigt.
- Die globale Transaktion IDs (GIDs) von zweiphasigen Commit-Befehlen wird jetzt als Konstanten angezeigt.
- Die Namen von DEALLOCATE-Anweisungen werden als Konstanten angezeigt.
- CALL-Parameter werden jetzt als Konstanten angezeigt.

Timeout:PgSleep

Das `Timeout:PgSleep`-Ereignis tritt ein, wenn ein Serverprozess die `pg_sleep`-Funktion aufgerufen hat und auf das Ablaufen des Sleep-Timeouts wartet.

Themen

- Unterstützte Engine-Versionen
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Dieses Wait-Ereignis tritt auf, wenn eine Anwendung, eine gespeicherte Funktion oder ein Benutzer eine SQL-Anweisung ausgibt, die eine der folgenden Funktionen aufruft:

- `pg_sleep`
- `pg_sleep_for`
- `pg_sleep_until`

Die vorhergehenden Funktionen verzögern die Ausführung, bis die angegebene Anzahl von Sekunden verstrichen ist. Beispielsweise pausiert `SELECT pg_sleep(1)` für 1 Sekunde. Weitere Informationen finden Sie unter Ausführung verzögern in der PostgreSQL-Dokumentation.

Aktionen

Identifizieren Sie die Anweisung, die die `pg_sleep`-Funktion ausgeführt hat. Überprüfen Sie, ob die Verwendung der Funktion angemessen ist.

Timeout:VacuumDelay

Das `Timeout:VacuumDelay`-Ereignis weist darauf hin, dass das Kostenlimit für Bereinigungs-I/O überschritten wurde und dass der Bereinigungsprozess in den Ruhezustand versetzt wurde. Bereinigungsoperationen werden für die im jeweiligen Kostenverzögerungsparameter angegebene Dauer unterbrochen und setzen ihren Betrieb danach fort. Für den manuellen Bereinigungsbefehl ist die Verzögerung im `vacuum_cost_delay`-Parameter angegeben. Für den Selbstbereinigungs-Daemon ist die Verzögerung im Parameter `autovacuum_vacuum_cost_delay parameter.` festgelegt.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Wahrscheinliche Ursachen für erhöhte Wartezeiten
- Aktionen

Unterstützte Engine-Versionen

Diese Warteereignisinformationen werden für alle Versionen von RDS für PostgreSQL unterstützt.

Kontext

PostgreSQL verfügt sowohl über einen Selbstbereinigungs-Daemon als auch über einen manuellen Bereinigungsbefehl. Der Selbstbereinigungsprozess ist für DB-Instances von RDS für PostgreSQL standardmäßig aktiviert. Der manuelle Bereinigungsbefehl wird je nach Bedarf verwendet, um beispielsweise Tabellen von toten Tupeln zu bereinigen oder neue Statistiken zu generieren.

Während des Bereinigungsvorgangs verwendet PostgreSQL einen internen Zähler, um die geschätzten Kosten zu verfolgen, während das System verschiedene I/O-Operationen durchführt.

Wenn der Zähler den im Kostenlimitparameter angegebenen Wert erreicht, hält der Prozess, der den Vorgang ausführt, für die kurze Dauer an, die im Kostenverzögerungsparameter angegeben ist. Dann setzt er den Zähler zurück und den Betrieb fort.

Der Bereinigungsprozess verfügt über Parameter, mit denen der Ressourcenverbrauch reguliert werden kann. Die Selbstbereinigung und der manuelle Bereinigungsbefehl haben jeweils eigene Parameter zur Einstellung des Kostenlimits. Sie haben auch jeweils eigene Parameter, um eine Kostenverzögerung festzulegen, d. h. eine Zeitspanne, in der die Bereinigung in den Ruhezustand versetzt wird, wenn das Limit erreicht ist. Auf diese Weise fungiert der Kostenverzögerungsparameter als Drosselungsmechanismus für den Ressourcenverbrauch. Die Beschreibung dieser Parameter finden Sie in den folgenden Listen.

Parameter, die die Drosselung des Selbstbereinigungs-Daemons beeinflussen

- `autovacuum_vacuum_cost_limit` – Gibt den Kostenlimitwert an, der bei Selbstbereinigungsoperationen verwendet werden soll. Wenn Sie die Einstellung für diesen Parameter erhöhen, kann der Bereinigungsprozess mehr Ressourcen verbrauchen und das `Timeout:VacuumDelay`-Warteereignis wird verringert.
- `autovacuum_vacuum_cost_delay` – Gibt den Kostenverzögerungswert an, der bei Selbstbereinigungsoperationen verwendet werden soll. Der Standardwert ist 2 Millisekunden. Wenn Sie den Verzögerungsparameter auf 0 festlegen, wird der Drosselungsmechanismus deaktiviert, sodass das `Timeout:VacuumDelay`-Warteereignis nicht angezeigt wird.

Weitere Informationen finden Sie unter Selbstbereinigung in der PostgreSQL-Dokumentation.

Parameter, die die Drosselung des manuellen Bereinigungsprozesses beeinflussen

- `vacuum_cost_limit` – Die Schwelle, bei der der Bereinigungsvorgang in den Ruhezustand versetzt wird. In der Standardeinstellung ist das Limit 200. Diese Zahl steht für die kumulierten Kostenschätzungen für zusätzliche I/O, die von verschiedenen Ressourcen benötigt werden. Wenn Sie diesen Wert erhöhen, verringert sich die Anzahl der `Timeout:VacuumDelay`-Warteereignisse.
- `vacuum_cost_delay` – Die Zeitspanne, die der Bereinigungsprozess ruht, wenn das Bereinigungskostenlimit erreicht ist. Die Standardeinstellung ist 0, was bedeutet, dass diese Funktion deaktiviert ist. Sie können diese Einstellung auf einen ganzzahligen Wert festlegen, um die Anzahl der Millisekunden anzugeben und damit diese Funktion zu aktivieren. Wir empfehlen jedoch, die Standardeinstellung beizubehalten.

Weitere Informationen zum `vacuum_cost_delay`-Parameter finden Sie unter Ressourcennutzung in der PostgreSQL-Dokumentation.

Weitere Informationen zur Konfiguration und Verwendung der Selbstbereinigung mit RDS für PostgreSQL finden Sie unter .

Wahrscheinliche Ursachen für erhöhte Wartezeiten

Die `Timeout:VacuumDelay` wird durch das Gleichgewicht zwischen Kostenlimitparametern (`vacuum_cost_limit`, `autovacuum_vacuum_cost_limit`) und Kostenverzögerungsparametern (`vacuum_cost_delay`, `autovacuum_vacuum_cost_delay`) beeinflusst, die die Dauer des Ruhezustands des Bereinigungsprozesses steuern. Durch die Erhöhung eines Parameterwerts für das Kostenlimit können mehr Ressourcen vom Bereinigungsprozess verbraucht werden, bevor er in den Ruhezustand versetzt wird. Das führt zu weniger `Timeout:VacuumDelay`-Warteereignissen. Wenn Sie einen der Verzögerungsparameter erhöhen, tritt das `Timeout:VacuumDelay`-Warteereignis häufiger und für längere Zeiträume auf.

Die `autovacuum_max_workers`-Parametereinstellung kann auch die Anzahl der `Timeout:VacuumDelay` erhöhen. Jeder zusätzliche Selbstbereinigungs-Worker-Prozess trägt zum internen Zählermechanismus bei, sodass das Limit schneller erreicht werden kann als mit einem einzelnen Selbstbereinigungs-Worker-Prozess. Da das Kostenlimit schneller erreicht wird, tritt die Kostenverzögerung häufiger in Kraft, was zu mehr`Timeout:VacuumDelay`-Warteereignissen führt. Weitere Informationen finden Sie unter autovacuum_max_workers in der PostgreSQL-Dokumentation.

Große Objekte, z. B. 500 GB oder mehr, lösen dieses Warteereignis ebenfalls aus, da es einige Zeit dauern kann, bis der Bereinigungsprozess die Verarbeitung großer Objekte abgeschlossen hat.

Aktionen

Wenn die Bereinigungsoperationen wie erwartet abgeschlossen werden, ist keine Abhilfe erforderlich. Mit anderen Worten, dieses Warteereignis weist nicht zwingend auf ein Problem hin. Es gibt an, dass der Bereinigungsprozess für den im Verzögerungsparameter angegebenen Zeitraum in den Ruhezustand versetzt wird, sodass Ressourcen für andere Prozesse verwendet werden können, die abgeschlossen werden müssen.

Wenn Sie möchten, dass Bereinigungsoperationen schneller abgeschlossen werden, können Sie die Verzögerungsparameter niedriger ansetzen. Dadurch wird die Ruhezeit des Bereinigungsprozesses verkürzt.

Optimiert RDSfür Postgre SQL mit proaktiven Erkenntnissen von Amazon DevOps Guru

DevOpsGuru Proactive Insights erkennt Bedingungen in Ihren , die Probleme verursachen können, und informiert Sie darüber, bevor sie auftreten. Proaktive Einblicke können Sie auf einen seit langem laufenden Leerlauf in der Transaktionsverbindung aufmerksam machen. Weitere Informationen zur Behebung von Problemen mit lang andauerndem Leerlauf bei Transaktionsverbindungen finden Sie unter Die Datenbank läuft seit langem inaktiv in Transaktionsverbindung

DevOpsGuru kann Folgendes tun:

- Vermeiden vieler häufig auftretender Datenbankprobleme durch Abgleich der Datenbankkonfiguration mit den allgemein empfohlenen Einstellungen.
- Warnen vor kritischen Problemen in der Flotte, die, wenn sie nicht überprüft werden, später zu größeren Problemen führen können.
- Benachrichtigung bei neu erkannten Problemen.

Jeder proaktive Einblick beinhaltet eine Analyse der Problemursache und Empfehlungen für Korrekturmaßnahmen.

Weitere Informationen zu Amazon DevOps Guru für Amazon RDS finden Sie unterAnalysieren von Leistungsanomalien mit Amazon DevOps Guru für Amazon RDS.

Die Datenbank läuft seit langem inaktiv in Transaktionsverbindung

Eine Verbindung zur Datenbank befindet sich sein mehr als 1800 Sekunden im Status `idle in transaction`.

Themen

- Unterstützte Engine-Versionen
- Kontext
- Mögliche Ursachen für dieses Problem
- Aktionen
- Relevante Metriken

Unterstützte Engine-Versionen

Diese Insight-Informationen werden für alle Versionen von RDSfür Postgre SQL unterstützt.

Kontext

Eine Transaktion im Status `idle in transaction` kann Sperren enthalten, die andere Abfragen blockieren. Sie kann auch verhindern, dass `VACUUM` (einschließlich Autovacuum) tote Zeilen bereinigt, was zu einer Überlastung von Index oder Tabellen oder zu einem Wraparound der Transaktions-ID führt.

Mögliche Ursachen für dieses Problem

Eine Transaktion, die in einer interaktiven Sitzung mit einemCOMMIT, BEGIN START TRANSACTION oder -Befehl initiiert wurde oder END nicht beendet wurde. ROLLBACK Dadurch wird die Transaktion in den Status `idle in transaction` versetzt.

Aktionen

Sie können ungenutzte Transaktionen finden, indem Sie `pg_stat_activity` abfragen.

Führen Sie in Ihrem SQL Client die folgende Abfrage aus, um alle Verbindungen im `idle in transaction` Status aufzulisten und sie nach Dauer zu ordnen:

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Abhängig von den Ursachen Ihres Einblicks empfehlen wir verschiedene Aktionen.

Themen

- Transaktion beenden
- Die Verbindung beenden
- Konfigurieren Sie den Parameter idle_in_transaction_session_timeout
- Überprüfen Sie den Status AUTOCOMMIT
- Überprüfen Sie die Transaktionslogik in Ihrem Anwendungscode

Transaktion beenden

Wenn Sie eine Transaktion in einer interaktiven Sitzung mit BEGIN oder initiieren STARTTRANSACTION, wechselt sie in den `idle in transaction` Status. Sie bleibt in diesem Zustand, bis Sie die Transaktion beenden, indem Sie einenCOMMIT,ROLLBACK, END Befehl ausgeben oder die Verbindung vollständig trennen, um die Transaktion rückgängig zu machen.

Die Verbindung beenden

Beenden Sie die Verbindung mit einer inaktiven Transaktion mit der folgenden Abfrage:

```
SELECT pg_terminate_backend(pid);
```

pid ist die Prozess-ID der Verbindung.

Konfigurieren Sie den Parameter idle_in_transaction_session_timeout

Stellen Sie den Parameter `idle_in_transaction_session_timeout` in der Parametergruppe ein. Der Vorteil der Konfiguration dieses Parameters besteht darin, dass kein manueller Eingriff erforderlich ist, um die lang dauernde inaktive Transaktion zu beenden. Weitere Informationen zu diesem Parameter finden Sie in der SQL Postgre-Dokumentation.

Die folgende Meldung wird in der SQL Postgre-Protokolldatei gemeldet, nachdem die Verbindung beendet wurde, wenn sich eine Transaktion länger als die angegebene Zeit im Status idle_in_transaction befindet.

```
FATAL: terminating connection due to idle in transaction timeout
```

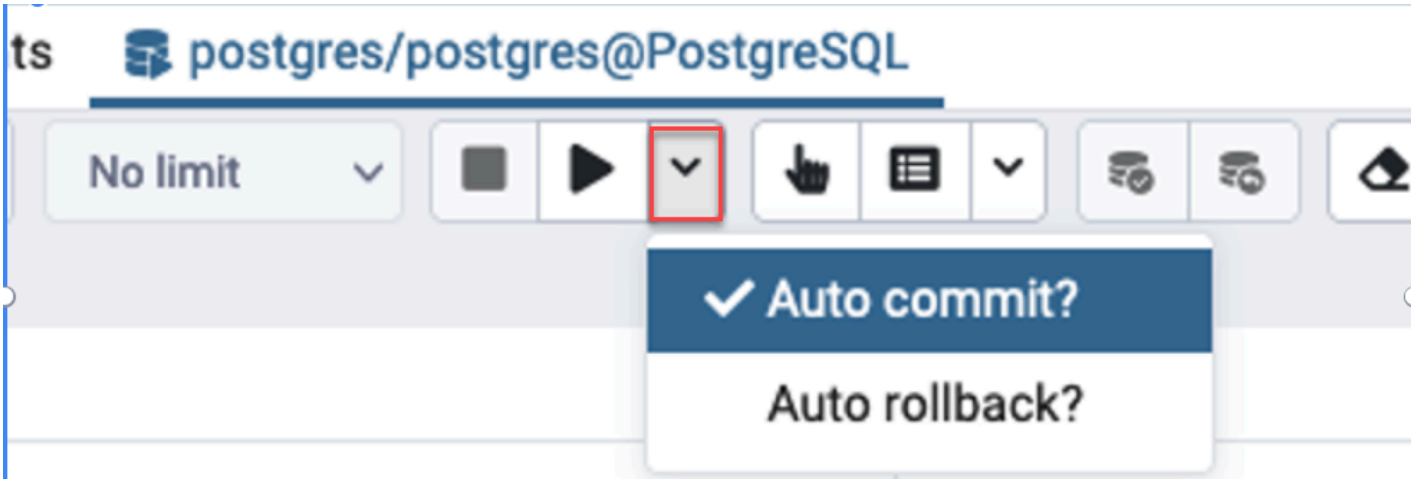
Überprüfen Sie den Status AUTOCOMMIT

AUTOCOMMITist standardmäßig aktiviert. Wenn dies jedoch versehentlich im Client deaktiviert wurde, stellen Sie sicher, dass Sie es wieder aktivieren.

- Führen Sie in Ihrem psql-Client den folgenden Befehl aus:

  ```
  postgres=> \set AUTOCOMMIT on
  ```

- Schalten Sie es in pgadmin ein, indem Sie die AUTOCOMMIT Option mit dem Abwärtspfeil auswählen.



Überprüfen Sie die Transaktionslogik in Ihrem Anwendungscode

Untersuchen Sie Ihre Anwendungslogik auf mögliche Probleme. Berücksichtigen Sie die folgenden Aktionen:

- Prüfen Sie, ob der JDBC Auto-Commit in Ihrer Anwendung auf true gesetzt ist. Erwägen Sie auch, explizite `COMMIT`-Befehle in Ihrem Code zu verwenden.
- Überprüfen Sie Ihre Fehlerbehandlungslogik, um festzustellen, ob eine Transaktion nach einem Fehler geschlossen wird.
- Prüfen Sie, ob Ihre Anwendung lange braucht, um die von einer Abfrage zurückgegebenen Zeilen zu verarbeiten, während die Transaktion geöffnet ist. Wenn dies der Fall ist, erwägen Sie, die Anwendung so zu codieren, dass die Transaktion geschlossen wird, bevor die Zeilen verarbeitet werden.
- Prüfen Sie, ob eine Transaktion viele lang laufende Operationen enthält. Wenn dies der Fall ist, teilen Sie eine einzelne Transaktion in mehrere Transaktionen auf.

Relevante Metriken

Die folgenden PI-Metriken beziehen sich auf diesen Einblick:

- idle_in_transaction_count – Anzahl der Sitzungen im Status `idle in transaction`.
- idle_in_transaction_max_time – Die Dauer der am längsten laufenden Transaktion im Status `idle in transaction`.

Verwenden von PostgreSQL-Erweiterungen mit Amazon RDS for PostgreSQL

Sie können die Funktionalität von PostgreSQL erweitern, indem Sie eine Vielzahl von Erweiterungen und Modulen installieren. Um beispielsweise mit Geodaten zu arbeiten, können Sie die PostGIS-Erweiterung installieren und verwenden. Weitere Informationen finden Sie unter Verwaltung von Geodaten mit der GIS Post-Erweiterung. Wenn Sie als anderes Beispiel die Dateneingabe für sehr große Tabellen verbessern möchten, können Sie die Partitionierung Ihrer Daten in Betracht ziehen, indem Sie die `pg_partman`-Erweiterung verwenden. Weitere Informationen hierzu finden Sie unter Verwalten von PostgreSQL-Partitionen mit der Erweiterung pg_partman.

> **Note**
>
> RDS for PostgreSQL unterstützt Trusted Language Extensions for PostgreSQL über die `pg_tle` Erweiterung, die Sie Ihrer DB-Instance hinzufügen können. Mithilfe dieser Erweiterung können Entwickler ihre eigenen PostgreSQL-Erweiterungen in einer sicheren Umgebung erstellen, was die Setup- und Konfigurationsanforderungen vereinfacht. Weitere Informationen zu den Versionen von RDS für PostgreSQL, die `pg_tle` Erweiterungen unterstützen, und weitere Informationen finden Sie unter. Arbeiten mit Trusted Language Extensions für PostgreSQL

In einigen Fällen bietet es sich an, anstatt eine Erweiterung zu installieren, ein bestimmtes Modul zur Liste der `shared_preload_libraries` in der benutzerdefinierten DB-Parametergruppe Ihrer DB-Instance von RDS für PostgreSQL hinzufügen. In der Regel lädt die standardmäßige DB-Cluster-Parametergruppe nur die `pg_stat_statements`. Es stehen jedoch weitere Module zur Verfügung, die der Liste hinzugefügt werden können. Sie können beispielsweise Planungsfunktionen hinzufügen, indem Sie das `pg_cron`-Modul hinzufügen, wie unter Planen der Wartung mit der PostgreSQL-Erweiterung pg_cron beschrieben. Als weiteres Beispiel können Sie Abfrageausführungspläne protokollieren, indem Sie das `auto_explain`-Modul laden. Weitere Informationen finden Sie im AWS Knowledge Center unter Protokollieren von Ausführungsplänen von Abfragen.

Abhängig von Ihrer Version von RDS for PostgreSQL erfordert die Installation einer Erweiterung möglicherweise `rds_superuser`-Berechtigungen wie folgt:

- Für RDS für PostgreSQL Versionen 12 und frühere Versionen erfordert das Installieren von Erweiterungen `rds_superuser`-Berechtigungen.

- Für RDS für PostgreSQL Version 13 und höher können Benutzer (Rollen) mit Erstellungsberechtigungen für eine bestimmte Datenbank-Instance vertrauenswürdige Erweiterungen installieren und verwenden. Eine Liste mit vertrauenswürdigen Erweiterungen finden Sie unter Vertrauenswürdige SQL Postgre-Erweiterungen.

Sie können auch genau angeben, welche Erweiterungen auf Ihrer DB-Instance von RDS for PostgreSQL installiert werden können, indem Sie sie im Parameter `rds.allowed_extensions` aufführen. Weitere Informationen finden Sie unter Einschränkung der Installation von Postgre-Erweiterungen SQL.

Weitere Informationen über die `rds_superuser`-Rolle finden Sie unter Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen.

Themen

- Verwenden der Funktionen aus der orafce-Erweiterung
- Verwenden der Amazon RDS Delegated Extension Support für Postgre SQL
- Verwalten von PostgreSQL-Partitionen mit der Erweiterung pg_partman
- Wird verwendet pgAudit , um Datenbankaktivitäten zu protokollieren
- Planen der Wartung mit der PostgreSQL-Erweiterung pg_cron
- Verwenden von pglogical, um Daten zwischen Instances zu synchronisieren
- Verwenden von „pgactive“ zur Unterstützung der Aktiv-Aktiv-Replikation
- Reduzieren von überflüssigen Daten in Tabellen und Indizes mit der Erweiterung pg_repack
- Erweiterung aktualisieren und verwenden PLV8
- Verwendung von PL/Rust zum Schreiben von PostgreSQL-Funktionen in der Sprache Rust
- Verwaltung von Geodaten mit der GIS Post-Erweiterung

Verwenden der Funktionen aus der orafce-Erweiterung

Die Orafce-Erweiterung bietet Funktionen und Operatoren, die eine Teilmenge von Funktionen und Paketen aus einer Oracle-Datenbank emulieren. Die Orace-Erweiterung erleichtert Ihnen die Portierung einer Oracle-Anwendung nach Postgre. SQL RDSfür SQL Postgre-Versionen 9.6.6 und höher wird diese Erweiterung unterstützt. Weitere Informationen zu Oracle finden Sie unter orafce on. GitHub

> **Note**
>
> RDSfür Postgre unterstützt das `utl_file` Paket SQL nicht, das Teil der Orafce-Erweiterung ist. Dies liegt daran, dass die `utl_file`-Schema-Funktionen Lese- und Schreiboperationen für Betriebssystem-Textdateien ermöglichen, wofür ein Superuser-Zugriff auf den zugrundeliegenden Host erforderlich ist. Als verwalteter Dienst bietet RDS Postgre SQL keinen Hostzugriff.

So verwenden Sie die orafce-Erweiterung

1. Stellen Sie unter Verwendung des Hauptbenutzernamens, der für die Erstellung der DB-Instance verwendet wurde, eine Verbindung mit der DB-Instance her.

   Wenn Sie orafce für eine andere Datenbank in derselben DB-Instance aktivieren möchten, verwenden Sie den Befehl `/c dbname`-psql. Mit diesem Befehl wechseln Sie nach dem Einleiten der Verbindung aus der primären Datenbank.
2. Aktivieren Sie die orafce-Erweiterung mit der Anweisung `CREATE EXTENSION`.

   ```
   CREATE EXTENSION orafce;
   ```
3. Übertragen Sie den Besitz der oracle-Schemas mit der Anweisung `ALTER SCHEMA` auf die rds_superuser-Rolle.

   ```
   ALTER SCHEMA oracle OWNER TO rds_superuser;
   ```

   Mit dem psql-Befehl `\dn` zeigen Sie die Liste der Eigentümer für das oracle-Schema an.

Verwenden der Amazon RDS Delegated Extension Support für Postgre SQL

Mit Amazon RDS Delegated Extension Support für Postgre SQL können Sie die Erweiterungsverwaltung an einen Benutzer delegieren, der kein Benutzer sein muss. `rds_superuser` Mit dieser delegierten Erweiterungsunterstützung `rds_extension` wird eine neue Rolle namens erstellt. Sie müssen diese einem Benutzer zuweisen, um andere Erweiterungen zu verwalten. Mit dieser Rolle können Erweiterungen erstellt, aktualisiert und gelöscht werden.

Sie können die Erweiterungen angeben, die auf Ihrer RDS DB-Instance installiert werden können, indem Sie sie im `rds.allowed_extensions` Parameter auflisten. Weitere Informationen finden Sie unter Verwenden von SQL Postgre-Erweiterungen mit Amazon RDS for SQL Postgre.

Sie können die Liste der verfügbaren Erweiterungen einschränken, die vom Benutzer mithilfe des Parameters `rds_extension` role using `rds.allowed_delegated_extensions` verwaltet werden können.

Die Unterstützung delegierter Erweiterungen ist in den folgenden Versionen verfügbar:

- Alle höheren Versionen
- 16.4 und höhere 16-Versionen
- 15.8 und höhere 15 Versionen
- 14.13 und höhere 14-Versionen
- 13.16 und höhere 13 Versionen
- 12.20 und höhere 12 Versionen

Themen

- Die Unterstützung für Delegate-Erweiterungen für einen Benutzer aktivieren
- Konfiguration, die bei der Unterstützung RDS delegierter Erweiterungen für Postgre verwendet wird SQL
- Die Unterstützung für die delegierte Erweiterung ausschalten
- Vorteile der Nutzung von Amazon RDS Delegated Extension Support
- Einschränkung der Unterstützung von Amazon RDS Delegated Extension für Postgre SQL
- Für bestimmte Erweiterungen sind Berechtigungen erforderlich

- Sicherheitsüberlegungen
- Die Kaskade der Drop-Erweiterung ist deaktiviert
- Beispielerweiterungen, die mithilfe der Unterstützung delegierter Erweiterungen hinzugefügt werden können

Die Unterstützung für Delegate-Erweiterungen für einen Benutzer aktivieren

Sie müssen die folgenden Schritte ausführen, um die Delegate-Erweiterungsunterstützung für einen Benutzer zu aktivieren:

1. Einem Benutzer eine **`rds_extension`** Rolle zuweisen — Connect zur Datenbank her als `rds_superuser` und führen Sie den folgenden Befehl aus:

   ```
   Postgres => grant rds_extension to user_name;
   ```

2. Liste der Erweiterungen festlegen, die delegierten Benutzern zur Verwaltung zur Verfügung stehen — `rds.allowed_delegated_extensions` Mit dieser Option können Sie `rds.allowed_extensions` im DB-Cluster-Parameter eine Teilmenge der verfügbaren Erweiterungen angeben. Sie können dies auf einer der folgenden Ebenen durchführen:

   - Im Cluster oder in der Instanzparametergruppe über den Befehl AWS Management Console oderAPI. Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.
   - Verwenden Sie den folgenden Befehl auf Datenbankebene:

     ```
     alter database database_name set rds.allowed_delegated_extensions =
      'extension_name_1,
                        extension_name_2,...extension_name_n';
     ```

   - Verwenden Sie den folgenden Befehl auf Benutzerebene:

     ```
     alter user user_name set rds.allowed_delegated_extensions = 'extension_name_1,
                        extension_name_2,...extension_name_n';
     ```

   > **Note**
   >
   > Sie müssen die Datenbank nicht neu starten, nachdem Sie den `rds.allowed_delegated_extensions` dynamischen Parameter geändert haben.

3. Erlauben Sie dem delegierten Benutzer den Zugriff auf Objekte, die während der Erstellung der Erweiterung erstellt wurden — Bestimmte Erweiterungen erstellen Objekte, für die zusätzliche Berechtigungen erteilt werden müssen, bevor der Benutzer mit der `rds_extension` Rolle auf sie zugreifen kann. Sie `rds_superuser` müssen dem delegierten Benutzer Zugriff auf diese Objekte gewähren. Eine der Optionen besteht darin, einen Ereignisauslöser zu verwenden, um dem delegierten Benutzer automatisch die Erlaubnis zu erteilen. Weitere Informationen finden Sie im Beispiel für einen Ereignisauslöser unterDie Unterstützung für die delegierte Erweiterung ausschalten.

Konfiguration, die bei der Unterstützung RDS delegierter Erweiterungen für Postgre verwendet wird SQL

Name der Konfiguration	Beschreibung	Standardwert	Hinweise	Wer kann die Erlaubnis ändern oder erteilen
`rds.ed_de ted_e sions`	Dieser Parameter schränkt die Erweiterungen ein, die eine rds_extension-Rolle in einer Datenbank verwalten kann. Es muss sich um eine Teilmenge von rds.allowed_extensions handeln.	leere Zeichenfolge	• Standardmäßig ist dieser Parameter eine leere Zeichenfolge, was bedeutet, dass keine Erweiterungen an Benutzer mit `rds_extension` delegiert wurden. • Jede unterstützte Erweiterung kann hinzugefügt werden, wenn der Benutzer dazu berechtigt ist. Setzen Sie dazu den `rds.allow`	rds_superuser

Name der Konfiguration	Beschreibung	Standardwert	Hinweise	Wer kann die Erlaubnis ändern oder erteilen
			`ed_delega ted_exten sions` Parameter auf eine Zeichenfo lge mit durch Kommas getrennte n Erweiteru ngsnamen. Indem Sie diesem Parameter eine Liste von Erweiterungen hinzufügen, identifizieren Sie explizit die Erweiteru ngen, die der Benutzer mit der `rds_exten sion` Rolle installieren kann. • Wenn diese Option auf gesetzt ist*, bedeutet dies, dass alle in aufgeführten Erweiterungen an Benutzer mit `rds_exten`	

Name der Konfiguration	Beschreibung	Standardwert	Hinweise	Wer kann die Erlaubnis ändern oder erteilen
			`sion` Rolle delegiert `rds_allowed_extensions` werden. Weitere Informationen zum Einrichten dieses Parameters finden Sie unterDie Unterstützung für Delegate-Erweiterungen für einen Benutzer aktivieren.	

Name der Konfiguration	Beschreibung	Standardwert	Hinweise	Wer kann die Erlaubnis ändern oder erteilen
`rds.allowed_extensions`	Mit diesem Parameter kann der Kunde die Erweiterungen einschränken, die in der DB-Instance installiert werden können. RDS Weitere Informationen finden Sie unter Einschränken der Installation von SQL Postgre-Erweiterungen	"*"	Standardmäßig ist dieser Parameter auf „*“ gesetzt, was bedeutet, dass alle Erweiterungen, die RDS für Postgre SQL und Aurora Postgre unterstützt SQL werden, von Benutzern mit den erforderlichen Rechten erstellt werden dürfen. Leer bedeutet, dass keine Erweiterungen in der RDS DB-Instance installiert werden können.	Administrator

Name der Konfiguration	Beschreibung	Standardwert	Hinweise	Wer kann die Erlaubnis ändern oder erteilen
`rds-delegated_nsionow_dr ascac`	Dieser Parameter steuert, ob Benutzer die Erweiterung mithilfe einer Kaskadenoption löschen können. `rds_extension`	aus	`rds-delegated_extensionallow_drop_cascade` ist standardmäßig auf `off` festgelegt. Dies bedeutet, dass Benutzer mit `rds_extension` der Option „Kaskade“ keine Erweiterung löschen dürfen. Um diese Fähigkeit zu gewähren, sollte der `rds.delegated_extensionallow_drop_cascade` Parameter auf on gesetzt werden.	rds_superuser

Die Unterstützung für die delegierte Erweiterung ausschalten

Teilweise ausschalten

Die delegierten Benutzer können keine neuen Erweiterungen erstellen, aber trotzdem bestehende Erweiterungen aktualisieren.

- Auf `rds.allowed_delegated_extensions` den Standardwert in der DB-Cluster-Parametergruppe zurücksetzen.
- Verwenden Sie den folgenden Befehl auf Datenbankebene:

```
alter database database_name reset rds.allowed_delegated_extensions;
```

- Verwenden Sie den folgenden Befehl auf Benutzerebene:

```
alter user user_name reset rds.allowed_delegated_extensions;
```

Vollständig ausschalten

Wenn Sie einem Benutzer die `rds_extension` Rolle entziehen, erhält der Benutzer wieder die Standardberechtigungen. Der Benutzer kann keine Erweiterungen mehr erstellen, aktualisieren oder löschen.

```
postgres => revoke rds_extension from user_name;
```

Beispiel für einen Ereignisauslöser

Wenn Sie einem delegierten Benutzer die Verwendung von Erweiterungen ermöglichen möchten, für die Berechtigungen für seine bei der Erstellung der Erweiterung erstellten Objekte erforderlich sind, können Sie das folgende Beispiel für einen Ereignisauslöser anpassen und nur die Erweiterungen hinzufügen, für die die delegierten Benutzer Zugriff auf die gesamte Funktionalität haben sollen. `rds_extension` Dieser Ereignistrigger kann auf Template1 (der Standardvorlage) erstellt werden, sodass alle mit Template1 erstellten Datenbanken über diesen Ereignistrigger verfügen. Wenn ein delegierter Benutzer die Erweiterung installiert, gewährt dieser Trigger automatisch das Eigentum an den von der Erweiterung erstellten Objekten.

```
CREATE OR REPLACE FUNCTION create_ext()

  RETURNS event_trigger AS $$

DECLARE

  schemaname TEXT;
  databaseowner TEXT;
```

```
  r RECORD;

BEGIN

  IF tg_tag = 'CREATE EXTENSION' and current_user != 'rds_superuser' THEN
    RAISE NOTICE 'SECURITY INVOKER';
    RAISE NOTICE 'user: %', current_user;
    FOR r IN SELECT * FROM pg_event_trigger_ddl_commands()
    LOOP
        CONTINUE WHEN r.command_tag != 'CREATE EXTENSION' OR r.object_type !=
 'extension';

        schemaname = (
            SELECT n.nspname
            FROM pg_catalog.pg_extension AS e
            INNER JOIN pg_catalog.pg_namespace AS n
            ON e.extnamespace = n.oid
            WHERE e.oid = r.objid
        );

        databaseowner = (
            SELECT pg_catalog.pg_get_userbyid(d.datdba)
            FROM pg_catalog.pg_database d
            WHERE d.datname = current_database()
        );
        RAISE NOTICE 'Record for event trigger %, objid: %,tag: %, current_user: %,
 schema: %, database_owenr: %', r.object_identity, r.objid, tg_tag, current_user,
 schemaname, databaseowner;
        IF r.object_identity = 'address_standardizer_data_us' THEN
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE %I.us_gaz TO
 %I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE %I.us_lex TO
 %I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE %I.us_rules
 TO %I WITH GRANT OPTION;', schemaname, databaseowner);
        ELSIF r.object_identity = 'dict_int' THEN
            EXECUTE format('ALTER TEXT SEARCH DICTIONARY %I.intdict OWNER TO %I;',
 schemaname, databaseowner);
        ELSIF r.object_identity = 'pg_partman' THEN
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE
 %I.part_config TO %I WITH GRANT OPTION;', schemaname, databaseowner);
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE
 %I.part_config_sub TO %I WITH GRANT OPTION;', schemaname, databaseowner);
```

```
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE
 %I.custom_time_partitions TO %I WITH GRANT OPTION;', schemaname, databaseowner);
        ELSIF r.object_identity = 'postgis_topology' THEN
            EXECUTE format('GRANT SELECT, UPDATE, INSERT, DELETE ON ALL TABLES IN
 SCHEMA topology TO %I WITH GRANT OPTION;', databaseowner);
            EXECUTE format('GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA topology TO
 %I WITH GRANT OPTION;', databaseowner);
            EXECUTE format('GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA topology TO %I
 WITH GRANT OPTION;', databaseowner);
            EXECUTE format('GRANT USAGE ON SCHEMA topology TO %I WITH GRANT OPTION;',
 databaseowner);
        END IF;
    END LOOP;
  END IF;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;

CREATE EVENT TRIGGER log_create_ext ON ddl_command_end EXECUTE PROCEDURE create_ext();
```

Vorteile der Nutzung von Amazon RDS Delegated Extension Support

Durch die Verwendung von Amazon RDS Delegated Extension Support für Postgre SQL delegieren Sie die Erweiterungsverwaltung auf sichere Weise an Benutzer, die diese Rolle nicht haben. `rds_superuser` Diese Funktion bietet die folgenden Vorteile:

- Sie können die Erweiterungsverwaltung ganz einfach an Benutzer Ihrer Wahl delegieren.
- Dafür ist keine `rds_superuser` Rolle erforderlich.
- Bietet die Möglichkeit, verschiedene Erweiterungen für verschiedene Datenbanken im selben DB-Cluster zu unterstützen.

Einschränkung der Unterstützung von Amazon RDS Delegated Extension für Postgre SQL

- Objekte, die während der Erstellung der Erweiterung erstellt wurden, benötigen möglicherweise zusätzliche Rechte, damit die Erweiterung ordnungsgemäß funktioniert.

- Einige Erweiterungen können standardmäßig nicht vom delegierten Erweiterungsbenutzer verwaltet werden, darunter die folgenden:`log_fdw`,,`pg_cron`,`pg_tle`,`pgactive`,`pglogical`, `postgis_rasterpostgis_tiger_geocoder`,`postgis_topology`.

Für bestimmte Erweiterungen sind Berechtigungen erforderlich

Um die folgenden Erweiterungen zu erstellen, zu verwenden oder zu aktualisieren, sollte der delegierte Benutzer über die erforderlichen Rechte für die folgenden Funktionen, Tabellen und Schemas verfügen.

Erwe ngen für die Eiger oder Bere unge erfor ich sind	Funktion	Tabellen	Schema	Wörterbuch für die Textsuche	Kommentar
addr tand er_da s	Keine	us_gaz, us_lex, us_lex, i.US_Regeln	Keine	Keine	Keine
amch	bt_index_check, bt_index_ parent_check	Keine	Keine	Keine	Keine
dict_i	Keine	Keine	Keine	verbieten	Keine
pg_p n	Keine	custom_ti me_partitions, part_config, part_config_sub	Keine	Keine	Keine

Erwe ngen für die Eiger oder Bere unge erfor ich sind	Funktion	Tabellen	Schema	Wörterbuch für die Textsuche	Kommentar
pg_st tatem	Keine	Keine	Keine	Keine	Keine
Beitra GIS	st_tileenvelope	spatial_ref_sys	Keine	Keine	Keine
postg aster	Keine	Keine	Keine	Keine	Keine
postg opolo	Keine	Topologie, Schicht	Topologie	Keine	der delegierte Benutzer muss der Datenbank besitzer sein
log_f	create_fo reign_tab le_for_log_file	Keine	Keine	Keine	Keine
rds_t	Verschlüs selungstyp für role_password_	Keine	Keine	Keine	Keine
postg iger_ oder	Keine	geocode_s ettings_default, geocode_s ettings	Tiger	Keine	Keine

Erwe ngen für die Eiger oder Bere unge erfor ich sind	Funktion	Tabellen	Schema	Wörterbuch für die Textsuche	Kommentar
pg_fr acem	pg_freespace	Keine	Keine	Keine	Keine
pg_vi lity	pg_visibility	Keine	Keine	Keine	Keine

Sicherheitsüberlegungen

Denken Sie daran, dass ein Benutzer mit `rds_extension` Rolle Erweiterungen für alle Datenbanken verwalten kann, für die er die Verbindungsberechtigung hat. Wenn beabsichtigt ist, dass ein delegierter Benutzer die Erweiterung für eine einzelne Datenbank verwaltet, empfiehlt es sich, alle öffentlichen Rechte für jede Datenbank zu entziehen und dann dem delegierten Benutzer explizit die Connect-Berechtigung für diese spezifische Datenbank zu gewähren.

Es gibt mehrere Erweiterungen, mit denen ein Benutzer auf Informationen aus mehreren Datenbanken zugreifen kann. Stellen Sie sicher, dass die Benutzer, die Sie gewähren, `rds_extension` über datenbankübergreifende Funktionen verfügen, bevor Sie diese Erweiterungen hinzufügen`rds.allowed_delegated_extensions`. `dblink`Stellen Sie beispielsweise `postgres_fdw` Funktionen für datenbankübergreifende Abfragen auf derselben Instanz oder auf Remote-Instances bereit. `log_fdw`liest die Protokolldateien der Postgres-Engine, die für alle Datenbanken in der Instanz gelten und möglicherweise langsame Abfragen oder Fehlermeldungen von mehreren Datenbanken enthalten. `pg_cron`ermöglicht die Ausführung von geplanten Hintergrundjobs auf der DB-Instance und kann Jobs so konfigurieren, dass sie in einer anderen Datenbank ausgeführt werden.

Die Kaskade der Drop-Erweiterung ist deaktiviert

Die Möglichkeit, die Erweiterung mit der Option „Kaskade“ durch einen Benutzer mit dieser `rds_extension` Rolle zu löschen, wird durch einen `rds.delegated_extension_allow_drop_cascade` Parameter gesteuert. `rds-delegated_extension_allow_drop_cascade` ist standardmäßig auf `off` festgelegt. Das bedeutet, dass Benutzer mit der `rds_extension` Rolle eine Erweiterung nicht mithilfe der Kaskadenoption löschen dürfen, wie in der folgenden Abfrage gezeigt.

```
DROP EXTENSION CASCADE;
```

Dadurch werden automatisch Objekte gelöscht, die von der Erweiterung abhängen, und wiederum alle Objekte, die von diesen Objekten abhängen. Der Versuch, die Cascade-Option zu verwenden, führt zu einem Fehler.

Um diese Fähigkeit zu gewähren, sollte der `rds.delegated_extension_allow_drop_cascade` Parameter auf on gesetzt werden.

Das Ändern des `rds.delegated_extension_allow_drop_cascade` dynamischen Parameters erfordert keinen Neustart der Datenbank. Sie können dies auf einer der folgenden Ebenen tun:

- Im Cluster oder in der Instanzparametergruppe über den Befehl AWS Management Console oderAPI.
- Verwenden Sie den folgenden Befehl auf Datenbankebene:

```
alter database database_name set rds.delegated_extension_allow_drop_cascade = 'on';
```

- Verwenden Sie den folgenden Befehl auf Benutzerebene:

```
alter role tenant_user set rds.delegated_extension_allow_drop_cascade = 'on';
```

Beispielerweiterungen, die mithilfe der Unterstützung delegierter Erweiterungen hinzugefügt werden können

- `rds_tools`

```
extension_test_db=> create extension rds_tools;
```

```
CREATE EXTENSION
extension_test_db=> SELECT * from rds_tools.role_password_encryption_type() where
 rolname = 'pg_read_server_files';
ERROR: permission denied for function role_password_encryption_type
```

- amcheck

```
extension_test_db=> CREATE TABLE amcheck_test (id int);
CREATE TABLE
extension_test_db=> INSERT INTO amcheck_test VALUES (generate_series(1,100000));
INSERT 0 100000
extension_test_db=> CREATE INDEX amcheck_test_btree_idx ON amcheck_test USING btree
 (id);
CREATE INDEX
extension_test_db=> create extension amcheck;
CREATE EXTENSION
extension_test_db=> SELECT bt_index_check('amcheck_test_btree_idx'::regclass);
ERROR: permission denied for function bt_index_check
extension_test_db=> SELECT bt_index_parent_check('amcheck_test_btree_idx'::regclass);
ERROR: permission denied for function bt_index_parent_check
```

- pg_freespacemap

```
extension_test_db=> create extension pg_freespacemap;
CREATE EXTENSION
extension_test_db=> SELECT * FROM pg_freespace('pg_authid');
ERROR: permission denied for function pg_freespace
extension_test_db=> SELECT * FROM pg_freespace('pg_authid',0);
ERROR: permission denied for function pg_freespace
```

- pg_visibility

```
extension_test_db=> create extension pg_visibility;
CREATE EXTENSION
extension_test_db=> select * from pg_visibility('pg_database'::regclass);
ERROR: permission denied for function pg_visibility
```

- postgres_fdw

```
extension_test_db=> create extension postgres_fdw;
CREATE EXTENSION
extension_test_db=> create server myserver foreign data wrapper postgres_fdw options
 (host 'foo', dbname 'foodb', port '5432');
```

```
ERROR: permission denied for foreign-data wrapper postgres_fdw
```

Verwalten von PostgreSQL-Partitionen mit der Erweiterung pg_partman

Die PostgreSQL-Tabellenpartitionierung bietet ein Framework für den leistungsstarken Umgang mit Dateneingaben und Berichten. Verwenden Sie die Partitionierung für Datenbanken, die eine sehr schnelle Eingabe großer Datenmengen erfordern. Die Partitionierung ermöglicht auch schnellere Abfragen großer Tabellen. Die Partitionierung hilft bei der Verwaltung von Daten, ohne die Datenbank-Instance zu beeinträchtigen, da sie weniger I/O-Ressourcen benötigt.

Durch die Partitionierung können Sie Daten zur Verarbeitung in benutzerdefinierte Blöcke aufteilen. Sie können beispielsweise Zeitreihendaten für Bereiche wie stündlich, täglich, wöchentlich, monatlich, vierteljährlich, jährlich, benutzerdefiniert oder eine beliebige Kombination davon partitionieren. Wenn Sie für ein Beispiel für Zeitreihendaten die Tabelle nach Stunden partitionieren, enthält jede Partition die Daten einer Stunde. Wenn Sie die Zeitreihentabelle nach Tag partitionieren, enthalten die Partitionen die Daten eines Tages und so weiter. Der Partitionsschlüssel steuert die Größe einer Partition.

Wenn Sie den SQL-Befehl `INSERT` oder `UPDATE` für eine partitionierte Tabelle verwenden, leitet die Datenbank-Engine die Daten an die entsprechende Partition weiter. PostgreSQL-Tabellenpartitionen, die die Daten speichern, sind untergeordnete Tabellen der Haupttabelle.

Während der Lesevorgänge der Datenbankabfrage untersucht der PostgreSQL-Optimierer die `WHERE`-Klausel der Abfrage und leitet den Datenbank-Scan nach Möglichkeit nur an die relevanten Partitionen weiter.

Beginnend mit Version 10 verwendet PostgreSQL deklarative Partitionierung, um die Tabellenpartitionierung zu implementieren. Dies wird auch als native PostgreSQL-Partitionierung bezeichnet. Vor PostgreSQL Version 10 wurden Auslöser verwendet, um Partitionen zu implementieren.

Die PostgreSQL-Tabellenpartitionierung bietet die folgenden Funktionen:

- Erstellung neuer Partitionen zu jeder Zeit.
- Variable Partitionsbereiche.
- Entfernbare und wiederverwendbare Partitionen mit DDL-Anweisungen (Data Definition Language).

  Zum Beispiel sind entfernbare Partitionen nützlich, um Verlaufsdaten aus der Hauptpartition zu entfernen, aber Verlaufsdaten für die Analyse zu behalten.
- Neue Partitionen erben die Eigenschaften der übergeordneten Datenbanktabelle, einschließlich folgender Eigenschaften:

  - Indizes
  - Primärschlüssel, die die Partitionsschlüsselspalte enthalten müssen
  - Fremdschlüssel
  - Einschränkungen prüfen
  - Referenzen
- Erstellen von Indizes für die vollständige Tabelle oder jede spezifische Partition.

Sie können das Schema für eine einzelne Partition nicht ändern. Sie können jedoch die übergeordnete Tabelle ändern (z. B. das Hinzufügen einer neuen Spalte), die auf Partitionen übertragen wird.

Themen

- Übersicht über die PostgreSQL-Erweiterung pg_partman
- Aktivieren der Erweiterung pg_partman
- Konfigurieren von Partitionen mit der create_parent-Funktion
- Konfigurieren der Partitionspflege mit der run_maintenance_proc-Funktion

Übersicht über die PostgreSQL-Erweiterung pg_partman

Sie können die PostgreSQL-Erweiterung `pg_partman` verwenden, um die Erstellung und Pflege von Tabellenpartitionen zu automatisieren. Weitere allgemeine Informationen finden Sie unter PG-Partitions-Manager in der `pg_partman`-Dokumentation.

> **Note**
>
> Die Erweiterung `pg_partman` wird auf den RDS-for-PostgreSQL-Versionen 12.5 und höher unterstützt.

Anstatt jede Partition manuell erstellen zu müssen, konfigurieren Sie `pg_partman` mit den folgenden Einstellungen:

- Zu partitionierende Tabelle
- Partitionstyp
- Partitionsschlüssel

- Granularität der Partition
- Optionen für die Erstellung und Verwaltung von Partitionen

Nachdem Sie eine partitionierte PostgreSQL-Tabelle erstellt haben, registrieren Sie diese mit `pg_partman`, indem Sie die Funktion `create_parent` aufrufen. Dadurch werden die erforderlichen Partitionen basierend auf den Parametern erstellt, die Sie an die Funktion übergeben.

Die Erweiterung `pg_partman` bietet auch die Funktion `run_maintenance_proc`, die Sie planmäßig aufrufen können, um Partitionen automatisch zu verwalten. Planen Sie, dass diese Funktion regelmäßig (z. B. stündlich) ausgeführt wird, um sicherzustellen, dass die richtigen Partitionen nach Bedarf erstellt werden. Sie können auch sicherstellen, dass Partitionen automatisch gelöscht werden.

Aktivieren der Erweiterung pg_partman

Wenn Sie mehrere Datenbanken innerhalb derselben PostgreSQL-DB-Instance haben, für die Sie Partitionen verwalten möchten, aktivieren Sie die Erweiterung `pg_partman` für jede Datenbank separat. Um die Erweiterung `pg_partman` für eine bestimmte Datenbank zu aktivieren, erstellen Sie das Partitionswartungsschema und dann die Erweiterung `pg_partman` wie folgt:

```
CREATE SCHEMA partman;
CREATE EXTENSION pg_partman WITH SCHEMA partman;
```

> **Note**
>
> Um die Erweiterung `pg_partman` zu erstellen, müssen Sie sicherstellen, dass Sie über `rds_superuser`-Berechtigungen verfügen.

Wenn Sie einen Fehler wie den folgenden erhalten, erteilen Sie dem Konto die Berechtigungen `rds_superuser` oder verwenden Sie Ihr Superuser-Konto.

```
ERROR: permission denied to create extension "pg_partman"
HINT: Must be superuser to create this extension.
```

Um die Berechtigungen `rds_superuser` zu erteilen, verbinden Sie sich mit Ihrem Superuser-Konto und führen Sie den folgenden Befehl aus.

```
GRANT rds_superuser TO user-or-role;
```

Für die Beispiele, die die Verwendung der Erweiterung pg_partman zeigen, verwenden wir die folgende Beispieldatenbanktabelle und Partition. Diese Datenbank verwendet eine partitionierte Tabelle basierend auf einem Zeitstempel. Ein `data_mart`-Schema enthält eine Tabelle mit dem Namen `events` mit einer Spalte namens `created_at`. Die folgenden Einstellungen sind in der `events`-Tabelle enthalten:

- Primärschlüssel `event_id` und `created_at`, die die Spalte zur Führung der Partition verwenden müssen.
- Eine CHECK-Beschränkung `ck_valid_operation` zum Durchsetzen von Werten für eine `operation`-Tabellenspalte.
- Zwei Fremdschlüssel, wobei einer `fk_orga_membership)` auf die externe Tabelle `organization` verweist und der andere (`fk_parent_event_id`) ein selbst referenzierter Fremdschlüssel ist.
- Zwei Indizes, wobei einer (`idx_org_id`) für den Fremdschlüssel und der andere (`idx_event_type`) für den Ereignistyp steht.

Die folgenden DDL-Anweisungen erstellen diese Objekte, die automatisch auf jeder Partition enthalten sind.

```
CREATE SCHEMA data_mart;
CREATE TABLE data_mart.organization ( org_id BIGSERIAL,
        org_name TEXT,
        CONSTRAINT pk_organization PRIMARY KEY (org_id)
    );

CREATE TABLE data_mart.events(
        event_id        BIGSERIAL,
        operation       CHAR(1),
        value           FLOAT(24),
        parent_event_id BIGINT,
        event_type      VARCHAR(25),
        org_id          BIGSERIAL,
        created_at      timestamp,
        CONSTRAINT pk_data_mart_event PRIMARY KEY (event_id, created_at),
        CONSTRAINT ck_valid_operation CHECK (operation = 'C' OR operation = 'D'),
        CONSTRAINT fk_orga_membership
            FOREIGN KEY(org_id)
```

```
            REFERENCES data_mart.organization (org_id),
        CONSTRAINT fk_parent_event_id
            FOREIGN KEY(parent_event_id, created_at)
            REFERENCES data_mart.events (event_id,created_at)
    ) PARTITION BY RANGE (created_at);

CREATE INDEX idx_org_id     ON  data_mart.events(org_id);
CREATE INDEX idx_event_type ON  data_mart.events(event_type);
```

Konfigurieren von Partitionen mit der create_parent-Funktion

Nachdem Sie die Erweiterung `pg_partman` aktiviert haben, verwenden Sie die `create_parent`-Funktion, um Partitionen innerhalb des Partitionswartungsschemas zu konfigurieren. Im folgenden Beispiel wird das `events`-Tabellenbeispiel verwendet, das in Aktivieren der Erweiterung pg_partman erstellt wurde. Rufen Sie die `create_parent`-Funktion wie folgt auf.

```
SELECT partman.create_parent(
 p_parent_table => 'data_mart.events',
 p_control      => 'created_at',
 p_type         => 'range',
 p_interval     => '1 day',
 p_premake      => 30);
```

Dabei werden die folgenden Parameter verwendet:

- `p_parent_table` – Die übergeordnete partitionierte Tabelle. Diese Tabelle muss bereits existieren und einschließlich des Schemas vollständig qualifiziert sein.
- `p_control` – Die Spalte, auf der die Partitionierung basieren soll. Der Datentyp muss ganzzahlig oder zeitbasiert sein.
- `p_type` – Der Typ ist entweder `'range'` oder `'list'`.
- `p_interval` – Das Zeitintervall oder der Ganzzahlbereich für jede Partition. Zu den Beispielwerten gehören `1 day` `1 hour`,, und so weiter.
- `p_premake` – Die Anzahl der Partitionen, die im Voraus erstellt werden müssen, um neue Inserts zu unterstützen.

Eine vollständige Beschreibung der Funktion `create_parent` finden Sie in der `pg_partman`-Dokumentation unter Creation Functions (Erstellungsfunktionen).

Konfigurieren der Partitionspflege mit der run_maintenance_proc-Funktion

Sie können Partitionswartungsvorgänge ausführen, um automatisch neue Partitionen zu erstellen, Partitionen zu trennen oder alte Partitionen zu entfernen. Die Partitionspflege beruht auf der Funktion `run_maintenance_proc` von der `pg_partman`-Erweiterung und der Erweiterung `pg_cron`, die einen internen Scheduler initiiert. Der `pg_cron`-Scheduler führt automatisch SQL-Anweisungen, -Funktionen und -Prozesse aus, die in Ihren Datenbanken definiert sind.

Im folgenden Beispiel wird das `events`-Tabellenbeispiel verwendet, das in Aktivieren der Erweiterung pg_partman erstellt wurde, um die automatische Ausführung der Partitionswartung festzulegen. Fügen Sie als Voraussetzung `pg_cron` zum Parameter `shared_preload_libraries` in der Parametergruppe der DB-Instance hinzu.

```
CREATE EXTENSION pg_cron;

UPDATE partman.part_config
SET infinite_time_partitions = true,
    retention = '3 months',
    retention_keep_table=true
WHERE parent_table = 'data_mart.events';
SELECT cron.schedule('@hourly', $$CALL partman.run_maintenance_proc()$$);
```

Im Folgenden finden Sie eine step-by-step Erläuterung des vorherigen Beispiels:

1. Ändern Sie die mit Ihrer DB-Instance verknüpfte Parametergruppe und fügen Sie `pg_cron` dem `shared_preload_libraries`-Parameterwert hinzu. Diese Änderung erfordert einen Neustart der DB-Instance, damit sie wirksam wird. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.
2. Führen Sie den Befehl `CREATE EXTENSION pg_cron;` mit einem Konto aus, das die `rds_superuser`-Berechtigungen besitzt. Dadurch wird die Erweiterung `pg_cron` aktiviert. Weitere Informationen finden Sie unter Planen der Wartung mit der PostgreSQL-Erweiterung pg_cron.
3. Führen Sie den Befehl `UPDATE partman.part_config` aus, um die `pg_partman`-Einstellungen für die Tabelle `data_mart.events` anzupassen.
4. Führen Sie den Befehl aus `SET . . .` um die `data_mart.events`-Tabelle mit den folgenden Klauseln zu konfigurieren:

a. `infinite_time_partitions = true,` – Konfiguriert die Tabelle so, dass automatisch neue Partitionen ohne Begrenzung erstellt werden können.

b. `retention = '3 months',` – Konfiguriert die Tabelle so, dass sie eine maximale Beibehaltung von drei Monaten hat.

c. `retention_keep_table=true` – Konfiguriert die Tabelle so, dass die Tabelle bei Fälligkeit der Aufbewahrungsfrist nicht automatisch gelöscht wird. Stattdessen werden Partitionen, die älter als die Aufbewahrungsfrist sind, nur von der übergeordneten Tabelle getrennt.

5. Führen Sie den Befehl aus `SELECT cron.schedule . . .` um einen `pg_cron`-Funktionsaufruf zu machen. Dieser Aufruf definiert, wie oft der Scheduler das `pg_partman`-Wartungsverfahren `partman.run_maintenance_proc` ausführt. In diesem Beispiel wird der Prozess stündlich ausgeführt.

Eine vollständige Beschreibung der `run_maintenance_proc`-Funktion finden Sie in der `pg_partman`-Dokumentation unter Maintenance Functions (Wartungsfunktionen).

Wird verwendet pgAudit , um Datenbankaktivitäten zu protokollieren

Finanzinstitute, Behörden und viele Branchen müssen Audit-Protokolle aufbewahren, um die gesetzlichen Bestimmungen zu erfüllen. Wenn Sie die Postgre SQL Audit-Erweiterung (pgAudit) mit Ihrem für SQL Postgre-DB-Instance verwenden, können Sie die detaillierten Aufzeichnungen erfassen, die normalerweise von Auditoren oder zur Erfüllung gesetzlicher Anforderungen benötigt werden. Sie können die pgAudit Erweiterung beispielsweise so einrichten, dass sie Änderungen an bestimmten Datenbanken und Tabellen nachverfolgt, den Benutzer, der die Änderung vorgenommen hat, und viele andere Details aufzeichnet.

Die pgAudit Erweiterung baut auf der Funktionalität der nativen SQL Postgre-Logging-Infrastruktur auf, indem sie die Protokollnachrichten um mehr Details erweitert. Mit anderen Worten, Sie verwenden für die Anzeige Ihres Audit-Protokolls den gleichen Ansatz wie für die Anzeige von Protokollmeldungen. Weitere Hinweise zur SQL Postgre-Protokollierung finden Sie unter. Datenbank-Protokolldateien von RDS für PostgreSQL

Die pgAudit Erweiterung löscht sensible Daten wie Klartext-Passwörter aus den Protokollen. Wenn Ihr für die SQL Postgre-DB-Instance so konfiguriert ist, dass Anweisungen der Datenmanipulationssprache (DML) protokolliert werden, wie unter beschriebenAktivieren der Abfrageprotokollierung für Ihre DB-Instance von RDS für PostgreSQL, können Sie das Problem mit Klartext-Passwörtern vermeiden, indem Sie die SQL Postgre Audit-Erweiterung verwenden.

Sie können das Auditing für Ihre Datenbank-Instances mit einem hohen Grad an Spezifität konfigurieren. Sie können alle Datenbanken und alle Benutzer überprüfen. Sie können auch festlegen, dass nur bestimmte Datenbanken, Benutzer und andere Objekte überprüft werden. Bestimmte Benutzer und Datenbanken können Sie auch explizit von der Prüfung ausschließen. Weitere Informationen finden Sie unter Benutzer oder Datenbanken von der Audit-Protokollierung ausschließen.

Angesichts der Menge an Details, die erfasst werden können, empfehlen wir, dass Sie bei Verwendung Ihren pgAudit Speicherverbrauch überwachen.

Die pgAudit Erweiterung wird von allen verfügbaren RDSfür Postgre-Versionen. SQL Eine Liste der pgAudit Versionen, die von den verfügbaren Versionen RDS für Postgre SQL unterstützt werden, finden Sie unter Erweiterungsversionen für Amazon RDS for Postgre SQL in den Versionshinweisen zu Amazon RDS for Postgre SQL.

Themen

- Einrichtung der pgAudit Erweiterung

- Überprüfen von Datenbankobjekten
- Benutzer oder Datenbanken von der Audit-Protokollierung ausschließen
- Referenz für die pgAudit Erweiterung

Einrichtung der pgAudit Erweiterung

Um die pgAudit Erweiterung auf Ihrem einzurichten, fügen Sie sie zunächst pgAudit zu den gemeinsam genutzten Bibliotheken in der benutzerdefinierten DB-Parametergruppe für Ihre RDS for Postgre-DB-Instance hinzu. SQL Weitere Informationen über das Erstellen einer benutzerdefinierten DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS. Als Nächstes installieren Sie die pgAudit Erweiterung. Abschließend geben Sie die Datenbanken und Objekte an, die Sie überprüfen möchten. Die Schritte in diesem Abschnitt veranschaulichen die Vorgehensweise. Sie können das AWS Management Console oder die AWS CLI.

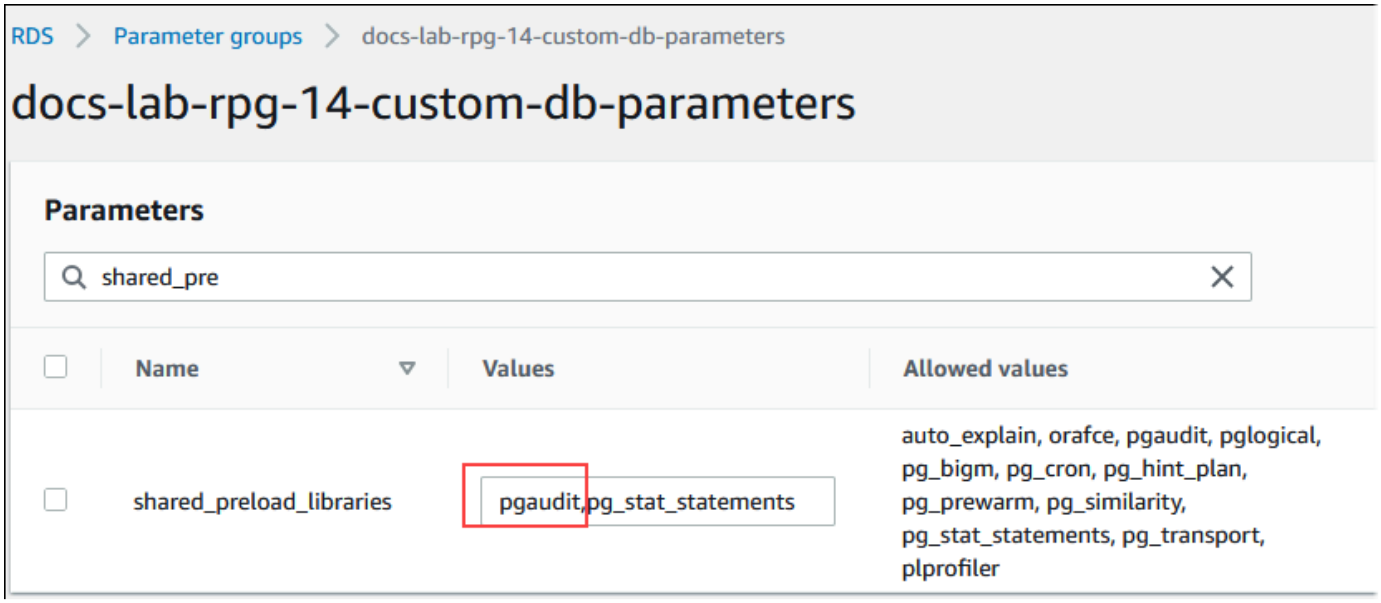
Sie müssen über Berechtigungen als `rds_superuser`-Rolle verfügen, um alle diese Aufgaben ausführen zu können.

Bei den folgenden Schritten wird davon ausgegangen, dass Ihr für die SQL Postgre-DB-Instance einer benutzerdefinierten zugeordnet ist.

Konsole

Um die Erweiterung einzurichten pgAudit

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die aus.
3. Öffnen Sie die Registerkarte Konfiguration für Ihre . RDSfür die SQL Postgre-DB-Instance. Suchen Sie in den Instance-Details den Link Parameter group (Parametergruppe).
4. Wählen Sie den Link, um die benutzerdefinierten Parameter zu öffnen, die mit Ihrem verknüpft sind. RDSfür die SQL Postgre-DB-Instance.
5. Geben Sie in das Suchfeld Parameters (Parameter) `shared_pre` ein, um den `shared_preload_libraries`-Parameter zu finden.
6. Wählen Sie Edit parameters (Parameter bearbeiten) aus, um auf die Eigenschaftswerte zuzugreifen.
7. Fügen Sie `pgaudit` der Liste im Feld Values (Werte) hinzu. Verwenden Sie ein Komma, um Elemente in der Werteliste zu trennen.



8. Starten Sie die für die SQL Postgre-DB-Instance neu, damit Ihre Änderung des `shared_preload_libraries` Parameters wirksam wird.
9. Wenn die Instance verfügbar ist, überprüfen Sie, ob sie pgAudit initialisiert wurde. Verwenden Sie`psql`, um eine Verbindung zur für SQL Postgre-DB-Instance herzustellen, und führen Sie dann den folgenden Befehl aus.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

10. Wenn Sie pgAudit initialisiert sind, können Sie die Erweiterung jetzt erstellen. Sie müssen die Erweiterung nach der Initialisierung der Bibliothek erstellen, da die `pgaudit` Erweiterung Ereignisauslöser für die Prüfung von Anweisungen in der Datendefinitionssprache (DDL) installiert.

```
CREATE EXTENSION pgaudit;
```

11. Schließen Sie die `psql`-Sitzung.

```
labdb=> \q
```

12. Melden Sie sich an bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.

13. Suchen Sie den `pgaudit.log`-Parameter in der Liste und legen Sie den entsprechenden Wert für Ihren Anwendungsfall fest. Wenn Sie beispielsweise den `pgaudit.log`-Parameter auf `write` festlegen, wie in der folgenden Abbildung gezeigt, werden Einfügungen, Aktualisierungen, Löschungen und einige andere Typänderungen im Protokoll erfasst.



Sie können auch einen der folgenden Werte für den `pgaudit.log`-Parameter auswählen.

- „none“: Dies ist der Standardwert. Es werden keine Datenbankänderungen protokolliert.
- „all“: Es wird alles protokolliert (Lesen, Schreiben, Funktion, Rolle, DDL, Verschiedenes).
- ddl — Protokolliert alle Anweisungen in der Datendefinitionssprache (DDL), die nicht in der `ROLE` Klasse enthalten sind.
- „function“: Protokolliert Funktionsaufrufe und `DO`-Blöcke.
- „misc“: Protokolliert verschiedene Befehle wie `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM` und `SET`.
- „read“: Protokolliert `SELECT` und `COPY`, wenn die Quelle eine Beziehung (z. B. eine Tabelle) oder eine Abfrage ist.
- „role“: Protokolliert Anweisungen in Bezug auf Rollen und Berechtigungen wie `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` und `DROP ROLE`.
- „write“: Protokolliert `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` und `COPY`, wenn das Ziel eine Beziehung (Tabelle) ist.

14. Wählen Sie Änderungen speichern.
15. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
16. Wählen Sie die für die SQL Postgre-DB-Instance aus der Datenbankliste aus.

AWS CLI

Zur Einrichtung pgAudit

Zur Einrichtung pgAudit mit dem AWS CLI, rufen Sie den modify-db-parameter-groupVorgang auf, um die Audit-Log-Parameter in Ihrer benutzerdefinierten Parametergruppe zu ändern, wie im folgenden Verfahren gezeigt.

1. Verwenden Sie Folgendes AWS CLI Befehl zum Hinzufügen `pgaudit` zum `shared_preload_libraries` Parameter.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Verwenden Sie Folgendes AWS CLI Befehl, um die für die SQL Postgre-DB-Instance neu zu starten, sodass die Pgaudit-Bibliothek initialisiert wird.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Wenn die Instance verfügbar ist, können Sie überprüfen, ob `pgaudit` initialisiert wurde. Verwenden Sie`psql`, um eine Verbindung zur für SQL Postgre-DB-Instance herzustellen, und führen Sie dann den folgenden Befehl aus.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

   Wenn Sie pgAudit initialisiert sind, können Sie die Erweiterung jetzt erstellen.

```
CREATE EXTENSION pgaudit;
```

4. Schließen Sie die `psql` Sitzung, damit Sie die verwenden können AWS CLI.

```
labdb=> \q
```

5. Verwenden Sie Folgendes AWS CLI Befehl, um die Klassen von Anweisungen anzugeben, die von der Sitzungsüberwachungsprotokollierung protokolliert werden sollen. Im Beispiel wird der `pgaudit.log`-Parameter auf `write` festgelegt, wodurch Einfügungen, Aktualisierungen und Löschungen im Protokoll erfasst werden.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.log,ParameterValue=write,ApplyMethod=pending-reboot" \
   --region aws-region
```

Sie können auch einen der folgenden Werte für den `pgaudit.log`-Parameter auswählen.

- „none“: Dies ist der Standardwert. Es werden keine Datenbankänderungen protokolliert.
- „all“: Es wird alles protokolliert (Lesen, Schreiben, Funktion, Rolle, DDL, Verschiedenes).
- ddl — Protokolliert alle Anweisungen der Datendefinitionssprache (DDL), die nicht in der `ROLE` Klasse enthalten sind.
- „function“: Protokolliert Funktionsaufrufe und `DO`-Blöcke.
- „misc“: Protokolliert verschiedene Befehle wie `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM` und `SET`.
- „read“: Protokolliert `SELECT` und `COPY`, wenn die Quelle eine Beziehung (z. B. eine Tabelle) oder eine Abfrage ist.
- „role“: Protokolliert Anweisungen in Bezug auf Rollen und Berechtigungen wie `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` und `DROP ROLE`.
- „write“: Protokolliert `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` und `COPY`, wenn das Ziel eine Beziehung (Tabelle) ist.

Starten Sie die für die SQL Postgre-DB-Instance wie folgt neu: AWS CLI Befehl.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Überprüfen von Datenbankobjekten

Wenn pgAudit es auf Ihrem für die SQL Postgre-DB-Instance eingerichtet und für Ihre Anforderungen konfiguriert ist, werden detailliertere Informationen im SQL Postgre-Protokoll erfasst. Während die standardmäßige SQL Postgre-Logging-Konfiguration beispielsweise das Datum und die Uhrzeit einer Änderung in einer Datenbanktabelle identifiziert, kann der Protokolleintrag bei der pgAudit Erweiterung das Schema, den Benutzer, der die Änderung vorgenommen hat, und andere Details enthalten, je nachdem, wie die Erweiterungsparameter konfiguriert sind. Sie können das Auditing einrichten, um Änderungen wie folgt zu verfolgen.

- Für jede Sitzung, nach Benutzer. Auf der Sitzungsebene können Sie den vollständig qualifizierten Befehlstext erfassen.
- Für jedes Objekt, nach Benutzer und nach Datenbank.

Die Objektüberwachungsfunktion wird aktiviert, wenn Sie die `rds_pgaudit`-Rolle in Ihrem System erstellen und diese Rolle dann dem `pgaudit.role`-Parameter in Ihrer benutzerdefinierten Parametergruppe hinzufügen. Standardmäßig ist der `pgaudit.role`-Parameter nicht festgelegt und der einzig zulässige Wert ist `rds_pgaudit`. Bei den folgenden Schritten wird davon ausgegangen, dass `pgaudit` initialisiert wurde und Sie die `pgaudit`-Erweiterung gemäß den Schritten unter Einrichtung der pgAudit Erweiterung erstellt haben.

```
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: statement: SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: AUDIT: SESSION,2,1,READ,SELECT,TABLE,public.support,"SELECT
feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;",<none>
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:LOG: QUERY STATISTICS
2022-10-07 23:36:51 UTC:52.95.4.10(14410):postgres@labdb:[1374]:DETAIL: ! system usage stats:
! 0.009494 s user, 0.007442 s system, 0.141985 s elapsed
! [0.022327 s user, 0.007442 s system total]
```

Wie in diesem Beispiel gezeigt, enthält die Zeile SESSION "LOGAUDIT::" unter anderem Informationen über die Tabelle und ihr Schema.

So richten Sie die Objektüberwachung ein

1. Wird verwendet`psql`, um eine Verbindung zur herzustellen. RDSfür die SQL Postgre-DB-Instance.

   ```
   psql --host=your-instance-name.aws-region.rds.amazonaws.com --port=5432 --
   username=postgrespostgres --password --dbname=labdb
   ```

2. Erstellen Sie mithilfe des folgenden Befehls eine Datenbankrolle mit dem Namen `rds_pgaudit`.

```
labdb=> CREATE ROLE rds_pgaudit;
CREATE ROLE
labdb=>
```

3. Schließen Sie die `psql`-Sitzung.

```
labdb=> \q
```

Verwenden Sie in den nächsten Schritten den AWS CLI um die Audit-Log-Parameter in Ihrer benutzerdefinierten Parametergruppe zu ändern.

4. Verwenden Sie Folgendes AWS CLI Befehl, auf den der `pgaudit.role` Parameter gesetzt werden soll`rds_pgaudit`. Standardmäßig ist dieser Parameter leer und der einzig zulässige Wert ist `rds_pgaudit`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot"
 \
   --region aws-region
```

5. Verwenden Sie Folgendes AWS CLI Befehl, um die für die SQL Postgre-DB-Instance neu zu starten, damit Ihre Änderungen an den Parametern wirksam werden.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

6. Führen Sie den folgenden Befehl aus, um zu bestätigen, dass `pgaudit.role` auf `rds_pgaudit` festgelegt ist.

```
SHOW pgaudit.role;
pgaudit.role
-----------------
rds_pgaudit
```

Um die pgAudit Protokollierung zu testen, können Sie mehrere Beispielbefehle ausführen, die Sie prüfen möchten. Sie könnten beispielsweise die folgenden Befehle ausführen.

```
CREATE TABLE t1 (id int);
GRANT SELECT ON t1 TO rds_pgaudit;
SELECT * FROM t1;
id
----
(0 rows)
```

Die Datenbankprotokolle sollten dann einen Eintrag ähnlich dem folgenden enthalten.

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

Weitere Informationen zur Anzeige der Protokolle finden Sie unter Überwachung von Amazon RDS Amazon.

Weitere Informationen über die pgAudit Erweiterung finden Sie pgAuditunter GitHub.

Benutzer oder Datenbanken von der Audit-Protokollierung ausschließen

Wie unter beschriebenDatenbank-Protokolldateien von RDS für PostgreSQL, verbrauchen SQL Postgre-Protokolle Speicherplatz. Durch die Verwendung der pgAudit Erweiterung wird das in Ihren Protokollen gesammelte Datenvolumen in unterschiedlichem Maße erhöht, je nachdem, welche Änderungen Sie verfolgen. Möglicherweise müssen Sie nicht jeden Benutzer oder jede Datenbank in Ihrem prüfen. RDSfür die SQL Postgre-DB-Instance.

Sie können Benutzer und Datenbanken von der Prüfung ausschließen, um die Auswirkungen auf Ihren Speicher zu minimieren und die unnötige Erfassung von Audit-Datensätzen zu vermeiden. Sie können die Protokollierung auch innerhalb einer bestimmten Sitzung ändern. In den nachstehenden Beispielen wird die Vorgehensweise dazu veranschaulicht.

> **Note**
>
> Parametereinstellungen auf Sitzungsebene haben Vorrang vor den Einstellungen in der für die Postgre-DB-Instance. SQL Wenn Sie nicht möchten, dass Datenbankbenutzer

Ihre Konfigurationseinstellungen für die Audit-Protokollierung umgehen, müssen Sie ihre Berechtigungen ändern.

Angenommen, Ihr für die SQL Postgre-DB-Instance ist so konfiguriert, dass für alle Benutzer und Datenbanken dasselbe Aktivitätsniveau geprüft wird. Sie entscheiden dann, dass Sie den Benutzer `myuser` nicht überprüfen möchten. Sie können die Überwachung für `myuser` mit dem folgenden SQL Befehl deaktivieren.

```
ALTER USER myuser SET pgaudit.log TO 'NONE';
```

Anschließend können Sie die folgende Abfrage verwenden, um die Spalte `user_specific_settings` auf `pgaudit.log` zu überprüfen und zu bestätigen, dass der Parameter auf `NONE` festgelegt ist.

```
SELECT
    usename AS user_name,
    useconfig AS user_specific_settings
FROM
    pg_user
WHERE
    usename = 'myuser';
```

Die Ausgabe sollte folgendermaßen aussehen.

```
 user_name | user_specific_settings
-----------+------------------------
 myuser    | {pgaudit.log=NONE}
(1 row)
```

Sie können die Protokollierung für einen bestimmten Benutzer während seiner Datenbanksitzung mit dem folgenden Befehl deaktivieren.

```
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'none';
```

Verwenden Sie die folgende Abfrage, um die Einstellungsspalte für eine bestimmte Benutzer- und Datenbankkombination auf pgaudit.log zu überprüfen.

```
SELECT
```

```
    usename AS "user_name",
    datname AS "database_name",
    pg_catalog.array_to_string(setconfig, E'\n') AS "settings"
FROM
    pg_catalog.pg_db_role_setting s
    LEFT JOIN pg_catalog.pg_database d ON d.oid = setdatabase
    LEFT JOIN pg_catalog.pg_user r ON r.usesysid = setrole
WHERE
    usename = 'myuser'
    AND datname = 'mydatabase'
ORDER BY
    1,
    2;
```

Die Ausgabe entspricht weitgehend der folgenden.

```
  user_name | database_name |     settings
-----------+---------------+-----------------
 myuser    | mydatabase    | pgaudit.log=none
(1 row)
```

Nachdem Sie das Auditing für `myuser` deaktiviert haben, entscheiden Sie, dass Sie Änderungen an `mydatabase` nicht verfolgen möchten. Sie können das Auditing für diese spezifische Datenbank mit dem folgenden Befehl deaktivieren.

```
ALTER DATABASE mydatabase SET pgaudit.log to 'NONE';
```

Verwenden Sie dann die folgende Abfrage, um in der Spalte database_specific_settings zu überprüfen, ob pgaudit.log auf gesetzt ist. NONE

```
SELECT
a.datname AS database_name,
b.setconfig AS database_specific_settings
FROM
pg_database a
FULL JOIN pg_db_role_setting b ON a.oid = b.setdatabase
WHERE
a.datname = 'mydatabase';
```

Die Ausgabe sollte folgendermaßen aussehen.

```
 database_name | database_specific_settings
---------------+----------------------------
 mydatabase    | {pgaudit.log=NONE}
(1 row)
```

Verwenden Sie den folgenden Befehl, um die Einstellungen wieder auf die Standardeinstellung für myuser festzulegen:

```
ALTER USER myuser RESET pgaudit.log;
```

Verwenden Sie den folgenden Befehl, um die Einstellungen wieder auf die Standardeinstellung für eine Datenbank festzulegen.

```
ALTER DATABASE mydatabase RESET pgaudit.log;
```

Verwenden Sie den folgenden Befehl, um Benutzer und Datenbank wieder auf die Standardeinstellung festzulegen.

```
ALTER USER myuser IN DATABASE mydatabase RESET pgaudit.log;
```

Sie können auch bestimmte Ereignisse im Protokoll erfassen, indem Sie `pgaudit.log` auf einen der anderen zulässigen Werte für den `pgaudit.log`-Parameter festlegen. Weitere Informationen finden Sie unter Liste der zulässigen Einstellungen für den `pgaudit.log`-Parameter.

```
ALTER USER myuser SET pgaudit.log TO 'read';
ALTER DATABASE mydatabase SET pgaudit.log TO 'function';
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'read,function'
```

Referenz für die pgAudit Erweiterung

Sie können den gewünschten Detaillierungsgrad für Ihr Audit-Protokoll angeben, indem Sie einen oder mehrere der in diesem Abschnitt aufgeführten Parameter ändern.

pgAudit Verhalten kontrollieren

Sie können die Audit-Protokollierung steuern, indem Sie einen oder mehrere der in der folgenden Tabelle aufgeführten Parameter ändern.

Parameter	Beschreibung
`pgaudit.log`	Gibt die Anweisungsklassen an, die durch die Sitzungs-Audit-Protokollierung erfasst werden. Zulässige Werte sind „ddl“, „function“, „misc“, „read“, „role“, „write“, „none“, „all“. Weitere Informationen finden Sie unter Liste der zulässigen Einstellungen für den `pgaudit.log` -Parameter.
`pgaudit.log_catalog`	Wenn diese Option aktiviert ist (auf 1 festgelegt), werden Anweisungen zum Audit-Trail hinzugefügt, wenn sich alle Beziehungen in einer Anweisung in pg_catalog befinden.
`pgaudit.log_level`	Gibt die Protokollstufe an, die für Protokolleinträge verwendet werden soll. Zulässige Werte: „debug5“, „debug4“, „debug3“, „debug2“, „debug1“, „info“, „notice“, „warning“, „log“
`pgaudit.log_parameter`	Wenn diese Option aktiviert ist (auf 1 festgelegt), werden die mit der Anweisung übergebenen Parameter im Audit-Protokoll erfasst.
`pgaudit.log_relation`	Wenn diese Option aktiviert ist (auf 1 gesetzt), erstellt das Auditprotokoll für die Sitzung einen separaten Protokolleintrag für jede Beziehung (TABLE, usw.)VIEW, auf die in einer SELECT DML ODER-Anweisung verwiesen wird.
`pgaudit.log_statement_once`	Gibt an, ob die Protokollierung den Anweisungstext und die Parameter mit dem ersten Protokolleintrag für eine Kombination aus Anweisung/Unteranweisung oder bei jedem Eintrag enthält.
`pgaudit.role`	Gibt die Hauptrolle an, die für die Objektüberwachungsprotokollierung verwendet werden soll. Der einzig zulässige Eintrag ist `rds_pgaudit` .

Liste der zulässigen Einstellungen für den **`pgaudit.log`**-Parameter

Wert	Beschreibung
none	Dies ist die Standardeinstellung. Es werden keine Datenbankänderungen protokolliert.
all	Protokolliert alles (Lesen, Schreiben, Funktion, Rolle, DDL, Verschied enes).
ddl	Protokolliert alle Anweisungen in der Datendefinitionssprache (DDL), die nicht in der `ROLE` Klasse enthalten sind.
function	Protokolliert Funktionsaufrufe und `DO`-Blöcke.
misc	Protokolliert verschiedene Befehle wie `DISCARD`, `FETCH`, `CHECKPOINT` , `VACUUM` und `SET`.
read	Protokolliert `SELECT` und `COPY`, wenn die Quelle eine Beziehung (z. B. eine Tabelle) oder eine Abfrage ist.
role	Protokolliert Anweisungen in Bezug auf Rollen und Berechtigungen wie `REVOKE`, `CREATE ROLE`, `ALTER ROLE` und `DROP ROLE`.
write	Protokolliert `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` und `COPY`, wenn das Ziel eine Beziehung (Tabelle) ist.

Um mehrere Ereignistypen mit der Sitzungsüberwachung zu protokollieren, verwenden Sie eine kommagetrennte Liste. Um alle Ereignistypen zu protokollieren, legen Sie `pgaudit.log` auf `ALL` fest. Starten Sie Ihre DB-Instance neu, um die Änderungen zu übernehmen.

Mit der Objektüberwachung können Sie die Überwachungsprotokollierung verfeinern, um mit bestimmten Beziehungen zu arbeiten. Sie können z. B. angeben, dass Sie eine Audit-Protokollierung für `READ`-Vorgänge in einer oder mehreren Tabellen wünschen.

Planen der Wartung mit der PostgreSQL-Erweiterung pg_cron

Sie können die PostgreSQL-Erweiterung `pg_cron` verwenden, um Wartungsbefehle innerhalb einer PostgreSQL-Datenbank zu planen. Weitere Informationen zu der Erweiterung finden Sie unter Was ist pg_cron? in der pg_cron-Dokumentation.

Die Erweiterung `pg_cron` wird von RDS-PostgreSQL-Engine-Versionen 12.5 und höher unterstützt.

Weitere Informationen zur Verwendung von `pg_cron` finden Sie unter Planen von Aufträgen mit pg_cron auf RDS für PostgreSQL oder Ihren Datenbanken der mit Aurora PostgreSQL kompatiblen Edition.

Themen

- Einrichten der pg_con-Erweiterung
- Gewähren von Berechtigungen zur Verwendung von pg_cron für Datenbankbenutzer
- Planen von pg_cron-Aufträgen
- Referenz für die pg_cron-Erweiterung

Einrichten der pg_con-Erweiterung

Richten Sie die Erweiterung `pg_cron` wie folgt ein:

1. Ändern Sie die benutzerdefinierte Parametergruppe, die mit Ihrer PostgreSQL-DB-Instance verknüpft ist, indem Sie `pg_cron` dem Parameterwert `shared_preload_libraries` hinzufügen.
   - Wenn Ihre DB-Instance von RDS für PostgreSQL den Parameter `rds.allowed_extensions` verwendet, um Erweiterungen, die installiert werden können, explizit aufzulisten, müssen Sie die Erweiterung `pg_cron` in die Liste aufnehmen. Nur bestimmte Versionen von RDS für PostgreSQL unterstützen den Parameter `rds.allowed_extensions`. Standardmäßig sind alle verfügbaren Erweiterungen zulässig. Weitere Informationen finden Sie unter Einschränkung der Installation von Postgre-Erweiterungen SQL.

   Starten Sie die PostgreSQL-DB-Instance neu, damit die Änderungen an der Parametergruppe in Kraft treten. Weitere Informationen zum Arbeiten mit Parametergruppen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.
2. Nachdem die PostgreSQL-DB-Instance neu gestartet wurde, führen Sie den folgenden Befehl mit einem Konto aus, das über `rds_superuser`-Berechtigungen verfügt. Wenn Sie beispielsweise

beim Erstellen Ihrer RDS-for-PostgreSQL-DB-Instance die Standardeinstellungen verwendet haben, verbinden Sie sich als Benutzer `postgres` und erstellen Sie die Erweiterung.

```
CREATE EXTENSION pg_cron;
```

Der Scheduler `pg_cron` ist in der standardmäßigen PostgreSQL-Datenbank mit dem Namen `postgres` festgelegt. Die Objekte `pg_cron` werden in dieser `postgres`-Datenbank erstellt und alle Planungsaktionen werden in dieser Datenbank ausgeführt.

3. Sie können die Standardeinstellungen verwenden oder Aufgaben für die Ausführung in anderen Datenbanken innerhalb Ihrer PostgreSQL-DB-Instance planen. Informationen zum Planen von Aufgaben für andere Datenbanken innerhalb Ihrer PostgreSQL-DB-Instance finden Sie im Beispiel unter Planen einer Cron-Aufgabe für eine andere als die Standard-Datenbank.

Gewähren von Berechtigungen zur Verwendung von pg_cron für Datenbankbenutzer

Zum Installieren der Erweiterung `pg_cron` sind `rds_superuser`-Berechtigungen erforderlich. Berechtigungen zur Verwendung von `pg_cron` können anderen Datenbankbenutzern (von einem Mitglied der Gruppe/Rolle `rds_superuser`) gewährt werden, damit diese ihre eigenen Aufträge planen können. Wir empfehlen Ihnen, dem `cron`-Schema Berechtigungen nur nach Bedarf, wenn es die Abläufe in Ihrer Produktionsumgebung verbessert, zu gewähren.

Führen Sie den folgenden Befehl aus, um einem Datenbankbenutzer Berechtigungen im `cron`-Schema zu erteilen:

```
postgres=> GRANT USAGE ON SCHEMA cron TO db-user;
```

Dies gibt die *`db-user`* Erlaubnis, auf das `cron` Schema zuzugreifen, um Cron-Jobs für die Objekte zu planen, für die sie Zugriffsberechtigungen haben. Wenn der Datenbankbenutzer keine Berechtigungen hat, schlägt der Auftrag fehl, nachdem die Fehlermeldung in der `postgresql.log`-Datei angezeigt wurde, wie nachfolgend dargestellt:

```
2020-12-08 16:41:00 UTC::@:[30647]:ERROR: permission denied for table table-name
2020-12-08 16:41:00 UTC::@:[27071]:LOG: background worker "pg_cron" (PID 30647) exited
 with exit code 1
```

Mit anderen Worten, stellen Sie sicher, dass Datenbankbenutzer, denen Berechtigungen für das `cron` Schema erteilt wurden, auch über Berechtigungen für die Objekte (Tabellen, Schemas usw.) verfügen, die sie planen möchten.

Die Details des Cron-Jobs und dessen Erfolg oder Misserfolg werden ebenfalls in der `cron.job_run_details` Tabelle erfasst. Weitere Informationen finden Sie unter Tabellen zum Planen von Jobs und zur Erfassung des Status .

Planen von pg_cron-Aufträgen

Die folgenden Abschnitte zeigen, wie Sie verschiedene Verwaltungsaufgaben mit `pg_cron`-Aufträgen planen können.

> **Note**
>
> Wenn Sie `pg_cron`-Aufträge erstellen, überprüfen Sie, dass die Einstellung `max_worker_processes` größer ist als die Anzahl von `cron.max_running_jobs`. Ein `pg_cron`-Auftrag schlägt fehl, wenn keine Hintergrund-Workerprozesse ausgeführt werden. Die Standardanzahl von `pg_cron`-Aufträgen ist `5`. Weitere Informationen finden Sie unter Parameter für die Verwaltung der pg_cron-Erweiterung.

Themen

- Bereinigen von Tabellen
- Löschen der Verlaufstabelle pg_cron
- Protokollieren von Fehlern nur in der Datei postgresql.log
- Planen einer Cron-Aufgabe für eine andere als die Standard-Datenbank

Bereinigen von Tabellen

Autovacuum übernimmt die Entfernung für die meisten Fälle Möglicherweise möchten Sie jedoch eine Bereinigung für eine bestimmte Tabelle zu einem Zeitpunkt Ihrer Wahl planen.

Weitere Informationen finden Sie auch unter, .

Es folgt ein Beispiel für die Verwendung der Funktion `cron.schedule` zum Einrichten eines Auftrags, der jeden Tag um 22:00 Uhr (GMT) `VACUUM FREEZE` auf eine bestimmte Tabelle anwenden soll.

```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts');
 schedule
----------
1
```

```
(1 row)
```

Nachdem das vorangehende Beispiel ausgeführt wurde, können Sie den Verlauf in der `cron.job_run_details`-Tabelle wie folgt überprüfen.

```
postgres=> SELECT * FROM cron.job_run_details;
jobid  | runid | job_pid | database | username | command                        |
 status    | return_message | start_time                    | end_time
-------+-------+---------+----------+----------+--------------------------------
+-----------+----------------+-------------------------------
+-------------------------------
 1     | 1     | 3395    | postgres | adminuser| vacuum freeze pgbench_accounts
 | succeeded | VACUUM         | 2020-12-04 21:10:00.050386+00 | 2020-12-04
 21:10:00.072028+00
(1 row)
```

Im Folgenden finden Sie eine Abfrage der `cron.job_run_details` Tabelle, um die fehlgeschlagenen Jobs zu sehen.

```
postgres=> SELECT * FROM cron.job_run_details WHERE status = 'failed';
jobid | runid | job_pid | database | username | command                        | status
 | return_message                                | start_time                    |
 end_time
------+-------+---------+----------+----------+--------------------------------+--------
+-----------------------------------------------+-------------------------------
+-------------------------------
 5    | 4     | 30339   | postgres | adminuser| vacuum freeze pgbench_account | failed
 | ERROR: relation "pgbench_account" does not exist | 2020-12-04 21:48:00.015145+00 |
 2020-12-04 21:48:00.029567+00
(1 row)
```

Weitere Informationen finden Sie unter Tabellen zum Planen von Jobs und zur Erfassung des Status .

Löschen der Verlaufstabelle pg_cron

Die `cron.job_run_details`-Tabelle enthält einen Verlauf von Cron-Aufgaben, die im Laufe der Zeit sehr groß werden können. Wir empfehlen Ihnen, eine Aufgabe zu planen, die diese Tabelle bereinigt. Beispielsweise kann es für die Fehlerbehebung ausreichen, die Eingaben einer Woche beizubehalten.

Im folgenden Beispiel wird die cron.schedule-Funktion verwendet, um eine Aufgabe zu planen, die jeden Tag um Mitternacht ausgeführt wird, um die `cron.job_run_details`-Tabelle zu bereinigen.

Die Aufgabe behält nur die letzten sieben Tage. Verwenden Sie Ihr `rds_superuser`-Konto, um die Aufgabe wie folgt zu planen.

```
SELECT cron.schedule('0 0 * * *', $$DELETE
    FROM cron.job_run_details
    WHERE end_time < now() - interval '7 days'$$);
```

Weitere Informationen finden Sie unter Tabellen zum Planen von Jobs und zur Erfassung des Status .

Protokollieren von Fehlern nur in der Datei postgresql.log

Wenn Sie verhindern möchten, dass in die `cron.job_run_details`-Tabelle geschrieben wird, ändern Sie die mit der PostgreSQL-DB-Instance verknüpfte Parametergruppe und deaktivieren Sie den Parameter `cron.log_run`. Es werden keine Schreibvorgänge der `pg_cron`-Erweiterung in die Tabelle mehr durchgeführt und nur noch Fehler in der `postgresql.log`-Datei erfasst. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Verwenden Sie den folgenden Befehl, um den Wert des `cron.log_run`-Parameters zu überprüfen.

```
postgres=> SHOW cron.log_run;
```

Weitere Informationen finden Sie unter Parameter für die Verwaltung der pg_cron-Erweiterung.

Planen einer Cron-Aufgabe für eine andere als die Standard-Datenbank

Die Metadaten für `pg_cron` werden alle in der PostgreSQL-Standarddatenbank mit dem Namen `postgres` gespeichert. Da Hintergrund-Worker für die Ausführung der Maintenance-Cron-Aufgaben verwendet werden, können Sie eine Aufgabe in jeder Ihrer Datenbanken innerhalb der PostgreSQL-DB-Instance planen.

> **Note**
>
> Nur Benutzer mit `rds_superuser` Rollen oder `rds_superuser` Rechten können alle Cron-Jobs in der Datenbank auflisten. Andere Benutzer können nur ihre eigenen Jobs in der `cron.job` Tabelle sehen.

1. Planen Sie die Aufgabe in der Cron-Datenbank wie gewohnt mit cron.schedule.

```
postgres=> SELECT cron.schedule('database1 manual vacuum', '29 03 * * *', 'vacuum
 freeze test_table');
```

2. Aktualisieren Sie als Benutzer mit der `rds_superuser`-Rolle die Datenbankspalte für die soeben erstellte Aufgabe, damit sie in einer anderen Datenbank innerhalb Ihrer PostgreSQL-DB-Instance ausgeführt wird.

```
postgres=> UPDATE cron.job SET database = 'database1' WHERE jobid = 106;
```

3. Überprüfen Sie dies, indem Sie die `cron.job`-Tabelle abfragen.

```
postgres=> SELECT * FROM cron.job;
jobid | schedule    | command                        | nodename  | nodeport |
 database | username  | active | jobname
------+-------------+--------------------------------+-----------+----------
+----------+-----------+--------+-------------------------
106   | 29 03 * * * | vacuum freeze test_table       | localhost | 8192     |
 database1| adminuser | t      | database1 manual vacuum
  1   | 59 23 * * * | vacuum freeze pgbench_accounts | localhost | 8192     |
 postgres | adminuser | t      | manual vacuum
(2 rows)
```

> **Note**
>
> In einigen Fällen können Sie eine Cron-Aufgabe hinzufügen, die Sie in einer anderen Datenbank ausführen möchten. Bevor Sie die richtige Datenbankspalte aktualisieren, versucht die Aufgabe in solchen Fällen möglicherweise, in der Standarddatenbank (`postgres`) ausgeführt zu werden. Wenn der Benutzername über Berechtigungen verfügt, wird die Aufgabe erfolgreich in der Standarddatenbank ausgeführt.

Referenz für die pg_cron-Erweiterung

Sie können die folgenden Parameter, Funktionen und Tabellen mit der `pg_cron`-Erweiterung verwenden. Weitere Informationen finden Sie unter Was ist pg_cron? in der pg_cron-Dokumentation.

Themen

- Parameter für die Verwaltung der pg_cron-Erweiterung

- Funktionsreferenz: cron.schedule
- Funktionsreferenz: cron.unschedule
- Tabellen zum Planen von Jobs und zur Erfassung des Status

Parameter für die Verwaltung der pg_cron-Erweiterung

Es folgt eine Liste der Parameter zur Steuerung des Erweiterungsverhaltens von `pg_cron`.

Parameter	Beschreibung
cron.database_name	Die Datenbank, in der `pg_cron`-Metadaten aufbewahrt werden.
`cron.host`	Der Hostname für die Verbindung mit PostgreSQL. Dieser Wert kann nicht verändert werden.
`cron.log_run`	Protokollieren Sie jeden ausgeführten Auftrag in der Tabelle `job_run_details` . Die Werte sind `on` oder `off`. Weitere Informationen finden Sie unter Tabellen zum Planen von Jobs und zur Erfassung des Status .
`cron.log_statement`	Protokolliert alle Cron-Anweisungen, bevor Sie ausgeführt werden. Die Werte sind `on` oder `off`.
`cron.max_running_jobs`	Die maximale Anzahl von Aufgaben, die gleichzeitig ausgeführt werden können.
`cron.use_background_workers`	Verwenden Sie Hintergrund-Worker anstelle von Client-Sitzungen. Dieser Wert kann nicht verändert werden.

Verwenden Sie den folgenden SQL-Befehl, um diese Parameter und ihre Werte anzuzeigen.

```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.%'
 ORDER BY name;
```

Funktionsreferenz: cron.schedule

Diese Funktion plant eine Cron-Aufgabe. Die Aufgabe ist anfänglich in der `postgres` Standarddatenbank geplant. Die Funktion gibt einen `bigint` Wert zurück, der den Aufgabenbezeichner darstellt. Informationen zum Planen von Aufgaben für die Ausführung in anderen Datenbanken innerhalb Ihrer PostgreSQL-DB-Instance finden Sie im Beispiel unter Planen einer Cron-Aufgabe für eine andere als die Standard-Datenbank.

Die Funktion hat zwei Syntaxformate.

Syntax

```
cron.schedule (job_name,
    schedule,
    command
);

cron.schedule (schedule,
    command
);
```

Parameter

Parameter	Beschreibung
`job_name`	Der Name der Cron-Aufgabe.
`schedule`	Text, der den Zeitplan für die Cron-Aufgabe angibt. Das Format ist das Standard-Cron-Format.
`command`	Text des auszuführenden Befehls.

Beispiele

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
```

```
 schedule
----------
      145
(1 row)

postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
 schedule
----------
      146
(1 row)
```

Funktionsreferenz: cron.unschedule

Diese Funktion löscht eine Cron-Aufgabe. Sie können entweder den `job_name` oder die `job_id` angeben. Eine Richtlinie stellt sicher, dass Sie der Besitzer sind, um den Plan für die Aufgabe zu entfernen. Die Funktion gibt einen Booleschen Wert zurück, der Erfolg oder Misserfolg anzeigt.

Die Funktion weist die folgende Syntax auf.

Syntax

```
cron.unschedule (job_id);

cron.unschedule (job_name);
```

Parameter

Parameter	Beschreibung
`job_id`	Ein Aufgabenbezeichner, der von der `cron.schedule` Funktion zurückgegeben wurde, als die Cron-Aufgabe geplant wurde.
`job_name`	Der Name einer Cron-Aufgabe, die mit der `cron.schedule` Funktion geplant wurde.

Beispiele

```
postgres=> SELECT cron.unschedule(108);
```

```
 unschedule
------------
 t
(1 row)

postgres=> SELECT cron.unschedule('test');
 unschedule
------------
 t
(1 row)
```

Tabellen zum Planen von Jobs und zur Erfassung des Status

Die folgenden Tabellen werden verwendet, um die Cron-Aufgaben zu planen und aufzuzeichnen, wie die Aufgaben abgeschlossen wurden.

Tabelle	Beschreibung
`cron.job`	Enthält die Metadaten zu jeder geplanten Aufgabe. Die meisten Interaktionen mit dieser Tabelle sollten über die Funktionen `cron.schedule` und `cron.unschedule` erfolgen. **Important** Wir empfehlen, dieser Tabelle keine Aktualisierungs- oder Einfügeberechtigungen zu gewähren. Dies würde es dem Benutzer ermöglichen, die `username`-Spalte zu aktualisieren, die als `rds-superuser` ausgeführt wird.
`cron.job_run_details`	Enthält Verlaufsdaten zu vergangenen geplanten Aufträgen , die ausgeführt wurden. Dies ist nützlich, um den Status, die Rückgabe von Nachrichten sowie die Start- und Endzeit des ausgeführten Auftrags zu untersuchen. **Note** Um zu verhindern, dass diese Tabelle ins Unendliche wächst, bereinigen Sie sie in regelmäßigen Abständen.

Tabelle	Beschreibung
	Ein Beispiel finden Sie unter Löschen der Verlaufstabelle pg_cron.

Verwenden von pglogical, um Daten zwischen Instances zu synchronisieren

Alle derzeit verfügbaren für SQL Postgre-Versionen unterstützen die `pglogical` Erweiterung. Die Erweiterung pglogical ist älter als die funktional ähnliche logische Replikationsfunktion, die von Postgre in Version 10 eingeführt wurde. SQL Weitere Informationen finden Sie unter Ausführen der logischen Replikation für Amazon RDS for PostgreSQL.

Die `pglogical` Erweiterung unterstützt die logische Replikation zwischen zwei oder mehr . RDSfür SQL Postgre-DB-Instances. Es unterstützt auch die Replikation zwischen verschiedenen SQL Postgre-Versionen und zwischen Datenbanken, die auf RDS SQL Postgre-DB-Instances und Aurora Postgre-DB-Clustern laufen. SQL Die `pglogical`-Erweiterung verwendet ein Publish-Subscribe-Modell, um Änderungen an Tabellen und anderen Objekten, z. B. Sequenzen, von einem Herausgeber auf einen Abonnenten zu replizieren. Sie stützt sich auf einen Replikationsslot, um sicherzustellen, dass Änderungen von einem Herausgeberknoten zu einem Abonnentenknoten synchronisiert werden. Dies ist wie folgt definiert.

- Der Publisher-Knoten ist der für die SQL Postgre-DB-Instance, der die Datenquelle ist, die auf andere Knoten repliziert werden soll. Der Herausgeberknoten definiert die Tabellen, die in einem Veröffentlichungssatz repliziert werden sollen.
- Der Abonnentenknoten ist der für die SQL Postgre-DB-Instance, der WAL Updates vom Herausgeber erhält. Der Abonnent erstellt ein Abonnement, um eine Verbindung zum Herausgeber herzustellen und die WAL dekodierten Daten abzurufen. Wenn der Abonnent das Abonnement erstellt, wird der Replikationsslot auf dem Herausgeberknoten erstellt.

Im Folgenden erfahren Sie, wie Sie die `pglogical`-Erweiterung einrichten.

Themen

- Anforderungen und Einschränkungen für die pglogical-Erweiterung
- Einrichten der pglogical-Erweiterung
- Einrichten der logischen Replikation für die DB-Instance von RDS für PostgreSQL
- Wiederherstellung der logischen Replikation nach einem Hauptversions-Upgrade
- Verwaltung logischer Replikationssteckplätze für for Postgre SQL
- Parameterreferenz für die pglogical-Erweiterung

Anforderungen und Einschränkungen für die pglogical-Erweiterung

Alle derzeit verfügbaren Versionen von for Postgre SQL unterstützen die `pglogical` Erweiterung.

Sowohl der Herausgeberknoten als auch der Abonnentenknoten müssen für die logische Replikation eingerichtet sein.

Die Tabellen, die Sie von einem Herausgeber auf einen Abonnenten replizieren möchten, müssen dieselben Namen und dasselbe Schema haben. Diese Tabellen müssen außerdem dieselben Spalten enthalten und diese müssen dieselben Datentypen verwenden. Sowohl die Herausgeber- als auch die Abonnententabelle müssen dieselben Primärschlüssel haben. Es wird empfohlen, nur die PRIMARY KEY als eindeutige Einschränkung zu verwenden.

Die Tabellen auf dem Abonnentenknoten können für Einschränkungen und Einschränkungen mehr Einschränkungen als für die Tabellen auf dem Herausgeberknoten CHECK enthalten. NOT NULL

Die `pglogical` Erweiterung bietet Funktionen wie die bidirektionale Replikation, die von der in Postgre SQL (Version 10 und höher) integrierten Funktion für logische Replikation nicht unterstützt werden. Weitere Informationen finden Sie unter SQLBidirektionale Postgre-Replikation mit pglogical.

Einrichten der pglogical-Erweiterung

Um die `pglogical` Erweiterung auf Ihrem einzurichten, fügen Sie sie `pglogical` zu den Shared Libraries in der benutzerdefinierten DB-Parametergruppe für Ihre RDS for Postgre-DB-Instance hinzu. SQL Sie müssen außerdem den Wert des `rds.logical_replication`-Parameters auf `1` festlegen, um die logische Dekodierung zu aktivieren. Abschließend erstellen Sie die Erweiterung in der Datenbank. Sie können das AWS Management Console oder das AWS CLI für diese Aufgaben verwenden.

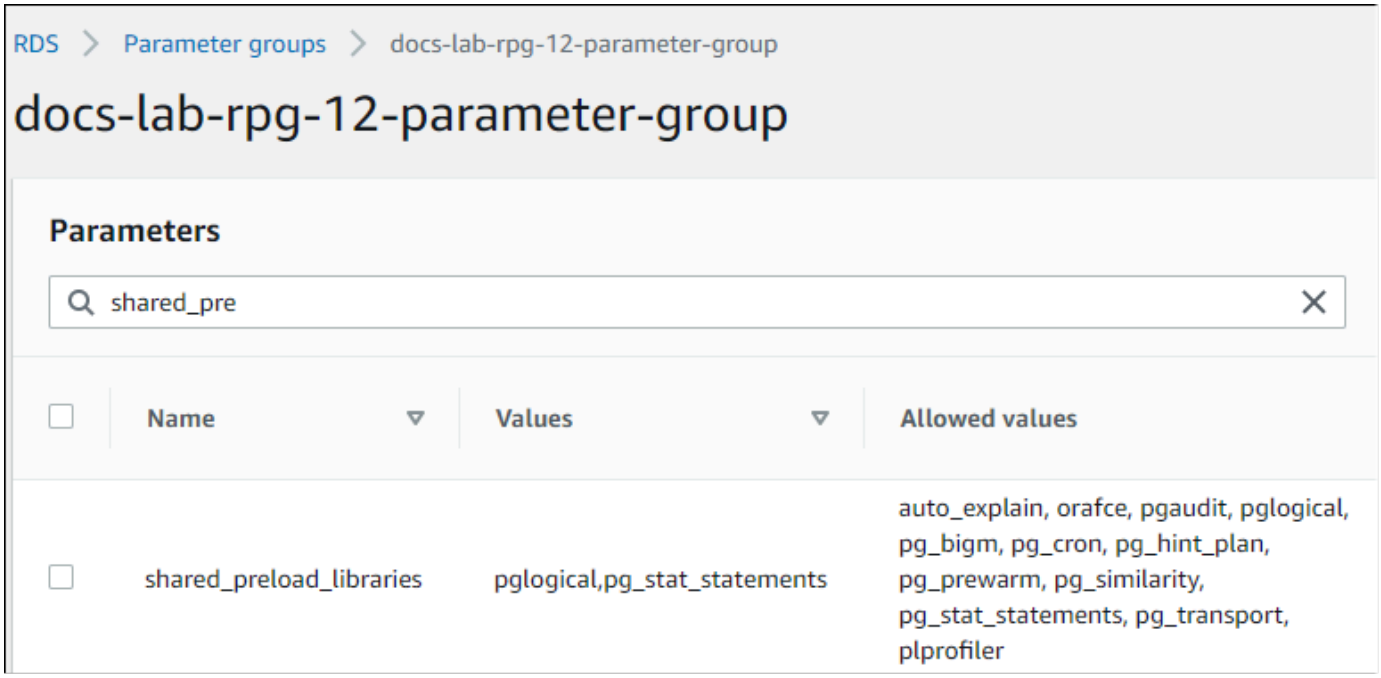
Sie müssen über Berechtigungen als `rds_superuser`-Rolle verfügen, um diese Aufgaben ausführen zu können.

Bei den folgenden Schritten wird davon ausgegangen, dass Ihr für die SQL Postgre-DB-Instance einer benutzerdefinierten zugeordnet ist. Weitere Informationen über das Erstellen einer benutzerdefinierten DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

Konsole

So richten Sie die pglogical-Erweiterung ein

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die aus.
3. Öffnen Sie die Registerkarte Konfiguration für Ihre . RDSfür die SQL Postgre-DB-Instance. Suchen Sie in den Instance-Details den Link Parameter group (Parametergruppe).
4. Wählen Sie den Link, um die benutzerdefinierten Parameter zu öffnen, die mit Ihrem verknüpft sind. RDSfür die SQL Postgre-DB-Instance.
5. Geben Sie in das Suchfeld Parameters (Parameter) `shared_pre` ein, um den `shared_preload_libraries`-Parameter zu finden.
6. Wählen Sie Edit parameters (Parameter bearbeiten) aus, um auf die Eigenschaftswerte zuzugreifen.
7. Fügen Sie `pglogical` der Liste im Feld Values (Werte) hinzu. Verwenden Sie ein Komma, um Elemente in der Werteliste zu trennen.



8. Suchen Sie den `rds.logical_replication`-Parameter und legen Sie ihn auf 1 fest, um die logische Replikation zu aktivieren.
9. Starten Sie die für die SQL Postgre-DB-Instance neu, damit Ihre Änderungen wirksam werden.

10. Wenn die Instance verfügbar ist, können Sie `psql` (oderpgAdmin) verwenden, um eine Verbindung zur herzustellen. RDSfür die SQL Postgre-DB-Instance.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

11. Führen Sie den folgenden Befehl aus, um zu überprüfen, dass pglogical initialisiert ist.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pglogical
(1 row)
```

12. Überprüfen Sie die Einstellung, die die logische Dekodierung ermöglicht, wie folgt.

```
SHOW wal_level;
wal_level
-----------
 logical
(1 row)
```

13. Erstellen Sie die Erweiterung wie folgt.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

14. Wählen Sie Änderungen speichern.
15. Öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
16. Wählen Sie die für die SQL Postgre-DB-Instance aus der Datenbankliste aus, um sie auszuwählen, und wählen Sie dann im Menü Aktionen die Option Reboot aus.

AWS CLI

So richten Sie die pglogical-Erweiterung ein

Um pglogical mithilfe von einzurichten AWS CLI, rufen Sie den modify-db-parameter-groupVorgang auf, um bestimmte Parameter in Ihrer benutzerdefinierten Parametergruppe zu ändern, wie im folgenden Verfahren gezeigt.

1. Verwenden Sie den folgenden AWS CLI Befehl, um den Parameter `pglogical` zu erweitern. `shared_preload_libraries`

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pglogical,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Verwenden Sie den folgenden AWS CLI Befehl, um auf `rds.logical_replication` zu setzen, 1 um die logische Dekodierungsfunktion für die zu aktivieren. RDSfür die Postgre-DB-Instance. SQL

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=rds.logical_replication,ParameterValue=1,ApplyMethod=pending-
reboot" \
   --region aws-region
```

3. Verwenden Sie den folgenden AWS CLI Befehl, um die neu zu starten, sodass die pglogische Bibliothek initialisiert wird.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

4. Wenn die Instance verfügbar ist, verwenden Sie `psql` sie, um eine Verbindung zur herzustellen. RDSfür die SQL Postgre-DB-Instance.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

5. Erstellen Sie die Erweiterung wie folgt.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

6. Starten Sie die für die SQL Postgre-DB-Instance mit dem folgenden AWS CLI Befehl neu.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Einrichten der logischen Replikation für die DB-Instance von RDS für PostgreSQL

Das folgende Verfahren zeigt Ihnen, wie Sie die logische Replikation zwischen zwei DB-Instances von RDS für PostgreSQL starten. Bei den Schritten wird davon ausgegangen, dass sowohl für die Quelle (Herausgeber) als auch für das Ziel (Abonnent) die `pglogical`-Erweiterung eingerichtet wurde, wie unter Einrichten der pglogical-Erweiterung beschrieben.

> **Note**
>
> Der `node_name` eines Abonnentenknotens kann nicht mit beginnen`rds`.

So erstellen Sie den Herausgeberknoten und die zu replizierenden Tabellen

Bei diesen Schritten wird vorausgesetzt, dass Ihre DB-Instance von RDS für PostgreSQL über eine Datenbank mit einer oder mehreren Tabellen verfügt, die Sie auf einen anderen Knoten replizieren möchten. Sie müssen die Tabellenstruktur des Herausgebers für den Abonnenten neu erstellen. Rufen Sie daher bei Bedarf zunächst die Tabellenstruktur ab. Verwenden Sie dazu den `psql`-Metabefehl `\d` *`tablename`* und erstellen Sie dann dieselbe Tabelle auf der Abonnenten-Instance. Mit dem folgenden Verfahren wird zu Demonstrationszwecken eine Beispieltabelle für den Herausgeber (Quelle) erstellt.

1. Verwenden Sie `psql`, um eine Verbindung zu der Instance herzustellen, die die Tabelle enthält, die Sie als Quelle für Abonnenten verwenden möchten.

   ```
   psql --host=source-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=labdb
   ```

   Wenn Sie noch keine vorhandene Tabelle haben, die Sie replizieren möchten, können Sie wie folgt eine Beispieltabelle erstellen.

   a. Erstellen Sie mit der folgenden SQL-Anweisung eine Beispieltabelle.

```
CREATE TABLE docs_lab_table (a int PRIMARY KEY);
```

b. Füllen Sie die Tabelle mit der folgenden SQL-Anweisung mit generierten Daten auf.

```
INSERT INTO docs_lab_table VALUES (generate_series(1,5000));
INSERT 0 5000
```

c. Stellen Sie sicher, dass Daten in der Tabelle vorhanden sind, indem Sie die folgende SQL-Anweisung verwenden.

```
SELECT count(*) FROM docs_lab_table;
```

2. Identifizieren Sie diese DB-Instance von RDS für PostgreSQL wie folgt als Herausgeberknoten.

```
SELECT pglogical.create_node(
    node_name := 'docs_lab_provider',
    dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
 dbname=labdb');
 create_node
-------------
   3410995529
(1 row)
```

3. Fügen Sie die Tabelle, die Sie replizieren möchten, zum Standardreplikationssatz hinzu. Weitere Informationen zu Replikationssätzen finden Sie unter Replikationssätze in der Dokumentation zur pglogical-Erweiterung.

```
SELECT pglogical.replication_set_add_table('default', 'docs_lab_table', 'true',
 NULL, NULL);
 replication_set_add_table
  --------------------------
  t
  (1 row)
```

Die Einrichtung des Herausgeberknotens ist abgeschlossen. Sie können jetzt den Abonnentenknoten einrichten, um die Updates vom Herausgeber zu erhalten.

So richten Sie den Abonnentenknoten ein und erstellen ein Abonnement für den Empfang von Updates

Bei diesen Schritten wird vorausgesetzt, dass die DB-Instance von RDS für PostgreSQL mit der `pglogical`-Erweiterung eingerichtet wurde. Weitere Informationen finden Sie unter Einrichten der pglogical-Erweiterung.

1. Verwenden Sie `psql`, um sich mit der Instance zu verbinden, die Updates vom Herausgeber erhalten soll.

```
psql --host=target-instance.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. Erstellen Sie auf der DB-Instance von RDS für PostgreSQL des Abonnenten dieselbe Tabelle, die auf dem Herausgeber vorhanden ist. In diesem Beispiel heißt die Tabelle `docs_lab_table`. Sie können die Tabelle wie folgt erstellen.

```
CREATE TABLE docs_lab_table (a int PRIMARY KEY);
```

3. Stellen Sie sicher, dass diese Tabelle leer ist.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  0
(1 row)
```

4. Identifizieren Sie diese DB-Instance von RDS für PostgreSQL wie folgt als Abonnentenknoten.

```
SELECT pglogical.create_node(
    node_name := 'docs_lab_target',
    dsn := 'host=target-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=********');
 create_node
-------------
   2182738256
(1 row)
```

5. Erstellen Sie das Abonnement.

```
SELECT pglogical.create_subscription(
   subscription_name := 'docs_lab_subscription',
```

```
   provider_dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=*******',
   replication_sets := ARRAY['default'],
   synchronize_data := true,
   forward_origins := '{}' );
 create_subscription
---------------------
1038357190
(1 row)
```

Wenn Sie diesen Schritt abschließen, werden die Daten aus der Tabelle des Herausgeberknotens in der Tabelle auf dem Abonnentenknoten erstellt. Mit der folgenden SQL-Abfrage können Sie überprüfen, ob dies der Fall ist.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  5000
(1 row)
```

Ab diesem Zeitpunkt werden Änderungen, die an der Tabelle auf dem Herausgeberknoten vorgenommen wurden, auf die Tabelle auf dem Abonnentenknoten repliziert.

Wiederherstellung der logischen Replikation nach einem Hauptversions-Upgrade

Bevor Sie ein Hauptversions-Upgrade eines durchführen können, der als Publisher-Knoten für die logische Replikation eingerichtet ist, müssen Sie alle Replikationssteckplätze löschen, auch die, die nicht aktiv sind. Es wird empfohlen, Datenbanktransaktionen vorübergehend vom Herausgeberknoten umzuleiten, die Replikationsslots zu löschen, den zu aktualisieren und dann die Replikation erneut einzurichten und neu zu starten.

Die Replikationsslots werden nur auf dem Herausgeberknoten gehostet. Der Knoten RDS für SQL Postgre-Abonnenten in einem logischen Replikationsszenario hat keine Steckplätze zum Löschen, aber er kann nicht auf eine Hauptversion aktualisiert werden, solange er als Abonnentenknoten mit einem Abonnement für den Herausgeber vorgesehen ist. Bevor Sie den SQL Abonnentenknoten RDS für Postgre aktualisieren, löschen Sie das Abonnement und den Knoten. Weitere Informationen finden Sie unter Verwaltung logischer Replikationssteckplätze für for Postgre SQL.

Feststellen, dass die logische Replikation unterbrochen wurde

Sie können feststellen, ob der Replikationsprozess unterbrochen wurde, indem Sie entweder den Herausgeberknoten oder den Abonnentenknoten wie folgt abfragen.

So überprüfen Sie den Herausgeberknoten

- Verwenden Sie `psql`, um eine Verbindung mit dem Herausgeberknoten herzustellen, und fragen Sie dann die `pg_replication_slots`-Funktion ab. Notieren Sie sich den Wert in der aktiven Spalte. Normalerweise wird `t` (true) zurückgegeben, was bedeutet, dass die Replikation aktiv ist. Wenn die Abfrage `f` (false) zurückgibt, ist dies ein Hinweis darauf, dass die Replikation an den Abonnenten gestoppt wurde.

```
SELECT slot_name,plugin,slot_type,active FROM pg_replication_slots;
                    slot_name              |      plugin      | slot_type | active
-------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_labcb4fa94_docs_lab3de412c | pglogical_output | logical   | f
(1 row)
```

So überprüfen Sie den Abonnentenknoten

Auf dem Abonnentenknoten können Sie den Status der Replikation auf drei verschiedene Arten überprüfen.

- Suchen Sie in den SQL Postgre-Protokollen auf dem Abonnentenknoten nach Fehlermeldungen. Das Protokoll identifiziert Fehler mit Meldungen, die den Exit-Code 1 enthalten, wie im Folgenden dargestellt.

```
2022-07-06 16:17:03 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 14610) exited with exit code 1
2022-07-06 16:19:44 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 21783) exited with exit code 1
```

- Fragen Sie die `pg_replication_origin`-Funktion ab. Stellen Sie mithilfe von `psql` eine Verbindung mit der Datenbank auf dem Abonnentenknoten her und fragen Sie die `pg_replication_origin`-Funktion wie folgt ab.

```
SELECT * FROM pg_replication_origin;
 roident | roname
---------+--------
```

```
(0 rows)
```

Die leere Ergebnismenge bedeutet, dass die Replikation unterbrochen wurde. Die Ausgabe sollte normalerweise folgendermaßen aussehen.

```
 roident |                      roname
---------+---------------------------------------------------
       1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

- Fragen Sie die `pglogical.show_subscription_status`-Funktion wie im folgenden Beispiel veranschaulicht ab.

```
SELECT subscription_name,status,slot_name FROM pglogical.show_subscription_status();
     subscription_name | status |              slot_name
-------+----------------+--------+-----------------------------------
 docs_lab_subscription | down   | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

Diese Ausgabe zeigt, dass die Replikation unterbrochen wurde. Ihr Status lautet down. Normalerweise zeigt die Ausgabe den Status `replicating` an.

Wenn Ihr logischer Replikationsprozess unterbrochen wurde, können Sie die Replikation wiederherstellen, indem Sie die folgenden Schritte ausführen.

So stellen Sie die logische Replikation zwischen Herausgeber- und Abonnentenknoten wieder her

Um die Replikation wiederherzustellen, trennen Sie zuerst den Abonnenten vom Herausgeberknoten und richten dann das Abonnement erneut ein, wie in diesen Schritten beschrieben.

1. Stellen Sie mit `psql` wie folgt eine Verbindung zum Abonnentenknoten her.

```
psql --host=222222222222.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. Deaktivieren Sie das Abonnement, indem Sie die `pglogical.alter_subscription_disable`-Funktion verwenden.

```
SELECT pglogical.alter_subscription_disable('docs_lab_subscription',true);
 alter_subscription_disable
```

```
----------------------------
 t
(1 row)
```

3. Rufen Sie die ID des Herausgeberknotens ab, indem Sie `pg_replication_origin` wie folgt abfragen.

```
SELECT * FROM pg_replication_origin;
 roident |               roname
---------+--------------------------------------
       1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

4. Verwenden Sie die Antwort aus dem vorherigen Schritt mit dem `pg_replication_origin_create`-Befehl, um die ID zuzuweisen, die vom Abonnement verwendet werden kann, wenn es erneut eingerichtet wurde.

```
SELECT pg_replication_origin_create('pgl_labdb_docs_labcb4fa94_docs_lab3de412c');
  pg_replication_origin_create
------------------------------
                            1
(1 row)
```

5. Aktivieren Sie das Abonnement, indem Sie seinen Namen mit dem Status `true` übergeben, wie im folgenden Beispiel veranschaulicht.

```
SELECT pglogical.alter_subscription_enable('docs_lab_subscription',true);
  alter_subscription_enable
---------------------------
 t
(1 row)
```

Prüfen Sie den Status des Knotens. Sein Status sollte `replicating` wie in diesem Beispiel gezeigt lauten.

```
SELECT subscription_name,status,slot_name
  FROM pglogical.show_subscription_status();
            subscription_name |   status    |              slot_name
------------------------------+-------------+-------------------------------------
 docs_lab_subscription         | replicating |
 pgl_labdb_docs_lab98f517b_docs_lab3de412c
```

```
(1 row)
```

Überprüfen Sie den Status des Replikationsslots des Abonnenten auf dem Herausgeberknoten. Die `active`-Spalte des Slots sollte den Wert `t` (true) zurückgeben, was darauf hinweist, dass die Replikation wiederhergestellt wurde.

```
SELECT slot_name,plugin,slot_type,active
  FROM pg_replication_slots;
                   slot_name              |      plugin      | slot_type | active
------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_lab98f517b_docs_lab3de412c | pglogical_output | logical   | t
(1 row)
```

Verwaltung logischer Replikationssteckplätze für for Postgre SQL

Bevor Sie ein Hauptversions-Upgrade auf der für die SQL Postgre-DB-Instance durchführen können, die in einem logischen Replikationsszenario als Publisher-Knoten dient, müssen Sie die Replikationsslots auf der Instance löschen. Bei der Vorabüberprüfung des Hauptversions-Upgrades werden Sie darüber informiert, dass das Upgrade erst fortgesetzt werden kann, wenn die Slots gelöscht wurden.

Um Slots aus Ihrer RDS SQL Postgre-DB-Instance zu löschen, kündigen Sie zuerst das Abonnement und dann den Steckplatz.

Um Replikationsslots zu identifizieren, die mit der `pglogical`-Erweiterung erstellt wurden, melden Sie sich bei jeder Datenbank an und rufen Sie die Namen der Knoten ab. Wenn Sie den Abonnentenknoten abfragen, erhalten Sie in der Ausgabe sowohl den Herausgeber- als auch den Abonnentenknoten, wie in diesem Beispiel gezeigt.

```
SELECT * FROM pglogical.node;
node_id   |     node_name
------------+------------------
 2182738256 | docs_lab_target
 3410995529 | docs_lab_provider
(2 rows)
```

Die Details zum Abonnement erhalten Sie mit der folgenden Abfrage.

```
SELECT sub_name,sub_slot_name,sub_target
```

```
  FROM pglogical.subscription;
 sub_name |         sub_slot_name          | sub_target
----------+--------------------------------+------------
  docs_lab_subscription     | pgl_labdb_docs_labcb4fa94_docs_lab3de412c | 2182738256
(1 row)
```

Sie können das Abonnement jetzt wie folgt kündigen.

```
SELECT pglogical.drop_subscription(subscription_name := 'docs_lab_subscription');
 drop_subscription
-------------------
                 1
(1 row)
```

Nachdem Sie das Abonnement gekündigt haben, können Sie den Knoten löschen.

```
SELECT pglogical.drop_node(node_name := 'docs-lab-subscriber');
 drop_node
-----------
 t
(1 row)
```

Sie können wie folgt überprüfen, dass der Knoten nicht mehr existiert.

```
SELECT * FROM pglogical.node;
 node_id | node_name
---------+-----------
(0 rows)
```

Parameterreferenz für die pglogical-Erweiterung

In der Tabelle finden Sie Parameter, die der `pglogical`-Erweiterung zugeordnet sind. Parameter wie `pglogical.conflict_log_level` und `pglogical.conflict_resolution` werden verwendet, um Aktualisierungskonflikte zu beheben. Konflikte können auftreten, wenn Änderungen lokal an denselben Tabellen vorgenommen werden, die Änderungen vom Herausgeber abonniert haben. Konflikte können auch in verschiedenen Szenarien auftreten, z. B. bei der bidirektionalen Replikation oder wenn mehrere Abonnenten vom selben Herausgeber replizieren. Weitere Informationen finden Sie unter SQLBidirektionale Postgre-Replikation mit pglogical.

Parameter	Beschreibung
pglogical.batch_inserts	Batch-Inserts wenn möglich Standardmäßig nicht festgelegt. In '1' ändern zum Einschalten, in '0' zum Ausschalten.
pglogical.conflict_log_level	Legt die Protokollstufe für die Protokollierung gelöster Konflikte fest. Unterstützte Zeichenfolgenwerte sind debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, panic.
pglogical.conflict_resolution	Legt die Methode fest, die verwendet werden soll, um Konflikte zu lösen, die lösbar sind. Unterstützte Zeichenfolgenwerte sind error, apply_remote, keep_local, last_update_wins, first_upd ate_wins.
pglogical.extra_connection_ options	Verbindungsoptionen zum Hinzufügen zu allen Peer-Knot enverbindungen
pglogical.synchronous_commit	Spezifischer synchroner Commit-Wert für pglogical
pglogical.use_spi	Verwenden Sie SPI (Server-Programmierschnittstelle) anstelle von Low-Level, um Änderungen vorzunehmen. API In '1' ändern zum Einschalten, in '0' zum Ausschalten. Weitere Informationen zu SPI finden Sie unter Server Programming Interface in der Postgre-Dokumentation. SQL

Verwenden von „pgactive“ zur Unterstützung der Aktiv-Aktiv-Replikation

Die Erweiterung „`pgactive`“ verwendet Aktiv-Aktiv-Replikation, um Schreibvorgänge auf mehreren RDS-für-PostgreSQL-Datenbanken zu unterstützen und zu koordinieren. Amazon RDS for PostgreSQL unterstützt die `pgactive` Erweiterung in den folgenden Versionen:

- RDS für PostgreSQL 17.0 und alle höheren Versionen
- RDS für PostgreSQL 16.1 und höhere 16-Versionen
- RDS für PostgreSQL 15.4-R2 und höhere 15-Versionen
- RDS für PostgreSQL 14.10 und höhere 14-Versionen
- RDS für PostgreSQL 13.13 und höhere 13-Versionen
- RDS für PostgreSQL 12.17 und höhere 12-Versionen
- RDS für PostgreSQL 11.22

> **Note**
>
> Wenn in einer Replikationskonfiguration Schreibvorgänge für mehr als eine Datenbank ausgeführt werden, sind Konflikte möglich. Weitere Informationen finden Sie unter Umgang mit Konflikten bei der Aktiv-Aktiv-Replikation.

Themen

- Einschränkungen für die pgactive-Erweiterung
- Initialisierung der „pgactive“-Erweiterungsfunktion
- Einrichten der Aktiv-Aktiv-Replikation für die DB-Instances von RDS für PostgreSQL
- Messung der Replikationsverzögerung zwischen pgactive-Mitgliedern
- Konfiguration der Parametereinstellungen für die pgactive-Erweiterung
- Aktiv-aktive Konflikte verstehen
- Das pgactive-Schema verstehen
- pgactive-Funktionsreferenz
- Umgang mit Konflikten bei der Aktiv-Aktiv-Replikation
- Umgang mit Sequenzen bei der Aktiv-Aktiv-Replikation

Einschränkungen für die pgactive-Erweiterung

- Alle Tabellen benötigen einen Primärschlüssel, andernfalls sind Aktualisierungen und Löschungen nicht zulässig. Die Werte in der Spalte „Primärschlüssel“ sollten nicht aktualisiert werden.
- Sequenzen können Lücken aufweisen und manchmal keine bestimmte Reihenfolge beachten. Sequenzen werden nicht repliziert. Weitere Informationen finden Sie unter Umgang mit Sequenzen bei der Aktiv-Aktiv-Replikation.
- DDL und große Objekte werden nicht repliziert.
- Sekundäre eindeutige Indizes können zu Datendivergenzen führen.
- Die Sortierung muss auf allen Knoten in der Gruppe identisch sein.
- Das Load Balancing zwischen den Knoten ist ein Anti-Muster.
- Große Transaktionen können zu Verzögerungen bei der Replikation führen.

Initialisierung der „pgactive“-Erweiterungsfunktion

Um die Erweiterungsfunktion „`pgactive`“ auf Ihrer RDS-für-PostgreSQL-DB-Instance zu initialisieren, setzen Sie den Wert des Parameters „`rds.enable_pgactive`“ auf `1` und erstellen Sie dann die Erweiterung in der Datenbank. Dadurch werden die Parameter „`rds.logical_replication`“ und „`track_commit_timestamp`“ automatisch aktiviert und der Wert von `wal_level` wird auf `logical` festgelegt.

Sie müssen über Berechtigungen als `rds_superuser`-Rolle verfügen, um diese Aufgaben ausführen zu können.

Sie können das AWS Management Console oder das verwenden AWS CLI , um den erforderlichen RDS für PostgreSQL-DB-Instances zu erstellen. Bei den folgenden Schritten wird davon ausgegangen, dass Ihre RDS-für-PostgreSQL-DB-Instance einer benutzerdefinierten DB-Parametergruppe zugeordnet ist. Informationen zum Erstellen einer benutzerdefinierten DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

Konsole

So initialisieren Sie die „pgactive“-Erweiterungsfunktion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Ihre DB-Instance von RDS für PostgreSQL aus.

3. Öffnen Sie die Registerkarte Configuration (Konfiguration) für Ihre RDS-für-PostgreSQL-DB-Instance. Suchen Sie in den Instance-Details nach dem Link DB-Instance-Parametergruppe.
4. Wählen Sie den Link aus, um die benutzerdefinierten Parameter zu öffnen, die Ihrer RDS-für-PostgreSQL-DB-Instance zugeordnet sind.
5. Suchen Sie den Parameter „`rds.enable_pgactive`“ und setzen Sie ihn auf „1“, um die „`pgactive`“-Funktion zu initialisieren.
6. Wählen Sie Änderungen speichern aus.
7. Wählen Sie im Navigationsbereich der Amazon-RDS-Konsole die Option Databases (Datenbanken) aus.
8. Wählen Sie Ihre RDS-für-PostgreSQL-DB-Instance aus und wählen Sie dann im Menü Aktionen die Option Neustart aus.
9. Bestätigen Sie den Neustart der DB-Instance, damit die Änderungen in Kraft treten.
10. Wenn die DB-Instance verfügbar ist, können Sie „`psql`“ oder einen anderen PostgreSQL-Client nutzen, um eine Verbindung mit der RDS-für-PostgreSQL-DB-Instance herzustellen.

    Im folgenden Beispiel wird davon ausgegangen, dass Ihre RDS for PostgreSQL-DB-Instance eine Standarddatenbank mit dem Namen hat. *`postgres`*

    ```
    psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
    username=postgres --password=PASSWORD --dbname=postgres
    ```

11. Führen Sie den folgenden Befehl aus, um zu überprüfen, dass „pgactive“ initialisiert ist.

    ```
    postgres=>SELECT setting ~ 'pgactive'
    FROM pg_catalog.pg_settings
    WHERE name = 'shared_preload_libraries';
    ```

    Wenn „`pgactive`“ in `shared_preload_libraries` ist, gibt der vorherige Befehl Folgendes zurück:

    ```
    ?column?
    ----------
     t
    ```

AWS CLI

So initialisieren Sie die „pgactive“-Erweiterungsfunktion

Um das `pgactive` mit dem zu initialisieren AWS CLI, rufen Sie den modify-db-parameter-groupVorgang auf, um bestimmte Parameter in Ihrer benutzerdefinierten Parametergruppe zu ändern, wie im folgenden Verfahren gezeigt.

1. Verwenden Sie den folgenden AWS CLI Befehl, `rds.enable_pgactive 1` um die `pgactive` Funktion für die RDS for PostgreSQL-DB-Instance zu initialisieren.

```
postgres=>aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=rds.enable_pgactive,ParameterValue=1,ApplyMethod=pending-reboot" \
   --region aws-region
```

2. Verwenden Sie den folgenden AWS CLI Befehl, um die RDS for PostgreSQL-DB-Instance neu zu starten, sodass die `pgactive` Bibliothek initialisiert wird.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Wenn die Instance verfügbar ist, stellen Sie über `psql` eine Verbindung mit der DB-Instance von RDS für PostgreSQL her.

```
psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master user --password=PASSWORD --dbname=postgres
```

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, dass „pgactive“ initialisiert ist.

```
postgres=>SELECT setting ~ 'pgactive'
FROM pg_catalog.pg_settings
WHERE name = 'shared_preload_libraries';
```

   Wenn „`pgactive`“ in `shared_preload_libraries` ist, gibt der vorherige Befehl Folgendes zurück:

```
?column?
```

```
----------
 t
```

Einrichten der Aktiv-Aktiv-Replikation für die DB-Instances von RDS für PostgreSQL

Das folgende Verfahren zeigt Ihnen, wie Sie die aktiv-aktive Replikation zwischen zwei , sofern verfügbar. `pgactive` Um das Beispiel für hohe Verfügbarkeit in mehreren Regionen auszuführen, müssen Sie Instances von Amazon RDS für PostgreSQL in zwei verschiedenen Regionen bereitstellen und VPC-Peering einrichten. Weitere Informationen finden Sie unter VPC-Peering.

> **Note**
>
> Beim Senden von Datenverkehr zwischen mehreren Regionen können zusätzliche Kosten anfallen.

Bei diesen Schritten wird davon ausgegangen, dass die RDS for PostgreSQL-DB-Instance mit der `pgactive` Erweiterung aktiviert wurde. Weitere Informationen finden Sie unter Initialisierung der „pgactive“-Erweiterungsfunktion.

So konfigurieren Sie die erste DB-Instance von RDS für PostgreSQL mit der „**`pgactive`**“-Erweiterung

Das folgende Beispiel zeigt, wie die „`pgactive`“-Gruppe erstellt wird, sowie weitere Schritte, die erforderlich sind, um die „`pgactive`“-Erweiterung auf der RDS-für-PostgreSQL-DB-Instance zu erstellen.

1. Verwenden Sie `psql` oder ein anderes Client-Tool, um eine Verbindung zu Ihrer ersten DB-Instance von RDS für PostgreSQL herzustellen.

   ```
   psql --host=firstinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password=PASSWORD --dbname=postgres
   ```

2. Erstellen Sie mit dem folgenden Befehl eine Datenbank auf der RDS-für-PostgreSQL-Instance:

   ```
   postgres=> CREATE DATABASE app;
   ```

3. Wechseln Sie mit dem folgenden Befehl die Verbindung zur neuen Datenbank:

```
\c app
```

4. Erstellen Sie mit den folgenden SQL-Anweisungen eine Beispieltabelle.

   a. Erstellen Sie mit der folgenden SQL-Anweisung eine Beispieltabelle.

   ```
   app=> CREATE SCHEMA inventory;
   CREATE TABLE inventory.products (
   id int PRIMARY KEY, product_name text NOT NULL,
   created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP);
   ```

   b. Füllen Sie die Tabelle mit der folgenden SQL-Anweisung mit Beispieldaten auf.

   ```
   app=> INSERT INTO inventory.products (id, product_name)
   VALUES (1, 'soap'), (2, 'shampoo'), (3, 'conditioner');
   ```

   c. Stellen Sie sicher, dass Daten in der Tabelle vorhanden sind, indem Sie die folgende SQL-Anweisung verwenden.

   ```
    app=>SELECT count(*) FROM inventory.products;

    count
   -------
    3
   ```

5. Erstellen Sie eine „`pgactive`“-Erweiterung für die bestehende Datenbank.

```
app=> CREATE EXTENSION pgactive;
```

6. Verwenden Sie die folgenden Befehle, um die Gruppe pgactive sicher zu erstellen und zu initialisieren:

```
app=>
-- connection info for endpoint1
CREATE SERVER pgactive_server_endpoint1
    FOREIGN DATA WRAPPER pgactive_fdw
    OPTIONS (host '<endpoint1>', dbname 'app');
CREATE USER MAPPING FOR postgres
    SERVER pgactive_server_endpoint1
    OPTIONS (user 'postgres', password '<password>');
      -- connection info for endpoint2
```

```
CREATE SERVER pgactive_server_endpoint2
    FOREIGN DATA WRAPPER pgactive_fdw
    OPTIONS (host '<endpoint2>', dbname 'app');
CREATE USER MAPPING FOR postgres
    SERVER pgactive_server_endpoint2
    OPTIONS (user 'postgres', password '<password>');
```

Jetzt können Sie die Replikationsgruppe initialisieren und diese erste Instanz hinzufügen:

```
SELECT pgactive.pgactive_create_group(
    node_name := 'endpoint1-app',
    node_dsn := 'user_mapping=postgres
 pgactive_foreign_server=pgactive_server_endpoint1'

);
```

Verwenden Sie die folgenden Befehle als alternative, aber weniger sichere Methode, um die pgactive-Gruppe zu erstellen und zu initialisieren:

```
app=> SELECT pgactive.pgactive_create_group(
    node_name := 'node1-app',
    node_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

node1-app ist der Name, den Sie vergeben, um einen Knoten in der „`pgactive`“-Gruppe eindeutig zu identifizieren.

> **Note**
>
> Um diesen Schritt erfolgreich auf einer DB-Instance durchzuführen, auf die öffentlich zugegriffen werden kann, müssen Sie den Parameter „`rds.custom_dns_resolution`“ aktivieren, indem Sie ihn auf 1 setzen.

7. Verwenden Sie den folgenden Befehl, um zu überprüfen, ob die DB-Instance bereit ist:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

Wird der Befehl erfolgreich ausgeführt, wird Ihnen die folgende Ausgabe angezeigt:

```
pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

So konfigurieren Sie die zweite RDS-für-PostgreSQL-Instance und fügen sie der „**pgactive**“-Gruppe hinzu

Das folgende Beispiel zeigt, wie Sie eine RDS-für-PostgreSQL-DB-Instance mit der „pgactive“-Gruppe verbinden können, sowie weitere Schritte, die zum Erstellen der „pgactive“-Erweiterung auf der DB-Instance erforderlich sind.

Bei diesen Schritten wird vorausgesetzt, dass andere DB-Instances von RDS für PostgreSQL mit der „pgactive“-Erweiterung eingerichtet wurde(n). Weitere Informationen finden Sie unter Initialisierung der „pgactive“-Erweiterungsfunktion.

1. Verwenden Sie psql, um sich mit der Instance zu verbinden, die Updates vom Herausgeber erhalten soll.

   ```
   psql --host=secondinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password=PASSWORD --dbname=postgres
   ```

2. Erstellen Sie mit dem folgenden Befehl eine Datenbank auf der zweiten RDS-für-PostgreSQL-Instance:

   ```
   postgres=> CREATE DATABASE app;
   ```

3. Wechseln Sie mit dem folgenden Befehl die Verbindung zur neuen Datenbank:

   ```
   \c app
   ```

4. Erstellen Sie die „pgactive“-Erweiterung für die bestehende Datenbank.

   ```
   app=> CREATE EXTENSION pgactive;
   ```

5. Fügen Sie die zweite RDS for PostgreSQL der pgactive Gruppe auf sicherere Weise hinzu, indem Sie die folgenden Befehle verwenden:

```
-- connection info for endpoint1
CREATE SERVER pgactive_server_endpoint1
    FOREIGN DATA WRAPPER pgactive_fdw
    OPTIONS (host '<endpoint1>', dbname 'app');
CREATE USER MAPPING FOR postgres
    SERVER pgactive_server_endpoint1
    OPTIONS (user 'postgres', password '<password>');

-- connection info for endpoint2
CREATE SERVER pgactive_server_endpoint2
    FOREIGN DATA WRAPPER pgactive_fdw
    OPTIONS (host '<endpoint2>', dbname 'app');
CREATE USER MAPPING FOR postgres
    SERVER pgactive_server_endpoint2
    OPTIONS (user 'postgres', password '<password>');
```

```
SELECT pgactive.pgactive_join_group(
    node_name := 'endpoint2-app',
    node_dsn := 'user_mapping=postgres
 pgactive_foreign_server=pgactive_server_endpoint2',
    join_using_dsn := 'user_mapping=postgres
 pgactive_foreign_server=pgactive_server_endpoint1'
);
```

Verwenden Sie die folgenden Befehle als alternative, aber weniger sichere Methode, um die zweite (RDS for PostgreSQL) der Gruppe hinzuzufügen. `pgactive`

```
app=> SELECT pgactive.pgactive_join_group(
node_name := 'node2-app',
node_dsn := 'dbname=app host=secondinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD',
join_using_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

node2-app ist der Name, den Sie vergeben, um einen Knoten in der „`pgactive`“-Gruppe eindeutig zu identifizieren.

6. Verwenden Sie den folgenden Befehl, um zu überprüfen, ob die DB-Instance bereit ist:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

Wird der Befehl erfolgreich ausgeführt, wird Ihnen die folgende Ausgabe angezeigt:

```
pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Wenn die erste RDS-für-PostgreSQL-Datenbank relativ groß ist, können Sie sehen, dass `pgactive.pgactive_wait_for_node_ready()` den Fortschrittsbericht des Wiederherstellungsvorgangs ausgibt. Die Ausgabe sieht folgendermaßen oder ähnlich aus:

```
NOTICE:  restoring database 'app', 6% of 7483 MB complete
NOTICE:  restoring database 'app', 42% of 7483 MB complete
NOTICE:  restoring database 'app', 77% of 7483 MB complete
NOTICE:  restoring database 'app', 98% of 7483 MB complete
NOTICE:  successfully restored database 'app' from node node1-app in
 00:04:12.274956
 pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Ab diesem Zeitpunkt werden die Daten zwischen den beiden DB-Instances durch „`pgactive`“ synchronisiert.

7. Sie können den folgenden Befehl verwenden, um zu überprüfen, ob die Datenbank der zweiten DB-Instance die Daten enthält:

```
app=> SELECT count(*) FROM inventory.products;
```

Wenn die Daten erfolgreich synchronisiert wurden, sehen Sie die folgende Ausgabe:

```
 count
-------
 3
```

8. Führen Sie den folgenden Befehl aus, um neue Werte einzufügen:

```
app=> INSERT INTO inventory.products (id, product_name) VALUES (4, 'lotion');
```

9. Stellen Sie eine Verbindung zur Datenbank der ersten DB-Instance her und führen Sie die folgende Abfrage aus:

```
app=> SELECT count(*) FROM inventory.products;
```

   Wenn die Aktiv-Aktiv-Replikation initialisiert ist, sieht die Ausgabe ungefähr wie folgt aus:

```
count
-------
 4
```

So trennen Sie eine DB-Instance von der „**`pgactive`**“-Gruppe und entfernen Sie daraus

Sie können eine DB-Instance mit den folgenden Schritten von der „`pgactive`“-Gruppe trennen und daraus entfernen:

1. Sie können die zweite DB-Instance mit dem folgenden Befehl von der ersten DB-Instance trennen:

```
app=> SELECT * FROM pgactive.pgactive_detach_nodes(ARRAY['node2-app']);
```

2. Entfernen Sie die „`pgactive`“-Erweiterung mit dem folgenden Befehl aus der zweiten DB-Instance:

```
app=> SELECT * FROM pgactive.pgactive_remove();
```

   So erzwingen Sie die Entfernung der Erweiterung:

```
app=> SELECT * FROM pgactive.pgactive_remove(true);
```

3. Löschen Sie die Erweiterung mit dem folgenden Befehl:

```
app=> DROP EXTENSION pgactive;
```

Messung der Replikationsverzögerung zwischen pgactive-Mitgliedern

Sie können die folgende Abfrage verwenden, um die Replikationsverzögerung zwischen den `pgactive` Mitgliedern anzuzeigen. Führen Sie diese Abfrage auf jedem `pgactive` Knoten aus, um sich ein vollständiges Bild zu machen.

```
app=> SELECT * FROM pgactive.pgactive_get_replication_lag_info();
#-[ RECORD 1 ]--------+-------------------------------------------
#node_name            | node2-app
#node_sysid           | 7481018224801653637
#application_name     | pgactive:7481018224801653637:send
#slot_name            | pgactive_16385_7481018224801653637_0_16385__
#active               | t
#active_pid           | 783486
#pending_wal_decoding | 0
#pending_wal_to_apply | 0
#restart_lsn          | 0/2108150
#confirmed_flush_lsn  | 0/2154690
#sent_lsn             | 0/2154690
#write_lsn            | 0/2154690
#flush_lsn            | 0/2154690
#replay_lsn           | 0/2154690
#-[ RECORD 2 ]--------+-------------------------------------------
#node_name            | node1-app
#node_sysid           | 7481018033434600853
#application_name     | pgactive:7481018033434600853:send
#slot_name            | pgactive_16385_7481018033434600853_0_16385__
#active               | t
#active_pid           | 783488
#pending_wal_decoding | 0
#pending_wal_to_apply | 0
#restart_lsn          | 0/20F5AD0
#confirmed_flush_lsn  | 0/214EF68
#sent_lsn             | 0/214EF68
#write_lsn            | 0/214EF68
#flush_lsn            | 0/214EF68
#replay_lsn           | 0/214EF68
```

Überwachen Sie mindestens die folgenden Diagnosen:

aktiv

Richten Sie Benachrichtigungen ein, wenn active den Wert false hat, was darauf hinweist, dass der Slot derzeit nicht verwendet wird (die Abonnenteninstanz hat die Verbindung zum Publisher getrennt).

pending_wal_decoding

Bei der logischen Replikation von PostgreSQL werden WAL-Dateien im Binärformat gespeichert. Der Herausgeber muss diese WAL-Änderungen dekodieren und in logische Änderungen umwandeln (z. B. Einfüge-, Aktualisierungs- oder Löschvorgänge).

Die Metrik pending_wal_decoding zeigt die Anzahl der WAL-Dateien, die auf der Herausgeberseite darauf warten, dekodiert zu werden.

Diese Zahl kann aufgrund der folgenden Faktoren steigen:

- Wenn der Abonnent nicht verbunden ist, wird der Status „Aktiv“ auf „Falsch“ gesetzt und der Wert pending_wal_decoding wird erhöht
- Der Slot ist aktiv, aber der Publisher kann mit der Menge der WAL-Änderungen nicht Schritt halten

pending_wal_to_apply

Die Metrik pending_wal_apply gibt die Anzahl der WAL-Dateien an, die auf Abonnentenseite darauf warten, angewendet zu werden.

Verschiedene Faktoren können verhindern, dass der Abonnent Änderungen vornimmt, und möglicherweise zu einem Szenario führen, dass die Festplatte voll ist:

- Schemaunterschiede — zum Beispiel, wenn Sie Änderungen im WAL-Stream für eine Tabelle mit dem Namen sample haben, diese Tabelle aber auf der Abonnentenseite nicht existiert
- Die Werte in den Primärschlüsselspalten wurden aktualisiert
- Sekundäre eindeutige Indizes können zu Datendivergenzen führen

Konfiguration der Parametereinstellungen für die pgactive-Erweiterung

Sie können die folgende Abfrage verwenden, um alle mit der „`pgactive`“-Erweiterung verknüpften Parameter anzuzeigen.

```
app=> SELECT * FROM pg_settings WHERE name LIKE 'pgactive.%';
```

Sie können die `pgactive` Erweiterung mit verschiedenen Parametern konfigurieren. Diese Parameter können entweder über die AWS Management Console oder die AWS CLI-Schnittstelle eingestellt werden.

Die wichtigsten Parameter der pgactive-Erweiterung

Die folgende Tabelle enthält eine Referenz für die Hauptparameter der `pgactive` Erweiterung:

Parameter	Einheit	Standard	Beschreibung
`pgactive.conflict_logging_include_tuples`	`boolean`	–	Protokolliert vollständige Tupelinformationen für die `pgactive` Erweiterung. **Note** Ein Serverneustart ist erforderlich, damit die Änderungen wirksam werden.
`pgactive.log_conflicts_to_table`	`boolean`	–	Legt fest, ob die `pgactive` Erweiterung die erkannten Konflikte in der `pgactive.pgactive_`

Parameter	Einheit	Standard	Beschreibung
			`conflict_history` Tabelle protokolliert. Weitere Informationen finden Sie unter Konfliktprotokollierung. **Note** Ein Serverneustart ist erforderlich, damit die Änderungen wirksam werden.

Parameter	Einheit	Standard	Beschreibung
`pgactive.log_conflicts_to_logfile`	`boolean`	–	Bestimmt, ob die `pgactive` Erweiterung die erkannten Konflikte in der PostgreSQL-Protokolldatei protokolliert. Weitere Informationen finden Sie unter Konfliktprotokollierung. **Note** Ein Serverneustart ist erforderlich, damit die Änderungen wirksam werden.

Parameter	Einheit	Standard	Beschreibung
`pgactive.synchronous_commit`	`boolean`	aus	Bestimmt das Commit-Verhalten für pgactive Apply-Worker. Wenn diese Option deaktiviert (aus) ist, führen Apply-Worker asynchrone Commits durch. Dadurch wird der PostgreSQL-Durchsatz bei Apply-Vorgängen verbessert, die Wiedergabebestätigungen an den Upstream jedoch verzögert. Die Einstellung auf `off` ist immer sicher und führt nicht zum Verlust oder Überspringen von Transaktionen. Diese Einstellung wirkt sich nur auf den Zeitpunkt der Festplattenleerungen auf

Parameter	Einheit	Standard	Beschreibung
			dem Downstream-Knoten und darauf aus, wann Bestätigungen flussaufwärts gesendet werden. Das System verzögert das Senden von Replay-Flush-Bestätigungen, bis Commits durch unabhängige Operationen wie Checkpoints oder regelmäßige Arbeiten auf die Festplatte geschrieben werden. Wenn im Upstream-Bereich jedoch der Downstream unter aufgeführt ist, `off` führt die Einstellung auf dazu`synchronous_standby_names` , dass synchrone Commits auf dem Upstream länger brauchen, bis sie dem

Parameter	Einheit	Standard	Beschreibung
			Client erfolgreich gemeldet werden. In diesem Fall setzen Sie den Parameter auf. on **Note** Selbst wenn dieser Parameter auf on mit den unter aufgeführten Knoten gesetzt istsynchronous_standby_names , können Replikationskonflikte in Active-Active-Konfigurationen trotzdem auftreten

Parameter	Einheit	Standard	Beschreibung
			. Das liegt daran, dass dem System das Sperren zwischen den Knoten und das globale Snapshot-Management fehlen, sodass gleichzeitige Transaktionen auf verschiedenen Knoten dasselbe Tupel modifizieren können. Darüber hinaus beginnen Transaktionen erst mit der

Parameter	Einheit	Standard	Beschreibung
			Replikation, nachdem sie auf dem Upstream-Knoten festgeschrieben wurden. Durch die Aktivierung von synchronem Commit wird die Erweiterung pgactive nicht in ein stets konsistentes System umgewandelt.

Parameter	Einheit	Standard	Beschreibung
pgactive.temp_dump_directory	string	–	Definiert den temporären Speicherpfad, der für das Klonen von Datenbanken bei der Ersteinrichtung erforderlich ist. Dieses Verzeichnis muss für den Postgres-Benutzer beschreibbar sein und über ausreichend Speicherplatz für einen vollständigen Datenbank-Dump verfügen. Das System verwendet diesen Speicherort nur bei der Ersteinrichtung der Datenbank mit logischen Kopiervorgängen. Dieser Parameter wird nicht von verwendet pgactive_

Parameter	Einheit	Standard	Beschreibung
			`init_copy command`.

Parameter	Einheit	Standard	Beschreibung
`pgactive.max_ddl_lock_delay`	`milliseconds`	`-1`	Gibt die maximale Wartezeit für die DDL-Sperre an, bevor gleichzeitige Schreibtransaktionen erzwungen werden. Der Standardwert ist `-1`, der den in `max_standby_streaming_delay` eingestellten Wert übernimmt. Dieser Parameter akzeptiert Zeiteinheiten. Sie können ihn beispielsweise für 10 Sekunden auf 10 Sekunden einstellen. Während dieser Wartezeit versucht das System, DDL-Sperren zu erlangen, während es darauf wartet, dass laufende

Parameter	Einheit	Standard	Beschreibung
			Schreibtransaktionen entweder festgeschrieben oder rückgängig gemacht werden. Weitere Informationen finden Sie unter DDL Locking.

Parameter	Einheit	Standard	Beschreibung
pgactive.ddl_lock_timeout	milliseconds	-1	Gibt an, wie lange ein DDL-Sperrversuch darauf wartet, die Sperre zu erhalten. Der Standardwert ist -1, der den in lock_timeout angegebenen Wert verwendet. Sie können diesen Parameter in Zeiteinheiten wie 10s für 10 Sekunden festlegen. Dieser Timer steuert nur die Wartezeit für den Erhalt einer DDL-Sperre. Sobald das System die Sperre erhält und den DDL-Vorgang startet, stoppt der Timer. Dieser Parameter begrenzt nicht die Gesamtdauer, für die eine DDL-Sperr

Parameter	Einheit	Standard	Beschreibung
			e aufrechte rhalten werden kann, oder die gesamte DDL-Betriebszeit. Um die Gesamtdau er des Vorgangs zu steuern, verwenden Sie `statement _timeout` stattdessen. Weitere Informati onen finden Sie unter DDL Locking.

Parameter	Einheit	Standard	Beschreibung
pgactive.debug_trace_ddl_locks_level	boolean	–	Setzt die standardmäßige Debug-Protokollebene für DDL-Sperrvorgänge in der Erweiterung außer Kraft. pgactive Wenn diese Einstellung konfiguriert ist, werden Meldungen im Zusammenhang mit DDL-Sperren auf der LOG-Debug-Ebene statt auf ihrer Standardebene ausgegeben. Verwenden Sie diesen Parameter, um DDL-Sperraktivitäten zu überwachen, ohne die Ausführlichkeit DEBUG1 - oder DEBUG2 Protokollebenen auf Ihrem gesamten Server zu aktivieren.

Parameter	Einheit	Standard	Beschreibung
			Verfügbare Protokollebenen, in aufsteigender Reihenfolge der Ausführlichkeit: • keine — DDL-Sperrmeldungen werden nur auf DEBUG1 und niedrigeren Serverprotokollebenen angezeigt. • statement — Fügt die LOG-Ausgabe für Versuche zur Erfassung von DDL-Sperren hinzu. • acquire_release — Zeichnet Sperrenerfassungs-, Freigabe- und Deklinationsereignisse sowie Peer-Node-Anwendungen von Remote-

Parameter	Einheit	Standard	Beschreibung
			DDL-Sperren auf. • Peers — Stellt zusätzliche Informationen zu DDL-Sperr verhandlu ngen zwischen Peer-Knoten bereit. • debug — Protokoll iert alle Aktivitäten im Zusammenh ang mit DDL-Sperren auf LOG-Ebene. Weitere Informati onen zu den Überwachu ngsoptionen finden Sie unter Überwachen globaler DDL-Sperren. Note Änderunge n an dieser Einstellu

Parameter	Einheit	Standard	Beschreibung
			ng werden wirksam, wenn Sie die Konfigura tion neu laden. Sie müssen den Server nicht neu starten.

Zusätzliche pgactive-Erweiterungsparameter

Die folgende Tabelle enthält weniger häufig verwendete und interne Konfigurationsoptionen, die für die `pgactive` Erweiterung verfügbar sind.

Parameter	Einheit	Standard	Beschreibung
`pgactive. debug_app ly_delay`	`integer`	–	Legt eine Anwendung sverzögerung (in Millisekunden) für konfiguri erte Verbindun gen fest, deren Eintrag keine explizite Anwendung sverzöger ung enthält.

Parameter	Einheit	Standard	Beschreibung
			`pgactive.pgactive_connections` Diese Verzögerung wird während der Knotenerstellung oder der Verbindungszeit festgelegt, und pgactive wird eine Transaktion auf Peer-Knoten erst wiedergeben, wenn mindestens die angegebene Anzahl von Millisekunden seit dem Festschreiben vergangen ist. Wird hauptsächlich zur Simulation von Netzwerken mit hoher Latenz in Testumgebungen verwendet, um die Entstehung von Konflikten zu vereinfachen. Beispiel: Bei

Parameter	Einheit	Standard	Beschreibung
			einer Verzögerung von 500 ms auf den Knoten A und B haben Sie nach dem Einfügen eines Werts auf Knoten A mindestens 500 ms Zeit, um eine widersprüchliche Einfügung auf Knoten B durchzuführen. **Note** Erfordert ein erneutes Laden des Servers oder einen Neustart der Apply-Worker, um wirksam zu werden.

Parameter	Einheit	Standard	Beschreibung
pgactive.connectability_check_duration	integer	–	Gibt die Dauer (in Sekunden) an, für die ein Datenbank-Worker versucht, während fehlgeschlagener Versuche Verbindungen herzustellen. Der Worker unternimmt einen Verbindungsversuch pro Sekunde, bis er erfolgreich ist oder diesen Timeout-Wert erreicht. Diese Einstellung ist nützlich, wenn die Datenbank-Engine gestartet wird, bevor der Worker bereit ist, Verbindungen herzustellen.

Parameter	Einheit	Standard	Beschreibung
`pgactive.skip_ddl_replication`	`boolean`	on	Steuert, wie DDL-Änderungen in Amazon RDS repliziert oder verarbeitet werden, wenn diese Option aktiviert ist. `pgactive` Wenn diese Option auf gesetzt iston, verarbeitet der Knoten DDL-Änderungen wie ein nicht-pgktiver Knoten. Für die Arbeit mit diesem Parameter gelten die folgenden Anforderungen: • Neue Knoten können keiner pgactive-Gruppe beitreten, wenn ihr `skip_ddl_replication` Wert vom Wert des Upstream-

Parameter	Einheit	Standard	Beschreibung
			Knotens abweicht. • Bestehende Knoten können pgactive-Worker nicht starten, wenn ihr Parameterwert nicht mit dem Upstream-Knoten übereinstimmt. • Alle pgactive-Mitglieder müssen denselben Parameterwert verwenden. Sie können diesen Parameter mit Superuser-Rechten auf zwei Arten ändern: global, lokal (Sitzungsebene). Note Wenn Sie

Parameter	Einheit	Standard	Beschreibung
			diesen Parameter falsch ändern, können Ihre Replikationseinrichtungen beschädigt werden.

Parameter	Einheit	Standard	Beschreibung
pgactive.do_not_replicate	boolean	–	Dieser Parameter ist nur für den internen Gebrauch bestimmt. Wenn Sie diesen Parameter in einer Transaktion festlegen, werden die Änderungen nicht auf andere Knoten in Ihrem DB-Cluster repliziert. **Note** Wenn Sie diesen Parameter falsch ändern, können Ihre Replikationseinrichtungen beschädigt werden.

Parameter	Einheit	Standard	Beschreibung
pgactive.discard_mismatched_row_attributes	boolean	–	Dieser Parameter ist nur für den Gebrauch durch Spezialisten vorgesehen. Es wird empfohlen, diesen Parameter nur zur Behebung bestimmter Replikationsprobleme zu verwenden. Verwenden Sie diesen Parameter in folgenden Fällen: • Der eingehende Replikationsstream enthält Zeilen mit mehr Spalten als Ihre lokale Tabelle. • Diese entfernten Zeilen enthalten Werte, die ungleich Null sind.

Parameter	Einheit	Standard	Beschreibung
			Diese Einstellung überschreibt die folgende Fehlermeldung und ermöglicht Datendivergenzen, sodass die Replikation fortgesetzt werden kann: `cannot right-pad mismatched attributes; attno %u is missing in local table and remote row has non-null, non-dropped value for this attribute` **Note** Wenn Sie diesen Parameter falsch ändern, können Ihre

Parameter	Einheit	Standard	Beschreibung
			Replikationseinrichtungen beschädigt werden.

Parameter	Einheit	Standard	Beschreibung
pgactive.debug_trace_replay	boolean	–	Wenn dieser Wert auf gesetzt iston, wird für jede Remote-Aktion, die nachgeschaltete Apply-Worker verarbeiten, eine Protokoll meldung ausgegeben. Zu den Protokollen gehören: • Änderungstyp • Name der betroffenen Tabelle • Anzahl der Änderungen seit dem Start der Transakti on • Transaktions-Commit LSN • Zeitstempel festschreiben • Kennung des Upstream-Knotens • Kennung des Weiterlei tungsknotens

Parameter	Einheit	Standard	Beschreibung
			(falls zutreffend) In den Protokollen werden auch DDL-Befehle und Tabellenabbrüche in der Warteschlange erfasst. para> Standardmäßig enthalten die Protokolle keine Zeilenfeldinhalte. Um Zeilenwerte in die Protokolle aufzunehmen, müssen Sie bei der Neukompilierung die folgenden Flags aktiviert haben: • VERBOSE_INSERT • VERBOSE_AKTUALISIEREN • VERBOSE_LÖSCHEN

Parameter	Einheit	Standard	Beschreibung
			Note Die Aktivierung dieser Protokollierungseinstellung kann sich auf die Leistung auswirken. Wir empfehlen, sie nur zu aktivieren, wenn sie zur Fehlerbehebung benötigt wird. Änderungen an dieser Einstellung werden wirksam, wenn Sie die Konfiguration neu

Parameter	Einheit	Standard	Beschreibung
			laden. Sie müssen den Server nicht neu starten.

Parameter	Einheit	Standard	Beschreibung
`pgactive.extra_apply_connection_options`		–	Sie können Verbindungsparameter für alle Peer-Node-Verbindungen mit pgactive-Knoten konfigurieren. Diese Parameter steuern Einstellungen wie Keepalives und SSL-Modi. Standardmäßig verwendet pgactive die folgenden Verbindungsparameter: • connect_timeout=30 • keepalives=1 • keepalives_idle=20 • keepalives_interval=20 • keepalives_count=5 Verwenden Sie den folgenden ähnlichen Befehl, um

Parameter	Einheit	Standard	Beschreibung
			die Standardparameter zu überschreiben: `pgactive.extra_apply_connection_options = 'keepalives=0'` Verbindungszeichenfolgen für einzelne Knoten haben Vorrang vor diesen Einstellungen und den integrierten Verbindungsoptionen von pgactive. Weitere Hinweise zu Verbindungszeichenfolgenformaten finden Sie unter libpq-Verbindungszeichenfolgen. Wir empfehlen, die Keepalive-Standard

Parameter	Einheit	Standard	Beschreibung
			einstellungen aktiviert zu lassen. Deaktivieren Sie Keepalives nur, wenn Sie Probleme mit großen Transaktionen haben, die über unzuverlässige Netzwerke abgeschlossen werden. **Note** Wir empfehlen, die Keepalive-Standardeinstellungen aktiviert zu lassen. Deaktivieren Sie Keepalives nur, wenn Sie Probleme mit großen

Parameter	Einheit	Standard	Beschreibung
			Transaktionen haben, die über unzuverlässige Netzwerke abgeschlossen werden. Änderungen an dieser Einstellung werden wirksam, wenn Sie die Konfiguration neu laden. Sie müssen den Server nicht neu starten.

Parameter	Einheit	Standard	Beschreibung
`pgactive.init_node_parallel_jobs (int)`		–	Gibt die Anzahl der parallel Jobs an, die bei logischen Knotenverbindungen mit der `pgactive.pgactive_join_group` Funktion verwendet werden `pg_restore` können `pg_dump` und können. Änderungen an dieser Einstellung werden wirksam, wenn Sie die Konfiguration neu laden. Sie müssen den Server nicht neu starten.

Parameter	Einheit	Standard	Beschreibung
`pgactive.max_nodes`	`int`	4	Gibt die maximale Anzahl von Knoten an, die in einer pgactive-Erweiterungsgruppe zulässig sind. Der Standardwert ist 4 Knoten. Sie müssen Folgendes berücksichtigen, wenn Sie den Wert dieses Parameters festlegen: • Alle Knoten in einer pgactive-Erweiterungsgruppe müssen denselben Parameterwert verwenden. • Ein neuer Knoten kann nicht beitreten, wenn sein Parameterwert vom Upstream-Knoten abweicht.

Parameter	Einheit	Standard	Beschreibung
			• Bestehende Knoten können pgactive Extension Worker nicht starten, wenn sich ihr Parameterwert vom Upstream-Knoten unterscheidet. • Größere Gruppen erfordern zusätzlichen Überwachungs- und Wartungsaufwand, daher sollte der Wert dieses Parameters mit Bedacht gewählt werden. Sie können diesen Parameter auf zwei Arten festlegen: in der Konfigura

Parameter	Einheit	Standard	Beschreibung
			tionsdatei mit dem `ALTER SYSTEM SET` folgenden Befehl Der Standardwert für diesen Parameter ist4, was bedeutet, dass zu jedem Zeitpunkt maximal 4 Knoten in der `pgactive` Erweiterungsgruppe zulässig sind. **Note** Die Änderung wird wirksam, nachdem Sie den Server neu gestartet haben.

Parameter	Einheit	Standard	Beschreibung
`pgactive.permit_node_identifier_getter_function_creation`	`boolean`	–	Dieser Parameter ist nur für den internen Gebrauch vorgesehen. Wenn sie aktiviert ist, ermöglicht die `pgactive` Erweiterung die Erstellung der Getter-Funktion pgactive Node Identifier.

Aktiv-aktive Konflikte verstehen

Wenn Sie pgactive im Active-Active-Modus verwenden, kann das Schreiben in dieselben Tabellen von mehreren Knoten aus zu Datenkonflikten führen. Während einige Clustersysteme verteilte Sperren verwenden, um gleichzeitigen Zugriff zu verhindern, verfolgt pgactive einen optimistischen Ansatz, der sich besser für geografisch verteilte Anwendungen eignet.

Einige Datenbank-Clustersysteme verhindern den gleichzeitigen Datenzugriff, indem sie verteilte Sperren verwenden. Dieser Ansatz funktioniert zwar, wenn sich Server in unmittelbarer Nähe befinden, unterstützt jedoch keine geografisch verteilten Anwendungen, da er für eine gute Leistung eine extrem niedrige Latenz erfordert. Anstatt verteilte Sperren zu verwenden (ein pessimistischer Ansatz), verwendet die Erweiterung pgactive einen optimistischen Ansatz. Das heißt:

- Hilft Ihnen, Konflikte nach Möglichkeit zu vermeiden.
- Ermöglicht das Auftreten bestimmter Arten von Konflikten.
- Ermöglicht die Konfliktlösung, wenn Konflikte auftreten.

Dieser Ansatz bietet Ihnen mehr Flexibilität beim Erstellen verteilter Anwendungen.

Wie Konflikte entstehen

Konflikte zwischen Knoten entstehen durch Abfolgen von Ereignissen, die nicht eintreten könnten, wenn alle beteiligten Transaktionen gleichzeitig auf demselben Knoten stattfänden. Da die Knoten erst nach dem Commit der Transaktion Änderungen austauschen, ist jede Transaktion auf dem Knoten, auf dem sie festgeschrieben wurde, einzeln gültig, wäre aber nicht gültig, wenn sie auf einem anderen Knoten ausgeführt würde, der in der Zwischenzeit andere Aufgaben ausgeführt hat. Da pgactive apply die Transaktion im Wesentlichen auf den anderen Knoten wiedergibt, kann der Wiedergabevorgang fehlschlagen, wenn ein Konflikt zwischen einer angewendeten Transaktion und einer Transaktion besteht, die auf dem empfangenden Knoten festgeschrieben wurde.

Der Grund, warum die meisten Konflikte nicht auftreten können, wenn alle Transaktionen auf einem einzigen Knoten ausgeführt werden, ist, dass PostgreSQL über Kommunikationsmechanismen zwischen Transaktionen verfügt, um dies zu verhindern, darunter:

- EINDEUTIGE Indizes
- SEQUENCEs
- Sperren von Zeilen und Beziehungen
- SERIALISIERBARE ABHÄNGIGKEITSVERFOLGUNG

All diese Mechanismen dienen der Kommunikation zwischen Transaktionen, um unerwünschte Parallelitätsprobleme zu vermeiden

pgactive erreicht eine geringe Latenz und verarbeitet Netzwerkpartitionen gut, da es keinen verteilten Transaktionsmanager oder Sperrmanager verwendet. Dies bedeutet jedoch, dass Transaktionen auf verschiedenen Knoten völlig isoliert voneinander ausgeführt werden. Während die Isolierung in der Regel die Datenbankkonsistenz verbessert, müssen Sie in diesem Fall die Isolation reduzieren, um Konflikte zu vermeiden.

Arten von Konflikten

Zu den Konflikten, die auftreten können, gehören:

Themen

- Konflikte mit PRIMARY KEY oder UNIQUE
- INSERT/INSERT-Konflikte
- INSERTsdie gegen mehrere UNIQUE-Einschränkungen verstoßen

- UPDATE/UPDATE-Konflikte
- UPDATE-Konflikte beim PRIMARY KEY
- UPDATEsdie gegen mehrere UNIQUE-Einschränkungen verstoßen
- Konflikte beim Aktualisieren/Löschen
- INSERT/UPDATE-Konflikte
- Konflikte löschen/löschen
- Konflikte mit Foreign Key Constraint
- Konflikte zwischen Ausschlussbeschränkungen
- Globale Datenkonflikte
- Sperrkonflikte und Deadlock-Abbrüche
- Divergierende Konflikte

Konflikte mit PRIMARY KEY oder UNIQUE

Zeilenkonflikte treten auf, wenn mehrere Operationen versuchen, denselben Zeilenschlüssel auf eine Weise zu ändern, die auf einem einzelnen Knoten nicht möglich ist. Diese Konflikte stellen die häufigste Art von Datenkonflikten dar.

pgactive löst erkannte Konflikte durch die last-update-wins Behandlung oder Ihren benutzerdefinierten Konflikthandler.

Zu den Zeilenkonflikten gehören:

- INSERT gegen INSERT
- INSERT gegen UPDATE
- AKTUALISIEREN gegen LÖSCHEN
- EINFÜGEN gegen LÖSCHEN
- LÖSCHEN gegen LÖSCHEN
- EINFÜGEN gegen LÖSCHEN

INSERT/INSERT-Konflikte

Dieser häufigste Konflikt tritt auf, wenn INSERTs auf zwei verschiedenen Knoten ein Tupel mit denselben PRIMARY KEY-Werten (oder identischen UNIQUE-Einschränkungswerten, wenn kein PRIMARY KEY vorhanden ist) erstellt wird.

pgactivelink löst INSERT-Konflikte, indem es den Zeitstempel des ursprünglichen Hosts verwendet, um das neueste Tupel beizubehalten. Sie können dieses Standardverhalten mit Ihrem benutzerdefinierten Konflikthandler außer Kraft setzen. Dieser Prozess erfordert zwar keine spezielle Administratoraktion, aber beachten Sie, dass pgactivelink eine der INSERT-Operationen auf allen Knoten verwirft. Es findet keine automatische Datenzusammenführung statt, es sei denn, Ihr benutzerdefinierter Handler implementiert sie.

Der pgactivelink kann nur Konflikte lösen, bei denen es sich um eine einzige Einschränkungsverletzung handelt. Wenn ein INSERT gegen mehrere UNIQUE-Einschränkungen verstößt, müssen Sie zusätzliche Strategien zur Konfliktlösung implementieren.

INSERTsdie gegen mehrere UNIQUE-Einschränkungen verstoßen

Ein INSERT/INSERT-Konflikt kann mehrere UNIQUE-Einschränkungen verletzen, einschließlich des PRIMARY KEY. pgactivelink kann nur Konflikte behandeln, die eine einzige UNIQUE-Einschränkung beinhalten. Wenn Konflikte mehrere UNIQUE-Einschränkungen verletzen, schlägt der Apply-Worker fehl und gibt den folgenden Fehler zurück:

```
multiple unique constraints violated by remotely INSERTed tuple.
```

In älteren Versionen führte diese Situation stattdessen zu dem Fehler „Konflikt um unterschiedliche Eindeutigkeit“.

Um diese Konflikte zu lösen, müssen Sie manuelle Maßnahmen ergreifen. LÖSCHEN Sie entweder die widersprüchlichen lokalen Tupel oder AKTUALISIEREN Sie sie, um Konflikte mit dem neuen Remote-Tupel zu beseitigen. Beachten Sie, dass Sie sich möglicherweise mit mehreren in Konflikt stehenden Tupeln befassen müssen. Derzeit bietet pgactivelink keine integrierte Funktionalität zum Ignorieren, Verwerfen oder Zusammenführen von Tupeln, die gegen mehrere eindeutige Einschränkungen verstoßen.

> **Note**
>
> Weitere Informationen finden Sie unter UPDATEs Diese verstoßen gegen mehrere UNIQUE-Einschränkungen.

UPDATE/UPDATE-Konflikte

Dieser Konflikt tritt auf, wenn zwei Knoten gleichzeitig dasselbe Tupel ändern, ohne seinen PRIMÄRSCHLÜSSEL zu ändern. pgactivelink löst diese Konflikte mithilfe von last-update-wins

Logik oder Ihrem benutzerdefinierten Konflikthandler, falls definiert. Ein PRIMÄRSCHLÜSSEL ist für den Tupelabgleich und die Konfliktlösung unerlässlich. Für Tabellen ohne PRIMARY KEY lehnt pgactivelink UPDATE-Operationen mit dem folgenden Fehler ab:

```
Cannot run UPDATE or DELETE on table (tablename) because it does not have a primary key.
```

UPDATE-Konflikte beim PRIMARY KEY

pgactive hat Einschränkungen beim Umgang mit PRIMARY KEY-Updates. Sie können zwar eine UPDATE-Operation für einen PRIMÄRSCHLÜSSEL ausführen, aber pgactive kann Konflikte nicht automatisch mithilfe der last-update-wins Logik für diese Operationen lösen. Sie müssen sicherstellen, dass Ihre PRIMARY KEY-Aktualisierungen nicht mit vorhandenen Werten in Konflikt geraten. Wenn bei PRIMARY KEY-Aktualisierungen Konflikte auftreten, werden diese zu divergierenden Konflikten, die Ihr manuelles Eingreifen erfordern. Weitere Informationen zum Umgang mit diesen Situationen finden Sie unter. Divergierende Konflikte

UPDATEsdie gegen mehrere UNIQUE-Einschränkungen verstoßen

pgactivelink kann keine last-update-wins Konfliktlösung anwenden, wenn ein eingehendes UPDATE gegen mehrere UNIQUE-Einschränkungen oder PRIMARY KEY-Werte verstößt. Dieses Verhalten ähnelt INSERT-Vorgängen mit mehreren Einschränkungsverletzungen. Diese Situationen führen zu unterschiedlichen Konflikten, die Ihr manuelles Eingreifen erfordern. Weitere Informationen finden Sie unter Divergierende Konflikte.

Konflikte beim Aktualisieren/Löschen

Diese Konflikte treten auf, wenn UPDATEs ein Knoten eine Zeile und ein anderer Knoten gleichzeitig DELETEs vorhanden ist. In diesem Fall tritt bei der Wiedergabe ein UPDATE/DELETE-Konflikt auf. Die Lösung besteht darin, alle AKTUALISIERUNGEN, die nach einem DELETE-Befehl eintreffen, zu verwerfen, sofern Ihr benutzerdefinierter Konflikthandler nichts anderes festlegt.

pgactivelink benötigt einen PRIMÄRSCHLÜSSEL, um Tupel abzugleichen und Konflikte zu lösen. Für Tabellen ohne PRIMARY KEY lehnt es DELETE-Operationen mit dem folgenden Fehler ab:

```
Cannot run UPDATE or DELETE on table (tablename) because it does not have a primary key.
```

> **Note**
>
> pgactivelink kann nicht zwischen Konflikten unterscheiden. UPDATE/DELETE and INSERT/UPDATE In beiden Fällen wirkt sich ein UPDATE auf eine Zeile aus, die nicht existiert. Aufgrund der asynchronen Replikation und der fehlenden Reihenfolge der Wiedergabe zwischen den Knoten kann pgactivelink nicht feststellen, ob das UPDATE für eine neue Zeile (INSERT noch nicht empfangen) oder für eine gelöschte Zeile gilt. In beiden Szenarien verwirft pgactivink das UPDATE.

INSERT/UPDATE-Konflikte

Dieser Konflikt kann in Umgebungen mit mehreren Knoten auftreten. Er tritt auf, wenn INSERTs ein Knoten eine Zeile, ein zweiter Knoten UPDATEs sie und ein dritter Knoten das UPDATE vor dem ursprünglichen INSERT empfängt. Standardmäßig löst pgactivelink diese Konflikte, indem es das UPDATE verwirft, sofern Ihr benutzerdefinierter Konflikt-Trigger nichts anderes angibt. Beachten Sie, dass diese Lösungsmethode zu Dateninkonsistenzen zwischen den Knoten führen kann. Weitere Informationen zu ähnlichen Szenarien und deren Handhabung finden Sie unter. Konflikte beim Aktualisieren/Löschen

Konflikte löschen/löschen

Dieser Konflikt tritt auf, wenn zwei verschiedene Knoten gleichzeitig dasselbe Tupel löschen. pgactivelink betrachtet diese Konflikte als harmlos, da beide DELETE-Operationen dasselbe Endergebnis haben. In diesem Szenario ignoriert pgactivink sicher eine der DELETE-Operationen, ohne die Datenkonsistenz zu beeinträchtigen.

Konflikte mit Foreign Key Constraint

FOREIGN KEY-Einschränkungen können zu Konflikten führen, wenn Remotetransaktionen auf vorhandene lokale Daten angewendet werden. Diese Konflikte treten typischerweise auf, wenn Transaktionen in einer anderen Reihenfolge als in ihrer logischen Reihenfolge auf den Ausgangsknoten angewendet werden.

Standardmäßig wendet pgactive Änderungen mit session_replication_role as an, wodurch Fremdschlüsselprüfungen während der Replikation `replica` umgangen werden. In Active-Active-Konfigurationen kann dies zu Verletzungen von Fremdschlüsseln führen. Die meisten Verstöße sind vorübergehend und werden behoben, sobald die Replikation abgeschlossen ist. Allerdings kann es

zu fehlerhaften Fremdschlüsseln kommen, weil pgactive das knotenübergreifende Sperren von Zeilen nicht unterstützt.

Dieses Verhalten ist partitionstoleranten asynchronen aktiv-aktiven Systemen inhärent. Beispielsweise könnte Knoten A eine neue untergeordnete Zeile einfügen, während Knoten B gleichzeitig seine übergeordnete Zeile löscht. Das System kann diese Art der gleichzeitigen Änderung knotenübergreifend nicht verhindern.

Um Fremdschlüsselkonflikte zu minimieren, empfehlen wir Folgendes:

- Beschränken Sie Fremdschlüsselbeziehungen auf eng verwandte Entitäten.
- Ändern Sie verwandte Entitäten, wenn möglich, von einem einzigen Knoten aus.
- Wählen Sie Entitäten aus, die selten geändert werden müssen.
- Implementieren Sie die Parallelitätssteuerung auf Anwendungsebene für Änderungen.

Konflikte zwischen Ausschlussbeschränkungen

pgactive link unterstützt keine Ausschlussbeschränkungen und schränkt deren Erstellung ein.

> **Note**
>
> Wenn Sie eine bestehende eigenständige Datenbank in eine pgactivink-Datenbank konvertieren, löschen Sie manuell alle Ausschlussbeschränkungen.

In einem verteilten asynchronen System kann nicht garantiert werden, dass kein Satz von Zeilen gegen die Einschränkung verstößt. Dies liegt daran, dass alle Transaktionen auf verschiedenen Knoten vollständig isoliert sind. Ausschlussbeschränkungen können zu Deadlocks bei der Wiedergabe führen, sodass die Wiedergabe aufgrund von Verstößen gegen Ausschlussbeschränkungen nicht von einem Knoten zum anderen fortgesetzt werden kann.

Wenn Sie pgactive Link zwingen, eine Ausschlussbeschränkung zu erstellen, oder wenn Sie bei der Konvertierung einer eigenständigen Datenbank in pgactive Link bestehende Einschränkungen nicht löschen, ist es wahrscheinlich, dass die Replikation unterbrochen wird. Um den Replikationsfortschritt wiederherzustellen, entfernen oder ändern Sie die lokalen Tupel, die mit einem eingehenden Remote-Tupel in Konflikt stehen, sodass die Remotetransaktion angewendet werden kann.

Globale Datenkonflikte

Bei der Verwendung von pgactivelink können Konflikte auftreten, wenn Knoten unterschiedliche globale systemweite PostgreSQL-Daten haben, z. B. Rollen. Diese Konflikte können dazu führen, dass Operationen — in erster Linie DDL — auf einem Knoten erfolgreich ausgeführt und festgeschrieben werden, aber nicht auf andere Knoten angewendet werden.

Wenn ein Benutzer auf einem Knoten existiert, auf einem anderen aber nicht, können Replikationsprobleme auftreten:

- Node1 hat einen Benutzer namens`fred`, aber dieser Benutzer existiert nicht auf Node2
- Wenn eine Tabelle auf Node1 `fred` erstellt wird, wird die Tabelle mit dem Eigentümer repliziert `fred`
- Wenn dieser DDL-Befehl auf Node2 angewendet wird, schlägt er fehl, weil der Benutzer nicht existiert `fred`
- Dieser Fehler generiert einen FEHLER in den PostgreSQL-Protokollen auf Node2 und erhöht den Zähler. `pgactive.pgactive_stats.nr_rollbacks`

Lösung: Erstellen Sie den Benutzer auf Node2. `fred` Der Benutzer benötigt keine identischen Berechtigungen, sondern muss auf beiden Knoten vorhanden sein.

Wenn eine Tabelle auf einem Knoten existiert, auf einem anderen aber nicht, schlagen Datenänderungsvorgänge fehl:

- Node1 hat eine Tabelle mit dem Namen`foo`, die auf Node2 nicht existiert
- Alle DML-Operationen in der `foo` Tabelle auf Node1 schlagen fehl, wenn sie auf Node2 repliziert werden

Lösung: Erstellen Sie die Tabelle `foo` auf Node2 mit derselben Struktur.

> **Note**
>
> pgactivelink repliziert derzeit keine CREATE USER-Befehle oder DDL-Operationen. Die DDL-Replikation ist für eine future Version geplant.

Sperrkonflikte und Deadlock-Abbrüche

Da pgactive-Apply-Prozesse wie normale Benutzersitzungen funktionieren, folgen sie den Standardregeln für das Sperren von Zeilen und Tabellen. Dies kann dazu führen, dass pgactivink-Apply-Prozesse auf Sperren warten, die von Benutzertransaktionen oder anderen Apply-Prozessen gehalten werden.

Die folgenden Arten von Sperren können sich auf Apply-Prozesse auswirken:

- Explizites Sperren auf Tabellenebene (LOCK TABLE...) durch Benutzersitzungen
- Explizites Sperren auf Zeilenebene (SELECT... FOR UPDATE/FOR SHARE) nach Benutzersitzungen
- Sperren mit Fremdschlüsseln
- Implizites Sperren aufgrund von Zeilen UPDATEs INSERTs, oder DELETEs, entweder aufgrund lokaler Aktivitäten oder aufgrund von Angriffen von anderen Servern

Deadlocks können auftreten zwischen:

- Ein pgactivelink-Apply-Prozess und eine Benutzertransaktion
- Zwei Apply-Prozesse

Wenn Deadlocks auftreten, beendet der Deadlock-Detektor von PostgreSQL eine der problematischen Transaktionen. Wenn der Prozess des Apply-Workers pgactivelink beendet wird, versucht er es automatisch erneut, was in der Regel erfolgreich ist.

> **Note**
>
> - Diese Probleme sind vorübergehend und erfordern im Allgemeinen kein Eingreifen des Administrators. Wenn ein Apply-Prozess durch eine Sperre einer Benutzersitzung im Leerlauf für einen längeren Zeitraum blockiert wird, können Sie die Benutzersitzung beenden, um die Replikation fortzusetzen. Diese Situation ähnelt der Situation, wenn ein Benutzer eine lange Sperre hält, die sich auf eine andere Benutzersitzung auswirkt.
> - Um Verzögerungen bei der Wiedergabe im Zusammenhang mit Sperren zu identifizieren, aktivieren Sie die `log_lock_waits` Funktion in PostgreSQL.

Divergierende Konflikte

Divergierende Konflikte treten auf, wenn Daten, die knotenübergreifend identisch sein sollten, unerwartet voneinander abweichen. Diese Konflikte sollten zwar nicht auftreten, aber mit der aktuellen Implementierung können nicht alle zuverlässig verhindert werden.

> **Note**
>
> Das Ändern des PRIMÄRSCHLÜSSELS einer Zeile kann zu unterschiedlichen Konflikten führen, wenn ein anderer Knoten den Schlüssel derselben Zeile ändert, bevor alle Knoten die Änderung verarbeiten. Vermeiden Sie es, Primärschlüssel zu ändern, oder beschränken Sie Änderungen auf einen bestimmten Knoten. Weitere Informationen finden Sie unter the section called “UPDATE-Konflikte beim PRIMARY KEY ”.

Divergierende Konflikte, die Zeilendaten betreffen, erfordern in der Regel ein Eingreifen des Administrators. Um diese Konflikte zu lösen, müssen Sie die Daten auf einem Knoten manuell so anpassen, dass sie einem anderen entsprechen, und gleichzeitig die Replikation mithilfe von vorübergehend deaktivieren. `pgactive.pgactive_do_not_replicate` Diese Konflikte sollten nicht auftreten, wenn Sie pgactive wie dokumentiert verwenden und Einstellungen oder Funktionen vermeiden, die als unsicher markiert sind.

Als Administrator müssen Sie diese Konflikte manuell lösen. Je nach Konflikttyp müssen Sie erweiterte Optionen wie verwenden`pgactive.pgactive_do_not_replicate`. Verwenden Sie diese Optionen mit Vorsicht, da eine unsachgemäße Verwendung die Situation verschlimmern kann. Aufgrund der Vielzahl möglicher Konflikte können wir keine allgemeingültigen Lösungsansätze bereitstellen.

Divergierende Konflikte treten auf, wenn sich Daten, die auf verschiedenen Knoten identisch sein sollten, unerwartet unterscheiden. Diese Konflikte sollten zwar nicht auftreten, aber nicht alle dieser Konflikte können in der aktuellen Implementierung zuverlässig verhindert werden.

Konflikte vermeiden oder tolerieren

In den meisten Fällen können Sie ein geeignetes Anwendungsdesign verwenden, um Konflikte zu vermeiden oder Ihre Anwendung konflikttolerant zu machen.

Konflikte treten nur auf, wenn gleichzeitige Operationen auf mehreren Knoten ausgeführt werden. Um Konflikte zu vermeiden:

- Schreiben Sie nur auf einen Knoten
- Schreiben Sie in unabhängige Datenbankuntergruppen auf jedem Knoten (weisen Sie beispielsweise jedem Knoten ein eigenes Schema zu)

Verwenden Sie bei INSERT- und INSERT-Konflikten globale Sequenzen, um Konflikte vollständig zu vermeiden.

Wenn Konflikte für Ihren Anwendungsfall nicht akzeptabel sind, sollten Sie die Implementierung verteilter Sperren auf Anwendungsebene in Betracht ziehen. Oft ist es am besten, Ihre Anwendung so zu gestalten, dass sie mit den Konfliktlösungsmechanismen von pgactive funktioniert, anstatt zu versuchen, alle Konflikte zu verhindern. Weitere Informationen finden Sie unter the section called “Arten von Konflikten”.

Protokollierung von Konflikten

pgactivelink protokolliert Konfliktereignisse in der `pgactive.pgactive_conflict_history` Tabelle, um Ihnen bei der Diagnose und Behandlung von Active-Active-Konflikten zu helfen. Die Konfliktprotokollierung in dieser Tabelle erfolgt nur, wenn Sie den Wert auf true setzen. `pgactive.log_conflicts_to_table` Die Erweiterung pgactive protokolliert auch Konflikte in der PostgreSQL-Protokolldatei, wenn log_min_messages auf oder gesetzt ist, unabhängig von der Einstellung. `LOG` `lower` `pgactive.log_conflicts_to_table`

Verwenden Sie die Tabelle mit dem Konfliktverlauf, um:

- Messen Sie, wie häufig Ihre Anwendung Konflikte verursacht
- Identifizieren Sie, wo Konflikte auftreten
- Verbessern Sie Ihre Anwendung, um die Konfliktrate zu reduzieren
- Erkennen Sie Fälle, in denen Konfliktlösungen nicht zu den gewünschten Ergebnissen führen
- Ermitteln Sie, wo Sie benutzerdefinierte Konfliktauslöser oder Änderungen am Anwendungsdesign benötigen

Bei Zeilenkonflikten können Sie optional Zeilenwerte protokollieren. Dies wird durch die `pgactive.log_conflicts_to_table` Einstellung gesteuert. Beachten Sie:

- Dies ist eine globale datenbankweite Option
- Es gibt keine Kontrolle über die Protokollierung von Zeilenwerten pro Tabelle

- Für Feldnummern, Array-Elemente oder Feldlängen gelten keine Beschränkungen
- Die Aktivierung dieser Funktion ist möglicherweise nicht ratsam, wenn Sie mit Zeilen mit mehreren Megabyte arbeiten, die Konflikte auslösen könnten

Da die Konfliktverlaufstabelle Daten aus jeder Tabelle in der Datenbank enthält (jede mit potenziell unterschiedlichen Schemas), werden protokollierte Zeilenwerte als JSON-Felder gespeichert. Die JSON-Datei wird mithilfe von SQL erstellt`row_to_json`, ähnlich wie wenn sie direkt aus SQL aufgerufen wird. PostgreSQL bietet keine `json_to_row` Funktion, daher benötigen Sie tabellenspezifischen Code (in usw.)PL/pgSQL, PL/Python, PL/Perl, um ein zusammengesetztes Tupel aus dem protokollierten JSON zu rekonstruieren.

> **Note**
>
> Die Support von benutzerdefinierten Konflikten ist als future Erweiterungsfunktion geplant.

Das pgactive-Schema verstehen

Das pgactive-Schema verwaltet die Active-Active-Replikation in RDS für PostgreSQL. Dieses Schema enthält Tabellen, in denen Informationen zur Replikationskonfiguration und zum Status gespeichert werden.

> **Note**
>
> Das pgactive-Schema entwickelt sich weiter und kann sich ändern. Ändern Sie die Daten in diesen Tabellen nicht direkt.

Zu den wichtigsten Tabellen im pgactive-Schema gehören:

- `pgactive_nodes`— Speichert Informationen über Knoten in der aktiv-aktiven Replikationsgruppe.
- `pgactive_connections`— Speichert Verbindungsdetails für jeden Knoten.

pgactive_nodes

Die Datei pgactive_nodes speichert Informationen über die Knoten, die an der aktiv-aktiven Replikationsgruppe teilnehmen.

Spalte	Typ	Kollation	NULL-fähig	Standard
node_sysid	text	–	nicht null	–
node_timeline	OID	–	nicht null	–
node_dboid	OID	–	nicht null	–
node_status	char	–	nicht null	–
Knotenname	text	–	nicht null	–
node_dsn	text	–	nicht null	–
node_init _from_dsn	text	–	nicht null	–
node_read _only	boolesch	–	–	false
node_seq_id	smallint	–	nicht null	–

node_sysid

Eindeutige ID für einen Knoten, generiert während oder `pgactive_create_group` `pgactive_join_group`

node_status

Bereitschaft des Knotens:

- b - Beginn der Einrichtung
- i - initialisieren
- c - Aufholen
- o — Outbound-Slots erstellen
- r - bereit
- k - getötet

> **Note**
>
> Diese Spalte gibt nicht an, ob ein Knoten verbunden oder getrennt ist.

Knotenname

Vom Benutzer bereitgestellter eindeutiger Knotenname.

node_dsn

Verbindungszeichenfolge oder Name der Benutzerzuordnung

node_init_from_dsn

DSN, aus dem dieser Knoten erstellt wurde.

pgactive_connection

Die pgactive_connections speichert Verbindungsdetails für jeden Knoten.

Spalte	Typ	Kollation	NULL-fähig	Standard
conn_sysid	text	Keine	nicht null	Keine
conn_timeline	OID	Keine	nicht null	Keine
conn_dboid	OID	Keine	nicht null	Keine
conn_dsn	text	Keine	nicht null	Keine
conn_apply_delay	Ganzzahl	Keine	Keine	Keine
conn_replication_sets	text	Keine	Keine	Keine

conn_sysid

Knotenbezeichner für den Knoten, auf den sich dieser Eintrag bezieht.

conn_dsn

Wie pgactive.pgactive_nodes. `node_dsn`

conn_apply_delay

Falls gesetzt, warten Sie Millisekunden, bevor jede Transaktion vom Remote-Knoten angewendet wird. Hauptsächlich zum Debuggen. Bei Null gilt der globale Standard.

Mit Replikationssätzen arbeiten

Replikationssätze bestimmen, welche Tabellen in Replikationsvorgänge aufgenommen oder von ihnen ausgeschlossen werden sollen. Standardmäßig werden alle Tabellen repliziert, sofern Sie mit den folgenden Funktionen nichts anderes angeben:

- `pgactive_exclude_table_replication_set()`- Schließt angegebene Tabellen von der Replikation aus
- `pgactive_include_table_replication_set()`- Schließt bestimmte Tabellen in die Replikation ein

> **Note**
>
> Bevor Sie Replikationssätze konfigurieren, sollten Sie Folgendes berücksichtigen:
>
> - Sie können das Ein- oder Ausschließen von Tabellen erst nach der Ausführung`pgactive_create_group()`, aber vorher konfigurieren`pgactive_join_group()`.
> - Nachdem Sie es verwendet haben`pgactive_exclude_table_replication_set()`, können Sie es nicht mehr verwenden`pgactive_include_table_replication_set()`.
> - Nachdem Sie es verwendet haben`pgactive_include_table_replication_set()`, können Sie es nicht mehr verwenden`pgactive_exclude_table_replication_set()`.

Das System behandelt neu erstellte Tabellen je nach Ihrer ursprünglichen Konfiguration unterschiedlich:

- Wenn Sie Tabellen ausgeschlossen haben: Alle neuen Tabellen, die danach erstellt wurden, `pgactive_join_group()` werden automatisch in die Replikation einbezogen
- Wenn Sie Tabellen eingeschlossen haben: Alle neuen Tabellen, die danach erstellt wurden, `pgactive_join_group()` werden automatisch von der Replikation ausgeschlossen.

Verwenden Sie die `pgactive.pgactive_get_table_replication_sets()` Funktion, um die Replikationssatz-Konfiguration für eine bestimmte Tabelle anzuzeigen.

pgactive-Funktionsreferenz

Im Folgenden finden Sie eine Liste der pgactive-Funktionen mit ihren Parametern, Rückgabewerten und praktischen Anwendungshinweisen, die Ihnen helfen, sie effektiv zu nutzen:

get_last_applied_xact_info

Ruft die zuletzt angewendeten Transaktionsinformationen für einen angegebenen Knoten ab.

Argumente

- sysid (Text) — Zeitleisten-OID
- dboid (OID)

Rückgabetyp

Es zeichnet Folgendes auf:

- last_applied_xact_id (OID)
- last_applied_xact_committs (Zeitstempel mit Zeitzone)
- last_applied_xact_at (Zeitstempel mit Zeitzone)

Nutzungshinweise

Verwenden Sie diese Funktion, um die zuletzt angewendeten Transaktionsinformationen für einen bestimmten Knoten abzurufen.

pgactive_apply_pause

Unterbricht den Vorgang zum Anwenden der Replikation.

Argumente

Keine

Rückgabetyp

boolesch

Nutzungshinweise

Rufen Sie diese Funktion auf, um den Vorgang zum Anwenden der Replikation anzuhalten.

pgactive_apply_resume

Setzt den Vorgang zum Anwenden der Replikation fort.

Argumente

Keine

Rückgabetyp

void

Nutzungshinweise

Rufen Sie diese Funktion auf, um den Vorgang zum Anwenden der Replikation fortzusetzen.

pgactive_is_apply_paused

Prüft, ob die Replikation, die Anwendung übernimmt, derzeit angehalten ist.

Argumente

Keine

Rückgabetyp

boolesch

Nutzungshinweise

Verwenden Sie diese Funktion, um zu überprüfen, ob das Anwenden der Replikation derzeit angehalten ist.

pgactive_create_group

Erzeugt eine pgactive-Gruppe, indem eine eigenständige Datenbank in den ursprünglichen Knoten konvertiert wird.

Argumente

- Knotenname (Text)
- node_dsn (Text)
- apply_delay Ganzzahl STANDARD NULL: :Ganzzahl — replication_sets text [] STANDARDARRAY ['default': :text]

Typ der Rückgabe

void

Nutzungshinweise

Erzeugt eine pgactive-Gruppe, indem eine eigenständige Datenbank in den ursprünglichen Knoten konvertiert wird. Die Funktion führt Plausibilitätsprüfungen durch, bevor der Knoten in einen pgactive-Knoten umgewandelt wird. Bevor Sie diese Funktion verwenden, stellen Sie sicher, dass Ihr PostgreSQL-Cluster ausreichend `max_worker_processes` zur Unterstützung von pgactive-Hintergrundworkern verfügbar ist.

pgactive_detach_nodes

Entfernt die angegebenen Knoten aus der Gruppe pgactive.

Argumente

- p_nodes (Text [])

Rückgabetyp

void

Nutzungshinweise

Verwenden Sie diese Funktion, um angegebene Knoten aus der pgactive-Gruppe zu entfernen.

pgactive_exclude_table_replication_set

Schließt eine bestimmte Tabelle von der Replikation aus.

Argumente

- p_relation (Regclass)

Rückgabetyp

void

Nutzungshinweise

Verwenden Sie diese Funktion, um eine bestimmte Tabelle von der Replikation auszuschließen.

pgactive_get_replication_lag_info

Ruft detaillierte Informationen zur Replikationsverzögerung ab, einschließlich Knotendetails, WAL-Status und LSN-Werten.

Argumente

Keine

Rückgabetyp

SETOF record - node_name text - node_sysid text - application_name text - slot_name text - active boolean - active_pid integer - pending_wal_decoding bigint - Ungefähre Größe der WAL in Byte, die auf dem Absenderknoten dekodiert werden soll - pending_wal_to_apply bigint - Ungefähre Größe der WAL in Byte, die auf den Empfangsknoten angewendet werden soll - restart_lsn pg_lsn - confirm med_flush_lsn pg_lsn - sent_lsn pg_lsn - write_lsn pg_lsn - flush_lsn pg_lsn - replay_lsn pg_lsn

Nutzungshinweise

Rufen Sie diese Funktion auf, um Informationen zur Replikationsverzögerung abzurufen, einschließlich Knotendetails, WAL-Status und LSN-Werten.

pgactive_get_stats

Ruft pgactive-Replikationsstatistiken ab.

Argumente

Keine

Rückgabetyp

SETOF-Datensatz - rep_node_id - rilocalid oid - riremoteid-Text - nr_commit bigint - nr_rollback bigint - nr_insert bigint - nr_insert_conflict bigint - nr_update bigint - nr_update bigint - nr_update_conflict bigint - nr_delete bigint - nr_delete_conflict bigint - nr_disconnect bigint

Nutzungshinweise

Verwenden Sie diese Funktion, um pgactive-Replikationsstatistiken abzurufen.

pgactive_get_table_replication_sets

Ruft die Replikationssatzkonfiguration für eine bestimmte Beziehung ab.

Argumente

- Beziehung (Regclass)

Rückgabetyp

SETOF-Datensatz

Nutzungshinweise

Rufen Sie diese Funktion auf, um die Replikationssatzkonfiguration für eine bestimmte Beziehung abzurufen.

pgactive_include_table_replication_set

Schließt eine bestimmte Tabelle in die Replikation ein.

Argumente

- p_relation (Regclass)

Rückgabetyp

void

Nutzungshinweise

Verwenden Sie diese Funktion, um eine bestimmte Tabelle in die Replikation einzubeziehen.

pgactive_join_group

Fügt einer vorhandenen pgactive-Gruppe einen Knoten hinzu.

Argumente

- Knotenname (Text)
- node_dsn (Text)

- join_using_dsn (Text)
- apply_delay (Ganzzahl, optional)
- replication_sets (Text [], Standard: ['Standard'])
- bypass_collation_check (boolean, Standard: false)
- bypass_node_identifier_creation (boolean, Standard: false)
- bypass_user_tables_check (boolean, Standard: false)

Typ der Rückgabe

void

Nutzungshinweise

Rufen Sie diese Funktion auf, um einer vorhandenen pgactive-Gruppe einen Knoten hinzuzufügen. Stellen Sie sicher, dass Ihr PostgreSQL-Cluster über ausreichend max_worker_processes für pgactive-Hintergrundworker verfügt.

pgactive_remove

Entfernt alle pgactive-Komponenten vom lokalen Knoten.

Argumente

- force (boolean, Standard: false)

Rückgabetyp

void

Nutzungshinweise

Rufen Sie diese Funktion auf, um alle pgactive-Komponenten vom lokalen Knoten zu entfernen.

pgactive_snowflake_id_nextval

Generiert knotenspezifische eindeutige Sequenzwerte.

Argumente

- regclass

Rückgabetyp

bigint

Nutzungshinweise

Verwenden Sie diese Funktion, um knotenspezifische eindeutige Sequenzwerte zu generieren.

pgactive_update_node_conninfo

Aktualisiert die Verbindungsinformationen für einen pgactive-Knoten.

Argumente

- node_name_to_update (Text)
- node_dsn_to_update (Text)

Rückgabetyp

void

Nutzungshinweise

Verwenden Sie diese Funktion, um die Verbindungsinformationen für einen pgactive-Knoten zu aktualisieren.

pgactive_wait_for_node_ready

Überwacht den Fortschritt der Gruppenerstellung oder des Beitritts.

Argumente

- Timeout (Ganzzahl, Standard: 0)
- progress_interval (Ganzzahl, Standard: 60)

Rückgabetyp

void

Nutzungshinweise

Rufen Sie diese Funktion auf, um den Fortschritt der Gruppenerstellung oder des Beitritts zu überwachen.

Umgang mit Konflikten bei der Aktiv-Aktiv-Replikation

Die „`pgactive`“-Erweiterung funktioniert pro Datenbank und nicht pro Cluster. Jede DB-Instance, die „`pgactive`“ verwendet, ist eine unabhängige Instance und kann Datenänderungen aus jeder Quelle

akzeptieren. Wenn eine Änderung an eine DB-Instance gesendet wird, schreibt PostgreSQL sie lokal fest und repliziert die Änderung dann mithilfe von „pgactive“ asynchron auf andere DB-Instances. Wenn zwei PostgreSQL-DB-Instances denselben Datensatz fast zur gleichen Zeit aktualisieren, kann ein Konflikt auftreten.

Die Erweiterung „pgactive“ bietet Mechanismen zur Konflikterkennung und automatischen Lösung. Sie verfolgt den Zeitstempel, zu dem die Transaktion auf beiden DB-Instances festgeschrieben wurde, und wendet die Änderung automatisch mit dem neuesten Zeitstempel an. Die Erweiterung „pgactive“ protokolliert auch, wenn in der Tabelle pgactive.pgactive_conflict_history ein Konflikt auftritt.

Sie pgactive.pgactive_conflict_history werden weiter wachsen. Möglicherweise möchten Sie eine Löschrichtlinie definieren. Dies kann erreicht werden, indem Sie regelmäßig einige Datensätze löschen oder ein Partitionierungsschema für diese Beziehung definieren (und später die gewünschten Partitionen trennen, löschen oder kürzen). Um die Löschrichtlinie regelmäßig zu implementieren, besteht eine Möglichkeit darin, die Erweiterung zu verwenden. pg_cron In den folgenden Informationen finden Sie ein Beispiel für die pg_cron Verlaufstabelle Scheduling maintenance with the PostgreSQL pg_cron extension.

Umgang mit Sequenzen bei der Aktiv-Aktiv-Replikation

Eine RDS-für-PostgreSQL-DB-Instance mit der Erweiterung „pgactive“ verwendet zwei verschiedene Sequenzmechanismen, um eindeutige Werte zu generieren.

Globale Sequenzen

Um eine globale Sequenz zu verwenden, erstellen Sie mit der Anweisung CREATE SEQUENCE eine lokale Sequenz. Verwenden Sie pgactive.pgactive_snowflake_id_nextval(seqname) statt usingnextval(seqname), um den nächsten eindeutigen Wert der Sequenz abzurufen.

Mit dem folgenden Beispiel wird eine globale Sequenz erstellt:

```
app=> CREATE TABLE gstest (
      id bigint primary key,
      parrot text
    );
```

```
app=>CREATE SEQUENCE gstest_id_seq OWNED BY gstest.id;
```

```
app=> ALTER TABLE gstest \
```

```
      ALTER COLUMN id SET DEFAULT \
      pgactive.pgactive_snowflake_id_nextval('gstest_id_seq');
```

Partitionierte Sequenzen

In Split-Step- oder partitionierten Sequenzen wird auf jedem Knoten eine normale PostgreSQL-Sequenz verwendet. Jede Sequenz wird um den gleichen Betrag inkrementiert und beginnt mit unterschiedlichen Offsets. Bei Schritt 100 generiert Knoten 1 beispielsweise die Sequenz 101, 201, 301 usw. und Knoten 2 generiert die Sequenz 102, 202, 302 usw. Dieses Schema funktioniert auch dann gut, wenn die Knoten über einen längeren Zeitraum nicht kommunizieren können. Es ist jedoch erforderlich, dass der Designer bei der Einrichtung des Schemas eine maximale Anzahl von Knoten festlegt und jeden Knoten einzeln konfiguriert. Fehler können leicht zu überlappenden Sequenzen führen.

Es ist relativ einfach, diesen Ansatz mit `pgactive` zu konfigurieren, indem Sie die gewünschte Sequenz auf einem Knoten wie folgt erstellen:

```
CREATE TABLE some_table (generated_value bigint primary key);
```

```
app=> CREATE SEQUENCE some_seq INCREMENT 100 OWNED BY some_table.generated_value;
```

```
app=> ALTER TABLE some_table ALTER COLUMN generated_value SET DEFAULT
 nextval('some_seq');
```

Rufen Sie dann `setval` auf jedem Knoten auf, um einen anderen Offset-Startwert wie folgt anzugeben.

```
app=>
-- On node 1
SELECT setval('some_seq', 1);

-- On node 2
SELECT setval('some_seq', 2);
```

Reduzieren von überflüssigen Daten in Tabellen und Indizes mit der Erweiterung pg_repack

Sie können die `pg_repack` Erweiterung verwenden, um Bloat aus Tabellen und Indizes als Alternative zu entfernen. `VACUUM FULL` Diese Erweiterung wird RDS für SQL Postgre-Versionen 9.6.3 und höher unterstützt. Weitere Informationen zur `pg_repack` Erweiterung und zum vollständigen Table Repack finden Sie in der GitHub Projektdokumentation.

Im `VACUUM FULL` Gegensatz dazu erfordert die `pg_repack` Erweiterung in den folgenden Fällen nur für einen kurzen Zeitraum während der Neuerstellung der Tabelle eine exklusive Sperre (AccessExclusiveLock):

- Erste Erstellung der Protokolltabelle — Eine Protokolltabelle wird erstellt, um Änderungen aufzuzeichnen, die während der ersten Kopie der Daten vorgenommen wurden, wie im folgenden Beispiel gezeigt:

```
postgres=>\dt+ repack.log_*
List of relations
-[ RECORD 1 ]-+----------
Schema        | repack
Name          | log_16490
Type          | table
Owner         | postgres
Persistence   | permanent
Access method | heap
Size          | 65 MB
Description   |
```

- Letzte swap-and-drop Phase.

Für den Rest des Neuaufbauvorgangs ist lediglich eine `ACCESS SHARE` Sperre für die Originaltabelle erforderlich, um Zeilen aus dieser Tabelle in die neue Tabelle zu kopieren. Dadurch können die DELETE Operationen INSERTUPDATE, und wie gewohnt weitergeführt werden.

Empfehlungen

Die folgenden Empfehlungen gelten, wenn Sie Bloat aus den Tabellen und Indizes mithilfe der `pg_repack` Erweiterung entfernen:

- Führen Sie das Umpacken außerhalb der Geschäftszeiten oder während eines Wartungsfensters durch, um die Auswirkungen auf die Leistung anderer Datenbankaktivitäten so gering wie möglich zu halten.
- Überwachen Sie blockierende Sitzungen während der Neuerstellungsaktivität genau und stellen Sie sicher, dass es keine Aktivität in der Originaltabelle gibt, die möglicherweise blockiert werden könntepg_repack, insbesondere in der letzten swap-and-drop Phase, in der eine exklusive Sperre für die Originaltabelle erforderlich ist. Weitere Informationen finden Sie unter Identifizieren, was eine Abfrage blockiert.

  Wenn Sie eine blockierende Sitzung sehen, können Sie sie nach reiflicher Überlegung mit dem folgenden Befehl beenden. Dies hilft bei der Fortsetzung des pg_repack Neuaufbaus:

  ```
  SELECT pg_terminate_backend(pid);
  ```

- Beim Anwenden der aufgelaufenen Änderungen aus der pg_repack's Protokolltabelle auf Systeme mit einer sehr hohen Transaktionsrate kann der Anwenden-Prozess möglicherweise nicht mit der Änderungsrate Schritt halten. In solchen Fällen pg_repack könnte das Antragsverfahren nicht abgeschlossen werden. Weitere Informationen finden Sie unter Überwachung der neuen Tabelle während des Repacks. Wenn Indizes stark aufgebläht sind, besteht eine alternative Lösung darin, nur den Index neu zu packen. Dies trägt auch dazu bei, VACUUM dass die Indexbereinigungszyklen schneller abgeschlossen werden.

  Ab SQL Postgre-Version 12 können Sie die Phase der Indexbereinigung manuell überspringen. VACUUM Ab Postgre Version 14 wird sie beim automatischen Notabsaugen automatisch übersprungen. SQL Auf diese Weise können Sie schneller VACUUM fertig werden, ohne den aufgeblähten Index zu entfernen, und ist nur für Notfallsituationen gedacht, z. B. zur Vermeidung von Wraparound-Vorgängen. VACUUM Weitere Informationen finden Sie unter Vermeidung von Aufblähungen in Indizes im Amazon Aurora Aurora-Benutzerhandbuch.

Voraussetzungen

- Die Tabelle muss eine Einschränkung haben PRIMARY KEY oder nicht UNIQUE Null.
- Die Erweiterungsversion muss für den Client und den Server identisch sein.
- Stellen Sie sicher, dass die RDS Instanz mehr FreeStorageSpace als die Gesamtgröße der Tabelle ohne Aufblähung hat. Stellen Sie sich als Beispiel die Gesamtgröße der Tabelle einschließlich Indizes mit 2 TB TOAST und die Gesamtgröße der Tabelle mit 1 TB vor. Der

erforderliche Wert `FreeStorageSpace` muss größer sein als der von der folgenden Berechnung zurückgegebene Wert:

`2TB (Table size) - 1TB (Table bloat) = 1TB`

Sie können die folgende Abfrage verwenden, um die Gesamtgröße der Tabelle zu überprüfen und daraus eine Aufblähung `pgstattuple` abzuleiten. Weitere Informationen finden Sie unter Diagnosing Table and Index Bloat im Amazon Aurora Aurora-Benutzerhandbuch

```
SELECT pg_size_pretty(pg_total_relation_size('table_name')) AS total_table_size;
```

Dieser Speicherplatz wird nach Abschluss der Aktivität zurückgewonnen.

- Stellen Sie sicher, dass die RDS Instance über genügend Rechen- und I/O-Kapazität verfügt, um den Repack-Vorgang abzuwickeln. Sie könnten erwägen, die Instance-Klasse zu skalieren, um ein optimales Leistungsgleichgewicht zu erzielen.

Um die **`pg_repack`** Erweiterung zu verwenden

1. Installieren Sie die `pg_repack` Erweiterung auf Ihrer RDS for SQL Postgre-DB-Instance, indem Sie den folgenden Befehl ausführen.

   ```
   CREATE EXTENSION pg_repack;
   ```

2. Führen Sie die folgenden Befehle aus, um Schreibzugriff auf temporäre Protokolltabellen zu gewähren, die von `pg_repack` erstellt wurden.

   ```
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT INSERT ON TABLES TO PUBLIC;
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT USAGE, SELECT ON SEQUENCES TO
    PUBLIC;
   ```

3. Stellen Sie mithilfe des `pg_repack` Client-Dienstprogramms eine Connect zur Datenbank her. Verwenden Sie ein Konto, das `rds_superuser`-Berechtigungen hat. Nehmen Sie beispielsweise an, dass die `rds_test`-Rolle `rds_superuser`-Berechtigungen hat. Die folgende Syntax gilt `pg_repack` für vollständige Tabellen, einschließlich aller Tabellenindizes in der `postgres` Datenbank.

   ```
   pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
    -k postgres
   ```

> **Note**
>
> Sie müssen die Verbindung mit der Option -k herstellen. Die Option -a wird nicht unterstützt.

Die Antwort des `pg_repack` Clients enthält Informationen zu den Tabellen auf der DB-Instance, die neu gepackt wurden.

```
INFO: repacking table "pgbench_tellers"
INFO: repacking table "pgbench_accounts"
INFO: repacking table "pgbench_branches"
```

4. Die folgende Syntax packt eine einzelne Tabelle `orders` einschließlich der Indizes in der Datenbank neu. `postgres`

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders -k postgres
```

Die folgende Syntax packt nur Indizes für `orders` Tabellen in der Datenbank neu. `postgres`

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders --only-indexes -k postgres
```

Überwachung der neuen Tabelle während des Repacks

- Die Größe der Datenbank wird bis zur swap-and-drop Phase des Repacks um die Gesamtgröße der Tabelle abzüglich des Bloats erhöht. Sie können die Wachstumsrate der Datenbankgröße überwachen, die Geschwindigkeit des Neupackens berechnen und die Zeit, die bis zum Abschluss der ersten Datenübertragung benötigt wird, grob abschätzen.

  Stellen Sie sich als Beispiel die Gesamtgröße der Tabelle mit 2 TB, die Größe der Datenbank mit 4 TB und die Gesamtgröße der Tabelle mit 1 TB vor. Der durch die Berechnung am Ende des Repack-Vorgangs zurückgegebene Wert für die Gesamtgröße der Datenbank lautet wie folgt:

  `2TB (Table size) + 4 TB (Database size) - 1TB (Table bloat) = 5TB`

Sie können die Geschwindigkeit des Repack-Vorgangs grob schätzen, indem Sie die Wachstumsrate in Byte zwischen zwei Zeitpunkten ermitteln. Wenn die Wachstumsrate 1 GB pro Minute beträgt, kann es etwa 1000 Minuten oder 16,6 Stunden dauern, bis die erste Tabellenerstellung abgeschlossen ist. Zusätzlich zur ersten Tabellenerstellung müssen `pg_repack` auch die aufgelaufenen Änderungen übernommen werden. Die dafür benötigte Zeit hängt von der Geschwindigkeit ab, mit der die laufenden Änderungen und die aufgelaufenen Änderungen angewendet werden.

> **Note**
>
> Sie können die `pgstattuple` Erweiterung verwenden, um die Aufblähung in der Tabelle zu berechnen. Weitere Informationen finden Sie unter pgstattuple.

- Die Anzahl der Zeilen in der `pg_repack's` Protokolltabelle gemäß dem Repack-Schema entspricht der Menge der Änderungen, die nach dem ersten Laden noch auf die neue Tabelle angewendet werden müssen.

  Sie können die `pg_repack's` Protokolltabelle eincheckenpg_stat_all_tables, um die Änderungen zu überwachen, die auf die neue Tabelle angewendet wurden. `pg_stat_all_tables.n_live_tup`gibt die Anzahl der Datensätze an, deren Übernahme auf die neue Tabelle noch aussteht. Weitere Informationen finden Sie unter pg_stat_all_tables.

```
postgres=>SELECT relname,n_live_tup FROM pg_stat_all_tables WHERE schemaname =
 'repack' AND relname ILIKE '%log%';

-[ RECORD 1 ]---------
relname    | log_16490
n_live_tup | 2000000
```

- Sie können die `pg_stat_statements` Erweiterung verwenden, um herauszufinden, wie viel Zeit für jeden Schritt des Repack-Vorgangs benötigt wird. Dies ist hilfreich bei der Vorbereitung auf die Anwendung desselben Upack-Vorgangs in einer Produktionsumgebung. Sie können die `LIMIT` Klausel anpassen, um die Ausgabe weiter zu erweitern.

```
postgres=>SELECT
     SUBSTR(query, 1, 100) query,
     round((round(total_exec_time::numeric, 6) / 1000 / 60),4)
 total_exec_time_in_minutes
```

```
 FROM
     pg_stat_statements
 WHERE
     query ILIKE '%repack%'
 ORDER BY
     total_exec_time DESC LIMIT 5;

 query                                                                    |
 total_exec_time_in_minutes
--------------------------------------------------------------------------
+---------------------------
 CREATE UNIQUE INDEX index_16493 ON repack.table_16490 USING btree (a) |
        6.8627
 INSERT INTO repack.table_16490 SELECT a FROM ONLY public.t1           |
        6.4150
 SELECT repack.repack_apply($1, $2, $3, $4, $5, $6)                    |
        0.5395
 SELECT repack.repack_drop($1, $2)                                     |
        0.0004
 SELECT repack.repack_swap($1)                                         |
        0.0004
(5 rows)
```

Das Umpacken ist ein vollständiger out-of-place Vorgang, sodass die Originaltabelle nicht beeinträchtigt wird und wir nicht mit unerwarteten Problemen rechnen, die eine Wiederherstellung der Originaltabelle erforderlich machen. Wenn das Umpacken unerwartet fehlschlägt, müssen Sie die Ursache des Fehlers untersuchen und ihn beheben.

Wenn das Problem behoben ist, löschen Sie die `pg_repack` Erweiterung, erstellen Sie sie in der Datenbank, in der sich die Tabelle befindet, und wiederholen Sie den Schritt. `pg_repack` Darüber hinaus spielen die Verfügbarkeit von Rechenressourcen und der gleichzeitige Zugriff auf die Tabelle eine entscheidende Rolle für den rechtzeitigen Abschluss des Repack-Vorgangs.

Erweiterung aktualisieren und verwenden PLV8

PLV8 ist eine vertrauenswürdige Javascript-Spracherweiterung für PostgreSQL. Sie können sie für gespeicherte Prozeduren, Trigger und anderen prozeduralen Code verwenden, der von SQL aus aufrufbar ist. Diese Spracherweiterung wird von allen aktuellen Releases von PostgreSQL unterstützt.

Wenn Sie PostgreSQL verwenden PLV8und auf eine neue PLV8 Version aktualisieren, nutzen Sie sofort die Vorteile der neuen Erweiterung. Führen Sie die folgenden Schritte aus, um Ihre Katalogmetadaten mit der neuen Version von zu synchronisieren. PLV8 Diese Schritte sind optional, aber wir empfehlen, sie durchzuführen, um Warnungen aufgrund fehlender Übereinstimmung von Metadaten zu vermeiden.

Durch den Upgrade-Vorgang werden alle Ihre vorhandenen PLV8 Funktionen gelöscht. Daher empfehlen wir, vor dem Upgrade einen Snapshot Ihrer DB-Instance von RDS for PostgreSQL zu erstellen. Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots für eine Single-AZ-DB-Instance für Amazon RDS.

> **Important**
>
> Ab PostgreSQL Version 18 wird Amazon RDS for PostgreSQL die Erweiterungen und PostgreSQL als veraltet betrachten. `plcoffee` `plls` Wir empfehlen Ihnen, die Verwendung von CoffeeScript und LiveScript in Ihren Anwendungen einzustellen, um sicherzustellen, dass Sie über einen Upgrade-Pfad für future Engine-Versions-Upgrades verfügen.

Um Ihre Katalogmetadaten mit einer neuen Version von zu synchronisieren PLV8

1. Überprüfen, ob Sie ein Update benötigen. Führen Sie dazu den folgenden Befehl aus, während Sie mit Ihrer Instance verbunden sind.

   ```
   SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
   ```

   Wenn Ihre Ergebnisse Werte für eine installierte Version enthalten, die niedriger ist als die Standardversion, setzen Sie dieses Verfahren fort, um Ihre Erweiterungen zu aktualisieren. Die folgende Ergebnismenge beispielsweise deutet darauf hin, dass Sie ein Update vornehmen sollten.

   ```
   name    | default_version | installed_version |                    comment
   ```

```
--------+----------------+------------------
+-------------------------------------------------
plls    | 2.1.0          | 1.5.3             | PL/LiveScript (v8) trusted
 procedural language
plcoffee| 2.1.0          | 1.5.3             | PL/CoffeeScript (v8) trusted
 procedural language
plv8    | 2.1.0          | 1.5.3             | PL/JavaScript (v8) trusted
 procedural language
(3 rows)
```

2. Erstellen Sie einen Snapshot Ihrer RDS-for-PostgreSQL-DB-Instance, wenn Sie dies noch nicht getan haben. Sie können die folgenden Schritte fortsetzen, während der Snapshot erstellt wird.
3. Ermitteln Sie die Anzahl der PLV8 Funktionen in Ihrer DB-Instance, sodass Sie nach dem Upgrade überprüfen können, ob sie alle vorhanden sind. Die folgende SQL-Abfrage gibt beispielsweise die Anzahl der in plv8, plcoffee und plls geschriebenen Funktionen zurück.

```
SELECT proname, nspname, lanname
FROM pg_proc p, pg_language l, pg_namespace n
WHERE p.prolang = l.oid
AND n.oid = p.pronamespace
AND lanname IN ('plv8','plcoffee','plls');
```

4. Erstellen Sie mit pg_dump eine Dump-Datei, die nur das Schema enthält. Erstellen Sie beispielsweise eine Datei auf Ihrem Client-Computer im Verzeichnis `/tmp`.

```
./pg_dump -Fc --schema-only -U master postgres >/tmp/test.dmp
```

In diesem Beispiel werden die folgenden Optionen verwendet:

- `-Fc` – Benutzerdefiniertes Format
- --schema-only – Erstellen Sie nur einen Dump von Befehlen, die zum Erstellen des Schemas erforderlich sind (in diesem Fall Funktionen)
- `-U` – Der RDS-Hauptbenutzername
- `database` – Der Datenbankname für unsere DB-Instance

Weitere Informationen zu pg_dump finden Sie unter pg_dump in der PostgreSQL-Dokumentation.

5. Extrahieren Sie die in der Dump-Datei vorhandene DDL-Anweisung „CREATE FUNCTION”. Im folgenden Beispiel wird der `grep`-Befehl verwendet, um die DDL-Anweisung zu extrahieren, die die Funktionen erstellt, und sie in einer Datei zu speichern. Sie verwenden diese in nachfolgenden Schritten, um die Funktionen neu zu erstellen.

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list
```

   Weitere Informationen zu pg_restore finden Sie unter pg_restore in der PostgreSQL-Dokumentation.

6. Verwerfen Sie die Funktionen und Erweiterungen. Im folgenden Beispiel werden alle PLV8 basierten Objekte gelöscht. Die Option cascade stellt sicher, dass alle Abhängigkeiten verworfen werden.

```
DROP EXTENSION plv8 CASCADE;
```

   Wenn Ihre PostgreSQL-Instance Objekte enthält, die auf plcoffee oder plls basieren, wiederholen Sie diesen Schritt für diese Erweiterungen.

7. Erstellen Sie die Erweiterungen. Das folgende Beispiel erstellt die plv8-, plcoffee- und plls-Erweiterungen.

```
CREATE EXTENSION plv8;
CREATE EXTENSION plcoffee;
CREATE EXTENSION plls;
```

8. Erstellen Sie die Funktionen unter Verwendung der Dump-Datei und der „Treiber“-Datei.

   Das folgende Beispiel erstellt die zuvor extrahierten Funktionen neu.

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. Prüfen Sie mithilfe der folgenden Abfrage, ob all Ihre Funktionen neu erstellt wurden.

```
SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
```

   Die PLV8 Version 2 fügt Ihrer Ergebnismenge die folgende zusätzliche Zeile hinzu:

```
    proname    |  nspname   | lanname
---------------+------------+----------
```

```
plv8_version  | pg_catalog | plv8
```

Verwendung von PL/Rust zum Schreiben von PostgreSQL-Funktionen in der Sprache Rust

PL/Rust is a trusted Rust language extension for PostgreSQL. You can use it for stored procedures, functions, and other procedural code that's callable from SQL. The PL/RustDie Spracherweiterung ist in den folgenden Versionen verfügbar:

- RDS für PostgreSQL 17.1 und höhere 17-Versionen
- RDS für PostgreSQL 16.1 und höhere 16-Versionen
- RDS für PostgreSQL 15.2-R2 und höhere 15-Versionen
- RDS für PostgreSQL 14.9 und höhere 14-Versionen
- RDS für PostgreSQL 13.12 und höhere 13-Versionen

Weitere Informationen finden Sie unter PL/Rust on. GitHub

Themen

- Einrichten von PL/Rust
- Erstellen von Funktionen mit PL/Rust
- Verwenden von Kisten mit PL/Rust
- Einschränkungen von PL/Rust

Einrichten von PL/Rust

Um die plrust-Erweiterung auf Ihrer DB-Instance zu installieren, fügen Sie plrust zum Parameter `shared_preload_libraries` in der DB-Parametergruppe hinzu, die mit Ihrer DB-Instance verknüpft ist. Wenn die plrust-Erweiterung installiert ist, können Sie Funktionen erstellen.

Damit der Parameter `shared_preload_libraries` geändert werden kann, muss Ihre DB-Instance mit einer benutzerdefinierten Parametergruppe verknüpft sein. Informationen zum Erstellen einer benutzerdefinierten DB-Parametergruppe finden Sie unter Parametergruppen für Amazon RDS.

Sie können die Plrust-Erweiterung mit dem oder dem AWS Management Console installieren. AWS CLI

Bei den folgenden Schritten wird davon ausgegangen, dass Ihre DB-Instance einer benutzerdefinierten DB-Parametergruppe zugeordnet ist.

Konsole

Installieren der plrust-Erweiterung im Parameter **`shared_preload_libraries`**

Führen Sie die folgenden Schritte mit einem Konto aus, das Mitglied der `rds_superuser`-Gruppe (Rolle) ist.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen Ihrer DB-Instance aus, um ihre Details anzuzeigen.
4. Öffnen Sie die Registerkarte Konfiguration für Ihre DB-Instance und suchen Sie den Link zur DB-Instance-Parametergruppe.
5. Wählen Sie den Link aus, um die benutzerdefinierten Parameter zu öffnen, die Ihrer DB-Instance zugeordnet sind.
6. Geben Sie in das Suchfeld Parameters (Parameter) `shared_pre` ein, um den **`shared_preload_libraries`**-Parameter zu finden.
7. Wählen Sie Edit parameters (Parameter bearbeiten) aus, um auf die Eigenschaftswerte zuzugreifen.
8. Fügen Sie der Liste im Feld Werte „plrust“ hinzu. Verwenden Sie ein Komma, um Elemente in der Werteliste zu trennen.
9. Starten Sie die DB-Instance neu, damit die Änderung am Parameter `shared_preload_libraries` in Kraft tritt. Der erste Neustart kann zusätzliche Zeit in Anspruch nehmen.
10. Wenn die Instance verfügbar ist, überprüfen Sie, ob „plrust“ initialisiert wurde. Stellen Sie mit `psql` eine Verbindung mit der DB-Instance her und führen Sie den folgenden Befehl aus.

    ```
    SHOW shared_preload_libraries;
    ```

    Ihre Ausgabe sollte wie folgt aussehen:

    ```
    shared_preload_libraries
    -------------------------
    ```

```
rdsutils,plrust
(1 row)
```

AWS CLI

Installieren der plrust-Erweiterung im Parameter shared_preload_libraries

Führen Sie die folgenden Schritte mit einem Konto aus, das Mitglied der `rds_superuser`-Gruppe (Rolle) ist.

1. Verwenden Sie den modify-db-parameter-group AWS CLI Befehl, um plrust zum `shared_preload_libraries` Parameter hinzuzufügen.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=plrust,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Verwenden Sie den reboot-db-instance AWS CLI Befehl, um die DB-Instance neu zu starten und die Plrust-Bibliothek zu initialisieren. Der erste Neustart kann zusätzliche Zeit in Anspruch nehmen.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Wenn die Instance verfügbar ist, können Sie überprüfen, ob „plrust“ initialisiert wurde. Stellen Sie mit `psql` eine Verbindung mit der DB-Instance her und führen Sie den folgenden Befehl aus.

```
SHOW shared_preload_libraries;
```

   Ihre Ausgabe sollte wie folgt aussehen:

```
shared_preload_libraries
--------------------------
rdsutils,plrust
(1 row)
```

Erstellen von Funktionen mit PL/Rust

PL/Rust kompiliert die Funktion als dynamische Bibliothek, lädt sie und führt sie aus.

Die folgende Rust-Funktion filtert Vielfache aus einem Array.

```
postgres=> CREATE LANGUAGE plrust;
CREATE EXTENSION
```

```
CREATE OR REPLACE FUNCTION filter_multiples(a BIGINT[], multiple BIGINT) RETURNS
 BIGINT[]
    IMMUTABLE STRICT
    LANGUAGE PLRUST AS
$$
    Ok(Some(a.into_iter().filter(|x| x.unwrap() % multiple != 0).collect()))
$$;

WITH gen_values AS (
SELECT ARRAY(SELECT * FROM generate_series(1,100)) as arr)
SELECT filter_multiples(arr, 3)
from gen_values;
```

Verwenden von Kisten mit PL/Rust

In RDS für PostgreSQL Versionen 16.3-R2 und höher, 15.7-R2 und höher 15 Versionen, 14.12-R2 und höher 14 Versionen und 13.15-R2 und höher 13 Versionen unterstützt PL/Rust zusätzliche Crates:

- `url`
- `regex`
- `serde`
- `serde_json`

In RDS für PostgreSQL Versionen 15.5-R2 und höher, 14.10-R2 und höher 14 Versionen und 13.13-R2 und höher 13 Versionen unterstützt PL/Rust zwei zusätzliche Crates:

- `croaring-rs`
- `num-bigint`

Ab den Versionen 15.4, 14.9 und 13.12 von Amazon RDS for PostgreSQL unterstützt PL/Rust die folgenden Crates:

- `aes`
- `ctr`
- `rand`

Für diese Kisten werden nur die Standardfunktionen unterstützt. Neue Versionen von RDS für PostgreSQL enthalten möglicherweise aktualisierte Versionen von Kisten und ältere Versionen von Kisten werden möglicherweise nicht mehr unterstützt.

Folgen Sie den bewährten Methoden für die Durchführung eines Hauptversions-Upgrades, um zu testen, ob Ihre PL/Rust-Funktionen mit der neuen Hauptversion kompatibel sind. Weitere Informationen finden Sie im Blog Bewährte Methoden für das Upgrade von Amazon RDS auf Haupt- und Nebenversionen von PostgreSQL und Upgrade einer PostgreSQL-DB-Engine für Amazon RDS im Amazon-RDS-Benutzerhandbuch.

Beispiele für die Verwendung von Abhängigkeiten bei der Erstellung einer PL/Rust-Funktion finden Sie unter Use dependencies.

Einschränkungen von PL/Rust

Standardmäßig können Datenbankbenutzer den folgenden Befehl nicht verwendenPL/Rust. To provide access to PL/Rust, sich als Benutzer mit der Berechtigung rds_superuser verbinden und ihn ausführen:

```
postgres=> GRANT USAGE ON LANGUAGE PLRUST TO user;
```

Verwaltung von Geodaten mit der GIS Post-Erweiterung

Post GIS ist eine Erweiterung von Postgre SQL zum Speichern und Verwalten von Geoinformationen. Weitere Informationen zu Post finden Sie GIS unter GISPost.net.

Ab Version 10.5 SQL unterstützt Postgre die Bibliothek libprotobuf 1.3.0, die von Post GIS für die Arbeit mit Vektorkacheldaten von Mapboxen verwendet wird.

Für die Einrichtung der Post-Erweiterung sind Rechte erforderlich. GIS `rds_superuser` Wir empfehlen Ihnen, einen Benutzer (eine Rolle) für die Verwaltung der GIS Post-Erweiterung und Ihrer Geodaten zu erstellen. Die GIS Post-Erweiterung und die zugehörigen Komponenten erweitern Postgre SQL um Tausende von Funktionen. Erwägen Sie, die GIS Post-Erweiterung in einem eigenen Schema zu erstellen, wenn dies für Ihren Anwendungsfall sinnvoll ist. Das folgende Beispiel zeigt, wie die Erweiterung in einer eigenen Datenbank installiert wird. Dies ist jedoch nicht erforderlich.

Themen

- Schritt 1: Erstellen Sie einen Benutzer (eine Rolle) zur Verwaltung der GIS Post-Erweiterung
- Schritt 2: Laden Sie die Post-Erweiterungen GIS
- Schritt 3: Übertragung der Inhaberschaft an den Erweiterungsschemas
- Schritt 4: Übertragen Sie den Besitz der GIS Post-Tabellen
- Schritt 5: Testen der Erweiterungen
- Schritt 6: Aktualisieren Sie die GIS Post-Erweiterung
- Versionen nach GIS der Erweiterung
- Aktualisierung von Post GIS 2 auf Post 3 GIS

Schritt 1: Erstellen Sie einen Benutzer (eine Rolle) zur Verwaltung der GIS Post-Erweiterung

Stellen Sie zunächst als Benutzer mit `rds_superuser` Rechten eine Verbindung zu Ihrer RDS for SQL Postgre-DB-Instance her. Wenn Sie beim Einrichten Ihrer Instance den Standardnamen beibehalten haben, stellen Sie eine Verbindung als `postgres` her.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --username=postgres
 --password
```

Erstellen Sie eine separate Rolle (Benutzer), um die GIS Post-Erweiterung zu verwalten.

```
postgres=>  CREATE ROLE gis_admin LOGIN PASSWORD 'change_me';
CREATE ROLE
```

Gewähren Sie dieser Rolle `rds_superuser`-Berechtigungen, damit die Rolle die Erweiterung installieren kann.

```
postgres=> GRANT rds_superuser TO gis_admin;
GRANT
```

Erstellen Sie eine Datenbank, die Sie für Ihre GIS Post-Artefakte verwenden können. Dieser Schritt ist optional. Oder Sie können in Ihrer Benutzerdatenbank ein Schema für die GIS Post-Erweiterungen erstellen, aber das ist auch nicht erforderlich.

```
postgres=> CREATE DATABASE lab_gis;
CREATE DATABASE
```

Gewähren Sie `gis_admin` alle Berechtigungen für die `lab_gis`-Datenbank.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_gis TO gis_admin;
GRANT
```

Beenden Sie die Sitzung und stellen Sie erneut eine Verbindung zu Ihrer RDS for SQL Postgre-DB-Instance als her. `gis_admin`

```
postgres=> psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=gis_admin --password --dbname=lab_gis
Password for user gis_admin:...
lab_gis=>
```

Fahren Sie mit der Einrichtung der Erweiterung fort, wie in den nächsten Schritten beschrieben.

Schritt 2: Laden Sie die Post-Erweiterungen GIS

Die GIS Post-Erweiterung umfasst mehrere verwandte Erweiterungen, die zusammenarbeiten, um Geodatenfunktionen bereitzustellen. Je nach Anwendungsfall benötigen Sie möglicherweise nicht alle Erweiterungen, die Sie in diesem Schritt erstellt haben.

Verwenden Sie `CREATE EXTENSION` Anweisungen, um die GIS Post-Erweiterungen zu laden.

```
CREATE EXTENSION postgis;
CREATE EXTENSION
CREATE EXTENSION postgis_raster;
CREATE EXTENSION
CREATE EXTENSION fuzzystrmatch;
CREATE EXTENSION
CREATE EXTENSION postgis_tiger_geocoder;
CREATE EXTENSION
CREATE EXTENSION postgis_topology;
CREATE EXTENSION
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION
```

Sie können die Ergebnisse überprüfen, indem Sie die im folgenden Beispiel gezeigte SQL Abfrage ausführen, in der die Erweiterungen und ihre Besitzer aufgelistet sind.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;
List of schemas
     Name     |   Owner
--------------+-----------
 public       | postgres
 tiger        | rdsadmin
 tiger_data   | rdsadmin
 topology     | rdsadmin
(4 rows)
```

Schritt 3: Übertragung der Inhaberschaft an den Erweiterungsschemas

Verwenden Sie die ALTER SCHEMA Anweisungen, um das Eigentum an den Schemas auf die `gis_admin` Rolle zu übertragen.

```
ALTER SCHEMA tiger OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA tiger_data OWNER TO gis_admin;
```

```
ALTER SCHEMA
ALTER SCHEMA topology OWNER TO gis_admin;
ALTER SCHEMA
```

Sie können den Eigentümerwechsel bestätigen, indem Sie die folgende SQL Abfrage ausführen. Oder Sie können den Metabefehl \dn über die psql-Befehlszeile verwenden.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;

       List of schemas
     Name     |     Owner
--------------+---------------
 public       | postgres
 tiger        | gis_admin
 tiger_data   | gis_admin
 topology     | gis_admin
(4 rows)
```

Schritt 4: Übertragen Sie den Besitz der GIS Post-Tabellen

> **Note**
>
> Ändern Sie nicht die Inhaberschaft der GIS Post-Funktionen. Ein ordnungsgemäßer Betrieb und future Upgrades von Post GIS setzen voraus, dass diese Funktionen das ursprüngliche Eigentum behalten. Weitere Informationen zu GIS Post-Berechtigungen finden Sie unter SQLPostgre-Sicherheit.

Verwenden Sie die folgende Funktion, um das Eigentum an den GIS Post-Tabellen auf die `gis_admin` Rolle zu übertragen. Führen Sie die folgende Anweisung aus der psql-Aufforderung aus, um die Funktion zu erstellen.

```
CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $f$ BEGIN EXECUTE
 $1; RETURN $1; END; $f$;
CREATE FUNCTION
```

Führen Sie als Nächstes die folgende Abfrage aus, um die `exec`-Funktion auszuführen, die wiederum die Anweisungen ausführt und die Berechtigungen ändert.

```
SELECT exec('ALTER TABLE ' || quote_ident(s.nspname) || '.' || quote_ident(s.relname)
 || ' OWNER TO gis_admin;')
  FROM (
    SELECT nspname, relname
    FROM pg_class c JOIN pg_namespace n ON (c.relnamespace = n.oid)
    WHERE nspname in ('tiger','topology') AND
    relkind IN ('r','S','v') ORDER BY relkind = 'S')
s;
```

Schritt 5: Testen der Erweiterungen

Um zu vermeiden, dass der Schemaname angegeben werden muss, fügen Sie das `tiger`-Schema Ihrem Suchpfad unter Verwendung des folgenden Befehls hinzu.

```
SET search_path=public,tiger;
SET
```

Testen Sie das `tiger` Schema mithilfe der folgenden SELECT Anweisung.

```
SELECT address, streetname, streettypeabbrev, zip
 FROM normalize_address('1 Devonshire Place, Boston, MA 02109') AS na;
address | streetname | streettypeabbrev |  zip
---------+------------+------------------+-------
       1 | Devonshire | Pl               | 02109
(1 row)
```

Weitere Informationen zu dieser Erweiterung finden Sie unter Tiger Geocoder in der GIS Post-Dokumentation.

Testen Sie den Zugriff auf das `topology`-Schema, indem Sie folgende `SELECT`-Anweisung verwenden. Das ruft die `createtopology`-Funktion zum Registrieren eines neuen Topologieobjekts (my_new_topo) mit der angegebenen Raumbezugskennung (26986) und der Standardtoleranz (0,5) auf. Weitere Informationen finden Sie CreateTopologyin der GIS Post-Dokumentation.

```
SELECT topology.createtopology('my_new_topo',26986,0.5);
 createtopology
----------------
              1
```

```
(1 row)
```

Schritt 6: Aktualisieren Sie die GIS Post-Erweiterung

Jede neue Version von Postgre SQL unterstützt eine oder mehrere Versionen der GIS Post-Erweiterung, die mit dieser Version kompatibel sind. Durch ein Upgrade der SQL Postgre-Engine auf eine neue Version wird die GIS Post-Erweiterung nicht automatisch aktualisiert. Bevor Sie die SQL Postgre-Engine aktualisieren, aktualisieren Sie Post normalerweise GIS auf die neueste verfügbare Version für die aktuelle SQL Postgre-Version. Details hierzu finden Sie unter Versionen nach GIS der Erweiterung.

Nach dem Upgrade der SQL Postgre-Engine aktualisieren Sie die GIS Post-Erweiterung erneut auf die Version, die für die neu aktualisierte SQL Postgre-Engine-Version unterstützt wird. So führen Sie ein Hauptversions-Upgrade für RDS for PostgreSQL durch.

Sie können jederzeit auf Ihrem nach verfügbaren Updates der GIS Post-Erweiterungsversion suchen. Führen Sie dazu den folgenden Befehl aus. Diese Funktion ist mit Post GIS 2.5.0 und höheren Versionen verfügbar.

```
SELECT postGIS_extensions_upgrade();
```

Wenn Ihre Anwendung die neueste GIS Post-Version nicht unterstützt, können Sie eine ältere Version von Post, GIS die in Ihrer Hauptversion verfügbar ist, wie folgt installieren.

```
CREATE EXTENSION postgis VERSION "2.5.5";
```

Wenn Sie von einer älteren Version auf eine bestimmte GIS Post-Version aktualisieren möchten, können Sie auch den folgenden Befehl verwenden.

```
ALTER EXTENSION postgis UPDATE TO "2.5.5";
```

Abhängig von der Version, von der Sie ein Upgrade durchführen, müssen Sie diese Funktion möglicherweise noch einmal verwenden. Das Ergebnis des ersten Durchlaufs der Funktion bestimmt, ob eine zusätzliche Upgrade-Funktion benötigt wird. Dies ist beispielsweise bei einem Upgrade von Post GIS 2 auf Post GIS 3 der Fall. Weitere Informationen finden Sie unter Aktualisierung von Post GIS 2 auf Post 3 GIS.

Wenn Sie diese Erweiterung aktualisiert haben, um sich auf ein Upgrade der Hauptversion der SQL Postgre-Engine vorzubereiten, können Sie mit anderen vorbereitenden Aufgaben fortfahren. Weitere

Informationen finden Sie unter So führen Sie ein Hauptversions-Upgrade für RDS for PostgreSQL durch.

Versionen nach GIS der Erweiterung

Wir empfehlen, dass Sie die Versionen aller Erweiterungen GIS wie Post installieren, die unter Postgre aufgeführt sind. SQL Erweiterungsversionen für Amazon RDS for Postgre SQL in den SQLVersionshinweisen zu Amazon RDS for Postgre. Sie können mithilfe des folgenden Befehls auflisten, welche Versionen in Ihrer Version verfügbar sind.

```
SELECT * FROM pg_available_extension_versions WHERE name='postgis';
```

Versionsinformationen finden Sie in den folgenden Abschnitten der SQLVersionshinweise zu Amazon RDS for Postgre:

- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 16 RDS
- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 15 RDS
- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 14 RDS
- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 13 RDS
- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 12 RDS
- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 11 RDS
- Von Amazon unterstützte Erweiterungen der SQL Postgre-Version 10 RDS
- Von Amazon unterstützte SQL Postgre-Erweiterungen der Version 9.6.x RDS

Aktualisierung von Post GIS 2 auf Post 3 GIS

Ab Version 3.0 ist die GIS Post-Raster-Funktionalität jetzt eine separate Erweiterung`postgis_raster`. Diese Erweiterung hat einen eigenen Installations- und Upgrade-Pfad. Dadurch werden Dutzende von Funktionen, Datentypen und anderen Artefakten, die für die Rasterbildverarbeitung erforderlich sind, aus der `postgis`-Kernerweiterung entfernt. Das heißt, wenn Ihr Anwendungsfall keine Raster-Verarbeitung erfordert, müssen Sie die `postgis_raster`-Erweiterung nicht installieren.

Im folgenden Upgrade-Beispiel extrahiert der erste Upgrade-Befehl die Raster-Funktionalität in die `postgis_raster`-Erweiterung. Ein zweiter Upgrade-Befehl ist dann erforderlich, um ein Upgrade von `postgis_raster` auf die neue Version durchzuführen.

Um von Post GIS 2 auf Post GIS 3 zu aktualisieren

1. Identifizieren Sie die Standardversion von PostGIS, die für die SQL Postgre-Version auf Ihrem verfügbar ist. RDSfür die Postgre-DB-Instance. SQL Führen Sie dazu die folgende Abfrage durch.

```
SELECT * FROM pg_available_extensions
    WHERE default_version > installed_version;
  name   | default_version | installed_version |                          comment
---------+-----------------+-------------------
+------------------------------------------------------------
 postgis | 3.1.4           | 2.3.7             | PostGIS geometry and geography
 spatial types and functions
(1 row)
```

2. Identifizieren Sie die Versionen von Post, die in jeder Datenbank auf installiert sind. Ihre RDS für SQL Postgre-DB-Instance. Anders gesagt: Fragen Sie jede Benutzerdatenbank wie folgt ab.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
    AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
  Name   | Version | Schema |                            Description
---------+---------+--------
+------------------------------------------------------------------------
 postgis | 2.3.7   | public | PostGIS geometry, geography, and raster spatial types
 and functions
(1 row)
```

   Diese Diskrepanz zwischen der Standardversion (Post GIS 3.1.4) und der installierten Version (Post GIS 2.3.7) bedeutet, dass Sie die Post-Erweiterung aktualisieren müssen. GIS

```
ALTER EXTENSION postgis UPDATE;
ALTER EXTENSION
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
```

3. Führen Sie die folgende Abfrage aus, um sicherzustellen, dass sich die Raster-Funktionalität jetzt in einem eigenen Paket befindet.

```
SELECT
    probin,
    count(*)
FROM
    pg_proc
WHERE
    probin LIKE '%postgis%'
GROUP BY
    probin;
          probin          | count
--------------------------+-------
 $libdir/rtpostgis-2.3    | 107
 $libdir/postgis-3        | 487
(2 rows)
```

Die Ausgabe zeigt, dass es immer noch einen Unterschied zwischen den Versionen gibt. Die GIS Post-Funktionen sind Version 3 (postgis-3), während die Raster-Funktionen (rtpostgis) Version 2 (rtpostgis-2.3) sind. Um das Upgrade abzuschließen, führen Sie den Upgrade-Befehl erneut wie folgt aus.

```
postgres=> SELECT postgis_extensions_upgrade();
```

Die Warnmeldungen können Sie ohne Bedenken ignorieren. Führen Sie die folgende Abfrage erneut aus, um zu überprüfen, ob das Upgrade abgeschlossen ist. Das Upgrade ist abgeschlossen, wenn Post GIS und alle zugehörigen Erweiterungen nicht als aktualisierungsbedürftig gekennzeichnet sind.

```
SELECT postgis_full_version();
```

4. Verwenden Sie die folgende Abfrage, um den abgeschlossenen Upgrade-Vorgang und die separat gepackten Erweiterungen anzuzeigen und zu überprüfen, ob ihre Versionen übereinstimmen.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
        AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
      Name      | Version | Schema |                            Description
----------------+---------+--------
+-------------------------------------------------------------------------
 postgis        | 3.1.5   | public | PostGIS geometry, geography, and raster
 spatial types and functions
 postgis_raster | 3.1.5   | public | PostGIS raster types and functions
(2 rows)
```

   Die Ausgabe zeigt, dass die Post GIS 2-Erweiterung auf Post GIS 3 aktualisiert wurde und dass beide `postgis` und die jetzt separate `postgis_raster` Erweiterung Version 3.1.5 sind.

Wenn Sie nach Abschluss dieses Upgrades die Raster-Funktionalität nicht verwenden möchten, können Sie die Erweiterung wie folgt löschen.

```
DROP EXTENSION postgis_raster;
```

Arbeiten mit den unterstützten Foreign Data Wrappern für RDS SQL

Ein Foreign Data Wrapper (FDW) ist ein bestimmter Erweiterungstyp, der den Zugriff auf externe Daten ermöglicht. Die `oracle_fdw` Erweiterung ermöglicht es beispielsweise Ihrer , mit Oracle-Datenbanken zu arbeiten. SQL Ein weiteres Beispiel: Mithilfe der SQL nativen `postgres_fdw` Postgre-Erweiterung können Sie auf Daten zugreifen, die in SQL Postgre-DB-Instances gespeichert sind, die sich außerhalb Ihrer RDS for Postgre-DB-Instance befinden. SQL

Im Folgenden finden Sie Informationen zu verschiedenen unterstützten Postgre-Wrappern für ausländische Daten. SQL

Themen

- Verwenden der Erweiterung log_fdw für den Zugriff auf das DB-Protokoll mit SQL
- Verwenden der postgres_fdw-Erweiterung für den Zugriff auf externe Daten
- Arbeiten mit My SQL databases mithilfe der Erweiterung mysql_fdw
- Arbeiten mit Oracle-Datenbanken unter Verwendung der Erweiterung oracle_fdw
- Arbeiten mit SQL Serverdatenbanken mithilfe der Erweiterung tds_fdw

Verwenden der Erweiterung log_fdw für den Zugriff auf das DB-Protokoll mit SQL

Der für die SQL Postgre-DB-Instance unterstützt die `log_fdw` Erweiterung, mit der Sie über eine SQL Schnittstelle auf Ihr Datenbank-Engine-Protokoll zugreifen können. Die `log_fdw`-Erweiterung umfasst zwei neue Funktionen, die das Erstellen von Fremdtabellen für Datenbankprotokolle erleichtern:

- `list_postgres_log_files` führt die Dateien im Datenbankprotokollverzeichnis und die Dateigröße in Bytes auf.
- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` erstellt eine Fremdtabelle für die angegebene Datei in der aktuellen Datenbank.

Alle durch `log_fdw` erstellten Funktionen gehören `rds_superuser`. Mitglieder der `rds_superuser`-Rolle können anderen Datenbankbenutzern Zugriff auf diese Funktionen gewähren.

Standardmäßig werden die Protokolldateien von im Format `stderr` (Standardfehler) generiert, wie im `log_destination` Parameter angegeben. Für diesen Parameter gibt es nur zwei Optionen: `stderr` und `csvlog` (kommagetrennte Werte,CSV). Wenn Sie die `csvlog` Option zum Parameter hinzufügen, RDS generiert Amazon `stderr` sowohl als auch `csvlog` Protokolle. Dies kann die Speicherkapazität Ihres DB-Clusters beeinträchtigen. Daher müssen Sie sich der anderen Parameter bewusst sein, die sich auf die Protokollbehandlung auswirken. Weitere Informationen finden Sie unter Festlegen des Protokollziels (`stderr`, `csvlog`).

Ein Vorteil der Generierung von `csvlog`-Protokollen ist, dass Sie mit der `log_fdw`-Erweiterung Fremdtabellen erstellen können, bei denen die Daten ordentlich in verschiedene Spalten aufgeteilt sind. Dazu muss Ihre Instance einer benutzerdefinierten DB-Parametergruppe zugeordnet sein, damit Sie die Einstellung für `log_destination` ändern können. Weitere Informationen zur Vorgehensweise finden Sie unter Arbeiten mit Parametern auf der DB-Instance von RDS for PostgreSQL.

Im folgenden Beispiel wird angenommen, dass der `log_destination`-Parameter `cvslog` enthält.

So verwenden Sie die Erweiterung log_fdw:

1. Installieren Sie die `log_fdw`-Erweiterung.

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. Erstellen Sie den Protokollserver als Fremddaten-Wrapper.

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. Wählen Sie aus einer Liste an Protokolldateien "alle" aus.

```
postgres=> SELECT * FROM list_postgres_log_files() ORDER BY 1;
```

Beispiel-Antwort.

```
          file_name           | file_size_bytes
```

```
------------------------------+-----------------
 postgresql.log.2023-08-09-22.csv |            1111
 postgresql.log.2023-08-09-23.csv |            1172
 postgresql.log.2023-08-10-00.csv |            1744
 postgresql.log.2023-08-10-01.csv |            1102
(4 rows)
```

4. Erstellen Sie eine Tabelle mit einer einzigen 'log_entry'-Spalte für die ausgewählte Datei.

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
     'log_server', 'postgresql.log.2023-08-09-22.csv');
```

Die Antwort liefert keine weiteren Details außer, dass die Tabelle jetzt existiert.

```
-----------------------------------
(1 row)
```

5. Wählen Sie ein Beispiel der Protokolldatei aus. Mit dem folgenden Code werden die Protokollzeit und die Fehlermeldungsbeschreibung abgerufen.

```
postgres=> SELECT log_time, message FROM my_postgres_error_log ORDER BY 1;
```

Beispiel-Antwort.

```
             log_time             |                                 message
----------------------------------
+-------------------------------------------------------------------------------
Tue Aug 09 15:45:18.172 2023 PDT | ending log output to stderr
Tue Aug 09 15:45:18.175 2023 PDT | database system was interrupted; last known up
 at 2023-08-09 22:43:34 UTC
Tue Aug 09 15:45:18.223 2023 PDT | checkpoint record is at 0/90002E0
Tue Aug 09 15:45:18.223 2023 PDT | redo record is at 0/90002A8; shutdown FALSE
Tue Aug 09 15:45:18.223 2023 PDT | next transaction ID: 0/1879; next OID: 24578
Tue Aug 09 15:45:18.223 2023 PDT | next MultiXactId: 1; next MultiXactOffset: 0
Tue Aug 09 15:45:18.223 2023 PDT | oldest unfrozen transaction ID: 1822, in
 database 1
(7 rows)
```

Verwenden der postgres_fdw-Erweiterung für den Zugriff auf externe Daten

Auf die Daten in einer Tabelle auf einem Remote-Datenbank-Server können Sie mit der Erweiterung postgres_fdw zugreifen. Wenn Sie eine Remote-Verbindung ausgehend von einer PostgreSQL-DB-Instance einrichten, ist der Zugriff auch für das Lesereplikat verfügbar.

Verwenden Sie postgres_fdw wie folgt für den Zugriff auf einen Remote-Datenbank-Server:

1. Installieren Sie die Erweiterung postgres_fdw.

```
CREATE EXTENSION postgres_fdw;
```

2. Erstellen Sie einen Fremddaten-Server mit CREATE SERVER.

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. Erstellen Sie ein Benutzer-Mapping, um die Rolle zu identifizieren, die auf dem Remote-Server verwendet werden soll.

> **Important**
>
> Um das Passwort so zu ändern, dass es nicht in den Protokollen erscheint, legen Sie es auf `log_statement=none` Sitzungsebene fest. Durch die Einstellung auf Parameterebene wird das Passwort nicht geschwärzt.

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. Erstellen Sie eine Tabelle, die der Tabelle auf dem Remote-Server zugewiesen ist.

```
CREATE FOREIGN TABLE foreign_table (
        id integer NOT NULL,
        data text)
SERVER foreign_server
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

Arbeiten mit My SQL databases mithilfe der Erweiterung mysql_fdw

Um von Ihrem auf eine My -kompatible Datenbank zuzugreifen, können Sie die Erweiterung installieren und verwenden. `mysql_fdw` Mit diesem Foreign Data Wrapper können Sie mit RDS MySQL, Aurora MySQL, MariaDB und anderen My SQL -kompatiblen Datenbanken arbeiten. Die Verbindung vom für die SQL Postgre-DB-Instance zur SQL My-Datenbank wird je nach Client- und Serverkonfiguration auf Best-Effort-Basis verschlüsselt. Sie können die Verschlüsselung jedoch erzwingen, wenn Sie möchten. Weitere Informationen finden Sie unter Verwenden der Verschlüsselung während der Übertragung mit der Erweiterung.

Die `mysql_fdw` Erweiterung wird von 13.6 und höheren Versionen unterstützt. Sie unterstützt Auswahlen, Einfügungen, Aktualisierungen und Löschungen aus einer RDS for Postgre-DB in Tabellen auf einer My -kompatiblen Datenbank-InstanceSQL. SQL

Themen

- DB für die Verwendung der Erweiterung mysql_fdw einrichten
- Beispiel: Arbeiten mit einer Datenbank von SQL
- Verwenden der Verschlüsselung während der Übertragung mit der Erweiterung

DB für die Verwendung der Erweiterung mysql_fdw einrichten

Die Einrichtung der `mysql_fdw` Erweiterung auf Ihrem für die SQL Postgre-DB-Instance umfasst das Laden der Erweiterung in Ihre und das anschließende Erstellen des Verbindungspunkts zur My SQL DB-Instance. Für diese Aufgabe benötigen Sie die folgenden Details zur My SQL DB-Instance:

- Hostname oder Endpunkt. Bei einem für My SQL DB-Instance können Sie den Endpunkt mithilfe der Konsole finden. Wählen Sie die Registerkarte „Connectivity & security“ (Konnektivität und Sicherheit) aus und sehen Sie im Abschnitt „Endpoint and port“ (Endpunkt und Port) nach.
- Port-Nummer. Die Standard-Portnummer für My SQL ist 3306.
- Name der Datenbank. Die DB-ID.

Sie müssen auch Zugriff auf die Sicherheitsgruppe oder die Zugriffskontrollliste (ACL) für den SQL My-Port 3306 gewähren. Sowohl der für die SQL Postgre-DB-Instance und die RDS for My SQL DB-Instance benötigen Zugriff auf Port 3306. Wenn der Zugriff nicht korrekt konfiguriert ist, wird beim Versuch, eine Verbindung zur My SQL -kompatiblen Tabelle herzustellen, eine Fehlermeldung ähnlich der folgenden angezeigt:

```
ERROR: failed to connect to MySQL: Can't connect to MySQL server on 'hostname.aws-
region.rds.amazonaws.com:3306' (110)
```

Im folgenden Verfahren erstellen Sie (als `rds_superuser`-Konto) den fremden Server. Anschließend gewähren Sie bestimmten Benutzern Zugriff auf den fremden Server. Diese Benutzer erstellen dann ihre eigenen Zuordnungen zu den entsprechenden SQL Benutzerkonten vom Typ Meine Benutzer, um mit der My SQL DB-Instance zu arbeiten.

Um mysql_fdw für den Zugriff auf einen My Database-Server zu verwenden SQL

1. Stellen Sie mithilfe eines Kontos, das die `rds_superuser` Rolle hat, eine Connect zu Ihrer SQL Postgre-DB-Instance her. Wenn Sie bei der Erstellung Ihres die Standardeinstellungen akzeptiert haben, lautet der Benutzername`postgres`, und Sie können mit dem `psql` Befehlszeilentool wie folgt eine Verbindung herstellen:

   ```
   psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Installieren Sie die `mysql_fdw`-Erweiterung wie folgt:

   ```
   postgres=> CREATE EXTENSION mysql_fdw;
   CREATE EXTENSION
   ```

Nachdem die Erweiterung auf Ihrem für die SQL Postgre-DB-Instance installiert wurde, richten Sie den ausländischen Server ein, der die Verbindung zu einer My SQL database bereitstellt.

So erstellen Sie den fremden Server

Führen Sie diese Aufgaben auf dem aus. Die Schritte setzen voraus, dass Sie als Benutzer mit `rds_superuser`-Berechtigungen wie `postgres` verbunden sind.

1. Erstellen Sie einen Fremdserver im :

   ```
   postgres=> CREATE SERVER mysql-db FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'db-
   name.111122223333.aws-region.rds.amazonaws.com', port '3306');
   CREATE SERVER
   ```

2. Gewähren Sie den entsprechenden Benutzern Zugriff auf den fremden Server. Dies sollten keine Administratorbenutzer sein, d. h. Benutzer ohne `rds_superuser`-Rolle.

```
postgres=> GRANT USAGE ON FOREIGN SERVER mysql-db to user1;
GRANT
```

SQLPostgre-Benutzer erstellen und verwalten ihre eigenen Verbindungen zur SQL My-Datenbank über den fremden Server.

Beispiel: Arbeiten mit einer Datenbank von SQL

Angenommen, Sie haben eine einfache Tabelle auf einer für SQL Postgre-DB-Instance. Ihre SQL Benutzer von für Postgre möchten (`SELECT`), `INSERTUPDATE`, und `DELETE` Elemente in dieser Tabelle abfragen. Gehen Sie davon aus, dass die `mysql_fdw` Erweiterung auf Ihrer RDS for SQL Postgre-DB-Instance erstellt wurde, wie im vorherigen Verfahren beschrieben. Nachdem Sie als Benutzer mit `rds_superuser` Rechten eine Verbindung zur RDS for SQL Postgre-DB-Instance hergestellt haben, können Sie mit den folgenden Schritten fortfahren.

1. Erstellen Sie auf der für die SQL Postgre-DB-Instance einen fremden Server:

   ```
   test=> CREATE SERVER mysqldb FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'your-
   DB.aws-region.rds.amazonaws.com', port '3306');
   CREATE SERVER
   ```

2. Gewähren Sie einem Benutzer, der keine `rds_superuser`-Berechtigungen hat, die Nutzung, zum Beispiel `user1`:

   ```
   test=> GRANT USAGE ON FOREIGN SERVER mysqldb TO user1;
   GRANT
   ```

3. Connect als *`user1`*und erstellen Sie dann eine Zuordnung zum Benutzer „Mein SQL Benutzer“:

   ```
   test=> CREATE USER MAPPING FOR user1 SERVER mysqldb OPTIONS (username 'myuser',
    password 'mypassword');
   CREATE USER MAPPING
   ```

4. Erstellen Sie eine Fremdtabelle, die mit der SQL Tabelle Meine verknüpft ist:

   ```
   test=> CREATE FOREIGN TABLE mytab (a int, b text) SERVER mysqldb OPTIONS (dbname
    'test', table_name '');
   CREATE FOREIGN TABLE
   ```

5. Führen Sie eine einfache Abfrage mit der fremden Tabelle aus:

```
test=> SELECT * FROM mytab;
a |   b
---+-------
1 | apple
(1 row)
```

6. Sie können Daten zur SQL Tabelle Meine hinzufügen, ändern und daraus entfernen. Beispielsweise:

```
test=> INSERT INTO mytab values (2, 'mango');
INSERT 0 1
```

Führen Sie die SELECT-Abfrage noch einmal aus, um die Ergebnisse zu sehen:

```
test=> SELECT * FROM mytab ORDER BY 1;
 a |   b
---+-------
1 | apple
2 | mango
(2 rows)
```

Verwenden der Verschlüsselung während der Übertragung mit der Erweiterung

Die Verbindung SQL von zu My SQL verwendet standardmäßig Verschlüsselung bei der Übertragung (TLS/SSL). Die Verbindung wird jedoch nicht verschlüsselt, wenn sich die Client- und Serverkonfiguration unterscheiden. Sie können die Verschlüsselung für alle ausgehenden Verbindungen erzwingen, indem Sie die `REQUIRE SSL` Option RDS für Meine SQL Benutzerkonten angeben. Derselbe Ansatz funktioniert auch für MariaDB- und Aurora SQL My-Benutzerkonten.

Für Meine SQL Benutzerkonten, die so konfiguriert sind`REQUIRE SSL`, schlägt der Verbindungsversuch fehl, wenn keine sichere Verbindung hergestellt werden kann.

Um die Verschlüsselung für bestehende Benutzerkonten vom SQL Typ Meine Datenbank zu erzwingen, können Sie den `ALTER USER` Befehl verwenden. Die Syntax variiert je nach SQL Version von My, wie in der folgenden Tabelle dargestellt. Weitere Informationen finden Sie ALTERUSERin Mein SQL Referenzhandbuch.

Mein SQL 5.7, Mein SQL 8.0	Mein SQL 5.6
`ALTER USER 'user'@'%' REQUIRE SSL;`	`GRANT USAGE ON *.* to 'user'@'%' REQUIRE SSL;`

Weitere Informationen zur Erweiterung `mysql_fdw` finden Sie in der mysql_fdw-Dokumentation.

Arbeiten mit Oracle-Datenbanken unter Verwendung der Erweiterung oracle_fdw

Um von Ihrer RDS-für-PostgreSQL-DB-Instance auf eine Oracle-Datenbank zuzugreifen, können Sie die Erweiterung `oracle_fdw` installieren und verwenden. Diese Erweiterung ist ein Fremddaten-Wrapper für Oracle-Datenbanken. Weitere Informationen zu dieser Erweiterung finden Sie in der oracle_fdw-Dokumentation.

Die Erweiterung `oracle_fdw` wird von den RDS-für-PostgreSQL-Versionen 12.7, 13.3 und höher unterstützt.

Themen

- Aktivieren der Erweiterung oracle_fdw
- Beispiel: Verwendung eines fremden Servers, der mit einer Amazon RDS for Oracle Database verknüpft ist
- Datenverschlüsselung während der Übertragung
- Informationen zur Ansicht pg_user_mappings und zu Berechtigungen

Aktivieren der Erweiterung oracle_fdw

So verwenden Sie die Erweiterung oracle_fdw:

Aktivieren der Erweiterung oracle_fdw

- Führen Sie den folgenden Befehl mit einem Konto aus, das die `rds_superuser`-Berechtigungen besitzt.

```
CREATE EXTENSION oracle_fdw;
```

Beispiel: Verwendung eines fremden Servers, der mit einer Amazon RDS for Oracle Database verknüpft ist

Das folgende Beispiel zeigt die Verwendung eines fremden Servers, der mit einer Amazon-RDS-für-Oracle-Datenbank verknüpft ist.

So erstellen Sie einen fremden Server, der mit einer RDS-for-Oracle-Datenbank verknüpft ist:

1. Beachten Sie Folgendes auf der RDS-for-Oracle-DB-Instance:
   - Endpunkt
   - Port
   - Datenbankname
2. Erstellen Sie einen fremden Server.

```
test=> CREATE SERVER oradb FOREIGN DATA WRAPPER oracle_fdw OPTIONS (dbserver
 '//endpoint:port/DB_name');
CREATE SERVER
```

3. Gewähren Sie einem Benutzer, der keine `rds_superuser`-Berechtigungen hat, die Nutzung, zum Beispiel `user1`.

```
test=> GRANT USAGE ON FOREIGN SERVER oradb TO user1;
GRANT
```

4. Verbinden Sie sich als `user1` und erstellen Sie ein Mapping zu einem Oracle-Benutzer.

```
test=> CREATE USER MAPPING FOR user1 SERVER oradb OPTIONS (user 'oracleuser',
 password 'mypassword');
CREATE USER MAPPING
```

5. Erstellen einer fremden Tabelle, die mit einer Oracle-Tabelle verknüpft ist.

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER oradb OPTIONS (table 'MYTABLE');
CREATE FOREIGN TABLE
```

6. Fragen Sie die fremde Tabelle ab.

```
test=>  SELECT * FROM mytab;
a
```

```
---
1
(1 row)
```

Wenn die Abfrage den folgenden Fehler meldet, überprüfen Sie Ihre Sicherheitsgruppe und Zugriffssteuerungsliste (Access Control List, ACL), um sicherzustellen, dass beide Instances kommunizieren können.

```
ERROR: connection for foreign table "mytab" cannot be established
DETAIL: ORA-12170: TNS:Connect timeout occurred
```

Datenverschlüsselung während der Übertragung

PostgreSQL-to-Oracle Die Verschlüsselung bei der Übertragung basiert auf einer Kombination von Client- und Serverkonfigurationsparametern. Ein Beispiel für Oracle 21c finden Sie unter Informationen zu den Werten für Verschlüsselung und Integrität in der Oracle-Dokumentation. Der für oracle_fdw auf Amazon RDS verwendete Client ist mit konfiguriert`ACCEPTED`, was bedeutet, dass die Verschlüsselung von der Konfiguration des Oracle-Datenbankservers abhängt und für die Verschlüsselung die Oracle Security Library (libnnz) verwendet wird.

Wenn sich Ihre Datenbank auf RDS for Oracle befindet, finden Sie weitere Informationen zum Konfigurieren der Verschlüsselung unter Oracle Native Network Encryption.

Informationen zur Ansicht pg_user_mappings und zu Berechtigungen

Der PostgreSQL-Katalog `pg_user_mapping` speichert das Mapping eines Benutzers von RDS for PostgreSQL für den Benutzer auf einem fremden Datenserver (Remote). Der Zugriff auf den Katalog ist eingeschränkt, aber Sie verwenden die Ansicht `pg_user_mappings`, um die Mappings zu sehen. Das folgende Beispiel zeigt, wie Berechtigungen für eine Oracle-Beispieldatenbank gelten. Diese Informationen gelten im Allgemeinen für jeden fremden Daten-Wrapper.

In der folgenden Ausgabe finden Sie Rollen und Berechtigungen, die drei verschiedenen Beispielbenutzern zugeordnet sind. Benutzer `rdssu1` und `rdssu2` sind Mitglieder der `rds_superuser`-Rolle und `user1` nicht. In dem Beispiel wird der `psql`-Metabefehl `\du` verwendet, um vorhandene Rollen aufzulisten.

```
test=>  \du
                                                    List of roles
```

```
    Role name    |                         Attributes                         |
                  Member of
-----------------+------------------------------------------------------------
+-----------------------------------------------------------
 rdssu1          |                                                            |
 {rds_superuser}
 rdssu2          |                                                            |
 {rds_superuser}
 user1           |                                                            | {}
```

Alle Benutzer, einschließlich Benutzer mit `rds_superuser`-Berechtigungen, dürfen ihre eigenen Benutzerzuordnungen (`umoptions`) in der `pg_user_mappings`-Tabelle anzeigen. Wie im folgenden Beispiel gezeigt, wird trotz `rdssu1rds_superuser`-Berechtigungen ein Fehler gemeldet, wenn `rdssu1` versucht, alle Benutzerzuordnungen abzurufen:

```
test=> SELECT * FROM pg_user_mapping;
ERROR: permission denied for table pg_user_mapping
```

Im Folgenden sind einige Beispiele aufgeführt.

```
test=> SET SESSION AUTHORIZATION rdssu1;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     | {user=oracleuser,password=mypwd}
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)

test=> SET SESSION AUTHORIZATION rdssu2;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     | {user=oracleuser,password=mypwd}
 (3 rows)

test=> SET SESSION AUTHORIZATION user1;
SET
```

```
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |           umoptions
-------+-------+---------+--------+------------+-------------------------------
 16414 | 16411 | oradb   |  16412 | user1      | {user=oracleuser,password=mypwd}
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)
```

Aufgrund von Unterschieden in der Implementierung von `information_schema._pg_user_mappings` und `pg_catalog.pg_user_mappings` erfordert ein manuell erstelltes `rds_superuser` zusätzliche Berechtigungen zum Anzeigen von Passwörtern in `pg_catalog.pg_user_mappings`.

Es sind keine zusätzlichen Berechtigungen für `rds_superuser` nötig, um Kennwörter in `information_schema._pg_user_mappings` anzuzeigen.

Benutzer, die nicht über die `rds_superuser`-Rolle verfügen können Passwörter in `pg_user_mappings` nur unter den folgenden Bedingungen anzeigen:

- Der aktuelle Benutzer ist der zugeordnete Benutzer und besitzt den Server oder besitzt die `USAGE`-Berechtigung dafür.
- Der aktuelle Benutzer ist der Serverbesitzer und das Mapping ist für `PUBLIC`.

Arbeiten mit SQL Serverdatenbanken mithilfe der Erweiterung tds_fdw

Sie können die SQL `tds_fdw` Postgre-Erweiterung verwenden, um auf Datenbanken zuzugreifen, die das Tabular Data Stream (TDS) -Protokoll unterstützen, z. B. Sybase- und Microsoft SQL Server-Datenbanken. Mit diesem Foreign Data Wrapper können Sie von Ihrem aus eine Verbindung zu Datenbanken herstellen, die das TDS Protokoll verwenden, einschließlich Amazon RDS for Microsoft Server. SQL Weitere Informationen finden Sie in der tds-fdw/tds_fdw-Dokumentation unter. GitHub

Die `tds_fdw` Erweiterung wird auf Amazon RDS für SQL Postgre-Versionen 14.2, 13.6 und höhere Versionen unterstützt.

Einrichten Ihrer Aurora SQL Postgre-DB für die Verwendung der tds_fdw-Erweiterung

In den folgenden Verfahren finden Sie ein Beispiel für die Einrichtung und Verwendung des . Bevor Sie eine Verbindung zu einer SQL Serverdatenbank herstellen können`tds_fdw`, benötigen Sie die folgenden Informationen für die Instance:

- Hostname oder Endpunkt. RDSBei einer SQL Server-DB-Instance können Sie den Endpunkt mithilfe der Konsole finden. Wählen Sie die Registerkarte „Connectivity & security“ (Konnektivität und Sicherheit) aus und sehen Sie im Abschnitt „Endpoint and port“ (Endpunkt und Port) nach.
- Port-Nummer. Die Standardportnummer für Microsoft SQL Server ist 1433.
- Name der Datenbank. Die DB-ID.

Sie müssen auch Zugriff auf die Sicherheitsgruppe oder die Zugriffskontrollliste (ACL) für den SQL Serverport 1433 gewähren. Sowohl der für die SQL Postgre-DB-Instance als auch die RDS for SQL Server-DB-Instance benötigen Zugriff auf Port 1433. Wenn der Zugriff nicht richtig konfiguriert ist, wird beim Versuch, den Microsoft SQL Server abzufragen, die folgende Fehlermeldung angezeigt:

```
ERROR: DB-Library error: DB #: 20009, DB Msg: Unable to connect:
Adaptive Server is unavailable or does not exist (mssql2019.aws-
region.rds.amazonaws.com), OS #: 0, OS Msg: Success, Level: 9
```

Um mit tds_fdw eine Verbindung zu einer Serverdatenbank herzustellen SQL

1. Stellen Sie eine Verbindung mit der her, indem Sie ein Konto verwenden, das die folgende Rolle hat: `rds_superuser`

   ```
   psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=test --password
   ```

2. Installieren Sie die `tds_fdw`-Erweiterung.

   ```
   test=> CREATE EXTENSION tds_fdw;
   CREATE EXTENSION
   ```

Nachdem die Erweiterung auf Ihrem für die SQL Postgre-DB-Instance installiert wurde, richten Sie den ausländischen Server ein.

So erstellen Sie den fremden Server

Führen Sie diese Aufgaben auf dem für die SQL Postgre-DB-Instance mit einem Konto aus, das über Berechtigungen verfügt`rds_superuser`.

1. Erstellen Sie einen Fremdserver im :

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS
 (servername 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database
 'tds_fdw_testing');
CREATE SERVER
```

Um SQLServer nebenbei auf ASCII Nicht-Daten zuzugreifen, erstellen Sie einen Serverlink mit der Option character_set im : SQL

```
test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS (servername
 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database 'tds_fdw_testing',
 character_set 'UTF-8');
CREATE SERVER
```

2. Gewähren Sie einem Benutzer, der keine `rds_superuser`-Rollenberechtigungen hat, Berechtigungen, zum Beispiel `user1`:

```
test=> GRANT USAGE ON FOREIGN SERVER sqlserverdb TO user1;
```

3. Stellen Sie eine Connect als Benutzer1 her und erstellen Sie eine Zuordnung zu einem SQL Serverbenutzer:

```
test=> CREATE USER MAPPING FOR user1 SERVER sqlserverdb OPTIONS (username
 'sqlserveruser', password 'password');
CREATE USER MAPPING
```

4. Erstellen Sie eine Fremdtabelle, die mit einer SQL Servertabelle verknüpft ist:

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER sqlserverdb OPTIONS (table
 'MYTABLE');
CREATE FOREIGN TABLE
```

5. Fragen Sie die fremde Tabelle ab:

```
test=> SELECT * FROM mytab;
 a
---
 1
(1 row)
```

Verwenden der Verschlüsselung während der Übertragung für die Verbindung

Die Verbindung von for Postgre SQL zum SQL Server verwendet je nach SQL Serverdatenbankkonfiguration Verschlüsselung bei der Übertragung (TLS/SSL). Wenn der SQL Server nicht für die Verschlüsselung konfiguriert ist, wird der RDS SQL For-Postgre-Client, der die Anfrage an die SQL Serverdatenbank stellt, auf unverschlüsselt zurückgesetzt.

Sie können die Verschlüsselung für die Verbindung zu RDS SQL Server-DB-Instances erzwingen, indem Sie den `rds.force_ssl` Parameter festlegen. Wie das geht, erfahren Sie unter Verbindungen zu Ihrer DB-Instance zur Nutzung erzwingen SSL. Weitere Informationen zur TLS Konfiguration vonSSL/RDSfür SQL Server finden Sie unter SSLMit einer Microsoft SQL Server-DB-Instance verwenden.

Arbeiten mit Trusted Language Extensions für PostgreSQL

Trusted Language Extensions für PostgreSQL ist ein Open-Source-Entwicklungskit für die Erstellung von PostgreSQL-Erweiterungen. Es ermöglicht Ihnen, leistungsstarke PostgreSQL-Erweiterungen zu erstellen und diese sicher auf Ihrer PostgreSQL-DB-Instance auszuführen. Mithilfe von Trusted Language Extensions (TLE) für PostgreSQL können Sie PostgreSQL-Erweiterungen erstellen, die dem dokumentierten Ansatz zur Erweiterung der PostgreSQL-Funktionalität folgen. Weitere Informationen finden Sie unter Packaging Related Objects in a Extension in der PostgreSQL-Dokumentation.

Ein wesentlicher Vorteil von TLE besteht darin, dass Sie es in Umgebungen verwenden können, die keinen Zugriff auf das der PostgreSQL-Instance zugrunde liegende Dateisystem bieten. Bisher war für die Installation einer neuen Erweiterung Zugriff auf das Dateisystem erforderlich. Mit TLE entfällt diese Einschränkung. Es bietet eine Entwicklungsumgebung für die Erstellung neuer Erweiterungen für jede PostgreSQL-Datenbank, einschließlich solcher, die auf Ihren DB-Instances von RDS für PostgreSQL ausgeführt werden.

TLE wurde entwickelt, um den Zugriff auf unsichere Ressourcen für die Erweiterungen zu verhindern, die Sie mit TLE erstellen. Die Laufzeitumgebung begrenzt die Auswirkungen eines Erweiterungsdefekts auf eine einzelne Datenbankverbindung. TLE verleiht Datenbankadministratoren auch eine detaillierte Kontrolle darüber, wer Erweiterungen installieren kann, und bietet ein Berechtigungsmodell für deren Ausführung.

TLE wird von den folgenden RDS-for-PostgreSQL-Versionen unterstützt:

- Version 17.1 und höher 17 Versionen
- Version 16.1 und höher 16 Versionen
- Version 15.2 und höher 15 Versionen
- Version 14.5 und höher 14 Versionen
- Version 13.12 und höher 13 Versionen

Die Entwicklungsumgebung und Laufzeit von Trusted Language Extensions sind als `pg_tle`-PostgreSQL-Erweiterung, Version 1.0.1, verpackt. Es unterstützt die Erstellung von Erweiterungen in Perl JavaScript, Tcl, PL/pgSQL und SQL. Sie installieren die `pg_tle`-Erweiterung in Ihrer DB-Instance von RDS für PostgreSQL auf die gleiche Weise wie andere PostgreSQL-Erweiterungen. Nach der Einrichtung von `pg_tle` können Entwickler damit neue PostgreSQL-Erweiterungen, sogenannte TLE-Erweiterungen, erstellen.

In den folgenden Themen finden Sie Informationen darüber, wie Sie Trusted Language Extensions einrichten und Ihre eigenen TLE-Erweiterungen erstellen.

Themen

- Terminologie
- Anforderungen für die Verwendung von Trusted Language Extensions für Postgre SQL
- Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem
- Überblick über vertrauenswürdige Spracherweiterungen für Postgre SQL
- TLEErweiterungen für erstellen SQL
- Löschen Ihrer TLE Erweiterungen aus einer Datenbank
- Deinstallation von Trusted Language Extensions für Postgre SQL
- Verwenden von PostgreSQL-Haken mit Ihren TLE-Erweiterungen
- Verwenden von benutzerdefinierten Datentypen in TLE
- Funktionsreferenz für Trusted Language Extensions for Postgre SQL
- Hakenreferenz für Trusted Language Extensions für PostgreSQL

Terminologie

Sehen Sie sich das folgende Glossar und die in diesem Thema verwendeten Begriffe an, damit Sie Trusted Language Extensions besser verstehen.

Vertrauenswürdige Spracherweiterungen für Postgre SQL

: Trusted Language Extensions for Postgre SQL ist der offizielle Name des Open-Source-Entwicklungskits, das als Erweiterung im `pg_tle` Paket enthalten ist. Es ist für die Verwendung auf jedem Postgre-System verfügbar. SQL Weitere Informationen finden Sie unter aws/pg_tle on. GitHub

Trusted Language Extensions

: Trusted Language Extensions ist der Kurzname für Trusted Language Extensions for Postgre. SQL Dieser verkürzte Name und seine Abkürzung (TLE) werden auch in dieser Dokumentation verwendet.

Vertrauenswürdige Sprache

Eine vertrauenswürdige Sprache ist eine Programmier- oder Skriptsprache mit bestimmten Sicherheitsattributen. Beispielsweise schränken vertrauenswürdige Sprachen in der Regel den Zugriff auf das Dateisystem und die Verwendung bestimmter Netzwerkeigenschaften ein. Das TLE Entwicklungskit wurde entwickelt, um vertrauenswürdige Sprachen zu unterstützen. Postgre SQL unterstützt mehrere verschiedene Sprachen, die zur Erstellung vertrauenswürdiger oder nicht vertrauenswürdiger Erweiterungen verwendet werden. Ein Beispiel finden Sie unter Trusted and Untrusted PL/Perl in der Postgre-Dokumentation. SQL Wenn Sie eine Erweiterung mithilfe von Trusted Language Extensions erstellen, verwendet die Erweiterung von sich aus vertrauenswürdige Sprachmechanismen.

TLEErweiterung

Eine TLEErweiterung ist eine SQL Postgre-Erweiterung, die mithilfe des Trusted Language Extensions (TLE) -Entwicklungskits erstellt wurde.

Anforderungen für die Verwendung von Trusted Language Extensions für Postgre SQL

Im Folgenden sind die Anforderungen für die Einrichtung und Verwendung des TLE Entwicklungskits aufgeführt.

- für SQL Postgre-Versionen — Trusted Language Extensions wird nur von für SQL Postgre-Versionen 13.12 und höher 13 Versionen, 14.5 und höher 14 Versionen und 15.2 und höhere Versionen unterstützt.
  - Upgrades der RDS for PostgreSQL-DB-Engine.
  - Wenn Sie noch keine Amazon RDS DB-Instance im haben, auf der Postgre ausgeführt wirdSQL, können Sie eine erstellen. Weitere Informationen finden Sie unter RDSInformationen zur SQL Postgre-DB-Instance finden Sie unter. Erstellen einer PostgreSQL-DB-Instance und Herstellen einer Verbindung
- Erfordert **`rds_superuser`**-Berechtigungen – Um die `pg_tle`-Erweiterung einzurichten und zu konfigurieren, muss Ihre Datenbankbenutzerrolle über die Berechtigungen der `rds_superuser`-Rolle verfügen. Standardmäßig wird diese Rolle dem `postgres` Benutzer zugewiesen, der den erstellt. RDSfür die SQL Postgre-DB-Instance.
- Erfordert eine benutzerdefinierte DB-Parametergruppe — Ihr für die SQL Postgre-DB-Instance muss mit einer benutzerdefinierten DB-Parametergruppe konfiguriert werden.

- Wenn Ihr für SQL Postgre-DB-Instance nicht mit einer benutzerdefinierten DB-Parametergruppe konfiguriert ist, sollten Sie eine erstellen und sie Ihre RDS für Postgre-DB-InstanceSQL. Eine kurze Zusammenfassung der Schritte finden Sie unter Erstellen und Anwenden einer benutzerdefinierten DB-Parametergruppe.
- Wenn Ihr für die SQL Postgre-DB-Instance bereits mit einer benutzerdefinierten DB-Parametergruppe konfiguriert ist, können Sie Trusted Language Extensions einrichten. Details hierzu finden Sie unter Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem .

Erstellen und Anwenden einer benutzerdefinierten DB-Parametergruppe

Gehen Sie wie folgt vor, um eine benutzerdefinierte DB-Parametergruppe zu erstellen und Ihren für die Verwendung durch die SQL Postgre-DB-Instance zu konfigurieren.

Konsole

Um eine benutzerdefinierte DB-Parametergruppe zu erstellen und sie mit Ihrem zu verwenden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im RDS Amazon-Menü die Option Parametergruppen aus.
3. Wählen Sie Parametergruppe erstellen.
4. Geben Sie auf der Seite Parameter group details Parametergruppendetails die folgenden Informationen ein.
   - Wählen Sie unter Parameter group family (Parametergruppenfamilie) die Option postgres14 aus.
   - Wählen Sie für Type (Typ) die Option DB Parameter Group (DB-Parametergruppe) aus.
   - Geben Sie Ihrer Parametergruppe unter Group name (Gruppenname) im Kontext Ihrer Operationen einen aussagekräftigen Namen.
   - Geben Sie unter Description (Beschreibung) eine nützliche Beschreibung ein, damit andere Mitglieder Ihres Teams sie leicht finden können.
5. Wählen Sie Create (Erstellen) aus. Ihre benutzerdefinierte DB-Parametergruppe wird in Ihrem erstellt AWS-Region. Sie können jetzt Ihren so ändern, dass er verwendet wird, indem Sie die nächsten Schritte ausführen.
6. Wählen Sie im RDS Amazon-Menü die Option Datenbanken aus.

7. Wählen Sie aus der Liste den für die SQL Postgre-DB-Instance TLE aus, mit dem Sie ihn verwenden möchten, und wählen Sie dann Modify aus.
8. Suchen Sie auf der Seite zum Suchen Sie auf der Seite zum Ändern der DB-Instance-Einstellungen im Abschnitt „Additional configuration“ (Zusätzliche Konfiguration) nach Database options (Datenbankoptionen) und wählen Sie Ihre benutzerdefinierte DB-Parametergruppe in der Auswahl aus.
9. Wählen Sie Continue (Weiter) aus, um die Änderung zu speichern.
10. Wählen Sie Sofort anwenden, damit Sie mit der Einrichtung des RDS fortfahren können, den die SQLPostgre-DB-Instance verwenden soll. TLE

Informationen zum weiteren Einrichten Ihres Systems für Trusted Language Extensions finden Sie unter Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem .

Weitere Informationen zum Arbeiten mit DB-Parametergruppen siehe DB-Parametergruppen für Amazon RDS Amazon.

AWS CLI

Sie können die Angabe des `--region` Arguments vermeiden, wenn Sie CLI Befehle verwenden, indem Sie Ihre AWS CLI mit Ihrer Standardeinstellung AWS-Region. Weitere Informationen finden Sie unter Grundlagen der Konfiguration im AWS Command Line Interface Benutzerleitfaden.

Um eine benutzerdefinierte DB-Parametergruppe zu erstellen und sie mit Ihrem zu verwenden

1. Verwenden Sie den create-db-parameter-group AWS CLI Befehl zum Erstellen einer benutzerdefinierten DB-Parametergruppe basierend auf für Ihr AWS-Region.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   aws rds create-db-parameter-group \
     --region aws-region \
     --db-parameter-group-name custom-params-for-pg-tle \
     --db-parameter-group-family postgres14 \
     --description "My custom DB parameter group for Trusted Language Extensions"
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   aws rds create-db-parameter-group ^
     --region aws-region ^
   ```

```
  --db-parameter-group-name custom-params-for-pg-tle ^
  --db-parameter-group-family postgres14 ^
  --description "My custom DB parameter group for Trusted Language Extensions"
```

Ihre benutzerdefinierte DB-Parametergruppe ist in Ihrem verfügbar AWS-Region, sodass Sie ändern können, um sie zu verwenden.

2. Verwenden Sie die modify-db-instance AWS CLI Befehl, um Ihre benutzerdefinierte DB-Parametergruppe auf anzuwenden. Ihre RDS für SQL Postgre-DB-Instance. Mit diesem Befehl wird die aktive Instanz sofort neu gestartet.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   aws rds modify-db-instance \
     --region aws-region \
     --db-instance-identifier your-instance-name \
     --db-parameter-group-name custom-params-for-pg-tle \
     --apply-immediately
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   aws rds modify-db-instance ^
     --region aws-region ^
     --db-instance-identifier your-instance-name ^
     --db-parameter-group-name custom-params-for-pg-tle ^
     --apply-immediately
   ```

Informationen zum weiteren Einrichten Ihres Systems für Trusted Language Extensions finden Sie unter Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem .

Weitere Informationen finden Sie unter Parametergruppen für Amazon RDS.

Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem

Bei den folgenden Schritten wird davon ausgegangen, dass Ihr für die SQL Postgre-DB-Instance einer benutzerdefinierten zugeordnet ist. Sie können das AWS Management Console oder die AWS CLI für diese Schritte.

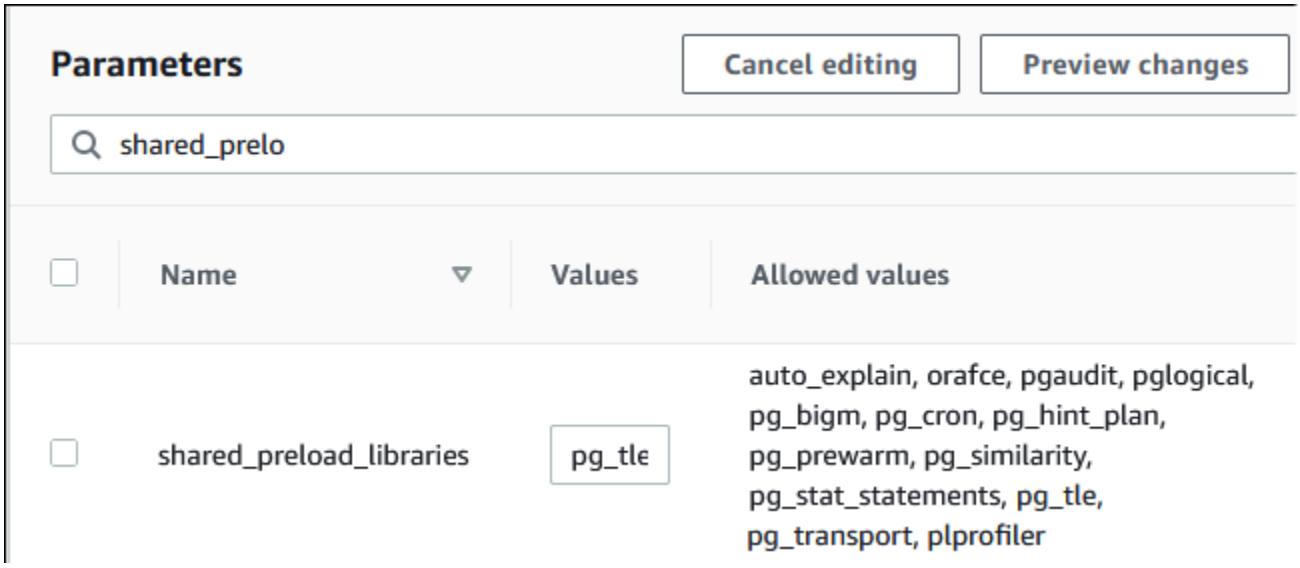
Wenn Sie Trusted Language Extensions in Ihrem für die SQL Postgre-DB-Instance einrichten, installieren Sie es in einer bestimmten Datenbank zur Verwendung durch die Datenbankbenutzer, die über Berechtigungen für diese Datenbank verfügen.

Konsole

So richten Sie Trusted Language Extensions ein

Führen Sie die folgenden Schritte mit einem Konto aus, das Mitglied der `rds_superuser`-Gruppe (Rolle) ist.

1. Melden Sie sich bei AWS Management Console und öffnen Sie die RDS Amazon-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die aus.
3. Öffnen Sie die Registerkarte Konfiguration für Ihre . RDSfür die SQL Postgre-DB-Instance. Suchen Sie in den Instance-Details den Link Parameter group (Parametergruppe).
4. Wählen Sie den Link, um die benutzerdefinierten Parameter zu öffnen, die mit Ihrem verknüpft sind. RDSfür die SQL Postgre-DB-Instance.
5. Geben Sie in das Suchfeld Parameters (Parameter) `shared_pre` ein, um den `shared_preload_libraries`-Parameter zu finden.
6. Wählen Sie Edit parameters (Parameter bearbeiten) aus, um auf die Eigenschaftswerte zuzugreifen.
7. Fügen Sie `pg_tle` der Liste im Feld Values (Werte) hinzu. Verwenden Sie ein Komma, um Elemente in der Werteliste zu trennen.



8. Starten Sie die für die SQL Postgre-DB-Instance neu, damit Ihre Änderung des `shared_preload_libraries` Parameters wirksam wird.
9. Wenn die Instance verfügbar ist, überprüfen Sie, ob `pg_tle` initialisiert wurde. Verwenden Sie`psql`, um eine Verbindung zur für SQL Postgre-DB-Instance herzustellen, und führen Sie dann den folgenden Befehl aus.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

10. Wenn die `pg_tle`-Erweiterung initialisiert ist, können Sie jetzt die Erweiterung erstellen.

```
CREATE EXTENSION pg_tle;
```

Sie können überprüfen, ob die Erweiterung installiert wurde, indem Sie den folgenden `psql`-Metabefehl verwenden.

```
labdb=> \dx
                          List of installed extensions
  Name   | Version |   Schema   |                Description
---------+---------+------------+-------------------------------------------
 pg_tle  | 1.0.1   | pgtle      | Trusted-Language Extensions for PostgreSQL
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
```

11. Weisen Sie die `pgtle_admin` Rolle dem primären Benutzernamen zu, den Sie bei der Einrichtung für Ihren für die SQL Postgre-DB-Instance erstellt haben. Wenn Sie die Standardeinstellung akzeptiert haben, lautet der Wert `postgres`.

```
labdb=> GRANT pgtle_admin TO postgres;
GRANT ROLE
```

Wie in folgendem Beispiel veranschaulicht, können Sie anhand des `psql`-Metabefehls überprüfen, ob die Gewährung erfolgt ist. In der Ausgabe werden nur die Rollen `pgtle_admin` und `postgres` angezeigt. Weitere Informationen finden Sie unter Die Rolle „rds_superuser“ verstehen.

```
labdb=> \du
```

```
                          List of roles
    Role name    |           Attributes            |              Member of
-----------------+---------------------------------
+---------------------------------
pgtle_admin     | Cannot login                     | {}
postgres        | Create role, Create DB          +| {rds_superuser,pgtle_admin}
                | Password valid until infinity    |...
```

12. Schließen Sie die `psql`-Sitzung mit dem `\q`-Metabefehl.

```
\q
```

Informationen zu den ersten Schritten beim Erstellen von TLE Erweiterungen finden Sie unter. Beispiel: Erstellen einer vertrauenswürdigen Spracherweiterung mit SQL

AWS CLI

Sie können die Angabe des `--region` Arguments bei der Verwendung von CLI Befehlen vermeiden, indem Sie AWS CLI mit Ihrer Standardeinstellung AWS-Region. Weitere Informationen finden Sie unter Grundlagen der Konfiguration im AWS Command Line Interface Benutzerleitfaden.

So richten Sie Trusted Language Extensions ein

1. Verwenden Sie die modify-db-parameter-group AWS CLI Befehl zum Hinzufügen `pg_tle` zum `shared_preload_libraries` Parameter.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pg_tle,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Verwenden Sie den reboot-db-instance AWS CLI Befehl, um die neu zu starten und die Bibliothek zu initialisieren. `pg_tle`

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Wenn die Instance verfügbar ist, können Sie überprüfen, ob `pg_tle` initialisiert wurde. Verwenden Sie`psql`, um eine Verbindung zur für SQL Postgre-DB-Instance herzustellen, und führen Sie dann den folgenden Befehl aus.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

   Sobald `pg_tle` initialisiert ist, können Sie die Erweiterung erstellen.

```
CREATE EXTENSION pg_tle;
```

4. Weisen Sie die `pgtle_admin` Rolle dem primären Benutzernamen zu, den Sie bei der Einrichtung für Ihren für die SQL Postgre-DB-Instance erstellt haben. Wenn Sie die Standardeinstellung akzeptiert haben, lautet der Wert `postgres`.

```
GRANT pgtle_admin TO postgres;
GRANT ROLE
```

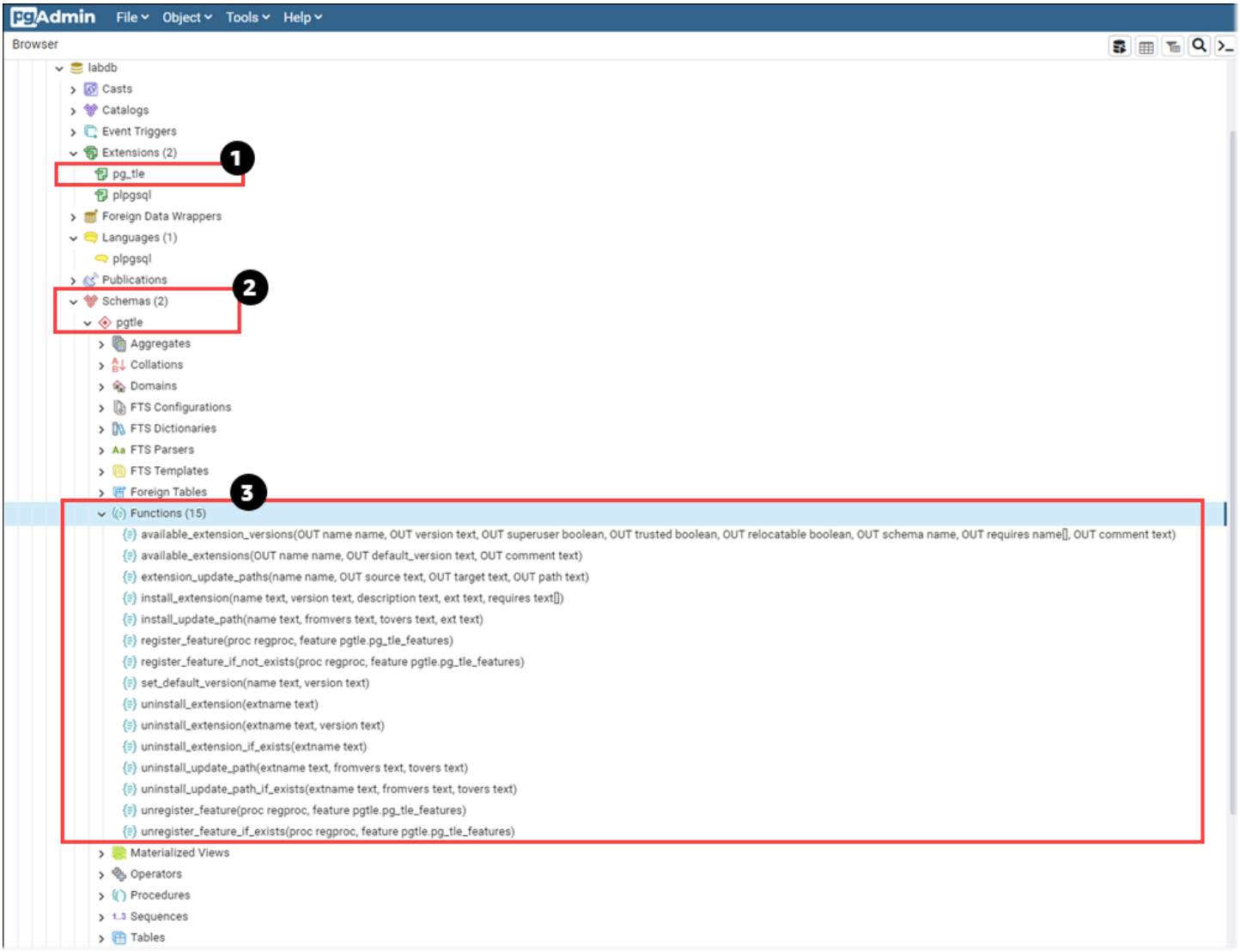
5. Schließen Sie die `psql`-Sitzung wie folgt.

```
labdb=> \q
```

Informationen zu den ersten Schritten beim Erstellen von TLE Erweiterungen finden Sie unter. [Beispiel: Erstellen einer vertrauenswürdigen Spracherweiterung mit SQL]

Überblick über vertrauenswürdige Spracherweiterungen für Postgre SQL

Trusted Language Extensions for Postgre SQL ist eine SQL Postgre-Erweiterung, die Sie in Ihrem auf die gleiche Weise installieren, wie Sie andere Postgre-Erweiterungen einrichten. SQL In der folgenden Abbildung einer Beispieldatenbank im pgAdmin Client-Tool können Sie sich einige der Komponenten ansehen, aus denen die Erweiterung besteht. `pg_tle`



Sie können die folgenden Details sehen.

1. Das SQL Entwicklungskit Trusted Language Extensions (TLE) für Postgre ist als `pg_tle` Erweiterung enthalten. Daher wird `pg_tle` den verfügbaren Erweiterungen für die Datenbank hinzugefügt, in der es installiert ist.
2. TLEhat sein eigenes Schema,`pgtle`. Dieses Schema enthält Hilfsfunktionen (3) für die Installation und Verwaltung der von Ihnen erstellten Erweiterungen.
3. TLEbietet über ein Dutzend Hilfsfunktionen für die Installation, Registrierung und Verwaltung Ihrer Erweiterungen. Weitere Informationen zu diesen Funktionen finden Sie unter Funktionsreferenz für Trusted Language Extensions for Postgre SQL.

Das `pg_tle`-Erweiterungspaket umfasst außerdem folgende Komponenten:

- Die **`pgtle_admin`**-Rolle – Die `pgtle_admin`-Rolle wird erstellt, wenn die `pg_tle`-Erweiterung installiert wird. Diese Rolle ist privilegiert und sollte entsprechend behandelt werden. Es wird dringend empfohlen, bei der Gewährung der `pgtle_admin`-Rolle an Datenbankbenutzer dem Prinzip der geringsten Berechtigung zu folgen. Mit anderen Worten, gewähren Sie die `pgtle_admin` Rolle nur Datenbankbenutzern, die berechtigt sind, neue TLE Erweiterungen zu erstellen, zu installieren und zu verwalten, wie `postgres` z.
- Die **`pgtle.feature_info`** Tabelle — Die `pgtle.feature_info` Tabelle ist eine geschützte Tabelle, die Informationen über Ihre TLEs Hooks und die von ihnen verwendeten benutzerdefinierten gespeicherten Prozeduren und Funktionen enthält. Wenn Sie über `pgtle_admin`-Berechtigungen verfügen, verwenden Sie die folgenden Funktionen von Trusted Language Extensions, um diese Informationen in der Tabelle hinzuzufügen und zu aktualisieren.
  - pgtle.register_feature
  - pgtle.register_feature_if_not_exists
  - pgtle.unregister_feature
  - pgtle.unregister_feature_if_exists

TLEErweiterungen für erstellen SQL

Sie können alle Erweiterungen, die Sie mit erstellen, TLE in jedem installieren, auf dem die `pg_tle` Erweiterung installiert ist. Die `pg_tle` Erweiterung ist auf die SQL Postgre-Datenbank beschränkt, in der sie installiert ist. Die Erweiterungen, mit denen Sie erstellen, TLE sind auf dieselbe Datenbank beschränkt.

Verwenden Sie die verschiedenen `pgtle` Funktionen, um den Code zu installieren, aus dem Ihre TLE Erweiterung besteht. Die folgenden Funktionen von Trusted Language Extensions erfordern alle die `pgtle_admin`-Rolle.

- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists
- pgtle.set_default_version
- pgtle.uninstall_extension(name)
- pgtle.uninstall_extension(name, version)

- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists
- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

Beispiel: Erstellen einer vertrauenswürdigen Spracherweiterung mit SQL

Das folgende Beispiel zeigt Ihnen, wie Sie eine TLE Erweiterung mit dem Namen erstellen`pg_distance`, die einige SQL Funktionen zur Berechnung von Entfernungen mithilfe verschiedener Formeln enthält. In der Liste finden Sie die Funktion zur Berechnung der Manhattan-Distanz und die Funktion zur Berechnung des euklidischen Abstands. Weitere Informationen zum Unterschied zwischen diesen Formeln finden Sie unter Taxi-Geometerie und Euklidische Geometrie in Wikipedia.

Sie können dieses Beispiel in Ihrem eigenen für die SQL Postgre-DB-Instance verwenden, wenn Sie die `pg_tle` Erweiterung wie unter beschrieben eingerichtet haben. Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem

> **Note**
>
> Sie benötigen die Rechte der `pgtle_admin`-Rolle, um dieses Verfahren ausführen zu können.

Um die Beispielerweiterung zu erstellen TLE

In den folgenden Schritten wird eine Beispieldatenbank namens `labdb` verwendet. Diese Datenbank gehört dem `postgres`-Hauptbenutzer. Die `postgres`-Rolle verfügt auch über die Berechtigungen der `pgtle_admin`-Rolle.

1. Wird verwendet`psql`, um eine Verbindung herzustellen. RDSfür die SQL Postgre-DB-Instance.

   ```
   psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
   --port=5432 --username=postgres --password --dbname=labdb
   ```

2. Erstellen Sie eine TLE Erweiterung mit dem Namen, `pg_distance` indem Sie den folgenden Code kopieren und in Ihre `psql` Sitzungskonsole einfügen.

```
SELECT pgtle.install_extension
(
 'pg_distance',
 '0.1',
  'Distance functions for two points',
$_pg_tle_$
    CREATE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8, norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL;

    CREATE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL;

    CREATE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL;
$_pg_tle_$
);
```

Die Ausgabe sollte folgendermaßen aussehen.

```
install_extension
---------------
 t
(1 row)
```

Die Artefakte, aus denen die `pg_distance`-Erweiterung besteht, sind jetzt in Ihrer Datenbank installiert. Zu diesen Artefakten gehören die Steuerdatei und der Code für die Erweiterung. Diese Elemente müssen vorhanden sein, damit die Erweiterung mit dem `CREATE EXTENSION`-Befehl erstellt werden kann. Mit anderen Worten, Sie müssen die Erweiterung nach wie vor erstellen, um ihre Funktionen Datenbankbenutzern zur Verfügung zu stellen.

3. Um die Erweiterung zu erstellen, verwenden Sie den `CREATE EXTENSION`-Befehl wie für jede andere Erweiterung. Wie bei anderen Erweiterungen muss der Datenbankbenutzer über die `CREATE`-Berechtigungen in der Datenbank verfügen.

```
CREATE EXTENSION pg_distance;
```

4. Um die `pg_distance` TLE Erweiterung zu testen, können Sie sie verwenden, um die Manhattan-Entfernung zwischen vier Punkten zu berechnen.

```
labdb=> SELECT manhattan_dist(1, 1, 5, 5);
8
```

Um den euklidischen Abstand zwischen derselben Menge von Punkten zu berechnen, können Sie Folgendes verwenden.

```
labdb=> SELECT euclidean_dist(1, 1, 5, 5);
5.656854249492381
```

Die `pg_distance`-Erweiterung lädt die Funktionen in die Datenbank und stellt sie allen Benutzern mit Berechtigungen für die Datenbank zur Verfügung.

Ändern Sie Ihre TLE Erweiterung

Um die Abfrageleistung für die in dieser TLE Erweiterung enthaltenen Funktionen zu verbessern, fügen Sie ihren Spezifikationen die folgenden beiden SQL Postgre-Attribute hinzu.

- `IMMUTABLE` – Das `IMMUTABLE`-Attribut stellt sicher, dass der Abfrageoptimierer Optimierungen verwenden kann, um die Antwortzeiten von Abfragen zu verbessern. Weitere Informationen finden Sie in der Postgre-Dokumentation unter Kategorien der Funktionsvolatilität. SQL
- `PARALLEL SAFE`— Das `PARALLEL SAFE` Attribut ist ein weiteres Attribut, das es Postgre SQL ermöglicht, die Funktion im Parallelmodus auszuführen. Weitere Informationen finden Sie CREATEFUNCTIONin der Postgre-Dokumentation. SQL

Im folgenden Beispiel können Sie sehen, wie die `pgtle.install_update_path` Funktion verwendet wird, um diese Attribute zu jeder Funktion hinzuzufügen, um eine Version `0.2` der `pg_distance` TLE Erweiterung zu erstellen. Weitere Informationen zu dieser Funktion finden

Sie unter pgtle.install_update_path. Sie benötigen die `pgtle_admin` Rolle, um diese Aufgabe auszuführen.

Um eine bestehende TLE Erweiterung zu aktualisieren und die Standardversion anzugeben

1. Connect für SQL Postgre-DB-Instance her, indem Sie `psql` oder ein anderes Client-Tool verwenden, z. pgAdmin

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. Ändern Sie die vorhandene TLE Erweiterung, indem Sie den folgenden Code kopieren und in Ihre `psql` Sitzungskonsole einfügen.

```
SELECT pgtle.install_update_path
(
 'pg_distance',
 '0.1',
 '0.2',
$_pg_tle_$
    CREATE OR REPLACE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8,
 norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;
$_pg_tle_$
);
```

Es wird eine Antwort ähnlich dem folgenden Beispiel angezeigt.

```
install_update_path
--------------------
 t
(1 row)
```

Sie können diese Version der Erweiterung zur Standardversion machen, sodass Datenbankbenutzer keine Version angeben müssen, wenn sie die Erweiterung in ihrer Datenbank erstellen oder aktualisieren.

3. Um anzugeben, dass die geänderte Version (Version 0.2) Ihrer TLE Erweiterung die Standardversion ist, verwenden Sie die `pgtle.set_default_version` Funktion wie im folgenden Beispiel gezeigt.

```
SELECT pgtle.set_default_version('pg_distance', '0.2');
```

Weitere Informationen zu dieser Funktion finden Sie unter pgtle.set_default_version.

4. Wenn der Code vorhanden ist, können Sie die installierte TLE Erweiterung auf die übliche Weise aktualisieren, indem Sie den `ALTER EXTENSION ... UPDATE` Befehl verwenden, wie hier gezeigt:

```
ALTER EXTENSION pg_distance UPDATE;
```

Löschen Ihrer TLE Erweiterungen aus einer Datenbank

Sie können Ihre TLE Erweiterungen löschen, indem Sie den `DROP EXTENSION` Befehl genauso verwenden wie bei anderen SQL Postgre-Erweiterungen. Durch das Löschen der Erweiterung werden die Installationsdateien, aus denen die Erweiterung besteht, nicht entfernt, sodass Benutzer die Erweiterung neu erstellen können. Gehen Sie wie folgt in zwei Schritten vor, um die Erweiterung und ihre Installationsdateien zu entfernen.

Um die TLE Erweiterung zu löschen und ihre Installationsdateien zu entfernen

1. Verwenden Sie `psql` oder ein anderes Client-Tool, um eine Verbindung zur herzustellen. RDSfür die SQL Postgre-DB-Instance.

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. Löschen Sie die Erweiterung wie jede andere SQL Postgre-Erweiterung.

```
DROP EXTENSION your-TLE-extension
```

Wenn Sie die `pg_distance`-Erweiterung beispielsweise wie in beschrieben Beispiel: Erstellen einer vertrauenswürdigen Spracherweiterung mit SQL erstellen, können Sie die Erweiterung wie folgt löschen.

```
DROP EXTENSION pg_distance;
```

Es wird eine Ausgabe angezeigt, die bestätigt, dass die Erweiterung gelöscht wurde, wie im Folgenden gezeigt.

```
DROP EXTENSION
```

Zu diesem Zeitpunkt ist die Erweiterung in der Datenbank nicht mehr aktiv. Die Installationsdateien und die Steuerdatei sind jedoch nach wie vor in der Datenbank verfügbar, sodass Datenbankbenutzer die Erweiterung erneut erstellen können, wenn sie möchten.

- Wenn Sie die Erweiterungsdateien unverändert lassen möchten, damit Datenbankbenutzer Ihre TLE Erweiterung erstellen können, können Sie hier aufhören.
- Wenn Sie alle Dateien, aus denen die Erweiterung besteht, entfernen möchten, fahren Sie mit dem nächsten Schritt fort.

3. Verwenden Sie die `pgtle.uninstall_extension`-Funktion, um alle Installationsdateien für Ihre Erweiterung zu entfernen. Diese Funktion entfernt dne gesamten Code und die Steuerdateien für Ihre Erweiterung.

```
SELECT pgtle.uninstall_extension('your-tle-extension-name');
```

Verwenden Sie beispielsweise den folgenden Befehl, um alle `pg_distance`-Installationsdateien zu entfernen.

```
SELECT pgtle.uninstall_extension('pg_distance');
```

```
 uninstall_extension
---------------------
 t
(1 row)
```

Deinstallation von Trusted Language Extensions für Postgre SQL

Wenn Sie damit keine eigenen TLE Erweiterungen mehr erstellen möchtenTLE, können Sie die `pg_tle` Erweiterung löschen und alle Artefakte entfernen. Diese Aktion beinhaltet das Löschen aller TLE Erweiterungen in der Datenbank und das Löschen des `pgtle` Schemas.

So löschen Sie die **`pg_tle`**-Erweiterung und ihr Schema aus einer Datenbank

1. Verwenden Sie `psql` oder ein anderes Client-Tool, um eine Verbindung zur herzustellen. RDSfür die SQL Postgre-DB-Instance.

   ```
   psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=dbname
   ```

2. Löschen Sie die `pg_tle`-Erweiterung aus der Datenbank. Wenn in der Datenbank Ihre eigenen TLE Erweiterungen noch in der Datenbank ausgeführt werden, müssen Sie diese Erweiterungen ebenfalls löschen. Dazu können Sie das `CASCADE`-Schlüsselwort verwenden, wie im Folgenden gezeigt.

   ```
   DROP EXTENSION pg_tle CASCADE;
   ```

   Wenn die `pg_tle`-Erweiterung in der Datenbank nicht mehr aktiv ist, müssen Sie das `CASCADE`-Schlüsselwort nicht verwenden.

3. Löschen Sie das `pgtle`-Schema. Mit dieser Aktion werden alle Verwaltungsfunktionen aus der Datenbank entfernt.

   ```
   DROP SCHEMA pgtle CASCADE;
   ```

   Der Befehl gibt nach Abschluss des Vorgangs Folgendes zurück.

   ```
   DROP SCHEMA
   ```

Die `pg_tle`-Erweiterung, ihr Schema und ihre Funktionen sowie alle Artefakte werden entfernt. Um neue Erweiterungen mit zu erstellenTLE, führen Sie den Einrichtungsvorgang erneut durch. Weitere Informationen finden Sie unter Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem .

Verwenden von PostgreSQL-Haken mit Ihren TLE-Erweiterungen

Ein Haken ist ein in PostgreSQL verfügbarer Callback-Mechanismus, der es Entwicklern ermöglicht, benutzerdefinierte Funktionen oder andere Routinen während regulärer Datenbankoperationen aufzurufen. Das TLE-Entwicklungskit unterstützt PostgreSQL-Haken, sodass Sie benutzerdefinierte Funktionen zur Laufzeit in das PostgreSQL-Verhalten integrieren können. Sie können beispielsweise einen Haken verwenden, um den Authentifizierungsprozess mit Ihrem eigenen benutzerdefinierten Code zu verknüpfen oder um den Planungs- und Ausführungsprozess für Abfragen Ihren spezifischen Bedürfnissen entsprechend anzupassen.

Ihre TLE-Erweiterungen können Haken verwenden. Wenn ein Haken einen globalen Gültigkeitsbereich hat, gilt er für alle Datenbanken. Wenn Ihre TLE-Erweiterung einen globalen Haken verwendet, müssen Sie Ihre TLE-Erweiterung daher in allen Datenbanken erstellen, auf die Ihre Benutzer zugreifen können.

Wenn Sie die `pg_tle`-Erweiterung verwenden, um Ihre eigenen Trusted Language Extensions zu erstellen, können Sie die verfügbaren Haken einer SQL-API verwenden, um die Funktionen Ihrer Erweiterung zu erstellen. Sie sollten alle Haken bei `pg_tle` registrieren. Für einige Haken müssen Sie möglicherweise auch verschiedene Konfigurationsparameter festlegen. Der `passcode`-Prüfungshaken kann beispielsweise auf ein, aus oder erforderlich festgelegt werden. Weitere Hinweise zu den spezifischen Anforderungen für verfügbare `pg_tle`-Haken finden Sie unter Hakenreferenz für Trusted Language Extensions für PostgreSQL.

Beispiel: Erstellen einer Erweiterung, die einen PostgreSQL-Haken verwendet

Das in diesem Abschnitt besprochene Beispiel verwendet einen PostgreSQL-Haken, um das bei bestimmten SQL-Vorgängen angegebene Passwort zu überprüfen, und verhindert, dass Datenbankbenutzer ihre Passwörter auf eines der in der `password_check.bad_passwords`-Tabelle enthaltenen Passwörter festlegen. Die Tabelle enthält die zehn am häufigsten verwendeten, aber leicht zu knackenden Optionen für Passwörter.

Um dieses Beispiel in Ihrer DB-Instance von RDS für PostgreSQL einzurichten, müssen Sie Trusted Language Extensions bereits installiert haben. Details hierzu finden Sie unter Einrichtung vertrauenswürdiger Spracherweiterungen in Ihrem .

So richten Sie das Beispiel für einen Haken für die Passwortüberprüfung ein

1. Verwenden Sie `psql`, um eine Verbindung mit der herzustellen. DB-Instance von RDS für PostgreSQL

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. Kopieren Sie den Code von Liste der Hakencodes für die Passwortüberprüfung und fügen Sie ihn in Ihre Datenbank ein.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;

  CREATE TABLE password_check.bad_passwords (plaintext) AS
  VALUES
    ('123456'),
    ('password'),
    ('12345678'),
    ('qwerty'),
    ('123456789'),
    ('12345'),
    ('1234'),
    ('111111'),
    ('1234567'),
    ('dragon');
  CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

  CREATE FUNCTION password_check.passcheck_hook(username text, password text,
 password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
  RETURNS void AS $$
    DECLARE
```

```
      invalid bool := false;
    BEGIN
      IF password_type = 'PASSWORD_TYPE_MD5' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE ('md5' || md5(bp.plaintext || username)) = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password
 dictionary';
        END IF;
      ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE bp.plaintext = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password
 dictionary';
        END IF;
      END IF;
    END
  $$ LANGUAGE plpgsql SECURITY DEFINER;

  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Wenn die Erweiterung in Ihre Datenbank geladen wurde, sehen Sie eine Ausgabe ähnlich wie die folgende.

```
 install_extension
-------------------
 t
(1 row)
```

3. Während Sie noch mit der Datenbank verbunden sind, können Sie jetzt die Erweiterung erstellen.

```
CREATE EXTENSION my_password_check_rules;
```

4. Mit dem folgenden `psql`-Metabefehl können Sie bestätigen, dass die Erweiterung in der Datenbank erstellt wurde.

```
\dx
                        List of installed extensions
          Name           | Version |   Schema   |
 Description
------------------------+---------+------------
+-------------------------------------------------------------
 my_password_check_rules | 1.0     | public     | Prevent use of any of the top-ten
 most common bad passwords
 pg_tle                  | 1.0.1   | pgtle      | Trusted-Language Extensions for
 PostgreSQL
 plpgsql                 | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

5. Öffnen Sie eine weitere Terminalsitzung, um mit dem zu arbeiten AWS CLI. Sie müssen Ihre benutzerdefinierte DB-Parametergruppe ändern, um den Haken für die Passwortüberprüfung zu aktivieren. Verwenden Sie dazu den modify-db-parameter-groupCLI-Befehl, wie im folgenden Beispiel gezeigt.

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Wenn der Parameter erfolgreich aktiviert wurde, wird eine Ausgabe wie die folgende angezeigt.

```
(
    "DBParameterGroupName": "docs-lab-parameters-for-tle"
}
```

Es kann einige Minuten dauern, bis die Änderung der Parametergruppe wirksam wird. Dieser Parameter ist jedoch dynamisch, sodass Sie die DB-Instance von RDS für PostgreSQL nicht neu starten müssen, damit die Einstellung wirksam wird.

6. Öffnen Sie die `psql`-Sitzung und fragen Sie die Datenbank ab, um zu überprüfen, ob der password_check-Haken aktiviert wurde.

```
labdb=> SHOW pgtle.enable_password_check;
pgtle.enable_password_check
----------------------------
on
(1 row)
```

Der Haken für die Passwortüberprüfung ist jetzt aktiv. Sie können dies testen, indem Sie eine neue Rolle erstellen und eines der schwachen Passwörter verwenden, wie im folgenden Beispiel gezeigt.

```
CREATE ROLE test_role PASSWORD 'password';
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 21 at RAISE
SQL statement "SELECT password_check.passcheck_hook(
    $1::pg_catalog.text,
    $2::pg_catalog.text,
    $3::pgtle.password_types,
    $4::pg_catalog.timestamptz,
    $5::pg_catalog.bool)"
```

Die Ausgabe wurde zur besseren Lesbarkeit formatiert.

Das folgende Beispiel zeigt, dass das interaktive `pgsql`-Metabefehlverhalten `\password` auch vom password_check-Haken beeinflusst wird.

```
postgres=> SET password_encryption TO 'md5';
SET
postgres=> \password
Enter new password for user "postgres":*****
Enter it again:*****
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 12 at RAISE
SQL statement "SELECT password_check.passcheck_hook($1::pg_catalog.text,
 $2::pg_catalog.text, $3::pgtle.password_types, $4::pg_catalog.timestamptz,
 $5::pg_catalog.bool)"
```

Sie können diese TLE-Erweiterung löschen und ihre Quelldateien deinstallieren, wenn Sie möchten. Weitere Informationen finden Sie unter Löschen Ihrer TLE Erweiterungen aus einer Datenbank.

Liste der Hakencodes für die Passwortüberprüfung

Der hier gezeigte Beispielcode definiert die Spezifikation für die `my_password_check_rules`-TLE-Erweiterung. Wenn Sie diesen Code kopieren und in Ihre Datenbank einfügen, wird der Code für die `my_password_check_rules`-Erweiterung in die Datenbank geladen und der `password_check`-Haken für die Verwendung durch die Erweiterung registriert.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;

  CREATE TABLE password_check.bad_passwords (plaintext) AS
  VALUES
    ('123456'),
    ('password'),
    ('12345678'),
    ('qwerty'),
    ('123456789'),
    ('12345'),
    ('1234'),
    ('111111'),
    ('1234567'),
    ('dragon');
  CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

  CREATE FUNCTION password_check.passcheck_hook(username text, password text,
 password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
  RETURNS void AS $$
    DECLARE
      invalid bool := false;
    BEGIN
      IF password_type = 'PASSWORD_TYPE_MD5' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
```

```
          WHERE ('md5' || md5(bp.plaintext || username)) = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
      ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE bp.plaintext = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
      END IF;
    END
  $$ LANGUAGE plpgsql SECURITY DEFINER;

  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Verwenden von benutzerdefinierten Datentypen in TLE

Postgre SQL unterstützt Befehle zur Registrierung neuer Basistypen (auch bekannt als Skalartypen), um komplexe Datenstrukturen in Ihrer Datenbank effizient handhaben zu können. Mit einem Basistyp können Sie anpassen, wie die Daten intern gespeichert werden und wie sie in und aus einer externen Textdarstellung konvertiert werden. Diese benutzerdefinierten Datentypen sind hilfreich, wenn Postgre erweitert wirdSQL, um funktionale Domänen zu unterstützen, in denen ein integrierter Typ wie Zahl oder Text keine ausreichende Suchsemantik bieten kann.

RDSfor Postgre SQL ermöglicht es Ihnen, benutzerdefinierte Datentypen in Ihrer vertrauenswürdigen Spracherweiterung zu erstellen und Funktionen zu definieren, die Operationen für diese neuen Datentypen unterstützen SQL und indexieren. Benutzerdefinierte Datentypen sind für die folgenden Versionen verfügbar:

- RDSfür Postgre SQL 15.4 und höhere Versionen (15)
- RDSfür Postgre SQL 14.9 und höhere 14-Versionen
- RDSfür Postgre SQL 13.12 und höhere 13-Versionen

Weitere Informationen finden Sie unter Trusted Language Base-Typen.

Funktionsreferenz für Trusted Language Extensions for Postgre SQL

Sehen Sie sich die folgende Referenzdokumentation zu den Funktionen an, die in Trusted Language Extensions for Postgre SQL verfügbar sind. Verwenden Sie diese Funktionen, um Ihre TLEErweiterungen, d. h. die Postgre-Erweiterungen, die Sie mit dem Trusted Language SQL Extensions Development Kit entwickeln, zu installieren, zu registrieren, zu aktualisieren und zu verwalten.

Funktionen

- pgtle.available_extensions
- pgtle.available_extension_versions
- pgtle.extension_update_paths
- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists
- pgtle.set_default_version
- pgtle.uninstall_extension(name)
- pgtle.uninstall_extension(name, version)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists
- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

pgtle.available_extensions

Die `pgtle.available_extensions`-Funktion ist eine Mengenrückgabefunktion. Sie gibt alle verfügbaren TLE Erweiterungen in der Datenbank zurück. Jede zurückgegebene Zeile enthält Informationen zu einer einzelnen TLE Erweiterung.

Funktionsprototyp

```
pgtle.available_extensions()
```

Rolle

Keine.

Argumente

Keine.

Output

- `name`— Der Name der TLE Erweiterung.
- `default_version`— Die Version der TLE Erweiterung, die verwendet werden `CREATE EXTENSION` soll, wenn sie ohne Angabe einer Version aufgerufen wird.
- `description`— Eine detailliertere Beschreibung der TLE Erweiterung.

Verwendungsbeispiel

```
SELECT * FROM pgtle.available_extensions();
```

pgtle.available_extension_versions

Die `available_extension_versions`-Funktion ist eine Mengenrückgabefunktion. Sie gibt eine Liste aller verfügbaren TLE Erweiterungen und ihrer Versionen zurück. Jede Zeile enthält Informationen zu einer bestimmten Version der angegebenen TLE Erweiterung, einschließlich der Frage, ob dafür eine bestimmte Rolle erforderlich ist.

Funktionsprototyp

```
pgtle.available_extension_versions()
```

Rolle

Keine.

Argumente

Keine.

Output

- `name`— Der Name der TLE Erweiterung.
- `version`— Die Version der TLE Erweiterung.
- `superuser`— Dieser Wert gilt immer `false` für Ihre TLE Erweiterungen. Die Berechtigungen, die zum Erstellen oder Aktualisieren der TLE Erweiterung erforderlich sind, sind dieselben wie für das Erstellen anderer Objekte in der angegebenen Datenbank.
- `trusted`— Dieser Wert gilt immer `false` für eine TLE Erweiterung.
- `relocatable`— Dieser Wert gilt immer `false` für eine TLE Erweiterung.
- `schema`— Gibt den Namen des Schemas an, in dem die TLE Erweiterung installiert ist.
- `requires`— Ein Array, das die Namen anderer Erweiterungen enthält, die von dieser TLE Erweiterung benötigt werden.
- `description`— Eine detaillierte Beschreibung der TLE Erweiterung.

Weitere Informationen zu Ausgabewerten finden Sie in der SQL Postgre-Dokumentation unter Zusammengehörige Objekte in eine Erweiterung verpacken > Erweiterungsdateien.

Verwendungsbeispiel

```
SELECT * FROM pgtle.available_extension_versions();
```

pgtle.extension_update_paths

Die `extension_update_paths`-Funktion ist eine Mengenrückgabefunktion. Sie gibt eine Liste aller möglichen Aktualisierungspfade für eine TLE Erweiterung zurück. Jede Zeile enthält die verfügbaren Upgrades oder Downgrades für diese TLE Erweiterung.

Funktionsprototyp

```
pgtle.extension_update_paths(name)
```

Rolle

Keine.

Argumente

`name`— Der Name der TLE Erweiterung, von der Upgrade-Pfade abgerufen werden sollen.

Output

- `source` – Die Quellversion für eine Aktualisierung.
- `target` – Die Zielversion für eine Aktualisierung.
- `path`— Der Upgrade-Pfad, der verwendet wird, um eine TLE Erweiterung von `source` `target` Version zu Version zu aktualisieren, `0.1--0.2` z. B.

Verwendungsbeispiel

```
SELECT * FROM pgtle.extension_update_paths('your-TLE');
```

pgtle.install_extension

Mit dieser `install_extension` Funktion können Sie die Artefakte, aus denen Ihre TLE Erweiterung besteht, in der Datenbank installieren. Danach kann sie mit dem `CREATE EXTENSION` Befehl erstellt werden.

Funktionsprototyp

```
pgtle.install_extension(name text, version text, description text, ext text, requires
 text[] DEFAULT NULL::text[])
```

Rolle

Keine.

Argumente

- `name`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.
- `version`— Die Version der TLE Erweiterung.
- `description`— Eine detaillierte Beschreibung der TLE Erweiterung. Diese Beschreibung wird im Feld `comment` in `pgtle.available_extensions()` angezeigt.
- `ext`— Der Inhalt der TLE Erweiterung. Dieser Wert enthält Objekte wie Funktionen.
- `requires`— Ein optionaler Parameter, der Abhängigkeiten für diese TLE Erweiterung angibt. Die `pg_tle`-Erweiterung wird automatisch als Abhängigkeit hinzugefügt.

Viele dieser Argumente sind dieselben wie diejenigen, die in einer Erweiterungssteuerungsdatei für die Installation einer SQL Postgre-Erweiterung im Dateisystem einer Postgre-Instanz enthalten sind. SQL Weitere Informationen finden Sie in der SQLPostgre-Dokumentation unter „Verwandte Objekte zu einer Erweiterung verpacken“ unter Erweiterungsdateien.

Output

Diese Funktion gibt bei Erfolg `OK` und bei Fehler `NULL` zurück.

- `OK`— Die TLE Erweiterung wurde erfolgreich in der Datenbank installiert.
- `NULL`— Die TLE Erweiterung wurde nicht erfolgreich in der Datenbank installiert.

Verwendungsbeispiel

```
SELECT pgtle.install_extension(
 'pg_tle_test',
 '0.1',
 'My first pg_tle extension',
$_pgtle_$
  CREATE FUNCTION my_test()
  RETURNS INT
  AS $$
    SELECT 42;
  $$ LANGUAGE SQL IMMUTABLE;
$_pgtle_$
);
```

pgtle.install_update_path

Die `install_update_path` Funktion stellt einen Aktualisierungspfad zwischen zwei verschiedenen Versionen einer TLE Erweiterung bereit. Diese Funktion ermöglicht es Benutzern Ihrer TLE Erweiterung, ihre Version mithilfe der `ALTER EXTENSION ... UPDATE` Syntax zu aktualisieren.

Funktionsprototyp

```
pgtle.install_update_path(name text, fromvers text, tovers text, ext text)
```

Rolle

`pgtle_admin`

Argumente

- `name`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.
- `fromvers`— Die Quellversion der TLE Erweiterung für das Upgrade.
- `tovers`— Die Zielversion der TLE Erweiterung für das Upgrade.
- `ext` – Der Inhalt der Aktualisierung. Dieser Wert enthält Objekte wie Funktionen.

Output

Keine.

Verwendungsbeispiel

```
SELECT pgtle.install_update_path('pg_tle_test', '0.1', '0.2',
  $_pgtle_$
    CREATE OR REPLACE FUNCTION my_test()
    RETURNS INT
    AS $$
      SELECT 21;
    $$ LANGUAGE SQL IMMUTABLE;
  $_pgtle_$
);
```

pgtle.register_feature

Die `register_feature` Funktion fügt der Tabelle das angegebene interne SQL Postgre-Feature hinzu. `pgtle.feature_info` SQLPostgre-Hooks sind ein Beispiel für eine interne SQL Postgre-Funktion. Das Trusted Language Extensions Development Kit unterstützt die Verwendung von SQL Postgre-Hooks. Derzeit unterstützt diese Funktion das folgende Feature.

- `passcheck`— Registriert den Passwort-Check-Hook bei Ihrer Prozedur oder Funktion, die das Verhalten von Postgre bei der Passwortprüfung anpasst. SQL

Funktionsprototyp

```
pgtle.register_feature(proc regproc, feature pg_tle_feature)
```

Rolle

`pgtle_admin`

Argumente

- `proc` – Der Name einer gespeicherten Prozedur oder Funktion, die für das Feature verwendet werden soll.
- `feature` – Der Name des `pg_tle`-Features (z. B. `passcheck`), das für die Funktion registriert werden soll.

Output

Keine.

Verwendungsbeispiel

```
SELECT pgtle.register_feature('pw_hook', 'passcheck');
```

pgtle.register_feature_if_not_exists

Die `pgtle.register_feature_if_not_exists` Funktion fügt der `pgtle.feature_info` Tabelle das angegebene SQL Postgre-Feature hinzu und identifiziert die TLE Erweiterung oder andere Prozedur oder Funktion, die das Feature verwendet. Weitere Informationen zu Haken und Trusted Language Extensions finden Sie unter Verwenden von PostgreSQL-Haken mit Ihren TLE-Erweiterungen.

Funktionsprototyp

```
pgtle.register_feature_if_not_exists(proc regproc, feature pg_tle_feature)
```

Rolle

`pgtle_admin`

Argumente

- `proc`— Der Name einer gespeicherten Prozedur oder Funktion, die die Logik (Code) enthält, die als Feature für Ihre TLE Erweiterung verwendet werden soll. Beispielsweise der `pw_hook`-Code.

- `feature`— Der Name der Postgre-Funktion, die für die SQL TLE Funktion registriert werden soll. Derzeit ist der `passcheck`-Haken das einzige verfügbare Feature. Weitere Informationen finden Sie unter Haken zur Passwortüberprüfung (Passcheck).

Output

Gibt `true` zurück, nachdem das Feature für die angegebene Erweiterung registriert wurde. Gibt `false` zurück, wenn das Feature bereits registriert ist.

Verwendungsbeispiel

```
SELECT pgtle.register_feature_if_not_exists('pw_hook', 'passcheck');
```

pgtle.set_default_version

Mit `set_default_version` dieser Funktion können Sie a `default_version` für Ihre TLE Erweiterung angeben. Sie können diese Funktion verwenden, um einen Upgrade-Pfad zu definieren und die Version als Standard für Ihre TLE Erweiterung festzulegen. Wenn Datenbankbenutzer Ihre TLE Erweiterung in den `ALTER EXTENSION ... UPDATE` Befehlen `CREATE EXTENSION` und angeben, wird diese Version Ihrer TLE Erweiterung in der Datenbank für diesen Benutzer erstellt.

Diese Funktion gibt bei Erfolg `true` zurück. Wenn die im `name` Argument angegebene TLE Erweiterung nicht existiert, gibt die Funktion einen Fehler zurück. In ähnlicher Weise gibt sie einen Fehler zurück, wenn die TLE Erweiterung nicht existiert. `version`

Funktionsprototyp

```
pgtle.set_default_version(name text, version text)
```

Rolle

`pgtle_admin`

Argumente

- `name`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.
- `version`— Die Version der TLE Erweiterung, die als Standard festgelegt werden soll.

Output

- `true`– Wenn die Standardversion erfolgreich festgelegt wurde, gibt die Funktion `true` zurück.
- `ERROR`— Gibt eine Fehlermeldung zurück, wenn eine TLE Erweiterung mit dem angegebenen Namen oder der angegebenen Version nicht existiert.

Verwendungsbeispiel

```
SELECT * FROM pgtle.set_default_version('my-extension', '1.1');
```

pgtle.uninstall_extension(name)

Die `uninstall_extension` Funktion entfernt alle Versionen einer TLE Erweiterung aus einer Datenbank. Diese Funktion verhindert, dass future `CREATE EXTENSION` Aufrufe die TLE Erweiterung installieren. Wenn die TLE Erweiterung nicht in der Datenbank existiert, wird ein Fehler ausgelöst.

Die `uninstall_extension` Funktion löscht keine TLE Erweiterung, die derzeit in der Datenbank aktiv ist. Um eine TLE Erweiterung zu entfernen, die derzeit aktiv ist, müssen Sie sie explizit aufrufen, `DROP EXTENSION` um sie zu entfernen.

Funktionsprototyp

```
pgtle.uninstall_extension(extname text)
```

Rolle

`pgtle_admin`

Argumente

- `extname`— Der Name der zu deinstallierenden TLE Erweiterung. Dieser Name ist derselbe, der verwendet wurde`CREATE EXTENSION`, um die TLE Erweiterung für die Verwendung in einer bestimmten Datenbank zu laden.

Output

Keine.

Verwendungsbeispiel

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test');
```

pgtle.uninstall_extension(name, version)

Die `uninstall_extension(name, version)` Funktion entfernt die angegebene Version der TLE Erweiterung aus der Datenbank. Diese Funktion verhindert`CREATE EXTENSION`, dass eine TLE Erweiterung installiert oder auf die angegebene Version aktualisiert wird. `ALTER EXTENSION` Diese Funktion entfernt auch alle Aktualisierungspfade für die angegebene Version der TLE Erweiterung. Diese Funktion deinstalliert die TLE Erweiterung nicht, wenn sie derzeit in der Datenbank aktiv ist. Sie müssen die TLE Erweiterung explizit aufrufen`DROP EXTENSION`, um sie zu entfernen. Informationen zum Deinstallieren aller Versionen einer TLE Erweiterung finden Sie unterpgtle.uninstall_extension(name).

Funktionsprototyp

```
pgtle.uninstall_extension(extname text, version text)
```

Rolle

`pgtle_admin`

Argumente

- `extname`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.
- `version`— Die Version der TLE Erweiterung, die aus der Datenbank deinstalliert werden soll.

Output

Keine.

Verwendungsbeispiel

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test', '0.2');
```

pgtle.uninstall_extension_if_exists

Die `uninstall_extension_if_exists` Funktion entfernt alle Versionen einer TLE Erweiterung aus einer bestimmten Datenbank. Wenn die TLE Erweiterung nicht existiert, kehrt die Funktion im Hintergrund zurück (es wird keine Fehlermeldung ausgegeben). Wenn die angegebene Erweiterung derzeit in einer Datenbank aktiv ist, wird sie von dieser Funktion nicht gelöscht. Sie müssen die TLE Erweiterung explizit aufrufen`DROP EXTENSION`, um sie zu entfernen, bevor Sie diese Funktion zur Deinstallation ihrer Artefakte verwenden können.

Funktionsprototyp

```
pgtle.uninstall_extension_if_exists(extname text)
```

Rolle

`pgtle_admin`

Argumente

- `extname`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.

Output

Die `uninstall_extension_if_exists`-Funktion gibt `true` nach der Deinstallation der angegebenen Erweiterung zurück. Wenn die angegebene Erweiterung nicht vorhanden ist, gibt die Funktion `false` zurück.

- `true`— Kehrt `true` nach der Deinstallation der TLE Erweiterung zurück.
- `false`— Gibt zurück`false`, wenn die TLE Erweiterung nicht in der Datenbank existiert.

Verwendungsbeispiel

```
SELECT * FROM pgtle.uninstall_extension_if_exists('pg_tle_test');
```

pgtle.uninstall_update_path

Die `uninstall_update_path` Funktion entfernt den spezifischen Aktualisierungspfad aus einer TLE Erweiterung. Dadurch wird verhindert, dass `ALTER EXTENSION ... UPDATE TO` diesen Pfad als Aktualisierungspfad verwendet.

Wenn die TLE Erweiterung derzeit von einer der Versionen auf diesem Aktualisierungspfad verwendet wird, verbleibt sie in der Datenbank.

Wenn der angegebene Aktualisierungspfad nicht vorhanden ist, gibt diese Funktion einen Fehler aus.

Funktionsprototyp

```
pgtle.uninstall_update_path(extname text, fromvers text, tovers text)
```

Rolle

`pgtle_admin`

Argumente

- `extname`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.
- `fromvers`— Die Quellversion der TLE Erweiterung, die im Aktualisierungspfad verwendet wird.
- `tovers`— Die Zielversion der TLE Erweiterung, die im Aktualisierungspfad verwendet wird.

Output

Keine.

Verwendungsbeispiel

```
SELECT * FROM pgtle.uninstall_update_path('pg_tle_test', '0.1', '0.2');
```

pgtle.uninstall_update_path_if_exists

Die `uninstall_update_path_if_exists` Funktion ähnelt `uninstall_update_path` insofern, als sie den angegebenen Aktualisierungspfad aus einer TLE Erweiterung entfernt. Wenn der Aktualisierungspfad jedoch nicht existiert, löst diese Funktion keine Fehlermeldung aus. Stattdessen gibt die Funktion `false` zurück.

Funktionsprototyp

```
pgtle.uninstall_update_path_if_exists(extname text, fromvers text, tovers text)
```

Rolle

`pgtle_admin`

Argumente

- `extname`— Der Name der TLE Erweiterung. Dieser Wert wird beim Aufrufen von `CREATE EXTENSION` verwendet.
- `fromvers`— Die Quellversion der TLE Erweiterung, die im Aktualisierungspfad verwendet wird.
- `tovers`— Die Zielversion der TLE Erweiterung, die im Aktualisierungspfad verwendet wird.

Output

- `true`— Die Funktion hat den Pfad für die TLE Erweiterung erfolgreich aktualisiert.
- `false`— Die Funktion konnte den Pfad für die TLE Erweiterung nicht aktualisieren.

Verwendungsbeispiel

```
SELECT * FROM pgtle.uninstall_update_path_if_exists('pg_tle_test', '0.1', '0.2');
```

pgtle.unregister_feature

Die `unregister_feature`-Funktion bietet eine Möglichkeit, Funktionen zu entfernen, die für die Verwendung von `pg_tle`-Features wurden, wie z. B. Haken. Weitere Informationen zur Registrierung eines Features erhalten Sie unter pgtle.register_feature.

Funktionsprototyp

```
pgtle.unregister_feature(proc regproc, feature pg_tle_features)
```

Rolle

`pgtle_admin`

Argumente

- `proc` – Der Name einer gespeicherten Funktion, für die ein `pg_tle`-Feature registriert werden soll.
- `feature` – Der Name des `pg_tle`-Features, das für die Funktion registriert werden soll. Beispielsweise ist `passcheck` ein Feature, das für die Verwendung durch die vertrauenswürdigen Spracherweiterungen, die Sie entwickeln, registriert werden kann. Weitere Informationen finden Sie unter Haken zur Passwortüberprüfung (Passcheck).

Output

Keine.

Verwendungsbeispiel

```
SELECT * FROM pgtle.unregister_feature('pw_hook', 'passcheck');
```

pgtle.unregister_feature_if_exists

Die `unregister_feature`-Funktion bietet eine Möglichkeit, Funktionen zu entfernen, die für die Verwendung von `pg_tle`-Features wurden, wie z. B. Haken. Weitere Informationen finden Sie unter Verwenden von PostgreSQL-Haken mit Ihren TLE-Erweiterungen. Gibt `true` zurück, nachdem die Registrierung der Funktion erfolgreich aufgehoben wurde. Gibt `false` zurück, wenn das Feature nicht registriert wurde.

Hinweise zur Registrierung von `pg_tle` Funktionen für Ihre TLE Erweiterungen finden Sie unterpgtle.register_feature.

Funktionsprototyp

```
pgtle.unregister_feature_if_exists('proc regproc', 'feature pg_tle_features')
```

Rolle

`pgtle_admin`

Argumente

- `proc` – Der Name der gespeicherten Funktion, für die ein `pg_tle`-Feature registriert wurde.

- `feature` – Der Name des `pg_tle`-Features, das mit der vertrauenswürdigen Spracherweiterung registriert wurde.

Output

Gibt `true` oder `false` wie folgt zurück.

- `true` – Die Funktion hat die Registrierung des Features in der Erweiterung erfolgreich aufgehoben.
- `false`— Die Funktion konnte die Registrierung der Funktion für die TLE Erweiterung nicht aufheben.

Verwendungsbeispiel

```
SELECT * FROM pgtle.unregister_feature_if_exists('pw_hook', 'passcheck');
```

Hakenreferenz für Trusted Language Extensions für PostgreSQL

Trusted Language Extensions für PostgreSQL unterstützt PostgreSQL-Haken. Ein Haken ist ein interner Callback-Mechanismus, der Entwicklern zur Erweiterung der Kernfunktionalität von PostgreSQL zur Verfügung steht. Durch die Verwendung von Haken können Entwickler ihre eigenen Funktionen oder Verfahren zur Verwendung bei verschiedenen Datenbankoperationen implementieren und so das Verhalten von PostgreSQL in gewisser Weise ändern. Sie können beispielsweise einen `passcheck`-Haken verwenden, um anzupassen, wie PostgreSQL die Passwörter behandelt, die beim Erstellen oder Ändern von Passwörtern für Benutzer (Rollen) angegeben werden.

In der folgenden Dokumentation erfahren Sie mehr über den Passcheck-Hook, der für Ihre TLE-Erweiterungen verfügbar ist. Weitere Informationen zu den verfügbaren Hooks, einschließlich des Client-Authentifizierungs-Hooks, finden Sie unter Trusted Language Extensions-Hooks.

Haken zur Passwortüberprüfung (Passcheck)

Der `passcheck`-Haken wird verwendet, um das PostgreSQL-Verhalten während der Passwortüberprüfung für die folgenden SQL-Befehle und `psql`-Metabefehle anzupassen.

- `CREATE ROLE` *`username`* `...PASSWORD` – Weitere Informationen finden Sie im Abschnitt CREATE ROLE der PostgreSQL-Dokumentation.

- `ALTER ROLE username...PASSWORD` – Weitere Informationen finden Sie im Abschnitt ALTER ROLE der PostgreSQL-Dokumentation.
- `\password username` – Dieser interaktive `psql`-Metabefehl ändert das Passwort für den angegebenen Benutzer auf sichere Weise, indem er das Passwort hasht, bevor die `ALTER ROLE ... PASSWORD`-Syntax transparent verwendet wird. Der Metabefehl ist ein sicherer Wrapper für den `ALTER ROLE ... PASSWORD`-Befehl, daher gilt der Haken für das Verhalten des`psql`-Metabefehls.

Ein Beispiel finden Sie unter Liste der Hakencodes für die Passwortüberprüfung.

Inhalt

- Funktionsprototyp
- Argumente
- Konfiguration
- Nutzungshinweise

Funktionsprototyp

```
passcheck_hook(username text, password text, password_type pgtle.password_types,
 valid_until timestamptz, valid_null boolean)
```

Argumente

Eine `passcheck`-Hakenfunktion verwendet die folgenden Argumente.

- `username` – Der Name (als Text) der Rolle (Benutzername), die ein Passwort festlegt.
- `password` – Das Klartext- oder Hash-Passwort. Das eingegebene Passwort sollte dem im `password_type` angegebenen Typ entsprechen.
- `password_type` – Geben Sie das `pgtle.password_type`-Format des Passworts an. Dieses Format kann einer der folgenden Optionen entsprechen.
  - `PASSWORD_TYPE_PLAINTEXT` – Ein Klartext-Passwort.
  - `PASSWORD_TYPE_MD5`— Ein Passwort, das mithilfe des Algorithmus MD5 (Message Digest 5) gehasht wurde.
  - `PASSWORD_TYPE_SCRAM_SHA_256` – Ein Passwort, das mit dem SCRAM-SHA-256-Algorithmus gehasht wurde.

- `valid_until` – Geben Sie den Zeitpunkt an, am dem das Passwort ungültig wird. Dieses Argument ist optional. Wenn Sie dieses Argument verwenden, geben Sie die Uhrzeit als `timestamptz`-Wert an.
- `valid_null` – Wenn dieser boolesche Wert auf `true` festgelegt ist, wird die `valid_until`-Option auf `NULL` eingestellt.

Konfiguration

Die Funktion `pgtle.enable_password_check` steuert, ob der Passcheck-Haken aktiv ist. Der Passcheck-Haken hat drei mögliche Einstellungen.

- `off` – Schaltet den `passcheck`-Haken für die Passwortüberprüfung aus. Dies ist der Standardwert.
- on – Aktiviert den `passcode`-Haken für die Passwortüberprüfung, sodass Passwörter mit der Tabelle verglichen werden.
- `require` – Erfordert, dass ein Passwortüberprüfungshaken definiert wird.

Nutzungshinweise

Um den `passcheck`-Haken ein- oder auszuschalten, müssen Sie die benutzerdefinierte DB-Parametergruppe für Ihre DB-Instance von RDS für PostgreSQL ändern.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
    --region aws-region ^
    --db-parameter-group-name your-custom-parameter-group ^
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Codebeispiele für Amazon RDS mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Amazon RDS mit einem AWS Software Development Kit (SDK) verwendet wird.

Bei Grundlagen handelt es sich um Code-Beispiele, die Ihnen zeigen, wie Sie die wesentlichen Vorgänge innerhalb eines Services ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Service-Funktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarios anzeigen.

Szenarien sind Code-Beispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Services aufrufen oder mit anderen AWS-Services kombinieren.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte

Hello Amazon RDS

Die folgenden Codebeispiele veranschaulichen die ersten Schritte mit Amazon RDS.

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.RDS;
using Amazon.RDS.Model;

namespace RDSActions;
```

```
public static class HelloRds
{
    static async Task Main(string[] args)
    {
        var rdsClient = new AmazonRDSClient();

        Console.WriteLine($"Hello Amazon RDS! Following are some of your DB
 instances:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first twenty DB instances.
        var response = await rdsClient.DescribeDBInstancesAsync(
            new DescribeDBInstancesRequest()
            {
                MaxRecords = 20 // Must be between 20 and 100.
            });

        foreach (var instance in response.DBInstances)
        {
            Console.WriteLine($"\tDB name: {instance.DBName}");
            Console.WriteLine($"\tArn: {instance.DBInstanceArn}");
            Console.WriteLine($"\tIdentifier: {instance.DBInstanceIdentifier}");
            Console.WriteLine();
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Code für die CMake Datei CMake Lists.txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS rds)

# Set this project's name.
project("hello_rds")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

     # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
 may need to uncomment this
                                 # and set the proper subdirectory to the
 executables' location.

     AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_rds.cpp)
```

```
target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Code für die Quelldatei „hello_rds.cpp“.

```
#include <aws/core/Aws.h>
#include <aws/rds/RDSClient.h>
#include <aws/rds/model/DescribeDBInstancesRequest.h>
#include <iostream>

/*
 *  A "Hello Rds" starter application which initializes an Amazon Relational
 Database Service (Amazon RDS) client and
 *  describes the Amazon RDS instances.
 *
 *  main function
 *
 *  Usage: 'hello_rds'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::RDS::RDSClient rdsClient(clientConfig);
        Aws::String marker;
        std::vector<Aws::String> instanceDBIDs;

        do {
            Aws::RDS::Model::DescribeDBInstancesRequest request;

            if (!marker.empty()) {
                request.SetMarker(marker);
            }
```

```
            Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
                    rdsClient.DescribeDBInstances(request);

            if (outcome.IsSuccess()) {
                for (auto &instance: outcome.GetResult().GetDBInstances()) {
                    instanceDBIDs.push_back(instance.GetDBInstanceIdentifier());
                }
                marker = outcome.GetResult().GetMarker();
            } else {
                result = 1;
                std::cerr << "Error with RDS::DescribeDBInstances. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!marker.empty());

        std::cout << instanceDBIDs.size() << " RDS instances found." <<
 std::endl;
        for (auto &instanceDBID: instanceDBIDs) {
            std::cout << "   Instance: " << instanceDBID << std::endl;
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für C++ API-Referenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/rds"
)

// main uses the AWS SDK for Go V2 to create an Amazon Relational Database
 Service (Amazon RDS)
// client and list up to 20 DB instances in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 rdsClient := rds.NewFromConfig(sdkConfig)
 const maxInstances = 20
 fmt.Printf("Let's list up to %v DB instances.\n", maxInstances)
 output, err := rdsClient.DescribeDBInstances(ctx,
  &rds.DescribeDBInstancesInput{MaxRecords: aws.Int32(maxInstances)})
 if err != nil {
  fmt.Printf("Couldn't list DB instances: %v\n", err)
  return
 }
 if len(output.DBInstances) == 0 {
  fmt.Println("No DB instances found.")
 } else {
  for _, instance := range output.DBInstances {
   fmt.Printf("DB instance %v has database %v.\n",
 *instance.DBInstanceIdentifier,
    *instance.DBName)
  }
```

```
 }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
```

```
        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
"""
Purpose
```

```
Shows how to use the AWS SDK for Python (Boto3) with the Amazon Relational
 Database Service
(Amazon RDS) to list the databases in your account.
"""

import boto3
from botocore.exceptions import ClientError

# Create an RDS client
rds_client = boto3.client("rds")

# Create a paginator for the describe_db_instances operation
paginator = rds_client.get_paginator("describe_db_instances")

try:
    # Use the paginator to get a list of DB instances
    response_iterator = paginator.paginate(
        PaginationConfig={
            "MaxItems": 123,
            "PageSize": 50,  # Adjust PageSize as needed
            "StartingToken": None,
        }
    )

    # Iterate through the pages of the response
    instances_found = False
    for page in response_iterator:
        if "DBInstances" in page and page["DBInstances"]:
            instances_found = True
            print("Your RDS instances are:")
            for db in page["DBInstances"]:
                print(db["DBInstanceIdentifier"])

    if not instances_found:
        print("No RDS instances found!")

except ClientError as e:
    print(f"Couldn't list RDS instances. Here's why: {e.response['Error']
['Message']}")
```

- Einzelheiten zur API finden Sie unter DBInstancesDescribe in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds'
require 'logger'

# RDSManager is a class responsible for managing RDS operations
# such as listing all RDS DB instances in the current AWS account.
class RDSManager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all RDS DB instances in the current AWS account.
  def list_db_instances
    @logger.info('Listing RDS DB instances')

    paginator = @client.describe_db_instances
    instances = []

    paginator.each_page do |page|
      instances.concat(page.db_instances)
    end

    if instances.empty?
      @logger.info('No instances found.')
    else
      @logger.info("Found #{instances.count} instance(s):")
      instances.each do |instance|
```

```
        @logger.info(" * #{instance.db_instance_identifier}
 (#{instance.db_instance_status})")
      end
    end
  end
end

if $PROGRAM_NAME == __FILE__
  rds_client = Aws::RDS::Client.new(region: 'us-west-2')
  manager = RDSManager.new(rds_client)
  manager.list_db_instances
end
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für Ruby API-Referenz.

Codebeispiele

- Grundlegende Beispiele für die Verwendung von Amazon RDS AWS SDKs
  - Hello Amazon RDS
  - Lernen Sie die Grundlagen von Amazon RDS mit einem AWS SDK kennen
  - Aktionen für Amazon RDS mit AWS SDKs
    - Verwendung CreateDBInstance mit einem AWS SDK oder CLI
    - Verwendung CreateDBParameterGroup mit einem AWS SDK oder CLI
    - Verwendung CreateDBSnapshot mit einem AWS SDK oder CLI
    - Verwendung DeleteDBInstance mit einem AWS SDK oder CLI
    - Verwendung DeleteDBParameterGroup mit einem AWS SDK oder CLI
    - Verwendung DescribeAccountAttributes mit einem AWS SDK oder CLI
    - Verwendung DescribeDBEngineVersions mit einem AWS SDK oder CLI
    - Verwendung DescribeDBInstances mit einem AWS SDK oder CLI
    - Verwendung DescribeDBParameterGroups mit einem AWS SDK oder CLI
    - Verwendung DescribeDBParameters mit einem AWS SDK oder CLI
    - Verwendung DescribeDBSnapshots mit einem AWS SDK oder CLI
    - Verwendung DescribeOrderableDBInstanceOptions mit einem AWS SDK oder CLI

    - Verwendung GenerateRDSAuthToken mit einem AWS SDK
    - Verwendung ModifyDBInstance mit einem AWS SDK oder CLI
    - Verwendung ModifyDBParameterGroup mit einem AWS SDK oder CLI
    - Verwendung RebootDBInstance mit einem AWS SDK oder CLI
- Szenarien für die Verwendung von Amazon RDS AWS SDKs
    - Erstellen eines Trackers für Aurora-Serverless-Arbeitsaufgaben
- Serverlose Beispiele für Amazon RDS
    - Herstellen einer Verbindung mit einer Amazon-RDS-Datenbank in einer Lambda-Funktion

Grundlegende Beispiele für die Verwendung von Amazon RDS AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie die Grundlagen von Amazon Relational Database Service mit AWS SDKs verwenden können.

Beispiele

- Hello Amazon RDS
- Lernen Sie die Grundlagen von Amazon RDS mit einem AWS SDK kennen
- Aktionen für Amazon RDS mit AWS SDKs
    - Verwendung CreateDBInstance mit einem AWS SDK oder CLI
    - Verwendung CreateDBParameterGroup mit einem AWS SDK oder CLI
    - Verwendung CreateDBSnapshot mit einem AWS SDK oder CLI
    - Verwendung DeleteDBInstance mit einem AWS SDK oder CLI
    - Verwendung DeleteDBParameterGroup mit einem AWS SDK oder CLI
    - Verwendung DescribeAccountAttributes mit einem AWS SDK oder CLI
    - Verwendung DescribeDBEngineVersions mit einem AWS SDK oder CLI
    - Verwendung DescribeDBInstances mit einem AWS SDK oder CLI
    - Verwendung DescribeDBParameterGroups mit einem AWS SDK oder CLI
    - Verwendung DescribeDBParameters mit einem AWS SDK oder CLI
    - Verwendung DescribeDBSnapshots mit einem AWS SDK oder CLI
    - Verwendung DescribeOrderableDBInstanceOptions mit einem AWS SDK oder CLI

- Verwendung GenerateRDSAuthToken mit einem AWS SDK
- Verwendung ModifyDBInstance mit einem AWS SDK oder CLI
- Verwendung ModifyDBParameterGroup mit einem AWS SDK oder CLI
- Verwendung RebootDBInstance mit einem AWS SDK oder CLI

Hello Amazon RDS

Die folgenden Codebeispiele veranschaulichen die ersten Schritte mit Amazon RDS.

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
using System;
using System.Threading.Tasks;
using Amazon.RDS;
using Amazon.RDS.Model;

namespace RDSActions;

public static class HelloRds
{
    static async Task Main(string[] args)
    {
        var rdsClient = new AmazonRDSClient();

        Console.WriteLine($"Hello Amazon RDS! Following are some of your DB
 instances:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first twenty DB instances.
        var response = await rdsClient.DescribeDBInstancesAsync(
            new DescribeDBInstancesRequest()
```

```
            {
                MaxRecords = 20 // Must be between 20 and 100.
            });

        foreach (var instance in response.DBInstances)
        {
            Console.WriteLine($"\tDB name: {instance.DBName}");
            Console.WriteLine($"\tArn: {instance.DBInstanceArn}");
            Console.WriteLine($"\tIdentifier: {instance.DBInstanceIdentifier}");
            Console.WriteLine();
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Code für die CMake Datei CMake Lists.txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS rds)

# Set this project's name.
project("hello_rds")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
```

```
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

     # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
 may need to uncomment this
                                    # and set the proper subdirectory to the
 executables' location.

     AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_rds.cpp)

target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Code für die Quelldatei „hello_rds.cpp“.

```
#include <aws/core/Aws.h>
#include <aws/rds/RDSClient.h>
#include <aws/rds/model/DescribeDBInstancesRequest.h>
#include <iostream>

/*
```

```
 *  A "Hello Rds" starter application which initializes an Amazon Relational
 Database Service (Amazon RDS) client and
 *  describes the Amazon RDS instances.
 *
 *  main function
 *
 *  Usage: 'hello_rds'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::RDS::RDSClient rdsClient(clientConfig);
        Aws::String marker;
        std::vector<Aws::String> instanceDBIDs;

        do {
            Aws::RDS::Model::DescribeDBInstancesRequest request;

            if (!marker.empty()) {
                request.SetMarker(marker);
            }

            Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
                    rdsClient.DescribeDBInstances(request);

            if (outcome.IsSuccess()) {
                for (auto &instance: outcome.GetResult().GetDBInstances()) {
                    instanceDBIDs.push_back(instance.GetDBInstanceIdentifier());
                }
                marker = outcome.GetResult().GetMarker();
            } else {
                result = 1;
                std::cerr << "Error with RDS::DescribeDBInstances. "
                          << outcome.GetError().GetMessage()
```

```
                            << std::endl;
                break;
            }
        } while (!marker.empty());

        std::cout << instanceDBIDs.size() << " RDS instances found." <<
 std::endl;
        for (auto &instanceDBID: instanceDBIDs) {
            std::cout << "   Instance: " << instanceDBID << std::endl;
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für C++ API-Referenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/rds"
)
```

```
// main uses the AWS SDK for Go V2 to create an Amazon Relational Database
 Service (Amazon RDS)
// client and list up to 20 DB instances in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 ctx := context.Background()
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 rdsClient := rds.NewFromConfig(sdkConfig)
 const maxInstances = 20
 fmt.Printf("Let's list up to %v DB instances.\n", maxInstances)
 output, err := rdsClient.DescribeDBInstances(ctx,
  &rds.DescribeDBInstancesInput{MaxRecords: aws.Int32(maxInstances)})
 if err != nil {
  fmt.Printf("Couldn't list DB instances: %v\n", err)
  return
 }
 if len(output.DBInstances) == 0 {
  fmt.Println("No DB instances found.")
 } else {
  for _, instance := range output.DBInstances {
   fmt.Printf("DB instance %v has database %v.\n",
 *instance.DBInstanceIdentifier,
    *instance.DBName)
  }
 }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
```

```
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
"""
Purpose

Shows how to use the AWS SDK for Python (Boto3) with the Amazon Relational
 Database Service
(Amazon RDS) to list the databases in your account.
"""

import boto3
from botocore.exceptions import ClientError
```

```
# Create an RDS client
rds_client = boto3.client("rds")

# Create a paginator for the describe_db_instances operation
paginator = rds_client.get_paginator("describe_db_instances")

try:
    # Use the paginator to get a list of DB instances
    response_iterator = paginator.paginate(
        PaginationConfig={
            "MaxItems": 123,
            "PageSize": 50,  # Adjust PageSize as needed
            "StartingToken": None,
        }
    )

    # Iterate through the pages of the response
    instances_found = False
    for page in response_iterator:
        if "DBInstances" in page and page["DBInstances"]:
            instances_found = True
            print("Your RDS instances are:")
            for db in page["DBInstances"]:
                print(db["DBInstanceIdentifier"])

    if not instances_found:
        print("No RDS instances found!")

except ClientError as e:
    print(f"Couldn't list RDS instances. Here's why: {e.response['Error']
['Message']}")
```

- Einzelheiten zur API finden Sie unter DBInstancesDescribe in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds'
require 'logger'

# RDSManager is a class responsible for managing RDS operations
# such as listing all RDS DB instances in the current AWS account.
class RDSManager
  def initialize(client)
    @client = client
    @logger = Logger.new($stdout)
  end

  # Lists and prints all RDS DB instances in the current AWS account.
  def list_db_instances
    @logger.info('Listing RDS DB instances')

    paginator = @client.describe_db_instances
    instances = []

    paginator.each_page do |page|
      instances.concat(page.db_instances)
    end

    if instances.empty?
      @logger.info('No instances found.')
    else
      @logger.info("Found #{instances.count} instance(s):")
      instances.each do |instance|
        @logger.info(" * #{instance.db_instance_identifier}
 (#{instance.db_instance_status})")
      end
    end
```

```
  end
end

if $PROGRAM_NAME == __FILE__
  rds_client = Aws::RDS::Client.new(region: 'us-west-2')
  manager = RDSManager.new(rds_client)
  manager.list_db_instances
end
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für Ruby API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Lernen Sie die Grundlagen von Amazon RDS mit einem AWS SDK kennen

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Erstellen Sie eine benutzerdefinierte DB-Parametergruppe und legen Sie Parameterwerte fest.
- Erstellen Sie eine DB-Instance, die zur Verwendung der Parametergruppe konfiguriert ist. Die DB-Instance enthält auch eine Datenbank.
- Erstellen Sie einen Snapshot der Instance.
- Löschen Sie die Instance und die Parametergruppe.

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Führen Sie ein interaktives Szenario an einem Prompt aus.

```
/// <summary>
/// Scenario for RDS DB instance example.
/// </summary>
public class RDSInstanceScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks:
    1.  Returns a list of the available DB engine families using the
 DescribeDBEngineVersionsAsync method.
    2.  Selects an engine family and creates a custom DB parameter group using
 the CreateDBParameterGroupAsync method.
    3.  Gets the parameter groups using the DescribeDBParameterGroupsAsync
 method.
    4.  Gets parameters in the group using the DescribeDBParameters method.
    5.  Parses and displays parameters in the group.
    6.  Modifies both the auto_increment_offset and auto_increment_increment
 parameters
        using the ModifyDBParameterGroupAsync method.
    7.  Gets and displays the updated parameters using the DescribeDBParameters
 method with a source of "user".
    8.  Gets a list of allowed engine versions using the
 DescribeDBEngineVersionsAsync method.
    9.  Displays and selects from a list of micro instance classes available for
 the selected engine and version.
    10. Creates an RDS DB instance that contains a MySql database and uses the
 parameter group
        using the CreateDBInstanceAsync method.
    11. Waits for DB instance to be ready using the DescribeDBInstancesAsync
 method.
    12. Prints out the connection endpoint string for the new DB instance.
    13. Creates a snapshot of the DB instance using the CreateDBSnapshotAsync
 method.
    14. Waits for DB snapshot to be ready using the DescribeDBSnapshots method.
    15. Deletes the DB instance using the DeleteDBInstanceAsync method.
    16. Waits for DB instance to be deleted using the DescribeDbInstances method.
    17. Deletes the parameter group using the DeleteDBParameterGroupAsync.
    */

    private static readonly string sepBar = new('-', 80);
```

```
    private static RDSWrapper rdsWrapper = null!;
    private static ILogger logger = null!;
    private static readonly string engine = "mysql";
    static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon RDS service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonRDS>()
                    .AddTransient<RDSWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder =>
        {
            builder.AddConsole();
        }).CreateLogger<RDSInstanceScenario>();

        rdsWrapper = host.Services.GetRequiredService<RDSWrapper>();

        Console.WriteLine(sepBar);
        Console.WriteLine(
            "Welcome to the Amazon Relational Database Service (Amazon RDS) DB
instance scenario example.");
        Console.WriteLine(sepBar);

        try
        {
            var parameterGroupFamily = await ChooseParameterGroupFamily();

            var parameterGroup = await
CreateDbParameterGroup(parameterGroupFamily);

            var parameters = await
DescribeParametersInGroup(parameterGroup.DBParameterGroupName,
                new List<string> { "auto_increment_offset",
"auto_increment_increment" });
```

```
            await ModifyParameters(parameterGroup.DBParameterGroupName,
parameters);

            await
DescribeUserSourceParameters(parameterGroup.DBParameterGroupName);

            var engineVersionChoice = await
ChooseDbEngineVersion(parameterGroupFamily);

            var instanceChoice = await ChooseDbInstanceClass(engine,
engineVersionChoice.EngineVersion);

            var newInstanceIdentifier = "Example-Instance-" + DateTime.Now.Ticks;

            var newInstance = await CreateRdsNewInstance(parameterGroup, engine,
engineVersionChoice.EngineVersion,
                instanceChoice.DBInstanceClass, newInstanceIdentifier);
            if (newInstance != null)
            {
                DisplayConnectionString(newInstance);

                await CreateSnapshot(newInstance);

                await DeleteRdsInstance(newInstance);
            }

            await DeleteParameterGroup(parameterGroup);

            Console.WriteLine("Scenario complete.");
            Console.WriteLine(sepBar);
        }
        catch (Exception ex)
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
        }
    }

    /// <summary>
    /// Choose the RDS DB parameter group family from a list of available
options.
    /// </summary>
    /// <returns>The selected parameter group family.</returns>
    public static async Task<string> ChooseParameterGroupFamily()
    {
```

```
        Console.WriteLine(sepBar);
        // 1. Get a list of available engines.
        var engines = await rdsWrapper.DescribeDBEngineVersions(engine);

        Console.WriteLine("1. The following is a list of available DB parameter
group families:");
        int i = 1;
        var parameterGroupFamilies = engines.GroupBy(e =>
e.DBParameterGroupFamily).ToList();
        foreach (var parameterGroupFamily in parameterGroupFamilies)
        {
            // List the available parameter group families.
            Console.WriteLine(
                $"\t{i}. Family: {parameterGroupFamily.Key}");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > parameterGroupFamilies.Count)
        {
            Console.WriteLine("Select an available DB parameter group family by
entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }
        var parameterGroupFamilyChoice = parameterGroupFamilies[choiceNumber -
1];
        Console.WriteLine(sepBar);
        return parameterGroupFamilyChoice.Key;
    }

    /// <summary>
    /// Create and get information on a DB parameter group.
    /// </summary>
    /// <param name="dbParameterGroupFamily">The DBParameterGroupFamily for the
new DB parameter group.</param>
    /// <returns>The new DBParameterGroup.</returns>
    public static async Task<DBParameterGroup> CreateDbParameterGroup(string
dbParameterGroupFamily)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine($"2. Create new DB parameter group with family
{dbParameterGroupFamily}:");
```

```
        var parameterGroup = await rdsWrapper.CreateDBParameterGroup(
            "ExampleParameterGroup-" + DateTime.Now.Ticks,
            dbParameterGroupFamily, "New example parameter group");

        var groupInfo =
            await rdsWrapper.DescribeDBParameterGroups(parameterGroup
                .DBParameterGroupName);

        Console.WriteLine(
            $"3. New DB parameter group: \n\t{groupInfo[0].Description}, \n\tARN
{groupInfo[0].DBParameterGroupArn}");
        Console.WriteLine(sepBar);
        return parameterGroup;
    }

    /// <summary>
    /// Get and describe parameters from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameterNames">Optional specific names of parameters to
describe.</param>
    /// <returns>The list of requested parameters.</returns>
    public static async Task<List<Parameter>> DescribeParametersInGroup(string
parameterGroupName, List<string>? parameterNames = null)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("4. Get some parameters from the group.");
        Console.WriteLine(sepBar);

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName);

        var matchingParameters =
            parameters.Where(p => parameterNames == null ||
parameterNames.Contains(p.ParameterName)).ToList();

        Console.WriteLine("5. Parameter information:");
        matchingParameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));
```

```
        Console.WriteLine(sepBar);

        return matchingParameters;
    }

    /// <summary>
    /// Modify a parameter from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameters">The parameters to modify.</param>
    /// <returns>Async task.</returns>
    public static async Task ModifyParameters(string parameterGroupName,
List<Parameter> parameters)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("6. Modify some parameters in the group.");

        foreach (var p in parameters)
        {
            if (p.IsModifiable && p.DataType == "integer")
            {
                int newValue = 0;
                while (newValue == 0)
                {
                    Console.WriteLine(
                        $"Enter a new value for {p.ParameterName} from the
allowed values {p.AllowedValues} ");

                    var choice = Console.ReadLine();
                    Int32.TryParse(choice, out newValue);
                }

                p.ParameterValue = newValue.ToString();
            }
        }

        await rdsWrapper.ModifyDBParameterGroup(parameterGroupName, parameters);

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Describe the user source parameters in the group.
    /// </summary>
```

```
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <returns>Async task.</returns>
    public static async Task DescribeUserSourceParameters(string
parameterGroupName)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("7. Describe user source parameters in the group.");

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName, "user");


        parameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);
    }


    /// <summary>
    /// Choose a DB engine version.
    /// </summary>
    /// <param name="dbParameterGroupFamily">DB parameter group family for engine
choice.</param>
    /// <returns>The selected engine version.</returns>
    public static async Task<DBEngineVersion> ChooseDbEngineVersion(string
dbParameterGroupFamily)
    {
        Console.WriteLine(sepBar);
        // Get a list of allowed engines.
        var allowedEngines =
            await rdsWrapper.DescribeDBEngineVersions(engine,
dbParameterGroupFamily);

        Console.WriteLine($"Available DB engine versions for parameter group
family {dbParameterGroupFamily}:");
        int i = 1;
        foreach (var version in allowedEngines)
        {
            Console.WriteLine(
```

```
                $"\t{i}. Engine: {version.Engine} Version
 {version.EngineVersion}.");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedEngines.Count)
        {
            Console.WriteLine("8. Select an available DB engine version by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var engineChoice = allowedEngines[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return engineChoice;
    }

    /// <summary>
    /// Choose a DB instance class for a particular engine and engine version.
    /// </summary>
    /// <param name="engine">DB engine for DB instance choice.</param>
    /// <param name="engineVersion">DB engine version for DB instance choice.</
param>
    /// <returns>The selected orderable DB instance option.</returns>
    public static async Task<OrderableDBInstanceOption>
 ChooseDbInstanceClass(string engine, string engineVersion)
    {
        Console.WriteLine(sepBar);
        // Get a list of allowed DB instance classes.
        var allowedInstances =
            await rdsWrapper.DescribeOrderableDBInstanceOptions(engine,
 engineVersion);

        Console.WriteLine($"8. Available micro DB instance classes for engine
 {engine} and version {engineVersion}:");
        int i = 1;

        // Filter to micro instances for this example.
        allowedInstances = allowedInstances
            .Where(i => i.DBInstanceClass.Contains("micro")).ToList();

        foreach (var instance in allowedInstances)
```

```
        {
            Console.WriteLine(
                $"\t{i}. Instance class: {instance.DBInstanceClass} (storage type
 {instance.StorageType})");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedInstances.Count)
        {
            Console.WriteLine("9. Select an available DB instance class by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var instanceChoice = allowedInstances[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return instanceChoice;
    }

    /// <summary>
    /// Create a new RDS DB instance.
    /// </summary>
    /// <param name="parameterGroup">Parameter group to use for the DB
 instance.</param>
    /// <param name="engineName">Engine to use for the DB instance.</param>
    /// <param name="engineVersion">Engine version to use for the DB instance.</
param>
    /// <param name="instanceClass">Instance class to use for the DB instance.</
param>
    /// <param name="instanceIdentifier">Instance identifier to use for the DB
 instance.</param>
    /// <returns>The new DB instance.</returns>
    public static async Task<DBInstance?> CreateRdsNewInstance(DBParameterGroup
 parameterGroup,
        string engineName, string engineVersion, string instanceClass, string
 instanceIdentifier)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine($"10. Create a new DB instance with identifier
 {instanceIdentifier}.");
        bool isInstanceReady = false;
        DBInstance newInstance;
```

```
        var instances = await rdsWrapper.DescribeDBInstances();
        isInstanceReady = instances.FirstOrDefault(i =>
            i.DBInstanceIdentifier == instanceIdentifier)?.DBInstanceStatus ==
"available";

        if (isInstanceReady)
        {
            Console.WriteLine("Instance already created.");
            newInstance = instances.First(i => i.DBInstanceIdentifier ==
instanceIdentifier);
        }
        else
        {
            Console.WriteLine("Please enter an admin user name:");
            var username = Console.ReadLine();

            Console.WriteLine("Please enter an admin password:");
            var password = Console.ReadLine();

            newInstance = await rdsWrapper.CreateDBInstance(
                "ExampleInstance",
                instanceIdentifier,
                parameterGroup.DBParameterGroupName,
                engineName,
                engineVersion,
                instanceClass,
                20,
                username,
                password
            );

            // 11. Wait for the DB instance to be ready.

            Console.WriteLine("11. Waiting for DB instance to be ready...");
            while (!isInstanceReady)
            {
                instances = await
rdsWrapper.DescribeDBInstances(instanceIdentifier);
                isInstanceReady = instances.FirstOrDefault()?.DBInstanceStatus ==
"available";
                newInstance = instances.First();
                Thread.Sleep(30000);
            }
        }
```

```
        Console.WriteLine(sepBar);
        return newInstance;
    }

    /// <summary>
    /// Display a connection string for an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to use to get a connection
 string.</param>
    public static void DisplayConnectionString(DBInstance instance)
    {
        Console.WriteLine(sepBar);
        // Display the connection string.
        Console.WriteLine("12. New DB instance connection string: ");
        Console.WriteLine(
            $"\n{engine} -h {instance.Endpoint.Address} -P
 {instance.Endpoint.Port} "
            + $"-u {instance.MasterUsername} -p [YOUR PASSWORD]\n");

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Create a snapshot from an RDS DB instance.
    /// </summary>
    /// <param name="instance">DB instance to use when creating a snapshot.</
param>
    /// <returns>The snapshot object.</returns>
    public static async Task<DBSnapshot> CreateSnapshot(DBInstance instance)
    {
        Console.WriteLine(sepBar);
        // Create a snapshot.
        Console.WriteLine($"13. Creating snapshot from DB instance
 {instance.DBInstanceIdentifier}.");
        var snapshot = await
 rdsWrapper.CreateDBSnapshot(instance.DBInstanceIdentifier, "ExampleSnapshot-" +
 DateTime.Now.Ticks);

        // Wait for the snapshot to be available
        bool isSnapshotReady = false;

        Console.WriteLine($"14. Waiting for snapshot to be ready...");
        while (!isSnapshotReady)
```

```
        {
            var snapshots = await
rdsWrapper.DescribeDBSnapshots(instance.DBInstanceIdentifier);
            isSnapshotReady = snapshots.FirstOrDefault()?.Status == "available";
            snapshot = snapshots.First();
            Thread.Sleep(30000);
        }

        Console.WriteLine(
            $"Snapshot {snapshot.DBSnapshotIdentifier} status is
{snapshot.Status}.");
        Console.WriteLine(sepBar);
        return snapshot;
    }

    /// <summary>
    /// Delete an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteRdsInstance(DBInstance newInstance)
    {
        Console.WriteLine(sepBar);
        // Delete the DB instance.
        Console.WriteLine($"15. Delete the DB instance
{newInstance.DBInstanceIdentifier}.");
        await rdsWrapper.DeleteDBInstance(newInstance.DBInstanceIdentifier);

        // Wait for the DB instance to delete.
        Console.WriteLine($"16. Waiting for the DB instance to delete...");
        bool isInstanceDeleted = false;

        while (!isInstanceDeleted)
        {
            var instance = await rdsWrapper.DescribeDBInstances();
            isInstanceDeleted = instance.All(i => i.DBInstanceIdentifier !=
newInstance.DBInstanceIdentifier);
            Thread.Sleep(30000);
        }

        Console.WriteLine("DB instance deleted.");
        Console.WriteLine(sepBar);
    }
```

```
    /// <summary>
    /// Delete a DB parameter group.
    /// </summary>
    /// <param name="parameterGroup">The parameter group to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteParameterGroup(DBParameterGroup
 parameterGroup)
    {
        Console.WriteLine(sepBar);
        // Delete the parameter group.
        Console.WriteLine($"17. Delete the DB parameter group
 {parameterGroup.DBParameterGroupName}.");
        await
 rdsWrapper.DeleteDBParameterGroup(parameterGroup.DBParameterGroupName);

        Console.WriteLine(sepBar);
    }
```

Wrapper-Methoden, die vom Szenario für DB-Instance-Aktionen verwendet werden.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 DB instance operations.
/// </summary>
public partial class RDSWrapper
{
    private readonly IAmazonRDS _amazonRDS;
    public RDSWrapper(IAmazonRDS amazonRDS)
    {
        _amazonRDS = amazonRDS;
    }


    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
 name.</param>
    /// <returns>List of DBEngineVersions.</returns>
```

```
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }



    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }
```

```
   /// <summary>
   /// Returns a list of DB instances.
   /// </summary>
   /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
   /// <returns>List of DB instances.</returns>
   public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
   {
       var results = new List<DBInstance>();
       var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
           new DescribeDBInstancesRequest
           {
               DBInstanceIdentifier = dbInstanceIdentifier
           });
       // Get the entire list using the paginator.
       await foreach (var instances in instancesPaginator.DBInstances)
       {
           results.Add(instances);
       }
       return results;
   }



   /// <summary>
   /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
   /// to determine when the DB instance is ready to use.
   /// </summary>
   /// <param name="dbName">Name for the DB instance.</param>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
```

```
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }



   /// <summary>
   /// Delete a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
   {
       var response = await _amazonRDS.DeleteDBInstanceAsync(
           new DeleteDBInstanceRequest()
           {
               DBInstanceIdentifier = dbInstanceIdentifier,
               SkipFinalSnapshot = true,
               DeleteAutomatedBackups = true
           });

       return response.DBInstance;
   }
```

Wrapper-Methoden, die vom Szenario für DB-Parametergruppen verwendet werden.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 parameter groups.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }



    /// <summary>
    /// Create a new DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="family">Family of the DB parameter group.</param>
    /// <param name="description">Description of the DB parameter group.</param>
    /// <returns>The new DB parameter group.</returns>
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
```

```
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }



    /// <summary>
    /// Update a DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }



    /// <summary>
    /// Delete a DB parameter group. The group cannot be a default DB parameter
 group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
```

```
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }



    /// <summary>
    /// Get a list of DB parameters from a specific parameter group.
    /// </summary>
    /// <param name="dbParameterGroupName">Name of a specific DB parameter
 group.</param>
    /// <param name="source">Optional source for selecting parameters.</param>
    /// <returns>List of parameter values.</returns>
    public async Task<List<Parameter>> DescribeDBParameters(string
 dbParameterGroupName, string source = null)
    {
        var results = new List<Parameter>();
        var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
            new DescribeDBParametersRequest()
            {
                DBParameterGroupName = dbParameterGroupName,
                Source = source
            });
        // Get the entire list using the paginator.
        await foreach (var parameters in paginateParameters.Parameters)
        {
            results.Add(parameters);
        }
        return results;
    }
```

Wrapper-Methoden, die vom Szenario für DB-Snapshot-Aktionen verwendet werden.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 snapshots.
```

```
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
 string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }



    /// <summary>
    /// Return a list of DB snapshots for a particular DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <returns>List of DB snapshots.</returns>
    public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
 dbInstanceIdentifier)
    {
        var results = new List<DBSnapshot>();
        var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
            new DescribeDBSnapshotsRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
```

```
        }
        return results;
    }
```

- API-Details finden Sie in den folgenden Themen der AWS SDK für .NET -API-Referenz.
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance
  - DBParameterGruppe löschen
  - DBEngineVersionen beschreiben
  - Beschreiben DBInstances
  - Beschreiben Sie DBParameter Gruppen
  - Beschreiben DBParameters
  - Beschreiben DBSnapshots
  - DescribeOrderableDBInstanceOptionen
  - DBParameterGruppe ändern

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

//! Routine which creates an Amazon RDS instance and demonstrates several
 operations
```

```
//! on that instance.
/*!
 \sa gettingStartedWithDBInstances()
 \param clientConfiguration: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::gettingStartedWithDBInstances(
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::RDS::RDSClient client(clientConfig);

    printAsterisksLine();
    std::cout << "Welcome to the Amazon Relational Database Service (Amazon RDS)"
              << std::endl;
    std::cout << "get started with DB instances demo." << std::endl;
    printAsterisksLine();

    std::cout << "Checking for an existing DB parameter group named '" <<
              PARAMETER_GROUP_NAME << "'." << std::endl;
    Aws::String dbParameterGroupFamily("Undefined");
    bool parameterGroupFound = true;
    {
        // 1. Check if the DB parameter group already exists.
        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }
        else if (outcome.GetError().GetErrorType() ==
                 Aws::RDS::RDSErrors::D_B_PARAMETER_GROUP_NOT_FOUND_FAULT) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' does not exist." << std::endl;
            parameterGroupFound = false;
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
```

```
            return false;
        }
    }

    if (!parameterGroupFound) {
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 2. Get available engine versions for the specified engine.
        if (!getDBEngineVersions(DB_ENGINE, NO_PARAMETER_GROUP_FAMILY,
                                 engineVersions, client)) {
            return false;
        }

        std::cout << "Getting available database engine versions for " <<
DB_ENGINE
                  << "."
                  << std::endl;
        std::vector<Aws::String> families;
        for (const Aws::RDS::Model::DBEngineVersion &version: engineVersions) {
            Aws::String family = version.GetDBParameterGroupFamily();
            if (std::find(families.begin(), families.end(), family) ==
                families.end()) {
                families.push_back(family);
                std::cout << "  " << families.size() << ": " << family <<
std::endl;
            }
        }

        int choice = askQuestionForIntRange("Which family do you want to use? ",
1,
                                            static_cast<int>(families.size()));
        dbParameterGroupFamily = families[choice - 1];
    }
    if (!parameterGroupFound) {
        // 3.  Create a DB parameter group.
        Aws::RDS::Model::CreateDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);

        if (outcome.IsSuccess()) {
```

```
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    printAsterisksLine();
    std::cout << "Let's set some parameter values in your parameter group."
              << std::endl;

    Aws::String marker;
    Aws::Vector<Aws::RDS::Model::Parameter> autoIncrementParameters;
    // 4.  Get the parameters in the DB parameter group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, AUTO_INCREMENT_PREFIX, NO_SOURCE,
                         autoIncrementParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    Aws::Vector<Aws::RDS::Model::Parameter> updateParameters;

    for (Aws::RDS::Model::Parameter &autoIncParameter: autoIncrementParameters) {
        if (autoIncParameter.GetIsModifiable() &&
            (autoIncParameter.GetDataType() == "integer")) {
            std::cout << "The " << autoIncParameter.GetParameterName()
                      << " is described as: " <<
                      autoIncParameter.GetDescription() << "." << std::endl;
            if (autoIncParameter.ParameterValueHasBeenSet()) {
                std::cout << "The current value is "
                          << autoIncParameter.GetParameterValue()
                          << "." << std::endl;
            }
            std::vector<int> splitValues = splitToInts(
                    autoIncParameter.GetAllowedValues(), '-');
            if (splitValues.size() == 2) {
                int newValue = askQuestionForIntRange(
                        Aws::String("Enter a new value in the range ") +
                        autoIncParameter.GetAllowedValues() + ": ",
```

```
                        splitValues[0], splitValues[1]);
                autoIncParameter.SetParameterValue(std::to_string(newValue));
                updateParameters.push_back(autoIncParameter);

            }
            else {
                std::cerr << "Error parsing " <<
autoIncParameter.GetAllowedValues()
                          << std::endl;
            }
        }
    }

    {
        // 5.  Modify the auto increment parameters in the group.
        Aws::RDS::Model::ModifyDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetParameters(updateParameters);

        Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
                client.ModifyDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully modified."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    std::cout
            << "You can get a list of parameters you've set by specifying a
source of 'user'."
            << std::endl;

    Aws::Vector<Aws::RDS::Model::Parameter> userParameters;
    // 6.  Display the modified parameters in the group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, NO_NAME_PREFIX, "user",
userParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
```

```
        return false;
    }

    for (const auto &userParameter: userParameters) {
        std::cout << "  " << userParameter.GetParameterName() << ", " <<
                  userParameter.GetDescription() << ", parameter value - "
                  << userParameter.GetParameterValue() << std::endl;
    }

    printAsterisksLine();
    std::cout << "Checking for an existing DB instance." << std::endl;

    Aws::RDS::Model::DBInstance dbInstance;
    // 7.  Check if the DB instance already exists.
    if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    if (dbInstance.DbInstancePortHasBeenSet()) {
        std::cout << "The DB instance already exists." << std::endl;
    }
    else {
        std::cout << "Let's create a DB instance." << std::endl;
        const Aws::String administratorName = askQuestion(
                "Enter an administrator username for the database: ");
        const Aws::String administratorPassword = askQuestion(
                "Enter a password for the administrator (at least 8 characters):
");
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 8.  Get a list of available engine versions.
        if (!getDBEngineVersions(DB_ENGINE, dbParameterGroupFamily,
engineVersions,
                                 client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "The available engines for your parameter group are:" <<
std::endl;

        int index = 1;
```

```
        for (const Aws::RDS::Model::DBEngineVersion &engineVersion:
engineVersions) {
            std::cout << "  " << index << ": " <<
engineVersion.GetEngineVersion()
                      << std::endl;
            ++index;
        }
        int choice = askQuestionForIntRange("Which engine do you want to use? ",
1,

static_cast<int>(engineVersions.size()));
        const Aws::RDS::Model::DBEngineVersion engineVersion =
engineVersions[choice -
                                                                              1];

        Aws::String dbInstanceClass;
        // 9.  Get a list of micro instance classes.
        if (!chooseMicroDBInstanceClass(engineVersion.GetEngine(),
                                        engineVersion.GetEngineVersion(),
                                        dbInstanceClass,
                                        client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "Creating a DB instance named '" << DB_INSTANCE_IDENTIFIER
                  << "' and database '" << DB_NAME << "'.\n"
                  << "The DB instance is configured to use your custom parameter
group '"
                  << PARAMETER_GROUP_NAME << "',\n"
                  << "selected engine version " <<
engineVersion.GetEngineVersion()
                  << ",\n"
                  << "selected DB instance class '" << dbInstanceClass << "',"
                  << " and " << DB_ALLOCATED_STORAGE << " GiB of " <<
DB_STORAGE_TYPE
                  << " storage.\nThis typically takes several minutes." <<
std::endl;

        Aws::RDS::Model::CreateDBInstanceRequest request;
        request.SetDBName(DB_NAME);
        request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetEngine(engineVersion.GetEngine());
```

```
        request.SetEngineVersion(engineVersion.GetEngineVersion());
        request.SetDBInstanceClass(dbInstanceClass);
        request.SetStorageType(DB_STORAGE_TYPE);
        request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
        request.SetMasterUsername(administratorName);
        request.SetMasterUserPassword(administratorPassword);

        Aws::RDS::Model::CreateDBInstanceOutcome outcome =
                client.CreateDBInstance(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB instance creation has started."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBInstance. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }
    }

    std::cout << "Waiting for the DB instance to become available." << std::endl;

    int counter = 0;
    // 11. Wait for the DB instance to become available.
    do {
        std::this_thread::sleep_for(std::chrono::seconds(1));
        ++counter;
        if (counter > 900) {
            std::cerr << "Wait for instance to become available timed out ofter "
                      << counter
                      << " seconds." << std::endl;
            cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
            return false;
        }

        dbInstance = Aws::RDS::Model::DBInstance();
        if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
            return false;
```

```
        }

        if ((counter % 20) == 0) {
            std::cout << "Current DB instance status is '"
                      << dbInstance.GetDBInstanceStatus()
                      << "' after " << counter << " seconds." << std::endl;
        }
    } while (dbInstance.GetDBInstanceStatus() != "available");

    if (dbInstance.GetDBInstanceStatus() == "available") {
        std::cout << "The DB instance has been created." << std::endl;
    }

    printAsterisksLine();

    // 12. Display the connection string that can be used to connect a 'mysql'
shell to the database.
    displayConnection(dbInstance);

    printAsterisksLine();

    if (askYesNoQuestion(
            "Do you want to create a snapshot of your DB instance (y/n)? ")) {
        Aws::String snapshotID(DB_INSTANCE_IDENTIFIER + "-" +
                               Aws::String(Aws::Utils::UUID::RandomUUID()));
        {
            std::cout << "Creating a snapshot named " << snapshotID << "." <<
std::endl;
            std::cout << "This typically takes a few minutes." << std::endl;

            // 13. Create a snapshot of the DB instance.
            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
```

```
                        << outcome.GetError().GetMessage()
                        << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
        }

        std::cout << "Waiting for snapshot to become available." << std::endl;

        Aws::RDS::Model::DBSnapshot snapshot;
        counter = 0;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 600) {
                std::cerr << "Wait for snapshot to be available timed out ofter "
                        << counter
                        << " seconds." << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }

            // 14. Wait for the snapshot to become available.
            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                        << outcome.GetError().GetMessage()
                        << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }

            if ((counter % 20) == 0) {
```

```
                std::cout << "Current snapshot status is '"
                          << snapshot.GetStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (snapshot.GetStatus() != "available");

        if (snapshot.GetStatus() != "available") {
            std::cout << "A snapshot has been created." << std::endl;
        }
    }

    printAsterisksLine();

    bool result = true;
    if (askYesNoQuestion(
            "Do you want to delete the DB instance and parameter group (y/n)? "))
 {
        result = cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
    }

    return result;
}


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
```

```
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
```

```
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.

    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }


        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
                client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
 engineVersions.begin(),
                                        engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!marker.empty());
```

```
    return true;
}


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

```
//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
```

```
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}

//! Routine which deletes resources created by the scenario.
/*!
\sa cleanUpResources()
\param parameterGroupName: A parameter group name, this may be empty.
\param dbInstanceIdentifier: A DB instance identifier, this may be empty.
\param client: 'RDSClient' instance.
\return bool: Successful completion.
*/
bool AwsDoc::RDS::cleanUpResources(const Aws::String &parameterGroupName,
                                   const Aws::String &dbInstanceIdentifier,
                                   const Aws::RDS::RDSClient &client) {
    bool result = true;
    if (!dbInstanceIdentifier.empty()) {
        {
            // 15. Delete the DB instance.
            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                    client.DeleteDBInstance(request);
```

```
            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
        }

        std::cout
                << "Waiting for DB instance to delete before deleting the
parameter group."
                << std::endl;
        std::cout << "This may take a while." << std::endl;

        int counter = 0;
        Aws::RDS::Model::DBInstance dbInstance;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 800) {
                std::cerr << "Wait for instance to delete timed out ofter " <<
counter
                          << " seconds." << std::endl;
                return false;
            }

            dbInstance = Aws::RDS::Model::DBInstance();
            // 16. Wait for the DB instance to be deleted.
            if (!describeDBInstance(dbInstanceIdentifier, dbInstance, client)) {
                return false;
            }

            if (dbInstance.DBInstanceIdentifierHasBeenSet() && (counter % 20) ==
0) {
                std::cout << "Current DB instance status is '"
                          << dbInstance.GetDBInstanceStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (dbInstance.DBInstanceIdentifierHasBeenSet());
```

```
    }

    if (!parameterGroupName.empty()) {
        // 17. Delete the parameter group.
        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);

        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
    }

    return result;
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK für C++ -API-Referenz.
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance
  - DBParameterGruppe löschen
  - DBEngineVersionen beschreiben
  - Beschreiben DBInstances
  - Beschreiben Sie DBParameter Gruppen
  - Beschreiben DBParameters
  - Beschreiben DBSnapshots
  - DescribeOrderableDBInstanceOptionen

- DBParameterGruppe ändern

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Führen Sie ein interaktives Szenario an einem Prompt aus.

```
import (
 "context"
 "fmt"
 "log"
 "sort"
 "strconv"
 "strings"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/demotools"
 "github.com/awsdocs/aws-doc-sdk-examples/gov2/rds/actions"
 "github.com/google/uuid"
)

// GetStartedInstances is an interactive example that shows you how to use the
 AWS SDK for Go
// with Amazon Relation Database Service (Amazon RDS) to do the following:
//
//  1. Create a custom DB parameter group and set parameter values.
//  2. Create a DB instance that is configured to use the parameter group. The DB
 instance
//     also contains a database.
//  3. Take a snapshot of the DB instance.
//  4. Delete the DB instance and parameter group.
```

```
type GetStartedInstances struct {
 sdkConfig  aws.Config
 instances  actions.DbInstances
 questioner demotools.IQuestioner
 helper     IScenarioHelper
 isTestRun  bool
}

// NewGetStartedInstances constructs a GetStartedInstances instance from a
 configuration.
// It uses the specified config to get an Amazon RDS
// client and create wrappers for the actions used in the scenario.
func NewGetStartedInstances(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
 helper IScenarioHelper) GetStartedInstances {
 rdsClient := rds.NewFromConfig(sdkConfig)
 return GetStartedInstances{
  sdkConfig:  sdkConfig,
  instances:  actions.DbInstances{RdsClient: rdsClient},
  questioner: questioner,
  helper:     helper,
 }
}

// Run runs the interactive scenario.
func (scenario GetStartedInstances) Run(ctx context.Context, dbEngine string,
 parameterGroupName string,
 instanceName string, dbName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the Amazon Relational Database Service (Amazon RDS) DB
 Instance demo.")
 log.Println(strings.Repeat("-", 88))

 parameterGroup := scenario.CreateParameterGroup(ctx, dbEngine,
 parameterGroupName)
 scenario.SetUserParameters(ctx, parameterGroupName)
 instance := scenario.CreateInstance(ctx, instanceName, dbEngine, dbName,
 parameterGroup)
```

```
 scenario.DisplayConnection(instance)
 scenario.CreateSnapshot(ctx, instance)
 scenario.Cleanup(ctx, instance, parameterGroup)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateParameterGroup shows how to get available engine versions for a
 specified
// database engine and create a DB parameter group that is compatible with a
// selected engine family.
func (scenario GetStartedInstances) CreateParameterGroup(ctx context.Context,
 dbEngine string,
 parameterGroupName string) *types.DBParameterGroup {

 log.Printf("Checking for an existing DB parameter group named %v.\n",
  parameterGroupName)
 parameterGroup, err := scenario.instances.GetParameterGroup(ctx,
 parameterGroupName)
 if err != nil {
  panic(err)
 }
 if parameterGroup == nil {
  log.Printf("Getting available database engine versions for %v.\n", dbEngine)
  engineVersions, err := scenario.instances.GetEngineVersions(ctx, dbEngine, "")
  if err != nil {
   panic(err)
  }

  familySet := map[string]struct{}{}
  for _, family := range engineVersions {
   familySet[*family.DBParameterGroupFamily] = struct{}{}
  }
  var families []string
  for family := range familySet {
   families = append(families, family)
  }
  sort.Strings(families)
  familyIndex := scenario.questioner.AskChoice("Which family do you want to use?
\n", families)
  log.Println("Creating a DB parameter group.")
  _, err = scenario.instances.CreateParameterGroup(
```

```
    ctx, parameterGroupName, families[familyIndex], "Example parameter group.")
   if err != nil {
    panic(err)
   }
   parameterGroup, err = scenario.instances.GetParameterGroup(ctx,
  parameterGroupName)
   if err != nil {
    panic(err)
   }
  }
  log.Printf("Parameter group %v:\n", *parameterGroup.DBParameterGroupFamily)
  log.Printf("\tName: %v\n", *parameterGroup.DBParameterGroupName)
  log.Printf("\tARN: %v\n", *parameterGroup.DBParameterGroupArn)
  log.Printf("\tFamily: %v\n", *parameterGroup.DBParameterGroupFamily)
  log.Printf("\tDescription: %v\n", *parameterGroup.Description)
  log.Println(strings.Repeat("-", 88))
  return parameterGroup
}

// SetUserParameters shows how to get the parameters contained in a custom
  parameter
// group and update some of the parameter values in the group.
func (scenario GetStartedInstances) SetUserParameters(ctx context.Context,
  parameterGroupName string) {
  log.Println("Let's set some parameter values in your parameter group.")
  dbParameters, err := scenario.instances.GetParameters(ctx, parameterGroupName,
  "")
  if err != nil {
   panic(err)
  }
  var updateParams []types.Parameter
  for _, dbParam := range dbParameters {
   if strings.HasPrefix(*dbParam.ParameterName, "auto_increment") &&
    *dbParam.IsModifiable && *dbParam.DataType == "integer" {
    log.Printf("The %v parameter is described as:\n\t%v",
     *dbParam.ParameterName, *dbParam.Description)
    rangeSplit := strings.Split(*dbParam.AllowedValues, "-")
    lower, _ := strconv.Atoi(rangeSplit[0])
    upper, _ := strconv.Atoi(rangeSplit[1])
    newValue := scenario.questioner.AskInt(
     fmt.Sprintf("Enter a value between %v and %v:", lower, upper),
     demotools.InIntRange{Lower: lower, Upper: upper})
    dbParam.ParameterValue = aws.String(strconv.Itoa(newValue))
    updateParams = append(updateParams, dbParam)
```

```
  }
 }
 err = scenario.instances.UpdateParameters(ctx, parameterGroupName, updateParams)
 if err != nil {
  panic(err)
 }
 log.Println("To get a list of parameters that you set previously, specify a
 source of 'user'.")
 userParameters, err := scenario.instances.GetParameters(ctx, parameterGroupName,
 "user")
 if err != nil {
  panic(err)
 }
 log.Println("Here are the parameters you set:")
 for _, param := range userParameters {
  log.Printf("\t%v: %v\n", *param.ParameterName, *param.ParameterValue)
 }
 log.Println(strings.Repeat("-", 88))
}

// CreateInstance shows how to create a DB instance that contains a database of a
// specified type. The database is also configured to use a custom DB parameter
 group.
func (scenario GetStartedInstances) CreateInstance(ctx context.Context,
 instanceName string, dbEngine string,
 dbName string, parameterGroup *types.DBParameterGroup) *types.DBInstance {

 log.Println("Checking for an existing DB instance.")
 instance, err := scenario.instances.GetInstance(ctx, instanceName)
 if err != nil {
  panic(err)
 }
 if instance == nil {
  adminUsername := scenario.questioner.Ask(
   "Enter an administrator username for the database: ", demotools.NotEmpty{})
  adminPassword := scenario.questioner.AskPassword(
   "Enter a password for the administrator (at least 8 characters): ", 7)
  engineVersions, err := scenario.instances.GetEngineVersions(ctx, dbEngine,
   *parameterGroup.DBParameterGroupFamily)
  if err != nil {
   panic(err)
  }
  var engineChoices []string
  for _, engine := range engineVersions {
```

```
  engineChoices = append(engineChoices, *engine.EngineVersion)
 }
 engineIndex := scenario.questioner.AskChoice(
  "The available engines for your parameter group are:\n", engineChoices)
 engineSelection := engineVersions[engineIndex]
 instOpts, err := scenario.instances.GetOrderableInstances(ctx,
*engineSelection.Engine,
  *engineSelection.EngineVersion)
 if err != nil {
  panic(err)
 }
 optSet := map[string]struct{}{}
 for _, opt := range instOpts {
  if strings.Contains(*opt.DBInstanceClass, "micro") {
   optSet[*opt.DBInstanceClass] = struct{}{}
  }
 }
 var optChoices []string
 for opt := range optSet {
  optChoices = append(optChoices, opt)
 }
 sort.Strings(optChoices)
 optIndex := scenario.questioner.AskChoice(
  "The available micro DB instance classes for your database engine are:\n",
optChoices)
 storageType := "standard"
 allocatedStorage := int32(5)
 log.Printf("Creating a DB instance named %v and database %v.\n"+
  "The DB instance is configured to use your custom parameter group %v,\n"+
  "selected engine %v,\n"+
  "selected DB instance class %v,"+
  "and %v GiB of %v storage.\n"+
  "This typically takes several minutes.",
  instanceName, dbName, *parameterGroup.DBParameterGroupName,
*engineSelection.EngineVersion,
  optChoices[optIndex], allocatedStorage, storageType)
 instance, err = scenario.instances.CreateInstance(
  ctx, instanceName, dbName, *engineSelection.Engine,
*engineSelection.EngineVersion,
  *parameterGroup.DBParameterGroupName, optChoices[optIndex], storageType,
  allocatedStorage, adminUsername, adminPassword)
 if err != nil {
  panic(err)
 }
```

```
  for *instance.DBInstanceStatus != "available" {
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(ctx, instanceName)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Instance created and available.")
 }
 log.Println("Instance data:")
 log.Printf("\tDBInstanceIdentifier: %v\n", *instance.DBInstanceIdentifier)
 log.Printf("\tARN: %v\n", *instance.DBInstanceArn)
 log.Printf("\tStatus: %v\n", *instance.DBInstanceStatus)
 log.Printf("\tEngine: %v\n", *instance.Engine)
 log.Printf("\tEngine version: %v\n", *instance.EngineVersion)
 log.Println(strings.Repeat("-", 88))
 return instance
}

// DisplayConnection displays connection information about a DB instance and tips
// on how to connect to it.
func (scenario GetStartedInstances) DisplayConnection(instance *types.DBInstance)
 {
 log.Println(
  "You can now connect to your database by using your favorite MySQL client.\n" +
   "One way to connect is by using the 'mysql' shell on an Amazon EC2 instance\n"
 +
   "that is running in the same VPC as your DB instance. Pass the endpoint,\n" +
   "port, and administrator username to 'mysql'. Then, enter your password\n" +
   "when prompted:")
 log.Printf("\n\tmysql -h %v -P %v -u %v -p\n",
  *instance.Endpoint.Address, instance.Endpoint.Port, *instance.MasterUsername)
 log.Println("For more information, see the User Guide for RDS:\n" +
  "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL")
 log.Println(strings.Repeat("-", 88))
}

// CreateSnapshot shows how to create a DB instance snapshot and wait until it's
 available.
func (scenario GetStartedInstances) CreateSnapshot(ctx context.Context, instance
 *types.DBInstance) {
 if scenario.questioner.AskBool(
  "Do you want to create a snapshot of your DB instance (y/n)? ", "y") {
```

```
  snapshotId := fmt.Sprintf("%v-%v", *instance.DBInstanceIdentifier,
 scenario.helper.UniqueId())
  log.Printf("Creating a snapshot named %v. This typically takes a few minutes.
\n", snapshotId)
  snapshot, err := scenario.instances.CreateSnapshot(ctx,
 *instance.DBInstanceIdentifier, snapshotId)
  if err != nil {
   panic(err)
  }
  for *snapshot.Status != "available" {
   scenario.helper.Pause(30)
   snapshot, err = scenario.instances.GetSnapshot(ctx, snapshotId)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Snapshot data:")
  log.Printf("\tDBSnapshotIdentifier: %v\n", *snapshot.DBSnapshotIdentifier)
  log.Printf("\tARN: %v\n", *snapshot.DBSnapshotArn)
  log.Printf("\tStatus: %v\n", *snapshot.Status)
  log.Printf("\tEngine: %v\n", *snapshot.Engine)
  log.Printf("\tEngine version: %v\n", *snapshot.EngineVersion)
  log.Printf("\tDBInstanceIdentifier: %v\n", *snapshot.DBInstanceIdentifier)
  log.Printf("\tSnapshotCreateTime: %v\n", *snapshot.SnapshotCreateTime)
  log.Println(strings.Repeat("-", 88))
 }
}

// Cleanup shows how to clean up a DB instance and DB parameter group.
// Before the DB parameter group can be deleted, all associated DB instances must
 first be deleted.
func (scenario GetStartedInstances) Cleanup(
 ctx context.Context, instance *types.DBInstance, parameterGroup
 *types.DBParameterGroup) {

 if scenario.questioner.AskBool(
  "\nDo you want to delete the database instance and parameter group (y/n)? ",
 "y") {
  log.Printf("Deleting database instance %v.\n", *instance.DBInstanceIdentifier)
  err := scenario.instances.DeleteInstance(ctx, *instance.DBInstanceIdentifier)
  if err != nil {
   panic(err)
  }
  log.Println(
```

```
    "Waiting for the DB instance to delete. This typically takes several
 minutes.")
  for instance != nil {
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(ctx,
 *instance.DBInstanceIdentifier)
   if err != nil {
    panic(err)
   }
  }
  log.Printf("Deleting parameter group %v.",
 *parameterGroup.DBParameterGroupName)
  err = scenario.instances.DeleteParameterGroup(ctx,
 *parameterGroup.DBParameterGroupName)
  if err != nil {
   panic(err)
  }
 }
}

// IScenarioHelper abstracts the function from a scenario so that it
// can be mocked for unit testing.
type IScenarioHelper interface {
 Pause(secs int)
 UniqueId() string
}
type ScenarioHelper struct{}

// Pause waits for the specified number of seconds.
func (helper ScenarioHelper) Pause(secs int) {
 time.Sleep(time.Duration(secs) * time.Second)
}

// UniqueId returns a new UUID.
func (helper ScenarioHelper) UniqueId() string {
 return uuid.New().String()
}
```

Definieren Sie Funktionen, die vom Szenario aufgerufen werden, um Amazon-RDS-Aktionen zu verwalten.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}


// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(ctx context.Context,
 parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  ctx, &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
  return nil, err
 } else {
  return &output.DBParameterGroups[0], err
 }
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
```

```
func (instances *DbInstances) CreateParameterGroup(
 ctx context.Context, parameterGroupName string, parameterGroupFamily string,
 description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(ctx,
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(ctx context.Context,
 parameterGroupName string) error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(ctx,
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}



// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(ctx context.Context,
 parameterGroupName string, source string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
```

```
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(ctx)
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(ctx context.Context,
 parameterGroupName string, params []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(ctx,
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}



// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(ctx context.Context, instanceName
 string, snapshotName string) (
 *types.DBSnapshot, error) {
```

```
 output, err := instances.RdsClient.CreateDBSnapshot(ctx,
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(ctx context.Context, snapshotName
 string) (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(ctx,
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(ctx context.Context, instanceName
 string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(ctx,
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
```

```
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
 }
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(ctx context.Context, instanceName
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(ctx,
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
 }
}



// DeleteInstance deletes a DB instance.
```

```
func (instances *DbInstances) DeleteInstance(ctx context.Context, instanceName
 string) error {
 _, err := instances.RdsClient.DeleteDBInstance(ctx, &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      aws.Bool(true),
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}



// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(ctx context.Context, engine
 string, parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(ctx,
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}



// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(ctx context.Context, engine
 string, engineVersion string) (
 []types.OrderableDBInstanceOption, error) {
```

```
 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(ctx)
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
 }
 return instanceOptions, err
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK für Go -API-Referenz.
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance
  - DBParameterGruppe löschen
  - DBEngineVersionen beschreiben
  - Beschreiben DBInstances
  - Beschreiben Sie DBParameter Gruppen
  - Beschreiben DBParameters
  - Beschreiben DBSnapshots
  - DescribeOrderableDBInstanceOptionen
  - DBParameterGruppe ändern

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Führen Sie mehrere Operationen aus.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotRequest;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotResponse;
import software.amazon.awssdk.services.rds.model.DBEngineVersion;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.DBParameterGroup;
import software.amazon.awssdk.services.rds.model.DBSnapshot;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsResponse;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsResponse;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.OrderableDBInstanceOption;
import software.amazon.awssdk.services.rds.model.Parameter;
```

```
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbParameterGroupRequest;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 * This Java example performs these tasks:
 *
 * 1. Returns a list of the available DB engines.
 * 2. Selects an engine family and create a custom DB parameter group.
 * 3. Gets the parameter groups.
 * 4. Gets parameters in the group.
 * 5. Modifies the auto_increment_offset parameter.
 * 6. Gets and displays the updated parameters.
 * 7. Gets a list of allowed engine versions.
 * 8. Gets a list of micro instance classes available for the selected engine.
 * 9. Creates an RDS database instance that contains a MySql database and uses
```

```
 * the parameter group.
 * 10. Waits for the DB instance to be ready and prints out the connection
 * endpoint value.
 * 11. Creates a snapshot of the DB instance.
 * 12. Waits for an RDS DB snapshot to be ready.
 * 13. Deletes the RDS DB instance.
 * 14. Deletes the parameter group.
 */
public class RDSScenario {
    public static long sleepTime = 20;
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException {
        final String usage = """

                Usage:
                    <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
 <dbName> <dbSnapshotIdentifier> <secretName>

                Where:
                    dbGroupName - The database group name.\s
                    dbParameterGroupFamily - The database parameter group name
 (for example, mysql8.0).
                    dbInstanceIdentifier - The database instance identifier\s
                    dbName - The database name.\s
                    dbSnapshotIdentifier - The snapshot identifier.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials"
                """;

        if (args.length != 6) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbGroupName = args[0];
        String dbParameterGroupFamily = args[1];
        String dbInstanceIdentifier = args[2];
        String dbName = args[3];
        String dbSnapshotIdentifier = args[4];
        String secretName = args[5];

        Gson gson = new Gson();
```

```
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
        String masterUsername = user.getUsername();
        String masterUserPassword = user.getPassword();

        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon RDS example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("1. Return a list of the available DB engines");
        describeDBEngines(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create a custom parameter group");
        createDBParameterGroup(rdsClient, dbGroupName, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get the parameter group");
        describeDbParameterGroups(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get the parameters in the group");
        describeDbParameters(rdsClient, dbGroupName, 0);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Modify the auto_increment_offset parameter");
        modifyDBParas(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Display the updated value");
        describeDbParameters(rdsClient, dbGroupName, -1);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("7. Get a list of allowed engine versions");
        getAllowedEngines(rdsClient, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Get a list of micro instance classes available for
the selected engine");
        getMicroInstances(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "9. Create an RDS database instance that contains a MySql
database and uses the parameter group");
        String dbARN = createDatabaseInstance(rdsClient, dbGroupName,
dbInstanceIdentifier, dbName, masterUsername,
                masterUserPassword);
        System.out.println("The ARN of the new database is " + dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Wait for DB instance to be ready");
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Create a snapshot of the DB instance");
        createSnapshot(rdsClient, dbInstanceIdentifier, dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Wait for DB snapshot to be ready");
        waitForSnapshotReady(rdsClient, dbInstanceIdentifier,
dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Delete the DB instance");
        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Delete the parameter group");
        deleteParaGroup(rdsClient, dbGroupName, dbARN);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Scenario has successfully completed.");
        System.out.println(DASHES);

        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
            .region(region)
            .build();
    }

    public static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                .secretId(secretName)
                .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    // Delete the parameter group after database has been deleted.
    // An exception is thrown if you attempt to delete the para group while
database
    // exists.
    public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
            throws InterruptedException {
        try {
            boolean isDataDel = false;
            boolean didFind;
            String instanceARN;

            // Make sure that the database has been deleted.
            while (!isDataDel) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
                List<DBInstance> instanceList = response.dbInstances();
```

```
                int listSize = instanceList.size();
                didFind = false;
                int index = 1;
                for (DBInstance instance : instanceList) {
                    instanceARN = instance.dbInstanceArn();
                    if (instanceARN.compareTo(dbARN) == 0) {
                        System.out.println(dbARN + " still exists");
                        didFind = true;
                    }
                    if ((index == listSize) && (!didFind)) {
                        // Went through the entire list and did not find the
database ARN.
                        isDataDel = true;
                    }
                    Thread.sleep(sleepTime * 1000);
                    index++;
                }
            }

            // Delete the para group.
            DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .build();

            rdsClient.deleteDBParameterGroup(parameterGroupRequest);
            System.out.println(dbGroupName + " was deleted.");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Delete the DB instance.
    public static void deleteDatabaseInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();
```

```
            DeleteDbInstanceResponse response =
rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the snapshot instance is available.
    public static void waitForSnapshotReady(RdsClient rdsClient, String
dbInstanceIdentifier,
            String dbSnapshotIdentifier) {
        try {
            boolean snapshotReady = false;
            String snapshotReadyStr;
            System.out.println("Waiting for the snapshot to become available.");

            DescribeDbSnapshotsRequest snapshotsRequest =
DescribeDbSnapshotsRequest.builder()
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            while (!snapshotReady) {
                DescribeDbSnapshotsResponse response =
rdsClient.describeDBSnapshots(snapshotsRequest);
                List<DBSnapshot> snapshotList = response.dbSnapshots();
                for (DBSnapshot snapshot : snapshotList) {
                    snapshotReadyStr = snapshot.status();
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }

            System.out.println("The Snapshot is available!");
        } catch (RdsException | InterruptedException e) {
```

```
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Create an Amazon RDS snapshot.
    public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
        try {
            CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .build();

            CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
            System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            String endpoint = "";
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
```

```
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint().address();
                        instanceReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available! The connection
endpoint is " + endpoint);

        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Create a database instance and return the ARN of the database.
    public static String createDatabaseInstance(RdsClient rdsClient,
            String dbGroupName,
            String dbInstanceIdentifier,
            String dbName,
            String userName,
            String userPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                .dbInstanceIdentifier(dbInstanceIdentifier)
                .allocatedStorage(100)
                .dbName(dbName)
                .engine("mysql")
                .dbInstanceClass("db.t3.medium") // Updated to a supported class
                .engineVersion("8.0.32")         // Updated to a supported
version
                .storageType("gp2")             // Changed to General Purpose SSD
(gp2)
                .masterUsername(userName)
                .masterUserPassword(userPassword)
                .build();
```

```
            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());
            return response.dbInstance().dbInstanceArn();

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }

        return "";
    }

    // Get a list of micro instances.
    public static void getMicroInstances(RdsClient rdsClient) {
        try {
            DescribeOrderableDbInstanceOptionsRequest dbInstanceOptionsRequest =
DescribeOrderableDbInstanceOptionsRequest
                    .builder()
                    .engine("mysql")
                    .build();

            DescribeOrderableDbInstanceOptionsResponse response = rdsClient

.describeOrderableDBInstanceOptions(dbInstanceOptionsRequest);
            List<OrderableDBInstanceOption> orderableDBInstances =
response.orderableDBInstanceOptions();
            for (OrderableDBInstanceOption dbInstanceOption :
orderableDBInstances) {
                System.out.println("The engine version is " +
dbInstanceOption.engineVersion());
                System.out.println("The engine description is " +
dbInstanceOption.engine());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Get a list of allowed engine versions.
```

```
   public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
       try {
           DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .engine("mysql")
                   .build();

           DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
           List<DBEngineVersion> dbEngines = response.dbEngineVersions();
           for (DBEngineVersion dbEngine : dbEngines) {
               System.out.println("The engine version is " +
dbEngine.engineVersion());
               System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
           }

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }

   // Modify auto_increment_offset and auto_increment_increment parameters.
   public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
       try {
           Parameter parameter1 = Parameter.builder()
                   .parameterName("auto_increment_offset")
                   .applyMethod("immediate")
                   .parameterValue("5")
                   .build();

           List<Parameter> paraList = new ArrayList<>();
           paraList.add(parameter1);
           ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .parameters(paraList)
                   .build();

           ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
```

```
            System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Retrieve parameters in the group.
    public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
        try {
            DescribeDbParametersRequest dbParameterGroupsRequest;
            if (flag == 0) {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .build();
            } else {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .source("user")
                        .build();
            }

            DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
            List<Parameter> dbParameters = response.parameters();
            String paraName;
            for (Parameter para : dbParameters) {
                // Only print out information about either auto_increment_offset
or
                // auto_increment_increment.
                paraName = para.parameterName();
                if ((paraName.compareTo("auto_increment_offset") == 0)
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
```

```
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
        try {
            DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .maxRecords(20)
                    .build();

            DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
            List<DBParameterGroup> groups = response.dbParameterGroups();
            for (DBParameterGroup group : groups) {
                System.out.println("The group name is " +
group.dbParameterGroupName());
                System.out.println("The group description is " +
group.description());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
        try {
            CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .description("Created by using the AWS SDK for Java")
```

```
                    .build();

            CreateDbParameterGroupResponse response =
 rdsClient.createDBParameterGroup(groupRequest);
            System.out.println("The group name is " +
 response.dbParameterGroup().dbParameterGroupName());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void describeDBEngines(RdsClient rdsClient) {
        try {
            DescribeDbEngineVersionsRequest engineVersionsRequest =
 DescribeDbEngineVersionsRequest.builder()
                    .defaultOnly(true)
                    .engine("mysql")
                    .maxRecords(20)
                    .build();

            DescribeDbEngineVersionsResponse response =
 rdsClient.describeDBEngineVersions(engineVersionsRequest);
            List<DBEngineVersion> engines = response.dbEngineVersions();

            // Get all DBEngineVersion objects.
            for (DBEngineVersion engineOb : engines) {
                System.out.println("The name of the DB parameter group family for
 the database engine is "
                        + engineOb.dbParameterGroupFamily());
                System.out.println("The name of the database engine " +
 engineOb.engine());
                System.out.println("The version number of the database engine " +
 engineOb.engineVersion());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK for Java 2.x -API-Referenz.
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance
  - DBParameterGruppe löschen
  - DBEngineVersionen beschreiben
  - Beschreiben DBInstances
  - Beschreiben Sie DBParameter Gruppen
  - Beschreiben DBParameters
  - Beschreiben DBSnapshots
  - DescribeOrderableDBInstanceOptionen
  - DBParameterGruppe ändern

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
/**
Before running this code example, set up your development environment, including
 your credentials.

For more information, see the following documentation topic:

https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
```

```
This example requires an AWS Secrets Manager secret that contains the database
 credentials. If you do not create a
secret, this example will not work. For more details, see:

https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html

This example performs the following tasks:

1. Returns a list of the available DB engines by invoking the
 DescribeDbEngineVersions method.
2. Selects an engine family and create a custom DB parameter group by invoking
 the createDBParameterGroup method.
3. Gets the parameter groups by invoking the DescribeDbParameterGroups method.
4. Gets parameters in the group by invoking the DescribeDbParameters method.
5. Modifies both the auto_increment_offset and auto_increment_increment
 parameters by invoking the modifyDbParameterGroup method.
6. Gets and displays the updated parameters.
7. Gets a list of allowed engine versions by invoking the
 describeDbEngineVersions method.
8. Gets a list of micro instance classes available for the selected engine.
9. Creates an Amazon Relational Database Service (Amazon RDS) database instance
 that contains a MySQL database and uses the parameter group.
10. Waits for DB instance to be ready and prints out the connection endpoint
 value.
11. Creates a snapshot of the DB instance.
12. Waits for the DB snapshot to be ready.
13. Deletes the DB instance.
14. Deletes the parameter group.
 */

var sleepTime: Long = 20

suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
 <dbName> <dbSnapshotIdentifier><secretName>

        Where:
            dbGroupName - The database group name.
            dbParameterGroupFamily - The database parameter group name.
            dbInstanceIdentifier - The database instance identifier.
            dbName -  The database name.
```

```
            dbSnapshotIdentifier - The snapshot identifier.
            secretName - The name of the AWS Secrets Manager secret that contains
the database credentials.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val dbGroupName = args[0]
    val dbParameterGroupFamily = args[1]
    val dbInstanceIdentifier = args[2]
    val dbName = args[3]
    val dbSnapshotIdentifier = args[4]
    val secretName = args[5]

    val gson = Gson()
    val user = gson.fromJson(getSecretValues(secretName).toString(),
User::class.java)
    val username = user.username
    val userPassword = user.password

    println("1. Return a list of the available DB engines")
    describeDBEngines()

    println("2. Create a custom parameter group")
    createDBParameterGroup(dbGroupName, dbParameterGroupFamily)

    println("3. Get the parameter groups")
    describeDbParameterGroups(dbGroupName)

    println("4. Get the parameters in the group")
    describeDbParameters(dbGroupName, 0)

    println("5. Modify the auto_increment_offset parameter")
    modifyDBParas(dbGroupName)

    println("6. Display the updated value")
    describeDbParameters(dbGroupName, -1)

    println("7. Get a list of allowed engine versions")
    getAllowedEngines(dbParameterGroupFamily)
```

```
    println("8. Get a list of micro instance classes available for the selected
 engine")
    getMicroInstances()

    println("9. Create an RDS database instance that contains a MySql database
 and uses the parameter group")
    val dbARN = createDatabaseInstance(dbGroupName, dbInstanceIdentifier, dbName,
 username, userPassword)
    println("The ARN of the new database is $dbARN")

    println("10. Wait for DB instance to be ready")
    waitForDbInstanceReady(dbInstanceIdentifier)

    println("11. Create a snapshot of the DB instance")
    createDbSnapshot(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("12. Wait for DB snapshot to be ready")
    waitForSnapshotReady(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("13. Delete the DB instance")
    deleteDbInstance(dbInstanceIdentifier)

    println("14. Delete the parameter group")
    if (dbARN != null) {
        deleteParaGroup(dbGroupName, dbARN)
    }

    println("The Scenario has successfully completed.")
}

suspend fun deleteParaGroup(
    dbGroupName: String,
    dbARN: String,
) {
    var isDataDel = false
    var didFind: Boolean
    var instanceARN: String

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        // Make sure that the database has been deleted.
        while (!isDataDel) {
            val response = rdsClient.describeDbInstances()
            val instanceList = response.dbInstances
            val listSize = instanceList?.size
```

```
            isDataDel = false // Reset this value.
            didFind = false // Reset this value.
            var index = 1
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceARN = instance.dbInstanceArn.toString()
                    if (instanceARN.compareTo(dbARN) == 0) {
                        println("$dbARN still exists")
                        didFind = true
                    }
                    if (index == listSize && !didFind) {
                        // Went through the entire list and did not find the
 database name.
                        isDataDel = true
                    }
                    index++
                }
            }
        }

        // Delete the para group.
        val parameterGroupRequest =
            DeleteDbParameterGroupRequest {
                dbParameterGroupName = dbGroupName
            }
        rdsClient.deleteDbParameterGroup(parameterGroupRequest)
        println("$dbGroupName was deleted.")
    }
}

suspend fun deleteDbInstance(dbInstanceIdentifierVal: String) {
    val deleteDbInstanceRequest =
        DeleteDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            deleteAutomatedBackups = true
            skipFinalSnapshot = true
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}
```

```
// Waits until the snapshot instance is available.
suspend fun waitForSnapshotReady(
    dbInstanceIdentifierVal: String?,
    dbSnapshotIdentifierVal: String?,
) {
    var snapshotReady = false
    var snapshotReadyStr: String
    println("Waiting for the snapshot to become available.")

    val snapshotsRequest =
        DescribeDbSnapshotsRequest {
            dbSnapshotIdentifier = dbSnapshotIdentifierVal
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }

    while (!snapshotReady) {
        RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbSnapshots(snapshotsRequest)
            val snapshotList: List<DbSnapshot>? = response.dbSnapshots
            if (snapshotList != null) {
                for (snapshot in snapshotList) {
                    snapshotReadyStr = snapshot.status.toString()
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("The Snapshot is available!")
}

// Create an Amazon RDS snapshot.
suspend fun createDbSnapshot(
    dbInstanceIdentifierVal: String?,
    dbSnapshotIdentifierVal: String?,
) {
    val snapshotRequest =
        CreateDbSnapshotRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
```

```
            dbSnapshotIdentifier = dbSnapshotIdentifierVal
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbSnapshot(snapshotRequest)
        print("The Snapshot id is ${response.dbSnapshot?.dbiResourceId}")
    }
}

// Waits until the database instance is available.
suspend fun waitForDbInstanceReady(dbInstanceIdentifierVal: String?) {
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest =
        DescribeDbInstancesRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }
    var endpoint = ""
    while (!instanceReady) {
        RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint?.address.toString()
                        instanceReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("Database instance is available! The connection endpoint is
 $endpoint")
}

// Create a database instance and return the ARN of the database.
suspend fun createDatabaseInstance(
```

```
    dbGroupNameVal: String?,
    dbInstanceIdentifierVal: String?,
    dbNameVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
): String? {
    val instanceRequest =
        CreateDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            allocatedStorage = 100
            dbName = dbNameVal
            dbParameterGroupName = dbGroupNameVal
            engine = "mysql"
            dbInstanceClass = "db.t3.micro"
            engineVersion = "8.0.35"
            storageType = "gp2"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
        return response.dbInstance?.dbInstanceArn
    }
}

// Get a list of micro instances.
suspend fun getMicroInstances() {
    val dbInstanceOptionsRequest =
        DescribeOrderableDbInstanceOptionsRequest {
            engine = "mysql"
        }
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeOrderableDbInstanceOptions(dbInstanceOptionsRequest)
        val orderableDBInstances = response.orderableDbInstanceOptions
        if (orderableDBInstances != null) {
            for (dbInstanceOption in orderableDBInstances) {
                println("The engine version is
 ${dbInstanceOption.engineVersion}")
                println("The engine description is ${dbInstanceOption.engine}")
            }
        }
```

```
    }
}

// Get a list of allowed engine versions.
suspend fun getAllowedEngines(dbParameterGroupFamilyVal: String?) {
    val versionsRequest =
        DescribeDbEngineVersionsRequest {
            dbParameterGroupFamily = dbParameterGroupFamilyVal
            engine = "mysql"
        }
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(versionsRequest)
        val dbEngines: List<DbEngineVersion>? = response.dbEngineVersions
        if (dbEngines != null) {
            for (dbEngine in dbEngines) {
                println("The engine version is ${dbEngine.engineVersion}")
                println("The engine description is
 ${dbEngine.dbEngineDescription}")
            }
        }
    }
}

// Modify the auto_increment_offset parameter.
suspend fun modifyDBParas(dbGroupName: String) {
    val parameter1 =
        Parameter {
            parameterName = "auto_increment_offset"
            applyMethod = ApplyMethod.Immediate
            parameterValue = "5"
        }

    val paraList: ArrayList<Parameter> = ArrayList()
    paraList.add(parameter1)
    val groupRequest =
        ModifyDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
            parameters = paraList
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.modifyDbParameterGroup(groupRequest)
        println("The parameter group ${response.dbParameterGroupName} was
 successfully modified")
```

```
    }
}

// Retrieve parameters in the group.
suspend fun describeDbParameters(
    dbGroupName: String?,
    flag: Int,
) {
    val dbParameterGroupsRequest: DescribeDbParametersRequest
    dbParameterGroupsRequest =
        if (flag == 0) {
            DescribeDbParametersRequest {
                dbParameterGroupName = dbGroupName
            }
        } else {
            DescribeDbParametersRequest {
                dbParameterGroupName = dbGroupName
                source = "user"
            }
        }
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameters(dbParameterGroupsRequest)
        val dbParameters: List<Parameter>? = response.parameters
        var paraName: String
        if (dbParameters != null) {
            for (para in dbParameters) {
                // Only print out information about either auto_increment_offset
 or auto_increment_increment.
                paraName = para.parameterName.toString()
                if (paraName.compareTo("auto_increment_offset") == 0 ||
 paraName.compareTo("auto_increment_increment ") == 0) {
                    println("*** The parameter name is  $paraName")
                    System.out.println("*** The parameter value is
 ${para.parameterValue}")
                    System.out.println("*** The parameter data type is
 ${para.dataType}")
                    System.out.println("*** The parameter description is
 ${para.description}")
                    System.out.println("*** The parameter allowed values  is
 ${para.allowedValues}")
                }
            }
        }
    }
```

```
}

suspend fun describeDbParameterGroups(dbGroupName: String?) {
    val groupsRequest =
        DescribeDbParameterGroupsRequest {
            dbParameterGroupName = dbGroupName
            maxRecords = 20
        }
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameterGroups(groupsRequest)
        val groups = response.dbParameterGroups
        if (groups != null) {
            for (group in groups) {
                println("The group name is ${group.dbParameterGroupName}")
                println("The group description is ${group.description}")
            }
        }
    }
}

// Create a parameter group.
suspend fun createDBParameterGroup(
    dbGroupName: String?,
    dbParameterGroupFamilyVal: String?,
) {
    val groupRequest =
        CreateDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
            dbParameterGroupFamily = dbParameterGroupFamilyVal
            description = "Created by using the AWS SDK for Kotlin"
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbParameterGroup(groupRequest)
        println("The group name is
 ${response.dbParameterGroup?.dbParameterGroupName}")
    }
}

// Returns a list of the available DB engines.
suspend fun describeDBEngines() {
    val engineVersionsRequest =
        DescribeDbEngineVersionsRequest {
            defaultOnly = true
```

```
            engine = "mysql"
            maxRecords = 20
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(engineVersionsRequest)
        val engines: List<DbEngineVersion>? = response.dbEngineVersions

        // Get all DbEngineVersion objects.
        if (engines != null) {
            for (engineOb in engines) {
                println("The name of the DB parameter group family for the
 database engine is ${engineOb.dbParameterGroupFamily}.")
                println("The name of the database engine ${engineOb.engine}.")
                println("The version number of the database engine
 ${engineOb.engineVersion}")
            }
        }
    }
}

suspend fun getSecretValues(secretName: String?): String? {
    val valueRequest =
        GetSecretValueRequest {
            secretId = secretName
        }

    SecretsManagerClient.fromEnvironment { region = "us-west-2" }.use
 { secretsClient ->
        val valueResponse = secretsClient.getSecretValue(valueRequest)
        return valueResponse.secretString
    }
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Kotlin.
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance

- DBParameterGruppe löschen
- DBEngineVersionen beschreiben
- Beschreiben DBInstances
- Beschreiben Sie DBParameter Gruppen
- Beschreiben DBParameters
- Beschreiben DBSnapshots
- DescribeOrderableDBInstanceOptionen
- DBParameterGruppe ändern

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Führen Sie ein interaktives Szenario an einem Prompt aus.

```
class RdsInstanceScenario:
    """Runs a scenario that shows how to get started using Amazon RDS DB
 instances."""

    def __init__(self, instance_wrapper):
        """
        :param instance_wrapper: An object that wraps Amazon RDS DB instance
 actions.
        """
        self.instance_wrapper = instance_wrapper

    def create_parameter_group(self, parameter_group_name, db_engine):
        """
        Shows how to get available engine versions for a specified database
 engine and
        create a DB parameter group that is compatible with a selected engine
 family.
```

```
        :param parameter_group_name: The name given to the newly created
parameter group.
        :param db_engine: The database engine to use as a basis.
        :return: The newly created parameter group.
        """
        print(
            f"Checking for an existing DB instance parameter group named
{parameter_group_name}."
        )
        parameter_group = self.instance_wrapper.get_parameter_group(
            parameter_group_name
        )
        if parameter_group is None:
            print(f"Getting available database engine versions for {db_engine}.")
            engine_versions =
self.instance_wrapper.get_engine_versions(db_engine)
            families = list({ver["DBParameterGroupFamily"] for ver in
engine_versions})
            family_index = q.choose("Which family do you want to use? ",
families)
            print(f"Creating a parameter group.")
            self.instance_wrapper.create_parameter_group(
                parameter_group_name, families[family_index], "Example parameter
group."
            )
            parameter_group = self.instance_wrapper.get_parameter_group(
                parameter_group_name
            )
        print(f"Parameter group {parameter_group['DBParameterGroupName']}:")
        pp(parameter_group)
        print("-" * 88)
        return parameter_group

    def update_parameters(self, parameter_group_name):
        """
        Shows how to get the parameters contained in a custom parameter group and
        update some of the parameter values in the group.

        :param parameter_group_name: The name of the parameter group to query and
modify.
        """
        print("Let's set some parameter values in your parameter group.")
        auto_inc_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, name_prefix="auto_increment"
```

```
        )
        update_params = []
        for auto_inc in auto_inc_parameters:
            if auto_inc["IsModifiable"] and auto_inc["DataType"] == "integer":
                print(f"The {auto_inc['ParameterName']} parameter is described
as:")
                print(f"\t{auto_inc['Description']}")
                param_range = auto_inc["AllowedValues"].split("-")
                auto_inc["ParameterValue"] = str(
                    q.ask(
                        f"Enter a value between {param_range[0]} and
{param_range[1]}: ",
                        q.is_int,
                        q.in_range(int(param_range[0]), int(param_range[1])),
                    )
                )
                update_params.append(auto_inc)
        self.instance_wrapper.update_parameters(parameter_group_name,
update_params)
        print(
            "You can get a list of parameters you've set by specifying a source
of 'user'."
        )
        user_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, source="user"
        )
        pp(user_parameters)
        print("-" * 88)

    def create_instance(self, instance_name, db_name, db_engine,
parameter_group):
        """
        Shows how to create a DB instance that contains a database of a specified
        type and is configured to use a custom DB parameter group.

        :param instance_name: The name given to the newly created DB instance.
        :param db_name: The name given to the created database.
        :param db_engine: The engine of the created database.
        :param parameter_group: The parameter group that is associated with the
DB instance.
        :return: The newly created DB instance.
        """
        print("Checking for an existing DB instance.")
        db_inst = self.instance_wrapper.get_db_instance(instance_name)
```

```
        if db_inst is None:
            print("Let's create a DB instance.")
            admin_username = q.ask(
                "Enter an administrator user name for the database: ",
q.non_empty
            )
            admin_password = q.ask(
                "Enter a password for the administrator (at least 8 characters):
",
                q.non_empty,
            )
            engine_versions = self.instance_wrapper.get_engine_versions(
                db_engine, parameter_group["DBParameterGroupFamily"]
            )
            engine_choices = [ver["EngineVersion"] for ver in engine_versions]
            print("The available engines for your parameter group are:")
            engine_index = q.choose("Which engine do you want to use? ",
engine_choices)
            engine_selection = engine_versions[engine_index]
            print(
                "The available micro DB instance classes for your database engine
are:"
            )
            inst_opts = self.instance_wrapper.get_orderable_instances(
                engine_selection["Engine"], engine_selection["EngineVersion"]
            )
            inst_choices = list(
                {
                    opt["DBInstanceClass"]
                    for opt in inst_opts
                    if "micro" in opt["DBInstanceClass"]
                }
            )
            inst_index = q.choose(
                "Which micro DB instance class do you want to use? ",
inst_choices
            )
            group_name = parameter_group["DBParameterGroupName"]
            storage_type = "standard"
            allocated_storage = 5
            print(
                f"Creating a DB instance named {instance_name} and database
{db_name}.\n"
```

```
                f"The DB instance is configured to use your custom parameter
 group {group_name},\n"
                f"selected engine {engine_selection['EngineVersion']},\n"
                f"selected DB instance class {inst_choices[inst_index]},"
                f"and {allocated_storage} GiB of {storage_type} storage.\n"
                f"This typically takes several minutes."
            )
            db_inst = self.instance_wrapper.create_db_instance(
                db_name,
                instance_name,
                group_name,
                engine_selection["Engine"],
                engine_selection["EngineVersion"],
                inst_choices[inst_index],
                storage_type,
                allocated_storage,
                admin_username,
                admin_password,
            )
            while db_inst.get("DBInstanceStatus") != "available":
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(instance_name)
        print("Instance data:")
        pp(db_inst)
        print("-" * 88)
        return db_inst

    @staticmethod
    def display_connection(db_inst):
        """
        Displays connection information about a DB instance and tips on how to
        connect to it.

        :param db_inst: The DB instance to display.
        """
        print(
            "You can now connect to your database using your favorite MySql
 client.\n"
            "One way to connect is by using the 'mysql' shell on an Amazon EC2
 instance\n"
            "that is running in the same VPC as your DB instance. Pass the
 endpoint,\n"
            "port, and administrator user name to 'mysql' and enter your password
\n"
```

```
            "when prompted:\n"
        )
        print(
            f"\n\tmysql -h {db_inst['Endpoint']['Address']} -P
 {db_inst['Endpoint']['Port']} "
            f"-u {db_inst['MasterUsername']} -p\n"
        )
        print(
            "For more information, see the User Guide for Amazon RDS:\n"
            "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL"
        )
        print("-" * 88)

    def create_snapshot(self, instance_name):
        """
        Shows how to create a DB instance snapshot and wait until it's available.

        :param instance_name: The name of a DB instance to snapshot.
        """
        if q.ask(
            "Do you want to create a snapshot of your DB instance (y/n)? ",
 q.is_yesno
        ):
            snapshot_id = f"{instance_name}-{uuid.uuid4()}"
            print(
                f"Creating a snapshot named {snapshot_id}. This typically takes a
 few minutes."
            )
            snapshot = self.instance_wrapper.create_snapshot(snapshot_id,
 instance_name)
            while snapshot.get("Status") != "available":
                wait(10)
                snapshot = self.instance_wrapper.get_snapshot(snapshot_id)
            pp(snapshot)
            print("-" * 88)

    def cleanup(self, db_inst, parameter_group_name):
        """
        Shows how to clean up a DB instance and parameter group.
        Before the parameter group can be deleted, all associated DB instances
 must first
        be deleted.
```

```
        :param db_inst: The DB instance to delete.
        :param parameter_group_name: The DB parameter group to delete.
        """
        if q.ask(
            "\nDo you want to delete the DB instance and parameter group (y/n)?
",
            q.is_yesno,
        ):
            print(f"Deleting DB instance {db_inst['DBInstanceIdentifier']}.")

self.instance_wrapper.delete_db_instance(db_inst["DBInstanceIdentifier"])
            print(
                "Waiting for the DB instance to delete. This typically takes
several minutes."
            )
            while db_inst is not None:
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(
                    db_inst["DBInstanceIdentifier"]
                )
            print(f"Deleting parameter group {parameter_group_name}.")
            self.instance_wrapper.delete_parameter_group(parameter_group_name)

    def run_scenario(self, db_engine, parameter_group_name, instance_name,
db_name):
        logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

        print("-" * 88)
        print(
            "Welcome to the Amazon Relational Database Service (Amazon RDS)\n"
            "get started with DB instances demo."
        )
        print("-" * 88)

        parameter_group = self.create_parameter_group(parameter_group_name,
db_engine)
        self.update_parameters(parameter_group_name)
        db_inst = self.create_instance(
            instance_name, db_name, db_engine, parameter_group
        )
        self.display_connection(db_inst)
        self.create_snapshot(instance_name)
        self.cleanup(db_inst, parameter_group_name)
```

```
        print("\nThanks for watching!")
        print("-" * 88)


if __name__ == "__main__":
    try:
        scenario = RdsInstanceScenario(InstanceWrapper.from_client())
        scenario.run_scenario(
            "mysql",
            "doc-example-parameter-group",
            "doc-example-instance",
            "docexampledb",
        )
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Definieren Sie Funktionen, die vom Szenario aufgerufen werden, um Amazon-RDS-Aktionen zu verwalten.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.
```

```
        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
group
        family.

        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
```

```
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
```

```
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
Parameters=update_parameters
            )
```

```
        except ClientError as err:
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
```

```
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
list of
                                       engine versions to those that are
compatible with
                                       this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise
        else:
            return versions


    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
```

```
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
```

```
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
```

```
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Python (Boto3).
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance
  - DBParameterGruppe löschen
  - DBEngineVersionen beschreiben
  - Beschreiben DBInstances
  - Beschreiben Sie DBParameter Gruppen
  - Beschreiben DBParameters
  - Beschreiben DBSnapshots
  - DescribeOrderableDBInstanceOptionen
  - DBParameterGruppe ändern

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Die Datei `Package.swift`

```
// swift-tools-version: 5.9
//
// The swift-tools-version declares the minimum version of Swift required to
// build this package.

import PackageDescription

let package = Package(
    name: "rds-scenario",
    // Let Xcode know the minimum Apple platforms supported.
    platforms: [
        .macOS(.v13),
        .iOS(.v15)
    ],
    dependencies: [
        // Dependencies declare other packages that this package depends on.
        .package(
            url: "https://github.com/awslabs/aws-sdk-swift",
            from: "1.4.0"),
        .package(
            url: "https://github.com/apple/swift-argument-parser.git",
            branch: "main"
        )
    ],
    targets: [
        // Targets are the basic building blocks of a package, defining a module
 or a test suite.
        // Targets can depend on other targets in this package and products
        // from dependencies.
        .executableTarget(
            name: "rds-scenario",
```

```
            dependencies: [
                .product(name: "AWSRDS", package: "aws-sdk-swift"),
                .product(name: "ArgumentParser", package: "swift-argument-
parser")
            ],
            path: "Sources")

    ]
)
```

Die Swift-Codedatei,`entry.swift`.

```
// An example that shows how to use the AWS SDK for Swift to perform a variety
// of operations using Amazon Relational Database Service (RDS).
//

import ArgumentParser
import Foundation
import AWSRDS

struct ExampleCommand: ParsableCommand {
    @Option(help: "The AWS Region to run AWS API calls in.")
    var awsRegion = "us-east-1"
    @Option(help: "The username to use for the database administrator.")
    var dbUsername = "admin"
    @Option(help: "The password to use for the database administrator.")
    var dbPassword: String

    static var configuration = CommandConfiguration(
        commandName: "rds-scenario",
        abstract: """
        Performs various operations to demonstrate the use of Amazon RDS
 Instances
        using the AWS SDK for Swift.
        """,
        discussion: """
        """
    )

    /// Called by ``main()`` to run the bulk of the example.
    func runAsync() async throws {
```

```
        let example = try await Example(region: awsRegion, username: dbUsername,
 password: dbPassword)

        await example.run()
    }
}

class Example {
    let rdsClient: RDSClient

    // Storage for AWS RDS properties

    let dbUsername: String
    let dbPassword: String
    var dbInstanceIdentifier: String
    var dbSnapshotIdentifier: String
    var dbParameterGroupName: String
    var dbParameterGroup: RDSClientTypes.DBParameterGroup?
    var selectedEngineVersion: String?

    init(region: String, username: String, password: String) async throws{
        let rdsConfig = try await RDSClient.RDSClientConfiguration(region:
 region)
        rdsClient = RDSClient(config: rdsConfig)

        dbUsername = username
        dbPassword = password
        dbParameterGroupName = ""
        dbInstanceIdentifier = ""
        dbSnapshotIdentifier = ""
    }

    /// The example's main body.
    func run() async {
        var parameterGroupFamilies: Set<String> = []

        //=========================================================================
        // 1. Get available database engine families for MySQL.
        //=========================================================================

        let engineVersions = await getDBEngineVersions(engineName: "mysql")

        for version in engineVersions {
            if version.dbParameterGroupFamily != nil {
```

```
                parameterGroupFamilies.insert(version.dbParameterGroupFamily!)
            }
        }

        if engineVersions.count > 0 {
            selectedEngineVersion = engineVersions.last!.engineVersion
        } else {
            print("*** Unable to find a valid database engine version. Canceling
operations.")
            await cleanUp()
            return
        }

        print("Found \(parameterGroupFamilies.count) parameter group families:")
        for family in parameterGroupFamilies {
            print("    \(family)")
        }

        //========================================================================
        // 2. Select an engine family and create a custom DB parameter group.
        //    We select a family by sorting the set of family names, then
        //    choosing the last one.
        //========================================================================

        let sortedFamilies = parameterGroupFamilies.sorted()

        guard let selectedFamily = sortedFamilies.last else {
            print("*** Unable to find a database engine family. Canceling
operations.")
            await cleanUp()
            return
        }

        print("Selected database engine family \(selectedFamily)")

        dbParameterGroupName = tempName(prefix: "rds-example")
        print("Creating a database parameter group named \(dbParameterGroupName)
using \(selectedFamily)")
        dbParameterGroup = await createDBParameterGroup(groupName:
dbParameterGroupName,
                                                        familyName:
selectedFamily)

        //========================================================================
```

```
        // 3. Get the parameter group's details.
        //===========================================================================

        print("Getting the database parameter group list...")
        let dbParameterGroupList = await describeDBParameterGroups(groupName:
dbParameterGroupName)
        guard let dbParameterGroupList else {
            await cleanUp()
            return
        }

        print("Found \(dbParameterGroupList.count) parameter groups...")
        for group in dbParameterGroupList {
            print("    \(group.dbParameterGroupName ?? "<unknown>")")
        }
        print()

        //===========================================================================
        // 4. Get a list of the parameter group's parameters. This list is
        //    likely to be long, so use pagination. Find the
        //    auto_increment_offset and auto_increment_increment parameters.
        //===========================================================================

        let parameters = await describeDBParameters(groupName:
dbParameterGroupName)

        //===========================================================================
        // 5. Parse and display each parameter's name, description, and
        //    allowed values.
        //===========================================================================

        for parameter in parameters {
            let name = parameter.parameterName
            guard let name else {
                print("*** Unable to get parameter name!")
                continue
            }

            if name == "auto_increment_offset" || name ==
"auto_increment_increment" {
                print("Parameter \(name):")
                print("          Value: \(parameter.parameterValue ??
"<undefined>")")
                print("      Data type: \(parameter.dataType ?? "<unknown>")")
```

```
                print("    Description: \(parameter.description ?? "")")
                print(" Allowed values: \(parameter.allowedValues ??
"<unspecified")")
                print(String(repeating: "=", count: 78))
            }
        }

        //===========================================================================
        // 6. Modify both the auto_increment_offset and
        //    auto_increment_increment parameters in one call in the custom
        //    parameter group. Set their parameterValue fields to a new
        //    permitted value.
        //===========================================================================

        print("Setting auto_increment_offset and auto_increment_increment both to
5...")
        await modifyDBParameters(groupName: dbParameterGroupName)

        //===========================================================================
        // 7. Get and display the updated parameters, specifying a source of
        //    "user" to get only the modified parameters.
        //===========================================================================

        let updatedParameters = await describeDBParameters(groupName:
dbParameterGroupName, source: "user")

        for parameter in updatedParameters {
            let name = parameter.parameterName
            guard let name else {
                print("*** Unable to get parameter name!")
                continue
            }

            print("Parameter \(name):")
            print("          Value: \(parameter.parameterValue ??
"<undefined>")")
            print("      Data type: \(parameter.dataType ?? "<unknown>")")
            print("    Description: \(parameter.description ?? "")")
            print(" Allowed values: \(parameter.allowedValues ??
"<unspecified")")
            print(String(repeating: "=", count: 78))
        }

        //===========================================================================
```

```
        // 8. Get a list of allowed engine versions using
        //    DescribeRDSEngineVersions.
        //=========================================================================

        await listAllowedEngines(family: selectedFamily)

        //=========================================================================
        // 9. Get a list of micro instance classes available for the selected
        //    engine and engine version.
        //=========================================================================

        let dbInstanceClass = await chooseMicroInstance(engine: "mysql",
engineVersion: selectedEngineVersion)
        guard let dbInstanceClass else {
            print("Did not get a valid instance class. Canceling operations.")
            await cleanUp()
            return
        }

        //=========================================================================
        // 10. Create an RDS database that contains a MySQL database and uses
        //     the parameter group we created.
        //=========================================================================

        print("Creating the database instance...")

        guard let instanceClass = dbInstanceClass.dbInstanceClass else {
            print("Instance class name is unknown. Canceling operations.")
            await cleanUp()
            return
        }

        dbInstanceIdentifier = tempName(prefix: "sample-identifier")
        let dbInstanceArn = await createDBInstance(
            name: "SampleDatabase\(Int.random(in: 1000000..<1000000000))",
            instanceIdentifier: dbInstanceIdentifier,
            parameterGroupName: dbParameterGroupName,
            engine: "mysql",
            engineVersion: selectedEngineVersion!,
            instanceClass: instanceClass,
            username: dbUsername,
            password: dbPassword
        )
```

```
        if dbInstanceArn == nil {
            await cleanUp()
            return
        }

        //========================================================================
        // 11. Wait for the database instance to be ready by calling
        //     DescribeDBInstances repeatedly until it reports
        //     dbInstanceStatus as "available". This can take upwards of 10
        //     minutes, let the user know that.
        //========================================================================

        guard let endpoint = await waitUntilDBInstanceReady(instanceIdentifier:
dbInstanceIdentifier) else {
            print("\nDid not get a valid endpoint from AWS RDS.")
            await cleanUp()
            return
        }

        guard let endpointAddress = endpoint.address else {
            print("\nNo endpoint address returned.")
            await cleanUp()
            return
        }
        guard let endpointPort = endpoint.port else {
            print("\nNo endpoint port returned.")
            await cleanUp()
            return
        }

        //========================================================================
        // 12. Display connection information for the database instance.
        //========================================================================

        print("\nTo connect to the new database instance using 'mysql' from the
shell:")
        print("    mysql -h \(endpointAddress) -P \(endpointPort) -u
\(self.dbUsername)")

        //========================================================================
        // 13. Create a snapshot of the database instance.
        //========================================================================

        dbSnapshotIdentifier = tempName(prefix: "sample-snapshot")
```

```
        await createDBSnapshot(instanceIdentifier: dbInstanceIdentifier,
snapshotIdentifier: dbSnapshotIdentifier)

        //========================================================================
        // 14. Wait for the snapshot to be ready.
        //========================================================================

        await waitUntilDBSnapshotReady(instanceIdentifier: dbInstanceIdentifier,
snapshotIdentifier: dbSnapshotIdentifier)

        // That's it! Clean up and exit!

        print("Example complete! Cleaning up...")
        await cleanUp()
    }

    /// Clean up by discarding and closing down all allocated EC2 items.
    func cleanUp() async {
        print("Deleting the database instance \(dbInstanceIdentifier)...")
        await deleteDBInstance(instanceIdentifier: dbInstanceIdentifier)
        await waitUntilDBInstanceDeleted(instanceIdentifier:
dbInstanceIdentifier)

        print("Deleting the database parameter group \(dbParameterGroupName)...")
        await deleteDBParameterGroup(groupName: dbParameterGroupName)
    }

    /// Get all the database engine versions available for the specified
    /// database engine.
    ///
    /// - Parameter engineName: The name of the database engine to query.
    ///
    /// - Returns: An array of `RDSClientTypes.DBEngineVersion` structures,
    ///   each describing one supported version of the specified database.
    func getDBEngineVersions(engineName: String) async ->
[RDSClientTypes.DBEngineVersion] {
        do {
            let output = try await rdsClient.describeDBEngineVersions(
                input: DescribeDBEngineVersionsInput(
                    engine: engineName
                )
            )

            return output.dbEngineVersions ?? []
```

```
        } catch {
            return []
        }
    }

    /// Create a new database parameter group with the specified name.
    ///
    /// - Parameters:
    ///   - groupName: The name of the new parameter group.
    ///   - familyName: The name of the parameter group family.
    /// - Returns:
    func createDBParameterGroup(groupName: String, familyName: String) async ->
RDSClientTypes.DBParameterGroup? {
        do {
            let output = try await rdsClient.createDBParameterGroup(
                input: CreateDBParameterGroupInput(
                    dbParameterGroupFamily: familyName,
                    dbParameterGroupName: groupName,
                    description: "Created using the AWS SDK for Swift"
                )
            )
            return output.dbParameterGroup
        } catch {
            print("*** Error creating the parameter group:
\(error.localizedDescription)")
            return nil
        }
    }

    /// Get descriptions of the database parameter groups matching the given
    /// name.
    ///
    /// - Parameter groupName: The name of the parameter group to describe.
    ///
    /// - Returns: An array of [RDSClientTypes.DBParameterGroup] objects
    ///   describing the parameter group.
    func describeDBParameterGroups(groupName: String) async ->
[RDSClientTypes.DBParameterGroup]? {
        do {
            let output = try await rdsClient.describeDBParameterGroups(
                input: DescribeDBParameterGroupsInput(
                    dbParameterGroupName: groupName
                )
            )
```

```
            return output.dbParameterGroups
        } catch {
            print("*** Error getting the database parameter group's details:
\(error.localizedDescription)")
            return nil
        }
    }

    /// Returns the detailed parameter list for the specified database
    /// parameter group.
    ///
    /// - Parameters:
    ///   - groupName: The name of the parameter group to return parameters for.
    ///   - source: The types of parameters to return (`user`, `system`, or
    ///     `engine-default`).
    ///
    /// - Returns: An array of `RdSClientTypes.Parameter` objects, each
    ///   describing one of the group's parameters.
    func describeDBParameters(groupName: String, source: String? = nil) async ->
[RDSClientTypes.Parameter] {
        var parameterList: [RDSClientTypes.Parameter] = []

        do {
            let pages = rdsClient.describeDBParametersPaginated(
                input: DescribeDBParametersInput(
                    dbParameterGroupName: groupName,
                    source: source
                )
            )

            for try await page in pages {
                guard let parameters = page.parameters else {
                    return []
                }

                parameterList += parameters
            }
        } catch {
            print("*** Error getting database parameters:
\(error.localizedDescription)")
            return []
        }

        return parameterList
```

```
    }

    /// Demonstrates modifying two of the specified database parameter group's
    /// parameters.
    ///
    /// - Parameter groupName: The name of the parameter group to change
    ///   parameters for.
    func modifyDBParameters(groupName: String) async {
        let parameter1 = RDSClientTypes.Parameter(
            applyMethod: RDSClientTypes.ApplyMethod.immediate,
            parameterName: "auto_increment_offset",
            parameterValue: "5"
        )
        let parameter2 = RDSClientTypes.Parameter(
            applyMethod: RDSClientTypes.ApplyMethod.immediate,
            parameterName: "auto_increment_increment",
            parameterValue: "5"
        )

        let parameterList = [parameter1, parameter2]

        do {
            _ = try await rdsClient.modifyDBParameterGroup(
                input: ModifyDBParameterGroupInput(
                    dbParameterGroupName: groupName,
                    parameters: parameterList
                )
            )

            print("Successfully modified the parameter group \(groupName).")
        } catch {
            print("*** Error modifying the parameter group \(groupName):
\(error.localizedDescription)")
        }
    }

    /// Output a list of the database engine versions supported by the
    /// specified family.
    ///
    /// - Parameter family: The family for which to list allowed database
    ///   engines.
    func listAllowedEngines(family: String?) async {
        do {
            let output = try await rdsClient.describeDBEngineVersions(
```

```
                input: DescribeDBEngineVersionsInput(
                    dbParameterGroupFamily: family,
                    engine: "mysql"
                )
            )

            guard let engineVersions = output.dbEngineVersions else {
                print("No engine versions returned.")
                return
            }

            print("Found \(engineVersions.count) database engine versions:")
            for version in engineVersions {
                print("    \(version.engineVersion ?? "<unknown>"):
\(version.dbEngineDescription ?? "")")
            }
        } catch {
            print("*** Error getting database engine version list:
\(error.localizedDescription)")
            return
        }
    }

    /// Print a list of available database instances with "micro" in the class
    /// name, then return one of them to be used by other code.
    ///
    /// - Parameters:
    ///   - engine: The database engine for which to list database instance
    ///     classes.
    ///   - engineVersion: The database version for which to list instances.
    ///
    /// - Returns: An `RDSClientTypes.OrderableDBInstanceOption` describing
    ///   the selected instance type.
    func chooseMicroInstance(engine: String = "mysql", engineVersion: String? =
nil) async -> RDSClientTypes.OrderableDBInstanceOption? {
        do {
            let pages = rdsClient.describeOrderableDBInstanceOptionsPaginated(
                input: DescribeOrderableDBInstanceOptionsInput(
                    engine: engine,
                    engineVersion: engineVersion
                )
            )

            var optionsList: [RDSClientTypes.OrderableDBInstanceOption] = []
```

```
            for try await page in pages {
                guard let orderableDBInstanceOptions =
page.orderableDBInstanceOptions else {
                    continue
                }

                for dbInstanceOption in orderableDBInstanceOptions {
                    guard let className = dbInstanceOption.dbInstanceClass else {
                        continue
                    }
                    if className.contains("micro") {
                        optionsList.append(dbInstanceOption)
                    }
                }
            }

            print("Found \(optionsList.count) database instances of 'micro' class
types:")
            for dbInstanceOption in optionsList {
                print("    \(dbInstanceOption.engine ??
"<unknown>") \(dbInstanceOption.engineVersion ?? "<unknown>")
(\(dbInstanceOption.dbInstanceClass ?? "<unknown class>"))")
            }

            return optionsList[0]
        } catch {
            print("*** Error getting a list of orderable instance options:
\(error.localizedDescription)")
            return nil
        }
    }

    /// Create a new database instance.
    ///
    /// - Parameters:
    ///   - name: The name of the database to create.
    ///   - instanceIdentifier: The identifier to give the new database
    ///     instance.
    ///   - parameterGroupName: The name of the parameter group to associate
    ///     with the new database instance.
    ///   - engine: The database engine to use.
    ///   - engineVersion: The version of the database given by `engine` to
    ///     use.
```

```
    ///   - instanceClass: The memory and compute capacity of the database
    ///     instance, such as `db.m5.large``.
    ///   - username: The admin user's username to establish for the new
    ///     instance.
    ///   - password: The password to use for the specified user's access.
    ///
    /// - Returns: A string indicating the ARN of the newly created database
    ///   instance, or nil if the instance couldn't be created.
    func createDBInstance(name: String, instanceIdentifier: String,
parameterGroupName: String,
                          engine: String, engineVersion: String, instanceClass:
String,
                          username: String, password: String) async -> String? {
        do {
            let output = try await rdsClient.createDBInstance(
                input: CreateDBInstanceInput(
                    allocatedStorage: 100,
                    dbInstanceClass: instanceClass,
                    dbInstanceIdentifier: instanceIdentifier,
                    dbName: name,
                    dbParameterGroupName: parameterGroupName,
                    engine: engine,
                    engineVersion: engineVersion,
                    masterUserPassword: password,
                    masterUsername: username,
                    storageType: "gp2"
                )
            )

            guard let dbInstance = output.dbInstance else {
                print("*** Unable to get the database instance.")
                return nil
            }

            return dbInstance.dbInstanceArn
        } catch {
            print("*** An error occurred while creating the database instance:
\(error.localizedDescription)")
            return nil
        }
    }

    /// Wait until the specified database is available to use.
    ///
```

```
    /// - Parameter instanceIdentifier: The database instance identifier of the
    ///   database to wait for.
    func waitUntilDBInstanceReady(instanceIdentifier: String) async ->
RDSClientTypes.Endpoint? {
        do {
            putString("Waiting for the database instance to be ready to use. This
may take 10 minutes or more...")
            while true {
                let output = try await rdsClient.describeDBInstances(
                    input: DescribeDBInstancesInput(
                        dbInstanceIdentifier: instanceIdentifier
                    )
                )

                guard let instanceList = output.dbInstances else {
                    continue
                }

                for instance in instanceList {
                    let status = instance.dbInstanceStatus

                    guard let status else {
                        print("\nUnable to determine the status.")
                        continue
                    }

                    if status.contains("available") {
                        return instance.endpoint
                    } else {
                        putString(".")
                        do {
                            try await Task.sleep(for: .seconds(15))
                        } catch {
                            print("*** Error pausing the task!")
                        }
                    }
                }
            }
        } catch {
            print("*** Unable to wait until the database is ready:
\(error.localizedDescription)")
            return nil
        }
    }
```

```
   /// Create a snapshot of the specified name.
   ///
   /// - Parameters:
   ///   - instanceIdentifier: The identifier of the database instance to
   ///     snapshot.
   ///   - snapshotIdentifier: A unique identifier to give the newly-created
   ///     snapshot.
   func createDBSnapshot(instanceIdentifier: String, snapshotIdentifier: String)
async {
       do {
           let output = try await rdsClient.createDBSnapshot(
               input: CreateDBSnapshotInput(
                   dbInstanceIdentifier: instanceIdentifier,
                   dbSnapshotIdentifier: snapshotIdentifier
               )
           )

           guard let snapshot = output.dbSnapshot else {
               print("No snapshot returned.")
               return
           }

           print("The snapshot has been created with ID
\(snapshot.dbiResourceId ?? "<unknown>")")
       } catch {
           print("*** Unable to create the database snapshot named
\(snapshotIdentifier): \(error.localizedDescription)")
       }
   }

   /// Wait until the specified database snapshot is available to use.
   ///
   /// - Parameters:
   ///   - instanceIdentifier: The identifier of the database for which the
   ///     snapshot was taken.
   ///   - snapshotIdentifier: The identifier of the snapshot to wait for.
   func waitUntilDBSnapshotReady(instanceIdentifier: String, snapshotIdentifier:
String) async {
       var snapshotReady = false

       putString("Waiting for the snapshot to be ready...")

       do {
```

```
            while !snapshotReady {
                let output = try await rdsClient.describeDBSnapshots(
                    input: DescribeDBSnapshotsInput(
                        dbInstanceIdentifier: instanceIdentifier,
                        dbSnapshotIdentifier: snapshotIdentifier
                    )
                )

                guard let snapshotList = output.dbSnapshots else {
                    return
                }

                for snapshot in snapshotList {
                    guard let snapshotReadyStr = snapshot.status else {
                        return
                    }

                    if snapshotReadyStr.contains("available") {
                        snapshotReady = true
                        print()
                    } else {
                        putString(".")
                        do {
                            try await Task.sleep(for: .seconds(15))
                        } catch {
                            print("\n*** Error pausing the task!")
                        }
                    }
                }
            }
        } catch {
            print("\n*** Unable to wait for the database snapshot to be ready:
\(error.localizedDescription)")
        }
    }

    /// Delete the specified database instance.
    ///
    /// - Parameter instanceIdentifier: The identifier of the database
    ///   instance to delete.
    func deleteDBInstance(instanceIdentifier: String) async {
        do {
            _ = try await rdsClient.deleteDBInstance(
                input: DeleteDBInstanceInput(
```

```
                    dbInstanceIdentifier: instanceIdentifier,
                    deleteAutomatedBackups: true,
                    skipFinalSnapshot: true
                )
            )
        } catch {
            print("*** Error deleting the database instance
\(instanceIdentifier): \(error.localizedDescription)")
        }
    }

    /// Wait until the specified database instance has been deleted.
    ///
    /// - Parameter instanceIdentifier: The identifier of the database
    ///   instance to wait for.
    func waitUntilDBInstanceDeleted(instanceIdentifier: String) async {
        putString("Waiting for the database instance to be deleted. This may take
a few minutes...")
        do {
            var isDatabaseDeleted = false
            var foundInstance = false

            while !isDatabaseDeleted {
                let output = try await rdsClient.describeDBInstances(input:
DescribeDBInstancesInput())
                guard let instanceList = output.dbInstances else {
                    return
                }

                foundInstance = false

                for instance in instanceList {
                    guard let foundInstanceIdentifier =
instance.dbInstanceIdentifier else {
                        continue
                    }

                    if instanceIdentifier == foundInstanceIdentifier {
                        foundInstance = true
                        break
                    } else {
                        putString(".")
                        do {
                            try await Task.sleep(for: .seconds(15))
```

```
                        } catch {
                            print("\n*** Error pausing the task!")
                        }
                    }
                }
                if !foundInstance {
                    isDatabaseDeleted = true
                    print()
                }
            }
        } catch {
            print("\n*** Error waiting for the database instance to be deleted:
\(error.localizedDescription)")
        }
    }

    /// Delete the specified database parameter group.
    ///
    /// - Parameter groupName: The name of the parameter group to delete.
    func deleteDBParameterGroup(groupName: String) async {
        do {
            _ = try await rdsClient.deleteDBParameterGroup(
                input: DeleteDBParameterGroupInput(
                    dbParameterGroupName: groupName
                )
            )
        } catch {
            print("*** Error deleting the database parameter group \(groupName):
\(error.localizedDescription)")
        }
    }

    /// Generate and return a unique file name that begins with the specified
    /// string.
    ///
    /// - Parameters:
    ///   - prefix: Text to use at the beginning of the returned name.
    ///
    /// - Returns: A string containing a unique filename that begins with the
    ///   specified `prefix`.
    ///
    /// The returned name uses a random number between 1 million and 1 billion to
    /// provide reasonable certainty of uniqueness for the purposes of this
    /// example.
```

```
    func tempName(prefix: String) -> String {
        return "\(prefix)-\(Int.random(in: 1000000..<1000000000))"
    }

    /// Print a string to standard output without a trailing newline, and
    /// without buffering.
    ///
    /// - Parameter str: The string to output.
    func putString(_ str: String = "") {
        if str.length >= 1 {
            let data = str.data(using: .utf8)
            guard let data else {
                return
            }
            FileHandle.standardOutput.write(data)
        }
    }
}

/// The program's asynchronous entry point.
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
        }
    }
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS SDK für Swift.
  - CreateDBInstance
  - DBParameterGruppe erstellen
  - CreateDBSnapshot
  - LöschenDBInstance

- DBParameterGruppe löschen
- DBEngineVersionen beschreiben
- Beschreiben DBInstances
- Beschreiben Sie DBParameter Gruppen
- Beschreiben DBParameters
- Beschreiben DBSnapshots
- DescribeOrderableDBInstanceOptionen
- DBParameterGruppe ändern

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Aktionen für Amazon RDS mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie einzelne Amazon RDS-Aktionen mit ausführen AWS SDKs. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Diese Auszüge rufen die Amazon-RDS-API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Sie können Aktionen im Kontext unter Szenarien für die Verwendung von Amazon RDS AWS SDKs anzeigen.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der Amazon Relational Database Service API-Referenz.

Beispiele

- Verwendung CreateDBInstance mit einem AWS SDK oder CLI
- Verwendung CreateDBParameterGroup mit einem AWS SDK oder CLI
- Verwendung CreateDBSnapshot mit einem AWS SDK oder CLI
- Verwendung DeleteDBInstance mit einem AWS SDK oder CLI
- Verwendung DeleteDBParameterGroup mit einem AWS SDK oder CLI
- Verwendung DescribeAccountAttributes mit einem AWS SDK oder CLI
- Verwendung DescribeDBEngineVersions mit einem AWS SDK oder CLI

- Verwendung DescribeDBInstances mit einem AWS SDK oder CLI
- Verwendung DescribeDBParameterGroups mit einem AWS SDK oder CLI
- Verwendung DescribeDBParameters mit einem AWS SDK oder CLI
- Verwendung DescribeDBSnapshots mit einem AWS SDK oder CLI
- Verwendung DescribeOrderableDBInstanceOptions mit einem AWS SDK oder CLI
- Verwendung GenerateRDSAuthToken mit einem AWS SDK
- Verwendung ModifyDBInstance mit einem AWS SDK oder CLI
- Verwendung ModifyDBParameterGroup mit einem AWS SDK oder CLI
- Verwendung RebootDBInstance mit einem AWS SDK oder CLI

Verwendung **`CreateDBInstance`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateDBInstance` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
    /// to determine when the DB instance is ready to use.
    /// </summary>
    /// <param name="dbName">Name for the DB instance.</param>
```

```
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }
```

- Einzelheiten zur API finden Sie unter DBInstance In der AWS SDK für .NET API-Referenz erstellen.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::CreateDBInstanceRequest request;
    request.SetDBName(DB_NAME);
    request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetEngine(engineVersion.GetEngine());
    request.SetEngineVersion(engineVersion.GetEngineVersion());
    request.SetDBInstanceClass(dbInstanceClass);
    request.SetStorageType(DB_STORAGE_TYPE);
    request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
    request.SetMasterUsername(administratorName);
    request.SetMasterUserPassword(administratorPassword);

    Aws::RDS::Model::CreateDBInstanceOutcome outcome =
            client.CreateDBInstance(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB instance creation has started."
                  << std::endl;
    }
    else {
        std::cerr << "Error with RDS::CreateDBInstance. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }
```

- Einzelheiten zur API finden Sie unter DBInstance In der AWS SDK für C++ API-Referenz erstellen.

CLI

AWS CLI

Um eine DB-Instance zu erstellen

Im folgenden `create-db-instance` Beispiel werden die erforderlichen Optionen verwendet, um eine neue DB-Instance zu starten.

```
aws rds create-db-instance \
    --db-instance-identifier test-mysql-instance \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --master-username admin \
    --master-user-password secret99 \
    --allocated-storage 20
```

Ausgabe:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
```

```
            {
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
```

```
        },
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:mysql-5-7",
                "Status": "in-sync"
            }
        ],
        "PubliclyAccessible": true,
        "StorageType": "gp2",
        "DbInstancePort": 0,
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter Create DBInstance in AWS CLI Command Reference.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(ctx context.Context, instanceName
 string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(ctx,
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
```

```
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
 }
}
```

- Einzelheiten zur API finden Sie unter DBInstance In der AWS SDK für Go API-Referenz erstellen.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
```

```
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For more details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 *
 */

public class CreateDBInstance {
    public static long sleepTime = 20;

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbName> <secretName>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    dbName - The database name.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials."
                """;

        if (args.length != 3) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String dbName = args[1];
        String secretName = args[2];
        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        createDatabaseInstance(rdsClient, dbInstanceIdentifier, dbName,
user.getUsername(), user.getPassword());
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
                .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .build();
    }

    private static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                .secretId(secretName)
                .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    public static void createDatabaseInstance(RdsClient rdsClient,
            String dbInstanceIdentifier,
            String dbName,
            String userName,
```

```
            String userPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                .dbInstanceIdentifier(dbInstanceIdentifier)
                .allocatedStorage(100)
                .dbName(dbName)
                .engine("mysql")
                .dbInstanceClass("db.t3.medium") // Updated to a supported class
                .engineVersion("8.0.32")         // Updated to a supported
version
                .storageType("gp2")             // Changed to General Purpose SSD
(gp2)
                .masterUsername(userName)
                .masterUserPassword(userPassword)
                .build();

            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            // Loop until the cluster is ready.
            while (!instanceReady) {
```

```
                DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available"))
                        instanceReady = true;
                    else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available!");

        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DBInstance In der AWS SDK for Java 2.x API-Referenz erstellen.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
suspend fun createDatabaseInstance(
    dbInstanceIdentifierVal: String?,
    dbNamedbVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?,
) {
```

```
    val instanceRequest =
        CreateDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            allocatedStorage = 100
            dbName = dbNamedbVal
            engine = "mysql"
            dbInstanceClass = "db.t3.micro" // Use a supported instance class
            engineVersion = "8.0.39" // Use a supported engine version
            storageType = "gp2"
            masterUsername = masterUsernameVal
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the database instance is available.
suspend fun waitForInstanceReady(dbInstanceIdentifierVal: String?) {
    val sleepTime: Long = 20
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest =
        DescribeDbInstancesRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        while (!instanceReady) {
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        instanceReady = true
                    } else {
                        println("...$instanceReadyStr")
                        delay(sleepTime * 1000)
                    }
```

```
                }
            }
        }
        println("Database instance is available!")
    }
}
```

- API-Details finden Sie unter API-Referenz DBInstance im AWS SDK für Kotlin erstellen.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$dbClass = 'db.t2.micro';
$storage = 5;
$engine = 'MySQL';
$username = 'MyUser';
$password = 'MyPassword';

try {
    $result = $rdsClient->createDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
```

```
        'DBInstanceClass' => $dbClass,
        'AllocatedStorage' => $storage,
        'Engine' => $engine,
        'MasterUsername' => $username,
        'MasterUserPassword' => $password,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Einzelheiten zur API finden Sie unter DBInstance In der AWS SDK für PHP API-Referenz erstellen.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
```

```
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
```

```
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- API-Einzelheiten finden Sie unter Create DBInstance in AWS SDK for Python (Boto3) API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Create a new database instance.
    ///
    /// - Parameters:
    ///   - name: The name of the database to create.
    ///   - instanceIdentifier: The identifier to give the new database
    ///     instance.
```

```
    ///   - parameterGroupName: The name of the parameter group to associate
    ///     with the new database instance.
    ///   - engine: The database engine to use.
    ///   - engineVersion: The version of the database given by `engine` to
    ///     use.
    ///   - instanceClass: The memory and compute capacity of the database
    ///     instance, such as `db.m5.large``.
    ///   - username: The admin user's username to establish for the new
    ///     instance.
    ///   - password: The password to use for the specified user's access.
    ///
    /// - Returns: A string indicating the ARN of the newly created database
    ///   instance, or nil if the instance couldn't be created.
    func createDBInstance(name: String, instanceIdentifier: String,
parameterGroupName: String,
                          engine: String, engineVersion: String, instanceClass:
String,
                          username: String, password: String) async -> String? {
        do {
            let output = try await rdsClient.createDBInstance(
                input: CreateDBInstanceInput(
                    allocatedStorage: 100,
                    dbInstanceClass: instanceClass,
                    dbInstanceIdentifier: instanceIdentifier,
                    dbName: name,
                    dbParameterGroupName: parameterGroupName,
                    engine: engine,
                    engineVersion: engineVersion,
                    masterUserPassword: password,
                    masterUsername: username,
                    storageType: "gp2"
                )
            )

            guard let dbInstance = output.dbInstance else {
                print("*** Unable to get the database instance.")
                return nil
            }

            return dbInstance.dbInstanceArn
        } catch {
            print("*** An error occurred while creating the database instance:
\(error.localizedDescription)")
            return nil
```

```
        }
    }
```

- Einzelheiten zur API finden Sie in der Referenz Create DBInstance in AWS SDK for Swift API.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateDBParameterGroup`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateDBParameterGroup` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   /// <summary>
   /// Create a new DB parameter group. Use the action
DescribeDBParameterGroupsAsync
   /// to determine when the DB parameter group is ready to use.
   /// </summary>
   /// <param name="name">Name of the DB parameter group.</param>
   /// <param name="family">Family of the DB parameter group.</param>
   /// <param name="description">Description of the DB parameter group.</param>
   /// <returns>The new DB parameter group.</returns>
```

```
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe erstellen in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::CreateDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);
```

```
        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe erstellen in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um eine DB-Parametergruppe zu erstellen

Im folgenden `create-db-parameter-group` Beispiel wird eine DB-Parametergruppe erstellt.

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
    --description "My new parameter group"
```

Ausgabe:

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
        "Description": "My new parameter group",
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

Weitere Informationen finden Sie unter Erstellen einer DB-Parametergruppe im Amazon RDS-Benutzerhandbuch.

- API-Details finden Sie unter DBParameterGruppe erstellen in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 ctx context.Context, parameterGroupName string, parameterGroupFamily string,
 description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(ctx,
  &rds.CreateDBParameterGroupInput{
```

```
    DBParameterGroupName:   aws.String(parameterGroupName),
    DBParameterGroupFamily: aws.String(parameterGroupFamily),
    Description:            aws.String(description),
   })
  if err != nil {
   log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
   return nil, err
  } else {
   return output.DBParameterGroup, err
  }
 }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe erstellen in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
        try {
            CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .description("Created by using the AWS SDK for Java")
                    .build();

            CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
            System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());
```

```
        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe erstellen in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
```

```
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
group
        family.

        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- API-Einzelheiten finden Sie unter Create DBParameter Group in AWS SDK for Python (Boto3) API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Create a new database parameter group with the specified name.
    ///
    /// - Parameters:
    ///   - groupName: The name of the new parameter group.
    ///   - familyName: The name of the parameter group family.
    /// - Returns:
    func createDBParameterGroup(groupName: String, familyName: String) async ->
 RDSClientTypes.DBParameterGroup? {
        do {
            let output = try await rdsClient.createDBParameterGroup(
                input: CreateDBParameterGroupInput(
                    dbParameterGroupFamily: familyName,
                    dbParameterGroupName: groupName,
                    description: "Created using the AWS SDK for Swift"
                )
            )
            return output.dbParameterGroup
        } catch {
            print("*** Error creating the parameter group:
 \(error.localizedDescription)")
            return nil
        }
    }
```

- Einzelheiten zur API finden Sie unter Create DBParameter Group in AWS SDK for Swift API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`CreateDBSnapshot`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateDBSnapshot` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   /// <summary>
   /// Create a snapshot of a DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
   /// <returns>DB snapshot object.</returns>
   public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
string snapshotIdentifier)
   {
       var response = await _amazonRDS.CreateDBSnapshotAsync(
           new CreateDBSnapshotRequest()
           {
               DBSnapshotIdentifier = snapshotIdentifier,
               DBInstanceIdentifier = dbInstanceIdentifier
           });

       return response.DBSnapshot;
```

```
    }
```

- Einzelheiten zur API finden Sie unter DBSnapshot In der AWS SDK für .NET API-Referenz erstellen.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
```

```
            }
```

- Einzelheiten zur API finden Sie unter DBSnapshot In der AWS SDK für C++ API-Referenz erstellen.

CLI

AWS CLI

Um einen DB-Snapshot zu erstellen

Im folgenden `create-db-snapshot` Beispiel wird ein DB-Snapshot erstellt.

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

Ausgabe:

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "creating",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 0,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
```

```
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter Create DBSnapshot in AWS CLI Command Reference.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// CreateSnapshot creates a snapshot of a DB instance.
```

```
func (instances *DbInstances) CreateSnapshot(ctx context.Context, instanceName
 string, snapshotName string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(ctx,
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}
```

- Einzelheiten zur API finden Sie unter DBSnapshot In der AWS SDK für Go API-Referenz erstellen.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   // Create an Amazon RDS snapshot.
   public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
       try {
           CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                   .dbInstanceIdentifier(dbInstanceIdentifier)
                   .dbSnapshotIdentifier(dbSnapshotIdentifier)
                   .build();
```

```
            CreateDbSnapshotResponse response =
 rdsClient.createDBSnapshot(snapshotRequest);
            System.out.println("The Snapshot id is " +
 response.dbSnapshot().dbiResourceId());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DBSnapshot In der AWS SDK for Java 2.x API-Referenz erstellen.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$snapshotName = '<<{{backup_2018_12_25}}>>';

try {
    $result = $rdsClient->createDBSnapshot([
```

```
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBSnapshotIdentifier' => $snapshotName,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Einzelheiten zur API finden Sie unter DBSnapshot In der AWS SDK für PHP API-Referenz erstellen.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)
```

```
    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- API-Einzelheiten finden Sie unter Create DBSnapshot in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds' # v2: require 'aws-sdk'

# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
                                db_snapshot_identifier: id
                              })
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- Einzelheiten zur API finden Sie unter DBSnapshot In der AWS SDK für Ruby API-Referenz erstellen.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Create a snapshot of the specified name.
    ///
    /// - Parameters:
    ///   - instanceIdentifier: The identifier of the database instance to
    ///     snapshot.
    ///   - snapshotIdentifier: A unique identifier to give the newly-created
    ///     snapshot.
```

```
   func createDBSnapshot(instanceIdentifier: String, snapshotIdentifier: String)
async {
       do {
           let output = try await rdsClient.createDBSnapshot(
               input: CreateDBSnapshotInput(
                   dbInstanceIdentifier: instanceIdentifier,
                   dbSnapshotIdentifier: snapshotIdentifier
               )
           )

           guard let snapshot = output.dbSnapshot else {
               print("No snapshot returned.")
               return
           }

           print("The snapshot has been created with ID
\(snapshot.dbiResourceId ?? "<unknown>")")
       } catch {
           print("*** Unable to create the database snapshot named
\(snapshotIdentifier): \(error.localizedDescription)")
       }
   }
```

- Einzelheiten zur API finden Sie in der Referenz Create DBSnapshot in AWS SDK for Swift API.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteDBInstance`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteDBInstance` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
/// <summary>
/// Delete a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
{
    var response = await _amazonRDS.DeleteDBInstanceAsync(
        new DeleteDBInstanceRequest()
        {
            DBInstanceIdentifier = dbInstanceIdentifier,
            SkipFinalSnapshot = true,
            DeleteAutomatedBackups = true
        });

    return response.DBInstance;
}
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                    client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um eine DB-Instance zu löschen

Im folgenden `delete-db-instance` Beispiel wird die angegebene DB-Instance gelöscht, nachdem ein letzter DB-Snapshot mit dem Namen erstellt wurde`test-instance-final-snap`.

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

Ausgabe:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"
```

```
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(ctx context.Context, instanceName
 string) error {
 _, err := instances.RdsClient.DeleteDBInstance(ctx, &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      aws.Bool(true),
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void deleteDatabaseInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
```

```
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
 DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();

            DeleteDbInstanceResponse response =
 rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
 response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
suspend fun deleteDatabaseInstance(dbInstanceIdentifierVal: String?) {
    val deleteDbInstanceRequest =
        DeleteDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            deleteAutomatedBackups = true
            skipFinalSnapshot = true
        }
```

```
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}
```

- API-Details finden Sie unter Löschen DBInstance im AWS SDK für die Kotlin-API-Referenz.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-1'
]);

$dbIdentifier = '<<{{db-identifier}}>>';

try {
    $result = $rdsClient->deleteDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
```

```
}
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance in der AWS SDK für PHP API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
```

```
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- API-Einzelheiten finden Sie unter Delete DBInstance in AWS SDK for Python (Boto3) API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Delete the specified database instance.
    ///
    /// - Parameter instanceIdentifier: The identifier of the database
    ///   instance to delete.
    func deleteDBInstance(instanceIdentifier: String) async {
```

```
        do {
            _ = try await rdsClient.deleteDBInstance(
                input: DeleteDBInstanceInput(
                    dbInstanceIdentifier: instanceIdentifier,
                    deleteAutomatedBackups: true,
                    skipFinalSnapshot: true
                )
            )
        } catch {
            print("*** Error deleting the database instance
\(instanceIdentifier): \(error.localizedDescription)")
        }
    }
```

- Einzelheiten zur API finden Sie unter Löschen DBInstance im AWS SDK für die Swift-API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DeleteDBParameterGroup`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteDBParameterGroup` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Delete a DB parameter group. The group cannot be a default DB parameter
group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe löschen in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);
```

```
        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe löschen in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um eine DB-Parametergruppe zu löschen

Im folgenden command Beispiel wird eine DB-Parametergruppe gelöscht.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter Arbeiten mit DB-Parametergruppen im Amazon RDS-Benutzerhandbuch.

- API-Einzelheiten finden Sie unter DBParameterGruppe löschen in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(ctx context.Context,
 parameterGroupName string) error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(ctx,
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe löschen in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   // Delete the parameter group after database has been deleted.
   // An exception is thrown if you attempt to delete the para group while
database
   // exists.
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
               int listSize = instanceList.size();
               didFind = false;
               int index = 1;
               for (DBInstance instance : instanceList) {
                   instanceARN = instance.dbInstanceArn();
                   if (instanceARN.compareTo(dbARN) == 0) {
                       System.out.println(dbARN + " still exists");
                       didFind = true;
                   }
```

```
                    if ((index == listSize) && (!didFind)) {
                        // Went through the entire list and did not find the
database ARN.
                        isDataDel = true;
                    }
                    Thread.sleep(sleepTime * 1000);
                    index++;
                }
            }

            // Delete the para group.
            DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .build();

            rdsClient.deleteDBParameterGroup(parameterGroupRequest);
            System.out.println(dbGroupName + " was deleted.");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe löschen in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""
```

```
    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe löschen in der AWS API-Referenz für SDK for Python (Boto3).

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Delete the specified database parameter group.
    ///
    /// - Parameter groupName: The name of the parameter group to delete.
    func deleteDBParameterGroup(groupName: String) async {
        do {
            _ = try await rdsClient.deleteDBParameterGroup(
                input: DeleteDBParameterGroupInput(
                    dbParameterGroupName: groupName
                )
            )
        } catch {
            print("*** Error deleting the database parameter group \(groupName):
 \(error.localizedDescription)")
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe löschen im AWS SDK für die Swift-API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeAccountAttributes`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAccountAttributes` verwendet wird.

CLI

AWS CLI

Um Kontoattribute zu beschreiben

Im folgenden `describe-account-attributes` Beispiel werden die Attribute für das AWS Girokonto abgerufen.

```
aws rds describe-account-attributes
```

Ausgabe:

```
{
    "AccountQuotas": [
        {
            "Max": 40,
            "Used": 4,
            "AccountQuotaName": "DBInstances"
        },
        {
            "Max": 40,
            "Used": 0,
            "AccountQuotaName": "ReservedDBInstances"
        },
        {
            "Max": 100000,
            "Used": 40,
            "AccountQuotaName": "AllocatedStorage"
        },
        {
            "Max": 25,
            "Used": 0,
            "AccountQuotaName": "DBSecurityGroups"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
        },
        {
            "Max": 50,
            "Used": 1,
```

```
            "AccountQuotaName": "DBParameterGroups"
        },
        {
            "Max": 100,
            "Used": 3,
            "AccountQuotaName": "ManualSnapshots"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "EventSubscriptions"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBSubnetGroups"
        },
        {
            "Max": 20,
            "Used": 1,
            "AccountQuotaName": "OptionGroups"
        },
        {
            "Max": 20,
            "Used": 6,
            "AccountQuotaName": "SubnetsPerDBSubnetGroup"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "ReadReplicasPerMaster"
        },
        {
            "Max": 40,
            "Used": 1,
            "AccountQuotaName": "DBClusters"
        },
        {
            "Max": 50,
            "Used": 0,
            "AccountQuotaName": "DBClusterParameterGroups"
        },
        {
            "Max": 5,
```

```
            "Used": 0,
            "AccountQuotaName": "DBClusterRoles"
        }
    ]
}
```

- Einzelheiten zur API finden Sie unter DescribeAccountAttributes AWS CLIBefehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.AccountQuota;
import software.amazon.awssdk.services.rds.model.RdsException;
import
 software.amazon.awssdk.services.rds.model.DescribeAccountAttributesResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeAccountAttributes {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
```

```
        getAccountAttributes(rdsClient);
        rdsClient.close();
    }

    public static void getAccountAttributes(RdsClient rdsClient) {
        try {
            DescribeAccountAttributesResponse response =
 rdsClient.describeAccountAttributes();
            List<AccountQuota> quotasList = response.accountQuotas();
            for (AccountQuota quotas : quotasList) {
                System.out.println("Name is: " + quotas.accountQuotaName());
                System.out.println("Max value is " + quotas.max());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie DescribeAccountAttributesin der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
suspend fun getAccountAttributes() {
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeAccountAttributes(DescribeAccountAttributesRequest {})
        response.accountQuotas?.forEach { quotas ->
            val response = response.accountQuotas
```

```
            println("Name is: ${quotas.accountQuotaName}")
            println("Max value is ${quotas.max}")
        }
    }
}
```

- API-Details finden Sie DescribeAccountAttributesin der API-Referenz zum AWS SDK für Kotlin.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDBEngineVersions`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDBEngineVersions` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
```

```
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
 engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }
```

- Einzelheiten zur API finden Sie unter DBEngineVersionen beschreiben in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
```

```
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.

    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }


        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
                client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
 engineVersions.begin(),
                                        engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!marker.empty());
```

```
    return true;
}
```

- Einzelheiten zur API finden Sie unter DBEngineVersionen beschreiben in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um die DB-Engine-Versionen für die MySQL-DB-Engine zu beschreiben

Im folgenden `describe-db-engine-versions` Beispiel werden Details zu jeder DB-Engine-Version für die angegebene DB-Engine angezeigt.

```
aws rds describe-db-engine-versions \
    --engine mysql
```

Ausgabe:

```
{
    "DBEngineVersions": [
        {
            "Engine": "mysql",
            "EngineVersion": "5.5.46",
            "DBParameterGroupFamily": "mysql5.5",
            "DBEngineDescription": "MySQL Community Edition",
            "DBEngineVersionDescription": "MySQL 5.5.46",
            "ValidUpgradeTarget": [
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.53",
                    "Description": "MySQL 5.5.53",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.54",
```

```
                    "Description": "MySQL 5.5.54",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.57",
                    "Description": "MySQL 5.5.57",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                ...some output truncated...
            ]
        }
```

Weitere Informationen finden Sie unter Was ist Amazon Relational Database Service (Amazon RDS)? im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DBEngineVersionen beschreiben in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)
```

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(ctx context.Context, engine
 string, parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(ctx,
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}
```

- Einzelheiten zur API finden Sie unter DBEngineVersionen beschreiben in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    public static void describeDBEngines(RdsClient rdsClient) {
```

```
        try {
            DescribeDbEngineVersionsRequest engineVersionsRequest =
DescribeDbEngineVersionsRequest.builder()
                    .defaultOnly(true)
                    .engine("mysql")
                    .maxRecords(20)
                    .build();

            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
            List<DBEngineVersion> engines = response.dbEngineVersions();

            // Get all DBEngineVersion objects.
            for (DBEngineVersion engineOb : engines) {
                System.out.println("The name of the DB parameter group family for
the database engine is "
                        + engineOb.dbParameterGroupFamily());
                System.out.println("The name of the database engine " +
engineOb.engine());
                System.out.println("The version number of the database engine " +
engineOb.engineVersion());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DBEngineVersionen beschreiben in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
 list of
                                       engine versions to those that are
 compatible with
                                       this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
```

```
            )
            raise
        else:
            return versions
```

- Einzelheiten zur API finden Sie unter DBEngineDescribe Versions in AWS SDK for Python (Boto3) API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Get all the database engine versions available for the specified
    /// database engine.
    ///
    /// - Parameter engineName: The name of the database engine to query.
    ///
    /// - Returns: An array of `RDSClientTypes.DBEngineVersion` structures,
    ///   each describing one supported version of the specified database.
    func getDBEngineVersions(engineName: String) async ->
 [RDSClientTypes.DBEngineVersion] {
        do {
            let output = try await rdsClient.describeDBEngineVersions(
                input: DescribeDBEngineVersionsInput(
                    engine: engineName
                )
            )

            return output.dbEngineVersions ?? []
        } catch {
            return []
```

```
        }
    }
```

- Einzelheiten zur API finden Sie unter DBEngineVersionen im AWS SDK für die Swift-API-Referenz beschreiben.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDBInstances`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDBInstances` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Returns a list of DB instances.
    /// </summary>
    /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
    /// <returns>List of DB instances.</returns>
    public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
    {
```

```
        var results = new List<DBInstance>();
        var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
            new DescribeDBInstancesRequest
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });
        // Get the entire list using the paginator.
        await foreach (var instances in instancesPaginator.DBInstances)
        {
            results.Add(instances);
        }
        return results;
    }
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
```

```
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um eine DB-Instance zu beschreiben

Im folgenden `describe-db-instances` Beispiel werden Details zur angegebenen DB-Instance abgerufen.

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

Ausgabe:

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-
east-1.rds.amazonaws.com",
                "Port": 3306,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            ...some output truncated...
        }
    ]
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(ctx context.Context, instanceName
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(ctx,
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
 }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
```

```
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
suspend fun describeInstances() {
    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbInstances(DescribeDbInstancesRequest
 {})
        response.dbInstances?.forEach { instance ->
            println("Instance Identifier is ${instance.dbInstanceIdentifier}")
            println("The Engine is ${instance.engine}")
            println("Connection endpoint is ${instance.endpoint?.address}")
        }
    }
}
```

- Einzelheiten zur API finden Sie unter DBInstancesDescribe in AWS SDK for Kotlin API-Referenz.

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

try {
    $result = $rdsClient->describeDBInstances();
    foreach ($result['DBInstances'] as $instance) {
        print('<p>DB Identifier: ' . $instance['DBInstanceIdentifier']);
        print('<br />Endpoint: ' . $instance['Endpoint']["Address"]
            . ':' . $instance['Endpoint']["Port"]);
        print('<br />Current Status: ' . $instance["DBInstanceStatus"]);
        print('</p>');
    }
    print(" Raw Result ");
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für PHP API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
```

```
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst
```

- Einzelheiten zur API finden Sie unter DBInstancesDescribe in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds' # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
```

```
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
                          "name": i.id,
                          "status": i.db_instance_status
                        })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- Einzelheiten zur API finden Sie unter Describe DBInstances in der AWS SDK für Ruby API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Wait until the specified database is available to use.
    ///
    /// - Parameter instanceIdentifier: The database instance identifier of the
    ///   database to wait for.
    func waitUntilDBInstanceReady(instanceIdentifier: String) async ->
 RDSClientTypes.Endpoint? {
        do {
            putString("Waiting for the database instance to be ready to use. This
 may take 10 minutes or more...")
            while true {
                let output = try await rdsClient.describeDBInstances(
                    input: DescribeDBInstancesInput(
                        dbInstanceIdentifier: instanceIdentifier
```

```
                    )
                )

                guard let instanceList = output.dbInstances else {
                    continue
                }

                for instance in instanceList {
                    let status = instance.dbInstanceStatus

                    guard let status else {
                        print("\nUnable to determine the status.")
                        continue
                    }

                    if status.contains("available") {
                        return instance.endpoint
                    } else {
                        putString(".")
                        do {
                            try await Task.sleep(for: .seconds(15))
                        } catch {
                            print("*** Error pausing the task!")
                        }
                    }
                }
            }
        } catch {
            print("*** Unable to wait until the database is ready:
\(error.localizedDescription)")
            return nil
        }
    }
```

- Einzelheiten zur API finden Sie unter DBInstancesDescribe in AWS SDK for Swift API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDBParameterGroups`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDBParameterGroups` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppen beschreiben in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }

        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppen beschreiben in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um Ihre DB-Parametergruppe zu beschreiben

Im folgenden `describe-db-parameter-groups` Beispiel werden Details zu Ihren DB-Parametergruppen abgerufen.

```
aws rds describe-db-parameter-groups
```

Ausgabe:

```
{
    "DBParameterGroups": [
        {
            "DBParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default parameter group for aurora-mysql5.7",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
        },
        {
            "DBParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default parameter group for aurora-postgresql9.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
        },
        {
            "DBParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default parameter group for aurora5.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
        },
        {
            "DBParameterGroupName": "default.mariadb10.1",
            "DBParameterGroupFamily": "mariadb10.1",
            "Description": "Default parameter group for mariadb10.1",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
        },
```

```
        ...some output truncated...
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit DB-Parametergruppen im Amazon RDS-Benutzerhandbuch.

- API-Details finden Sie unter Describe DBParameter Groups in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(ctx context.Context,
 parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
```

```
  ctx, &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
  return nil, err
 } else {
  return &output.DBParameterGroups[0], err
 }
}
```

- Einzelheiten zur API finden Sie unter DBParameterGruppen beschreiben in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
       try {
           DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .maxRecords(20)
                   .build();
```

```
            DescribeDbParameterGroupsResponse response =
 rdsClient.describeDBParameterGroups(groupsRequest);
            List<DBParameterGroup> groups = response.dbParameterGroups();
            for (DBParameterGroup group : groups) {
                System.out.println("The group name is " +
 group.dbParameterGroupName());
                System.out.println("The group description is " +
 group.description());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppen beschreiben in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
```

```
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.

        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group
```

- Einzelheiten zur API finden Sie unter DBParameterDescribe Groups in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds' # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Einzelheiten zur API finden Sie unter DBParameterGruppen beschreiben in der AWS SDK für Ruby API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Get descriptions of the database parameter groups matching the given
    /// name.
    ///
    /// - Parameter groupName: The name of the parameter group to describe.
    ///
    /// - Returns: An array of [RDSClientTypes.DBParameterGroup] objects
    ///   describing the parameter group.
    func describeDBParameterGroups(groupName: String) async ->
 [RDSClientTypes.DBParameterGroup]? {
        do {
            let output = try await rdsClient.describeDBParameterGroups(
                input: DescribeDBParameterGroupsInput(
                    dbParameterGroupName: groupName
                )
            )
            return output.dbParameterGroups
        } catch {
            print("*** Error getting the database parameter group's details:
 \(error.localizedDescription)")
            return nil
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterDescribe Groups in AWS SDK for Swift API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDBParameters`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDBParameters` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
   /// <summary>
   /// Get a list of DB parameters from a specific parameter group.
   /// </summary>
   /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
   /// <param name="source">Optional source for selecting parameters.</param>
   /// <returns>List of parameter values.</returns>
   public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
   {
       var results = new List<Parameter>();
       var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
           new DescribeDBParametersRequest()
           {
               DBParameterGroupName = dbParameterGroupName,
               Source = source
           });
```

```
        // Get the entire list using the paginator.
        await foreach (var parameters in paginateParameters.Parameters)
        {
            results.Add(parameters);
        }
        return results;
    }
```

- Einzelheiten zur API finden Sie unter Describe DBParameters in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
```

```
                                const Aws::String &namePrefix,
                                const Aws::String &source,
                                Aws::Vector<Aws::RDS::Model::Parameter>
&parametersResult,
                                const Aws::RDS::RDSClient &client) {
   Aws::String marker;
   do {
       Aws::RDS::Model::DescribeDBParametersRequest request;
       request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
       if (!marker.empty()) {
           request.SetMarker(marker);
       }
       if (!source.empty()) {
           request.SetSource(source);
       }

       Aws::RDS::Model::DescribeDBParametersOutcome outcome =
               client.DescribeDBParameters(request);

       if (outcome.IsSuccess()) {
           const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                   outcome.GetResult().GetParameters();
           for (const Aws::RDS::Model::Parameter &parameter: parameters) {
               if (!namePrefix.empty()) {
                   if (parameter.GetParameterName().find(namePrefix) == 0) {
                       parametersResult.push_back(parameter);
                   }
               }
               else {
                   parametersResult.push_back(parameter);
               }
           }

           marker = outcome.GetResult().GetMarker();
       }
       else {
           std::cerr << "Error with RDS::DescribeDBParameters. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           return false;
       }
   } while (!marker.empty());

   return true;
```

```
}
```

- Einzelheiten zur API finden Sie unter Describe DBParameters in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um die Parameter in einer DB-Parametergruppe zu beschreiben

Im folgenden `describe-db-parameters` Beispiel werden die Details der angegebenen DB-Parametergruppe abgerufen.

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

Ausgabe:

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have
 only an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "auto_generate_certs",
            "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
```

```
            "ApplyMethod": "pending-reboot"
        },
        ...some output truncated...
    ]
}
```

Weitere Informationen finden Sie unter Arbeiten mit DB-Parametergruppen im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter Describe DBParameters in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(ctx context.Context,
 parameterGroupName string, source string) (
```

```
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(ctx)
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}
```

- Einzelheiten zur API finden Sie unter Describe DBParameters in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    // Retrieve parameters in the group.
    public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
        try {
```

```
            DescribeDbParametersRequest dbParameterGroupsRequest;
            if (flag == 0) {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .build();
            } else {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .source("user")
                        .build();
            }

            DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
            List<Parameter> dbParameters = response.parameters();
            String paraName;
            for (Parameter para : dbParameters) {
                // Only print out information about either auto_increment_offset
or
                // auto_increment_increment.
                paraName = para.parameterName();
                if ((paraName.compareTo("auto_increment_offset") == 0)
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Einzelheiten zur API finden Sie unter Describe DBParameters in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
 filtered
                            to contain only parameters that start with this
 prefix.
```

```
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters
```

- Einzelheiten zur API finden Sie unter DBParametersDescribe in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds' # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Einzelheiten zur API finden Sie unter Describe DBParameters in der AWS SDK für Ruby API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Returns the detailed parameter list for the specified database
    /// parameter group.
    ///
    /// - Parameters:
    ///   - groupName: The name of the parameter group to return parameters for.
    ///   - source: The types of parameters to return (`user`, `system`, or
    ///     `engine-default`).
    ///
    /// - Returns: An array of `RdSClientTypes.Parameter` objects, each
    ///   describing one of the group's parameters.
    func describeDBParameters(groupName: String, source: String? = nil) async ->
 [RDSClientTypes.Parameter] {
        var parameterList: [RDSClientTypes.Parameter] = []

        do {
            let pages = rdsClient.describeDBParametersPaginated(
                input: DescribeDBParametersInput(
                    dbParameterGroupName: groupName,
                    source: source
                )
            )

            for try await page in pages {
                guard let parameters = page.parameters else {
                    return []
                }

                parameterList += parameters
            }
        } catch {
```

```
            print("*** Error getting database parameters:
\(error.localizedDescription)")
            return []
        }

        return parameterList
    }
```

- Einzelheiten zur API finden Sie unter DBParametersDescribe in AWS SDK for Swift API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeDBSnapshots`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeDBSnapshots` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
/// <summary>
/// Return a list of DB snapshots for a particular DB instance.
/// </summary>
```

```
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <returns>List of DB snapshots.</returns>
    public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
dbInstanceIdentifier)
    {
        var results = new List<DBSnapshot>();
        var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
            new DescribeDBSnapshotsRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Einzelheiten zur API finden Sie unter Describe DBSnapshots in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);
```

```
            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
```

- Einzelheiten zur API finden Sie unter Describe DBSnapshots in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Beispiel 1: Um einen DB-Snapshot für eine DB-Instance zu beschreiben

Im folgenden `describe-db-snapshots` Beispiel werden die Details eines DB-Snapshots für eine DB-Instance abgerufen.

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

Ausgabe:

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "mydbsnapshot",
            "DBInstanceIdentifier": "mysqldb",
            "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
```

```
            "Engine": "mysql",
            "AllocatedStorage": 20,
            "Status": "available",
            "Port": 3306,
            "AvailabilityZone": "us-east-1f",
            "VpcId": "vpc-6594f31c",
            "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
            "MasterUsername": "mysqladmin",
            "EngineVersion": "5.6.37",
            "LicenseModel": "general-public-license",
            "SnapshotType": "manual",
            "OptionGroupName": "default:mysql-5-6",
            "PercentProgress": 100,
            "StorageType": "gp2",
            "Encrypted": false,
            "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
            "IAMDatabaseAuthenticationEnabled": false,
            "ProcessorFeatures": [],
            "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots im Amazon RDS-Benutzerhandbuch.

Beispiel 2: Um die Anzahl der manuell erstellten Snapshots zu ermitteln

Im folgenden `describe-db-snapshots` Beispiel wird der `length` Operator in der `--query` Option verwendet, um die Anzahl der manuellen Schnappschüsse zurückzugeben, die in einer bestimmten AWS Region aufgenommen wurden.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

Ausgabe:

```
35
```

Weitere Informationen finden Sie unter Erstellen eines DB-Snapshots im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter Beschreibung DBSnapshots in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(ctx context.Context, snapshotName
 string) (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(ctx,
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
```

```
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}
```

- Einzelheiten zur API finden Sie unter Describe DBSnapshots in der AWS SDK für Go API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_snapshot(self, snapshot_id):
        """
```

```
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Einzelheiten zur API finden Sie unter DBSnapshotsDescribe in AWS SDK for Python (Boto3) API-Referenz.

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
require 'aws-sdk-rds' # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
# snapshots.
#
```

```
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if
 error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
                                "id": s.snapshot_id,
                                "status": s.status
                              })
  end
  instance_snapshots
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- Einzelheiten zur API finden Sie unter Describe DBSnapshots in der AWS SDK für Ruby API-Referenz.

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Wait until the specified database snapshot is available to use.
    ///
    /// - Parameters:
    ///   - instanceIdentifier: The identifier of the database for which the
    ///     snapshot was taken.
    ///   - snapshotIdentifier: The identifier of the snapshot to wait for.
    func waitUntilDBSnapshotReady(instanceIdentifier: String, snapshotIdentifier:
 String) async {
```

```
        var snapshotReady = false

        putString("Waiting for the snapshot to be ready...")

        do {
            while !snapshotReady {
                let output = try await rdsClient.describeDBSnapshots(
                    input: DescribeDBSnapshotsInput(
                        dbInstanceIdentifier: instanceIdentifier,
                        dbSnapshotIdentifier: snapshotIdentifier
                    )
                )

                guard let snapshotList = output.dbSnapshots else {
                    return
                }

                for snapshot in snapshotList {
                    guard let snapshotReadyStr = snapshot.status else {
                        return
                    }

                    if snapshotReadyStr.contains("available") {
                        snapshotReady = true
                        print()
                    } else {
                        putString(".")
                        do {
                            try await Task.sleep(for: .seconds(15))
                        } catch {
                            print("\n*** Error pausing the task!")
                        }
                    }
                }
            }
        } catch {
            print("\n*** Unable to wait for the database snapshot to be ready:
\(error.localizedDescription)")
        }
    }
```

- Einzelheiten zur API finden Sie unter DBSnapshotsDescribe in AWS SDK for Swift API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`DescribeOrderableDBInstanceOptions`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeOrderableDBInstanceOptions` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
```

```
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }
```

- Einzelheiten zur API finden Sie unter DescribeOrderableDBInstanceOptionen in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
```

```
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
```

```
                    << outcome.GetError().GetMessage()
                    << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}
```

- Einzelheiten zur API finden Sie unter DescribeOrderableDBInstanceOptionen in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um bestellbare DB-Instance-Optionen zu beschreiben

Im folgenden `describe-orderable-db-instance-options` Beispiel werden Details zu den bestellbaren Optionen für DB-Instances abgerufen, auf denen die MySQL-DB-Engine ausgeführt wird.

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

Ausgabe:

```
{
    "OrderableDBInstanceOptions": [
```

```
        {
            "MinStorageSize": 5,
            "ReadReplicaCapable": true,
            "MaxStorageSize": 6144,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "SupportsIops": false,
            "AvailableProcessorFeatures": [],
            "MultiAZCapable": true,
            "DBInstanceClass": "db.m1.large",
            "Vpc": true,
            "StorageType": "gp2",
            "LicenseModel": "general-public-license",
            "EngineVersion": "5.5.46",
            "SupportsStorageEncryption": false,
            "SupportsEnhancedMonitoring": true,
            "Engine": "mysql",
            "SupportsIAMDatabaseAuthentication": false,
            "SupportsPerformanceInsights": false
        }
    ]
    ...some output truncated...
}
```

- Einzelheiten zur API finden Sie unter DescribeOrderableDBInstanceOptionen in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(ctx context.Context, engine
 string, engineVersion string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
```

```
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(ctx)
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
 }
 return instanceOptions, err
}
```

- Einzelheiten zur API finden Sie unter DescribeOrderableDBInstanceOptionen in der AWS SDK für Go API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
```

```
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts
```

- Einzelheiten zur API finden Sie unter DescribeOrderableDBInstanceOptionen in der AWS API-Referenz zum SDK for Python (Boto3).

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Print a list of available database instances with "micro" in the class
    /// name, then return one of them to be used by other code.
    ///
    /// - Parameters:
    ///   - engine: The database engine for which to list database instance
    ///     classes.
    ///   - engineVersion: The database version for which to list instances.
    ///
    /// - Returns: An `RDSClientTypes.OrderableDBInstanceOption` describing
    ///   the selected instance type.
    func chooseMicroInstance(engine: String = "mysql", engineVersion: String? =
 nil) async -> RDSClientTypes.OrderableDBInstanceOption? {
        do {
            let pages = rdsClient.describeOrderableDBInstanceOptionsPaginated(
                input: DescribeOrderableDBInstanceOptionsInput(
                    engine: engine,
                    engineVersion: engineVersion
                )
            )

            var optionsList: [RDSClientTypes.OrderableDBInstanceOption] = []

            for try await page in pages {
                guard let orderableDBInstanceOptions =
 page.orderableDBInstanceOptions else {
                    continue
                }

                for dbInstanceOption in orderableDBInstanceOptions {
                    guard let className = dbInstanceOption.dbInstanceClass else {
```

```
                    continue
                }
                if className.contains("micro") {
                    optionsList.append(dbInstanceOption)
                }
            }
        }

        print("Found \(optionsList.count) database instances of 'micro' class
types:")
        for dbInstanceOption in optionsList {
            print("    \(dbInstanceOption.engine ??
"<unknown>") \(dbInstanceOption.engineVersion ?? "<unknown>")
(\(dbInstanceOption.dbInstanceClass ?? "<unknown class>"))")
        }

        return optionsList[0]
    } catch {
        print("*** Error getting a list of orderable instance options:
\(error.localizedDescription)")
        return nil
    }
}
```

- Einzelheiten zur API finden Sie unter DescribeOrderableDBInstanceOptionen im AWS SDK für die Swift-API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`GenerateRDSAuthToken`** mit einem AWS SDK

Das folgende Codebeispiel zeigt, wie es verwendet wird`GenerateRDSAuthToken`.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

Verwenden Sie die RdsUtilitiesKlasse, um ein Authentifizierungstoken zu generieren.

```
public class GenerateRDSAuthToken {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <masterUsername>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUsername - The master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUsername = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        String token = getAuthToken(rdsClient, dbInstanceIdentifier,
 masterUsername);
        System.out.println("The token response is " + token);
    }

    public static String getAuthToken(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUsername) {
```

```
        RdsUtilities utilities = rdsClient.utilities();
        try {
            GenerateAuthenticationTokenRequest tokenRequest =
 GenerateAuthenticationTokenRequest.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .username(masterUsername)
                    .port(3306)
                    .hostname(dbInstanceIdentifier)
                    .build();

            return utilities.generateAuthenticationToken(tokenRequest);

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Einzelheiten zur API finden Sie unter Generate RDSAuth Token in der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ModifyDBInstance`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `ModifyDBInstance` verwendet wird.

CLI

AWS CLI

Beispiel 1: Um eine DB-Instance zu ändern

Das folgende `modify-db-instance` Beispiel verknüpft eine Optionsgruppe und eine Parametergruppe mit einer kompatiblen Microsoft SQL Server-DB-Instance. Der `--apply-`

`immediately` Parameter bewirkt, dass die Options- und Parametergruppen sofort verknüpft werden, anstatt bis zum nächsten Wartungsfenster zu warten.

```
aws rds modify-db-instance \
    --db-instance-identifier database-2 \
    --option-group-name test-se-2017 \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --apply-immediately
```

Ausgabe:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-2",
        "DBInstanceClass": "db.r4.large",
        "Engine": "sqlserver-se",
        "DBInstanceStatus": "available",

        ...output omitted...

        "DBParameterGroups": [
            {
                "DBParameterGroupName": "test-sqlserver-se-2017",
                "ParameterApplyStatus": "applying"
            }
        ],
        "AvailabilityZone": "us-west-2d",

        ...output omitted...

        "MultiAZ": true,
        "EngineVersion": "14.00.3281.6.v1",
        "AutoMinorVersionUpgrade": false,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "license-included",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "test-se-2017",
                "Status": "pending-apply"
            }
        ],
        "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
        "SecondaryAvailabilityZone": "us-west-2c",
```

```
        "PubliclyAccessible": true,
        "StorageType": "gp2",

        ...output omitted...

        "DeletionProtection": false,
        "AssociatedRoles": [],
        "MaxAllocatedStorage": 1000
    }
}
```

Weitere Informationen finden Sie unter Ändern einer Amazon RDS-DB-Instance im Amazon RDS-Benutzerhandbuch.

Beispiel 2: So ordnen Sie eine VPC-Sicherheitsgruppe einer DB-Instance zu

Das folgende `modify-db-instance` Beispiel ordnet eine bestimmte VPC-Sicherheitsgruppe zu und entfernt DB-Sicherheitsgruppen aus einer DB-Instance:

```
aws rds modify-db-instance \
    --db-instance-identifier dbName \
    --vpc-security-group-ids sg-ID
```

Ausgabe:

```
{
"DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
        "Address": "dbName.abcdefghijk.us-west-2.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "ABCDEFGHIJK1234"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
    "PreferredBackupWindow": "11:57-12:27",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
```

```
        {
            "VpcSecurityGroupId": "sg-ID",
            "Status": "active"
        }
    ],
    ... output omitted ...
    "MultiAZ": false,
    "EngineVersion": "8.0.35",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",

    ... output ommited ...
    }
}
```

Weitere Informationen finden Sie unter Steuern des Zugriffs mit Sicherheitsgruppen im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter Ändern DBInstance in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
```

```
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ModifyDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbSnapshotIdentifier>\s
                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUserPassword - The updated password that corresponds to
 the master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUserPassword = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        updateIntance(rdsClient, dbInstanceIdentifier, masterUserPassword);
        rdsClient.close();
    }

    public static void updateIntance(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUserPassword) {
        try {
            // For a demo - modify the DB instance by modifying the master
 password.
            ModifyDbInstanceRequest modifyDbInstanceRequest =
 ModifyDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .publiclyAccessible(true)
                    .masterUserPassword(masterUserPassword)
                    .build();
```

```
            ModifyDbInstanceResponse instanceResponse =
 rdsClient.modifyDBInstance(modifyDbInstanceRequest);
            System.out.print("The ARN of the modified database is: " +
 instanceResponse.dbInstance().dbInstanceArn());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Ändern DBInstance in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
suspend fun updateIntance(
    dbInstanceIdentifierVal: String?,
    masterUserPasswordVal: String?,
) {
    val request =
        ModifyDbInstanceRequest {
            dbInstanceIdentifier = dbInstanceIdentifierVal
            publiclyAccessible = true
            masterUserPassword = masterUserPasswordVal
        }

    RdsClient.fromEnvironment { region = "us-west-2" }.use { rdsClient ->
        val instanceResponse = rdsClient.modifyDbInstance(request)
        println("The ARN of the modified database is
 ${instanceResponse.dbInstance?.dbInstanceArn}")
```

```
    }
}
```

- API-Details finden Sie unter Modify DBInstance in AWS SDK for Kotlin API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`ModifyDBParameterGroup`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `ModifyDBParameterGroup` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- Erlernen der Grundlagen

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    /// <summary>
    /// Update a DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
```

```
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe ändern in der AWS SDK für .NET API-Referenz.

C++

SDK für C++

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::ModifyDBParameterGroupRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetParameters(updateParameters);

    Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
            client.ModifyDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully modified."
```

```
                    << std::endl;
        }
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe ändern in der AWS SDK für C++ API-Referenz.

CLI

AWS CLI

Um eine DB-Parametergruppe zu ändern

Im folgenden `modify-db-parameter-group` Beispiel wird der Wert des `clr enabled` Parameters in einer DB-Parametergruppe geändert. Der `--apply-immediately` Parameter bewirkt, dass die DB-Parametergruppe sofort geändert wird, anstatt bis zum nächsten Wartungsfenster zu warten.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Ausgabe:

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter DBParameterGruppe ändern in der AWS CLI Befehlsreferenz.

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import (
 "context"
 "errors"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/service/rds"
 "github.com/aws/aws-sdk-go-v2/service/rds/types"
)

type DbInstances struct {
 RdsClient *rds.Client
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(ctx context.Context,
 parameterGroupName string, params []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(ctx,
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe ändern in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
    // Modify auto_increment_offset and auto_increment_increment parameters.
    public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
        try {
            Parameter parameter1 = Parameter.builder()
                    .parameterName("auto_increment_offset")
                    .applyMethod("immediate")
                    .parameterValue("5")
                    .build();

            List<Parameter> paraList = new ArrayList<>();
            paraList.add(parameter1);
            ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .parameters(paraList)
                    .build();

            ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
            System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
```

```
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe ändern in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
```

```
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
Parameters=update_parameters
            )
        except ClientError as err:
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe modifizieren in der API-Referenz zum AWS SDK for Python (Boto3).

Swift

SDK für Swift

> **Note**
>
> Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import AWSRDS

    /// Demonstrates modifying two of the specified database parameter group's
    /// parameters.
    ///
    /// - Parameter groupName: The name of the parameter group to change
    ///   parameters for.
    func modifyDBParameters(groupName: String) async {
```

```
        let parameter1 = RDSClientTypes.Parameter(
            applyMethod: RDSClientTypes.ApplyMethod.immediate,
            parameterName: "auto_increment_offset",
            parameterValue: "5"
        )
        let parameter2 = RDSClientTypes.Parameter(
            applyMethod: RDSClientTypes.ApplyMethod.immediate,
            parameterName: "auto_increment_increment",
            parameterValue: "5"
        )

        let parameterList = [parameter1, parameter2]

        do {
            _ = try await rdsClient.modifyDBParameterGroup(
                input: ModifyDBParameterGroupInput(
                    dbParameterGroupName: groupName,
                    parameters: parameterList
                )
            )

            print("Successfully modified the parameter group \(groupName).")
        } catch {
            print("*** Error modifying the parameter group \(groupName):
\(error.localizedDescription)")
        }
    }
```

- Einzelheiten zur API finden Sie unter DBParameterGruppe ändern im AWS SDK für die Swift-API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **`RebootDBInstance`** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `RebootDBInstance` verwendet wird.

CLI

AWS CLI

Um eine DB-Instance neu zu starten

Das folgende `reboot-db-instance` Beispiel startet einen Neustart der angegebenen DB-Instance.

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

Ausgabe:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "rebooting",
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.############.us-
west-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z1PVIF0EXAMPLE"
        },

    ... output omitted...

    }
}
```

Weitere Informationen finden Sie unter Rebooting a DB Instance im Amazon RDS-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter Reboot DBInstance in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel- einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class RebootDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
```

```
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        rebootInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void rebootInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            RebootDbInstanceRequest rebootDbInstanceRequest =
 RebootDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            RebootDbInstanceResponse instanceResponse =
 rdsClient.rebootDBInstance(rebootDbInstanceRequest);
            System.out.print("The database " +
 instanceResponse.dbInstance().dbInstanceArn() + " was rebooted");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Einzelheiten zur API finden Sie unter Reboot DBInstance in der AWS SDK for Java 2.x API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für die Verwendung von Amazon RDS AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie gängige Szenarien in Amazon RDS mit implementieren AWS SDKs. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben

erledigen können, indem Sie mehrere Funktionen in Amazon RDS aufrufen oder mit anderen kombinieren AWS-Services. Jedes Szenario enthält einen Link zum vollständigen Quell-Code, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Szenarien zielen auf eine mittlere Erfahrungsebene ab, um Ihnen zu helfen, Service-Aktionen im Kontext zu verstehen.

Beispiele

- Erstellen eines Trackers für Aurora-Serverless-Arbeitsaufgaben

Erstellen eines Trackers für Aurora-Serverless-Arbeitsaufgaben

Die folgenden Code-Beispiele zeigen, wie eine Webanwendung erstellt wird, die Arbeitselemente in einer Amazon Aurora Serverless-Datenbank verfolgt und mithilfe von Amazon Simple Email Service (Amazon SES) Berichte sendet.

.NET

SDK für .NET

Zeigt, wie Sie mithilfe von Amazon Simple Email Service (Amazon SES) eine Webanwendung erstellen, die Arbeitsaufgaben in einer Amazon Aurora Aurora-Datenbank nachverfolgt und Berichte per E-Mail versendet. AWS SDK für .NET In diesem Beispiel wird ein mit React.js erstelltes Frontend verwendet, um mit RESTful einem.NET-Backend zu interagieren.

- Integrieren Sie eine React-Webanwendung in AWS Dienste.
- Auflisten, Hinzufügen, Aktualisieren und Löschen von Elementen in einer Aurora-Tabelle.
- Senden Sie einen E-Mail-Bericht über gefilterte Arbeitselemente mit Amazon SES.
- Stellen Sie Beispielressourcen mit dem mitgelieferten AWS CloudFormation Skript bereit und verwalten Sie sie.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- Aurora
- Amazon RDS
- Amazon RDS Data Service

- Amazon SES

C++

SDK für C++

Zeigt, wie eine Webanwendung erstellt wird, die in einer Amazon-Aurora-Serverless-Datenbank gespeicherte Arbeitselemente verfolgt und darüber berichtet.

Den vollständigen Quellcode und Anweisungen zur Einrichtung einer C++-REST-API, die Amazon Aurora Aurora-Serverless-Daten abfragt und von einer React-Anwendung verwendet werden kann, finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- Aurora
- Amazon RDS
- Amazon RDS Data Service
- Amazon SES

Java

SDK für Java 2.x

Zeigt, wie eine Webanwendung erstellt wird, die Arbeitselemente, die in einer Amazon RDS-Datenbank gespeichert sind, verfolgt und darüber berichtet.

Den vollständigen Quellcode und Anweisungen zur Einrichtung einer Spring REST-API, die Amazon Aurora Aurora-Serverless-Daten abfragt und von einer React-Anwendung verwendet werden kann, finden Sie im vollständigen Beispiel unter GitHub.

Den vollständigen Quellcode und Anweisungen zum Einrichten und Ausführen eines Beispiels, das die JDBC-API verwendet, finden Sie im vollständigen Beispiel unter. GitHub

In diesem Beispiel verwendete Dienste

- Aurora
- Amazon RDS
- Amazon RDS Data Service
- Amazon SES

JavaScript

SDK für JavaScript (v3)

Zeigt, wie Sie mit AWS SDK für JavaScript (v3) eine Webanwendung erstellen, die Arbeitsaufgaben in einer Amazon Aurora Aurora-Datenbank verfolgt und Berichte mithilfe von Amazon Simple Email Service (Amazon SES) per E-Mail versendet. In diesem Beispiel wird ein mit React.js erstelltes Frontend verwendet, um mit einem Express-Node.js-Backend zu interagieren.

- Integrieren Sie eine React.js Webanwendung mit AWS-Services.
- Auflisten, hinzufügen und aktualisieren von Elementen in einer Aurora-Tabelle.
- Senden Sie einen E-Mail-Bericht über gefilterte Arbeitselemente mit Amazon SES.
- Stellen Sie Beispielressourcen mit dem mitgelieferten AWS CloudFormation Skript bereit und verwalten Sie sie.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- Aurora
- Amazon RDS
- Amazon RDS Data Service
- Amazon SES

Kotlin

SDK für Kotlin

Zeigt, wie eine Webanwendung erstellt wird, die Arbeitselemente, die in einer Amazon RDS-Datenbank gespeichert sind, verfolgt und darüber berichtet.

Den vollständigen Quellcode und Anweisungen zur Einrichtung einer Spring REST-API, die Amazon Aurora Aurora-Serverless-Daten abfragt und von einer React-Anwendung verwendet werden kann, finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- Aurora

- Amazon RDS
- Amazon RDS Data Service
- Amazon SES

PHP

SDK für PHP

Zeigt, wie Sie mithilfe von Amazon Simple Email Service (Amazon SES) eine Webanwendung erstellen, die Arbeitselemente in einer Amazon RDS-Datenbank verfolgt und Berichte per E-Mail versendet. AWS SDK für PHP In diesem Beispiel wird ein mit React.js erstelltes Frontend verwendet, um mit einem RESTful PHP-Backend zu interagieren.

- Integrieren Sie eine React.js -Webanwendung in AWS Dienste.
- In einer Amazon-RDS-Tabelle können Sie Elemente auflisten, aktualisieren und löschen.
- Senden Sie einen E-Mail-Bericht über gefilterte Arbeitselemente mit Amazon SES.
- Stellen Sie Beispielressourcen mit dem mitgelieferten AWS CloudFormation Skript bereit und verwalten Sie sie.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- Aurora
- Amazon RDS
- Amazon RDS Data Service
- Amazon SES

Python

SDK für Python (Boto3)

Zeigt, wie Sie mithilfe von Amazon Simple Email Service (Amazon SES) einen REST-Service erstellen, der Arbeitselemente in einer Amazon Aurora Aurora-Serverless-Datenbank nachverfolgt und Berichte per E-Mail versendet. AWS SDK für Python (Boto3) In diesem Beispiel wird das Flask-Web-Framework für das HTTP-Routing verwendet und in eine React-Webseite integriert, um eine voll funktionsfähige Webanwendung zu präsentieren.

- Erstellen Sie einen Flask-REST-Service, der sich in integrieren lässt. AWS-Services
- Lesen, schreiben und aktualisieren Sie Arbeitsaufgaben, die in einer Aurora-Serverless-Datenbank gespeichert sind.
- Erstellen Sie ein AWS Secrets Manager Geheimnis, das Datenbankanmeldedaten enthält, und verwenden Sie es, um Aufrufe an die Datenbank zu authentifizieren.
- Verwenden Sie Amazon SES, um E-Mail-Berichte über Arbeitsaufgaben zu senden.

Den vollständigen Quellcode und Anweisungen zur Einrichtung und Ausführung finden Sie im vollständigen Beispiel unter GitHub.

In diesem Beispiel verwendete Dienste

- Aurora
- Amazon RDS
- Amazon RDS Data Service
- Amazon SES

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Serverlose Beispiele für Amazon RDS

Die folgenden Codebeispiele zeigen, wie Sie Amazon RDS mit verwenden AWS SDKs.

Beispiele

- Herstellen einer Verbindung mit einer Amazon-RDS-Datenbank in einer Lambda-Funktion

Herstellen einer Verbindung mit einer Amazon-RDS-Datenbank in einer Lambda-Funktion

Die folgenden Codebeispiele veranschaulichen, wie eine Lambda-Funktion implementiert wird, die eine Verbindung zu einer RDS-Datenbank herstellt. Die Funktion stellt eine einfache Datenbankanfrage und gibt das Ergebnis zurück.

.NET

SDK für .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mithilfe von.NET.

```
using System.Data;
using System.Text.Json;
using Amazon.Lambda.APIGatewayEvents;
using Amazon.Lambda.Core;
using MySql.Data.MySqlClient;

// Assembly attribute to enable the Lambda function's JSON input to be converted
 into a .NET class.
[assembly:
 LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambdaJsonSeri

namespace aws_rds;

public class InputModel
{
    public string key1 { get; set; }
    public string key2 { get; set; }
}

public class Function
{
    /// <summary>
    // Handles the Lambda function execution for connecting to RDS using IAM
 authentication.
    /// </summary>
    /// <param name="input">The input event data passed to the Lambda function</
param>
    /// <param name="context">The Lambda execution context that provides runtime
 information</param>
```

```
    /// <returns>A response object containing the execution result</returns>

    public async Task<APIGatewayProxyResponse>
FunctionHandler(APIGatewayProxyRequest request, ILambdaContext context)
    {
        // Sample Input: {"body": "{\"key1\":\"20\", \"key2\":\"25\"}"}
        var input = JsonSerializer.Deserialize<InputModel>(request.Body);

        /// Obtain authentication token
        var authToken = RDSAuthTokenGenerator.GenerateAuthToken(
            Environment.GetEnvironmentVariable("RDS_ENDPOINT"),
            Convert.ToInt32(Environment.GetEnvironmentVariable("RDS_PORT")),
            Environment.GetEnvironmentVariable("RDS_USERNAME")
        );

        /// Build the Connection String with the Token
        string connectionString =
$"Server={Environment.GetEnvironmentVariable("RDS_ENDPOINT")};" +

$"Port={Environment.GetEnvironmentVariable("RDS_PORT")};" +

$"Uid={Environment.GetEnvironmentVariable("RDS_USERNAME")};" +
                                  $"Pwd={authToken};";


        try
        {
            await using var connection = new MySqlConnection(connectionString);
            await connection.OpenAsync();

            const string sql = "SELECT @param1 + @param2 AS Sum";

            await using var command = new MySqlCommand(sql, connection);
            command.Parameters.AddWithValue("@param1", int.Parse(input.key1 ??
"0"));
            command.Parameters.AddWithValue("@param2", int.Parse(input.key2 ??
"0"));

            await using var reader = await command.ExecuteReaderAsync();
            if (await reader.ReadAsync())
            {
                int result = reader.GetInt32("Sum");
```

```
                //Sample Response: {"statusCode":200,"body":"{\"message\":\"The
 sum is: 45\"}","isBase64Encoded":false}
                return new APIGatewayProxyResponse
                {
                    StatusCode = 200,
                    Body = JsonSerializer.Serialize(new { message = $"The sum is:
 {result}" })
                };
            }

        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }

        return new APIGatewayProxyResponse
        {
            StatusCode = 500,
            Body = JsonSerializer.Serialize(new { error = "Internal server
 error" })
        };
    }
}
```

Go

SDK für Go V2

> **Note**
>
> Es gibt noch mehr dazu. GitHub Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mit Go.

```
/*
Golang v2 code here.
```

```
*/

package main

import (
 "context"
 "database/sql"
 "encoding/json"
 "fmt"
 "os"

 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
 _ "github.com/go-sql-driver/mysql"
)

type MyEvent struct {
 Name string `json:"name"`
}

func HandleRequest(event *MyEvent) (map[string]interface{}, error) {

 var dbName string = os.Getenv("DatabaseName")
 var dbUser string = os.Getenv("DatabaseUser")
 var dbHost string = os.Getenv("DBHost") // Add hostname without https
 var dbPort int = os.Getenv("Port")      // Add port number
 var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
 var region string = os.Getenv("AWS_REGION")

 cfg, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  panic("configuration error: " + err.Error())
 }

 authenticationToken, err := auth.BuildAuthToken(
  context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
 if err != nil {
  panic("failed to create authentication token: " + err.Error())
 }

 dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
  dbUser, authenticationToken, dbEndpoint, dbName,
 )
```

```
 db, err := sql.Open("mysql", dsn)
 if err != nil {
  panic(err)
 }

 defer db.Close()

 var sum int
 err = db.QueryRow("SELECT ?+? AS sum", 3, 2).Scan(&sum)
 if err != nil {
  panic(err)
 }
 s := fmt.Sprint(sum)
 message := fmt.Sprintf("The selected sum is: %s", s)

 messageBytes, err := json.Marshal(message)
 if err != nil {
  return nil, err
 }

 messageString := string(messageBytes)
 return map[string]interface{}{
  "statusCode": 200,
  "headers":    map[string]string{"Content-Type": "application/json"},
  "body":       messageString,
 }, nil
}

func main() {
 lambda.Start(HandleRequest)
}
```

Java

SDK für Java 2.x

> **Note**
>
> Es gibt noch mehr GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mit Java.

```
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.APIGatewayProxyRequestEvent;
import com.amazonaws.services.lambda.runtime.events.APIGatewayProxyResponseEvent;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rdsdata.RdsDataClient;
import software.amazon.awssdk.services.rdsdata.model.ExecuteStatementRequest;
import software.amazon.awssdk.services.rdsdata.model.ExecuteStatementResponse;
import software.amazon.awssdk.services.rdsdata.model.Field;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class RdsLambdaHandler implements
 RequestHandler<APIGatewayProxyRequestEvent, APIGatewayProxyResponseEvent> {

    @Override
    public APIGatewayProxyResponseEvent handleRequest(APIGatewayProxyRequestEvent
 event, Context context) {
        APIGatewayProxyResponseEvent response = new
 APIGatewayProxyResponseEvent();

        try {
            // Obtain auth token
            String token = createAuthToken();
```

```
            // Define connection configuration
            String connectionString = String.format("jdbc:mysql://%s:%s/%s?
useSSL=true&requireSSL=true",
                    System.getenv("ProxyHostName"),
                    System.getenv("Port"),
                    System.getenv("DBName"));

            // Establish a connection to the database
            try (Connection connection =
 DriverManager.getConnection(connectionString, System.getenv("DBUserName"),
 token);
                 PreparedStatement statement =
 connection.prepareStatement("SELECT ? + ? AS sum")) {

                statement.setInt(1, 3);
                statement.setInt(2, 2);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        int sum = resultSet.getInt("sum");
                        response.setStatusCode(200);
                        response.setBody("The selected sum is: " + sum);
                    }
                }
            }

        } catch (Exception e) {
            response.setStatusCode(500);
            response.setBody("Error: " + e.getMessage());
        }

        return response;
    }

    private String createAuthToken() {
        // Create RDS Data Service client
        RdsDataClient rdsDataClient = RdsDataClient.builder()
                .region(Region.of(System.getenv("AWS_REGION")))
                .credentialsProvider(DefaultCredentialsProvider.create())
                .build();

        // Define authentication request
        ExecuteStatementRequest request = ExecuteStatementRequest.builder()
                .resourceArn(System.getenv("ProxyHostName"))
```

```
                .secretArn(System.getenv("DBUserName"))
                .database(System.getenv("DBName"))
                .sql("SELECT 'RDS IAM Authentication'")
                .build();

        // Execute request and obtain authentication token
        ExecuteStatementResponse response =
 rdsDataClient.executeStatement(request);
        Field tokenField = response.records().get(0).get(0);

        return tokenField.stringValue();
    }
}
```

JavaScript

SDK für JavaScript (v3)

> **Note**
>
> Es gibt noch mehr dazu GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mithilfe von JavaScript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/*
Node.js code here.
*/
// ES6+ example
import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

async function createAuthToken() {
  // Define connection authentication parameters
  const dbinfo = {
```

```
    hostname: process.env.ProxyHostName,
    port: process.env.Port,
    username: process.env.DBUserName,
    region: process.env.AWS_REGION,

  }

  // Create RDS Signer object
  const signer = new Signer(dbinfo);

  // Request authorization token from RDS, specifying the username
  const token = await signer.getAuthToken();
  return token;
}

async function dbOps() {

  // Obtain auth token
  const token = await createAuthToken();
  // Define connection configuration
  let connectionConfig = {
    host: process.env.ProxyHostName,
    user: process.env.DBUserName,
    password: token,
    database: process.env.DBName,
    ssl: 'Amazon RDS'
  }
  // Create the connection to the DB
  const conn = await mysql.createConnection(connectionConfig);
  // Obtain the result of the query
  const [res,] = await conn.execute('select ?+? as sum', [3, 2]);
  return res;

}

export const handler = async (event) => {
  // Execute database flow
  const result = await dbOps();
  // Return result
  return {
    statusCode: 200,
    body: JSON.stringify("The selected sum is: " + result[0].sum)
  }
};
```

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mithilfe von TypeScript.

```
import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

// RDS settings
// Using '!' (non-null assertion operator) to tell the TypeScript compiler that
 the DB settings are not null or undefined,
const proxy_host_name = process.env.PROXY_HOST_NAME!
const port = parseInt(process.env.PORT!)
const db_name = process.env.DB_NAME!
const db_user_name = process.env.DB_USER_NAME!
const aws_region = process.env.AWS_REGION!


async function createAuthToken(): Promise<string> {

    // Create RDS Signer object
    const signer = new Signer({
        hostname: proxy_host_name,
        port: port,
        region: aws_region,
        username: db_user_name
    });

    // Request authorization token from RDS, specifying the username
    const token = await signer.getAuthToken();
    return token;
}

async function dbOps(): Promise<mysql.QueryResult | undefined> {
    try {
        // Obtain auth token
        const token = await createAuthToken();
        const conn = await mysql.createConnection({
            host: proxy_host_name,
            user: db_user_name,
            password: token,
            database: db_name,
```

```
            ssl: 'Amazon RDS' // Ensure you have the CA bundle for SSL connection
        });
        const [rows, fields] = await conn.execute('SELECT ? + ? AS sum', [3, 2]);
        console.log('result:', rows);
        return rows;
    }
    catch (err) {
        console.log(err);
    }
}

export const lambdaHandler = async (event: any): Promise<{ statusCode: number;
 body: string }> => {
    // Execute database flow
    const result = await dbOps();

    // Return error is result is undefined
    if (result == undefined)
        return {
            statusCode: 500,
            body: JSON.stringify(`Error with connection to DB host`)
        }

    // Return result
    return {
        statusCode: 200,
        body: JSON.stringify(`The selected sum is: ${result[0].sum}`)
    };
};
```

PHP

SDK für PHP

> **Note**
>
> Es gibt noch mehr dazu. GitHub Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mit PHP.

```
<?php
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0

# using bref/bref and bref/logger for simplicity

use Bref\Context\Context;
use Bref\Event\Handler as StdHandler;
use Bref\Logger\StderrLogger;
use Aws\Rds\AuthTokenGenerator;
use Aws\Credentials\CredentialProvider;

require __DIR__ . '/vendor/autoload.php';

class Handler implements StdHandler
{
    private StderrLogger $logger;
    public function __construct(StderrLogger $logger)
    {
        $this->logger = $logger;
    }


    private function getAuthToken(): string {
        // Define connection authentication parameters
        $dbConnection = [
            'hostname' => getenv('DB_HOSTNAME'),
            'port' => getenv('DB_PORT'),
            'username' => getenv('DB_USERNAME'),
            'region' => getenv('AWS_REGION'),
        ];

        // Create RDS AuthTokenGenerator object
        $generator = new
 AuthTokenGenerator(CredentialProvider::defaultProvider());

        // Request authorization token from RDS, specifying the username
        return $generator->createToken(
            $dbConnection['hostname'] . ':' . $dbConnection['port'],
            $dbConnection['region'],
```

```
            $dbConnection['username']
        );
    }

    private function getQueryResults() {
        // Obtain auth token
        $token = $this->getAuthToken();

        // Define connection configuration
        $connectionConfig = [
            'host' => getenv('DB_HOSTNAME'),
            'user' => getenv('DB_USERNAME'),
            'password' => $token,
            'database' => getenv('DB_NAME'),
        ];

        // Create the connection to the DB
        $conn = new PDO(

"mysql:host={$connectionConfig['host']};dbname={$connectionConfig['database']}",
            $connectionConfig['user'],
            $connectionConfig['password'],
            [
                PDO::MYSQL_ATTR_SSL_CA => '/path/to/rds-ca-2019-root.pem',
                PDO::MYSQL_ATTR_SSL_VERIFY_SERVER_CERT => true,
            ]
        );

        // Obtain the result of the query
        $stmt = $conn->prepare('SELECT ?+? AS sum');
        $stmt->execute([3, 2]);

        return $stmt->fetch(PDO::FETCH_ASSOC);
    }

    /**
     * @param mixed $event
     * @param Context $context
     * @return array
     */
    public function handle(mixed $event, Context $context): array
    {
        $this->logger->info("Processing query");
```

```
        // Execute database flow
        $result = $this->getQueryResults();

        return [
            'sum' => $result['sum']
        ];
    }
}

$logger = new StderrLogger();
return new Handler($logger);
```

Python

SDK für Python (Boto3)

> **Note**
>
> Es gibt noch mehr GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mit Python.

```
import json
import os
import boto3
import pymysql

# RDS settings
proxy_host_name = os.environ['PROXY_HOST_NAME']
port = int(os.environ['PORT'])
db_name = os.environ['DB_NAME']
db_user_name = os.environ['DB_USER_NAME']
aws_region = os.environ['AWS_REGION']


# Fetch RDS Auth Token
def get_auth_token():
    client = boto3.client('rds')
```

```
    token = client.generate_db_auth_token(
        DBHostname=proxy_host_name,
        Port=port
        DBUsername=db_user_name
        Region=aws_region
    )
    return token

def lambda_handler(event, context):
    token = get_auth_token()
    try:
        connection = pymysql.connect(
            host=proxy_host_name,
            user=db_user_name,
            password=token,
            db=db_name,
            port=port,
            ssl={'ca': 'Amazon RDS'}  # Ensure you have the CA bundle for SSL
 connection
        )

        with connection.cursor() as cursor:
            cursor.execute('SELECT %s + %s AS sum', (3, 2))
            result = cursor.fetchone()

        return result

    except Exception as e:
        return (f"Error: {str(e)}")  # Return an error message if an exception
 occurs
```

Ruby

SDK für Ruby

> **Note**
>
> Es gibt noch mehr GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon RDS-Datenbank in einer Lambda-Funktion mit Ruby.

```
# Ruby code here.

require 'aws-sdk-rds'
require 'json'
require 'mysql2'

def lambda_handler(event:, context:)
  endpoint = ENV['DBEndpoint'] # Add the endpoint without https"
  port = ENV['Port']           # 3306
  user = ENV['DBUser']
  region = ENV['DBRegion']     # 'us-east-1'
  db_name = ENV['DBName']

  credentials = Aws::Credentials.new(
    ENV['AWS_ACCESS_KEY_ID'],
    ENV['AWS_SECRET_ACCESS_KEY'],
    ENV['AWS_SESSION_TOKEN']
  )
  rds_client = Aws::RDS::AuthTokenGenerator.new(
    region: region,
    credentials: credentials
  )

  token = rds_client.auth_token(
    endpoint: endpoint+ ':' + port,
    user_name: user,
    region: region
  )

  begin
    conn = Mysql2::Client.new(
      host: endpoint,
      username: user,
      password: token,
      port: port,
      database: db_name,
      sslca: '/var/task/global-bundle.pem',
      sslverify: true,
      enable_cleartext_plugin: true
    )
```

```
    a = 3
    b = 2
    result = conn.query("SELECT #{a} + #{b} AS sum").first['sum']
    puts result
    conn.close
    {
      statusCode: 200,
      body: result.to_json
    }
  rescue => e
    puts "Database connection failed due to #{e}"
  end
end
```

Rust

SDK für Rust

> **Note**
>
> Es gibt noch mehr GitHub. Das vollständige Beispiel sowie eine Anleitung zum Einrichten und Ausführen finden Sie im Repository mit Serverless-Beispielen.

Herstellen einer Verbindung zu einer Amazon-RDS-Datenbank in einer Lambda-Funktion mit Rust.

```
use aws_config::BehaviorVersion;
use aws_credential_types::provider::ProvideCredentials;
use aws_sigv4::{
    http_request::{sign, SignableBody, SignableRequest, SigningSettings},
    sign::v4,
};
use lambda_runtime::{run, service_fn, Error, LambdaEvent};
use serde_json::{json, Value};
use sqlx::postgres::PgConnectOptions;
use std::env;
use std::time::{Duration, SystemTime};

const RDS_CERTS: &[u8] = include_bytes!("global-bundle.pem");
```

```
async fn generate_rds_iam_token(
    db_hostname: &str,
    port: u16,
    db_username: &str,
) -> Result<String, Error> {
    let config = aws_config::load_defaults(BehaviorVersion::v2024_03_28()).await;

    let credentials = config
        .credentials_provider()
        .expect("no credentials provider found")
        .provide_credentials()
        .await
        .expect("unable to load credentials");
    let identity = credentials.into();
    let region = config.region().unwrap().to_string();

    let mut signing_settings = SigningSettings::default();
    signing_settings.expires_in = Some(Duration::from_secs(900));
    signing_settings.signature_location =
 aws_sigv4::http_request::SignatureLocation::QueryParams;

    let signing_params = v4::SigningParams::builder()
        .identity(&identity)
        .region(&region)
        .name("rds-db")
        .time(SystemTime::now())
        .settings(signing_settings)
        .build()?;

    let url = format!(
        "https://{db_hostname}:{port}/?Action=connect&DBUser={db_user}",
        db_hostname = db_hostname,
        port = port,
        db_user = db_username
    );

    let signable_request =
        SignableRequest::new("GET", &url, std::iter::empty(),
 SignableBody::Bytes(&[]))
            .expect("signable request");

    let (signing_instructions, _signature) =
        sign(signable_request, &signing_params.into())?.into_parts();
```

```
    let mut url = url::Url::parse(&url).unwrap();
    for (name, value) in signing_instructions.params() {
        url.query_pairs_mut().append_pair(name, &value);
    }

    let response = url.to_string().split_off("https://".len());

    Ok(response)
}

#[tokio::main]
async fn main() -> Result<(), Error> {
    run(service_fn(handler)).await
}

async fn handler(_event: LambdaEvent<Value>) -> Result<Value, Error> {
    let db_host = env::var("DB_HOSTNAME").expect("DB_HOSTNAME must be set");
    let db_port = env::var("DB_PORT")
        .expect("DB_PORT must be set")
        .parse::<u16>()
        .expect("PORT must be a valid number");
    let db_name = env::var("DB_NAME").expect("DB_NAME must be set");
    let db_user_name = env::var("DB_USERNAME").expect("DB_USERNAME must be set");

    let token = generate_rds_iam_token(&db_host, db_port, &db_user_name).await?;

    let opts = PgConnectOptions::new()
        .host(&db_host)
        .port(db_port)
        .username(&db_user_name)
        .password(&token)
        .database(&db_name)
        .ssl_root_cert_from_pem(RDS_CERTS.to_vec())
        .ssl_mode(sqlx::postgres::PgSslMode::Require);

    let pool = sqlx::postgres::PgPoolOptions::new()
        .connect_with(opts)
        .await?;

    let result: i32 = sqlx::query_scalar("SELECT $1 + $2")
        .bind(3)
        .bind(2)
        .fetch_one(&pool)
        .await?;
```

```
    println!("Result: {:?}", result);

    Ok(json!({
        "statusCode": 200,
        "content-type": "text/plain",
        "body": format!("The selected sum is: {result}")
    }))
}
```

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unterVerwenden Sie diesen Dienst mit einem AWS SDK. Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Sicherheit in Amazon RDS

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einem Rechenzentrum und einer Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS -Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon RDS Amazon gelten, finden Sie unter AWS Services im Umfang nach Compliance-Programmen.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amazon RDS anwenden können. In den folgenden Themen erfahren Sie, wie Sie Amazon RDS Amazon konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen helfen, Ihre Amazon RDS Amazon zu überwachen und zu sichern.

Sie können den Zugriff auf Ihre Amazon RDS Amazon und Ihre Datenbanken in einem verwalten. Die Methode, mit der Sie den Zugriff verwalten, hängt davon ab, welche Art von Aufgabe der Benutzer mit Amazon RDS Amazon ausführen muss:

- Führen Sie Ihren in einer Virtual Private Cloud (VPC) aus, die auf dem Amazon VPC-Service basiert, um die größtmögliche Netzwerkzugriffskontrolle zu gewährleisten. Weitere Hinweise zum Erstellen eines in einer VPC finden Sie unterAmazon VPC und Amazon RDS.
- Verwenden Sie AWS Identity and Access Management (IAM) -Richtlinien, um Berechtigungen zuzuweisen, die festlegen, wer Amazon RDS Amazon verwalten darf. Sie können IAM beispielsweise verwenden, um zu bestimmen, wer erstellen, beschreiben, ändern und löschen, Ressourcen taggen oder Sicherheitsgruppen ändern darf.

- Verwenden Sie Sicherheitsgruppen, um zu steuern, welche IP-Adressen oder EC2 Amazon-Instances eine Verbindung zu Ihren Datenbanken in einem herstellen können. Wenn Sie zum ersten Mal einen erstellen, verhindert dessen Firewall jeglichen Datenbankzugriff, außer durch Regeln, die von einer zugehörigen Sicherheitsgruppe festgelegt wurden.
- Verwenden Sie Secure Socket Layer- (SSL) - oder Transport Layer Security (TLS) -Verbindungen mit DB-Instances, auf denen die Datenbankmodule Db2, MySQL, MariaDB, PostgreSQL, Oracle oder Microsoft SQL Server ausgeführt werden.
- Verwenden Sie die Amazon RDS, um Ihre DB-Instances und Snapshots im Ruhezustand zu sichern. Amazon RDS Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.
- Verwenden Sie Netzwerkverschlüsselung und Transparent Data Encryption mit Oracle-DB-Instances. Weitere Informationen finden Sie unter Oracle Native Network Encryption und Oracle Transparent Data Encryption
- Diese Funktionen arbeiten genauso, als würden sich die Datenbanken in Ihrem lokalen Netzwerk befinden.

> **Note**
>
> Sie müssen die Sicherheit nur für Ihre Anwendungsfälle konfigurieren. Sie müssen den Sicherheitszugriff für Prozesse, die Amazon RDS verwaltet, nicht konfigurieren. Dazu gehört das Erstellen von Backups, das Replizieren von Daten zwischen einer primären DB-Instance und einem Lesereplikat und andere Prozesse.

Weitere Informationen zur Verwaltung des Zugriffs auf Amazon RDS und Ihre Datenbanken in einem finden Sie in den folgenden Themen.

Themen

- Datenbankauthentifizierung mit Amazon RDS
- Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager
- Datenschutz in Amazon RDS
- Identitäts- und Zugriffsmanagement für Amazon RDS
- Protokollieren und Überwachen in Amazon RDS
- Konformitätsvalidierung für Amazon RDS

- Ausfallsicherheit in Amazon RDS
- Infrastruktursicherheit in Amazon RDS
- Amazon-RDS-API und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)
- Bewährte Methoden für die Sicherheit in Amazon RDS
- Zugriffskontrolle mit Sicherheitsgruppen
- Berechtigungen von Hauptbenutzerkonten
- Verwenden von serviceverknüpften Rollen für Amazon RDS
- Amazon VPC und Amazon RDS

Datenbankauthentifizierung mit Amazon RDS

Amazon RDS unterstützt mehrere Methoden zur Authentifizierung von Datenbankbenutzern.

Passwort-, Kerberos- und IAM-Datenbank-Authentifizierung verwenden verschiedene Methoden zur Authentifizierung bei der Datenbank. Daher kann sich ein bestimmter Benutzer mit nur einer einzigen Authentifizierungsmethode bei einer Datenbank anmelden.

Verwenden Sie für PostgreSQL nur eine der folgenden Rolleneinstellungen für einen Benutzer einer bestimmten Datenbank:

- Um die IAM-Datenbank-Authentifizierung zu verwenden, weisen Sie die `rds_iam`-Rolle dem Benutzer zu.
- Um Kerberos-Authentifizierung zu verwenden, weisen Sie die `rds_ad`-Rolle dem Benutzer zu.
- Um die Passwort-Authentifizierung zu verwenden, weisen Sie keine der beiden `rds_iam`- oder `rds_ad`- Rollen dem Benutzer zu.

Weisen Sie nicht beide Rollen `rds_iam` und `rds_ad` einem Benutzer einer PostgreSQL-Datenbank zu, weder direkt noch indirekt durch verschachtelten gewährtem Zugriff. Wenn die `rds_iam`-Rolle dem Hauptbenutzer hinzugefügt wird, hat die IAM-Authentifizierung Vorrang vor der Passwort-Authentifizierung, so dass sich der Hauptbenutzer als IAM-Benutzer anmelden muss.

> **Important**
>
> Wir empfehlen Ihnen, den Hauptbenutzer nicht direkt in Ihren Anwendungen zu verwenden. Bleiben Sie stattdessen bei der bewährten Methode, einen Datenbankbenutzer zu

verwenden, der mit den Mindestberechtigungen erstellt wurde, die für Ihre Anwendung erforderlich sind.

Themen

- Passwortauthentifizierung
- IAM-Datenbankauthentifizierung
- Kerberos-Authentifizierung

Passwortauthentifizierung

Mit der Passwortauthentifizierung führt Ihre Datenbank die gesamte Verwaltung von Benutzerkonten durch. Sie erstellen Benutzer mit SQL-Anweisungen wie `CREATE USER`, mit der entsprechenden Klausel, die von der DB-Engine zum Angeben von Kennwörtern benötigt wird. In MySQL lautet die Anweisung beispielsweise `CREATE USER` *`name`* `IDENTIFIED BY` *`password`*, während in PostgreSQL die Anweisung lautet. `CREATE USER` *`name`* `WITH PASSWORD` *`password`*

Mit der Passwortauthentifizierung steuert und authentifiziert Ihre Datenbank Benutzerkonten. Wenn eine DB-Engine über starke Passwortverwaltungsfunktionen verfügt, können diese die Sicherheit erhöhen. Die Datenbankauthentifizierung ist möglicherweise einfacher mit der Passwortauthentifizierung zu verwalten, wenn Sie kleine Benutzergemeinschaften haben. Da in diesem Fall Klartext-Passwörter generiert werden, AWS Secrets Manager kann die Integration mit die Sicherheit erhöhen.

Informationen zur Verwendung von Secrets Manager mit Amazon RDS finden Sie unter Creating a Basic Secret und Roting Secrets for supported Amazon RDS databases im AWS Secrets Manager User Guide. Informationen zum programmgesteuerten Abrufen Ihrer Secrets in Ihren benutzerdefinierten Anwendungen finden Sie unter Abrufen des Secret-Werts im AWS Secrets Manager -Benutzerhandbuch.

IAM-Datenbankauthentifizierung

Sie können sich bei Ihrem mithilfe der AWS Identity and Access Management (IAM-) Datenbankauthentifizierung authentifizieren. Stattdessen verwenden Sie ein Authentifizierungstoken.

Weitere Informationen zur IAM-Datenbankauthentifizierung, einschließlich Informationen zur Verfügbarkeit bestimmter DB-Engines, finden Sie unterIAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

Kerberos-Authentifizierung

Amazon RDS unterstützt die externe Authentifizierung von Datenbankbenutzern mithilfe von Kerberos und Microsoft Active Directory. Kerberos ist ein Netzwerk-Authentifizierungsprotokoll, das Tickets und symmetrische Schlüsselkryptographie verwendet, um die Notwendigkeit der Übertragung von Passwörtern über das Netzwerk zu vermeiden. Kerberos wurde in Active Directory integriert und wurde entwickelt, um Benutzer gegenüber Netzwerkressourcen wie Datenbanken zu authentifizieren.

Die Amazon RDS-Unterstützung für Kerberos und Active Directory bietet die Vorteile von Single Sign-On und zentraler Authentifizierung von Datenbankbenutzern. Sie können Ihre Benutzeranmeldeinformationen in Active Directory speichern.

Um Anmeldeinformationen aus Ihrem selbstverwalteten Active Directory zu verwenden, müssen Sie eine Vertrauensbeziehung zu dem AWS Directory Service für Microsoft Active Directory einrichten, mit dem der verbunden ist.

RDS für PostgreSQL und RDS für MySQL Aurora PostgreSQL und Aurora MySQL bidirektionale Forest-Trust-Beziehungen mit gesamtstrukturweiter Authentifizierung oder selektiver Authentifizierung.

In einigen Szenarien können Sie die Kerberos-Authentifizierung über eine externe Vertrauensbeziehung konfigurieren. Dazu muss Ihr selbstverwaltetes Active Directory über zusätzliche Einstellungen verfügen. Dies beinhaltet, ist aber nicht beschränkt auf Kerberos Forest Search Order.

Microsoft SQL Server- und PostgreSQL-DB-Instances unterstützen unidirektionale und bidirektionale Forest-Trust-Beziehungen. Oracle-DB-Instances unterstützen unidirektionale und bidirektionale externe Vertrauensbeziehungen sowie Forest-Trust-Beziehungen. Weitere Informationen finden Sie unter Zeitpunkt zum Erstellen einer Vertrauensstellung im AWS Directory Service - Administrationshandbuch.

Informationen zur Kerberos-Authentifizierung mit einer bestimmten Engine finden Sie im Folgenden:

- Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server
- KerberosAuthentifizierung für Amazon RDS for MySQL verwenden
- Konfigurieren der Kerberos-Authentifizierung für Amazon RDS for Oracle
- Verwenden der Kerberos-Authentifizierung mit Amazon RDS for PostgreSQL
- KerberosAuthentifizierung für Amazon RDS for Db2 verwenden .

> **Note**
>
> Derzeit wird die Kerberos-Authentifizierung für MariaDB DB-Instances nicht unterstützt.

Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager

Amazon RDS lässt sich in Secrets Manager integrieren, um Hauptbenutzerpasswörter für Ihre DB-Instances und Multi-AZ-DB-Cluster zu verwalten.

Themen

- Einschränkungen für die Integration von Secrets Manager in Amazon RDS
- Überblick über die Verwaltung von Masterbenutzer-Passwörtern mit AWS Secrets Manager
- Vorteile der Verwaltung von Hauptbenutzerpasswörtern mit Secrets Manager
- Erforderliche Berechtigungen für die Integration von Secrets Manager
- Durchsetzung der Verwaltung des Masterbenutzerkennworts durch RDS in AWS Secrets Manager
- Verwaltung des Hauptbenutzerpassworts für eine DB-Instance mit Secrets Manager
- Verwaltung des Master-Benutzerpassworts für eine RDS for Oracle-Mandantendatenbank mit Secrets Manager
- Verwaltung des Hauptbenutzerpassworts für einen Multi-AZ-DB-Cluster mit Secrets Manager
- Rotieren des Hauptbenutzerpasswort-Secrets für eine DB-Instance
- Rotieren des Hauptbenutzerpasswort-Secrets für einen Multi-AZ-DB-Cluster
- Anzeigen der Details zu einem Secret für eine DB-Instance
- Anzeigen der Details zu einem Secret für einen Multi-AZ-DB-Cluster
- Details zu einem geheimen Schlüssel für eine Tenant-Datenbank anzeigen
- Verfügbarkeit von Regionen und Versionen

Einschränkungen für die Integration von Secrets Manager in Amazon RDS

Die Verwaltung von Hauptpasswörtern mit Secrets Manager wird für die folgenden Funktionen nicht unterstützt:

- Erstellen einer Read Replica, wenn die Quell-DB oder der DB-Cluster Anmeldeinformationen mit Secrets Manager verwaltet. Dies gilt für alle DB-Engines außer RDS für SQL Server.
- Amazon Blue/Green RDS-Bereitstellungen
- Amazon RDS Custom
- Umstellung auf Oracle Data Guard

Überblick über die Verwaltung von Masterbenutzer-Passwörtern mit AWS Secrets Manager

Mit AWS Secrets Manager können Sie hartcodierte Anmeldeinformationen in Ihrem Code, einschließlich Datenbankkennwörtern, durch einen API-Aufruf an Secrets Manager ersetzen, um das Geheimnis programmgesteuert abzurufen. Weitere Informationen zu Secrets Manager finden Sie im Benutzerhandbuch für AWS Secrets Manager.

Wenn Sie Datenbankgeheimnisse in Secrets Manager speichern, AWS-Konto fallen Gebühren an. Informationen zu Preisen erhalten Sie unter AWS Secrets Manager -Preise.

Sie können angeben, dass RDS das Hauptbenutzerpasswort in Secrets Manager für eine DB-Instance von Amazon RDS oder einen Multi-AZ-DB-Cluster verwaltet, wenn Sie eine der folgenden Operationen ausführen:

- Erstellen einer DB-Instance.
- Erstellen Sie einen Multi-AZ-DB-Cluster
- Erstellen Sie eine Tenant-Datenbank in einem RDS für Oracle CDB
- Ändern einer DB-Instance
- Ändern Sie einen Multi-AZ-DB-Cluster
- Ändern Sie eine Tenant-Datenbank (nur RDS für Oracle)
- Eine DB-Instance aus Amazon S3 wiederherstellen
- Stellen Sie eine DB-Instance aus einem Snapshot oder zu einem bestimmten Zeitpunkt wieder her (nur RDS für Oracle)

Wenn Sie angeben, dass RDS das Hauptbenutzerpasswort in Secrets Manager verwaltet, generiert RDS das Passwort und speichert es in Secrets Manager. Sie können direkt mit dem Secret interagieren, um die Anmeldeinformationen für den Hauptbenutzer abzurufen. Sie können auch einen vom Kunden verwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Schlüssel verwenden, der von Secrets Manager bereitgestellt wird.

RDS verwaltet die Einstellungen für das Secret und rotiert das Secret standardmäßig alle sieben Tage. Sie können einige Einstellungen ändern, wie zum Beispiel den Rotationsplan. Wenn Sie eine DB-Instance löschen, der ein Secret in Secrets Manager verwaltet, werden das Secret und die zugehörigen Metadaten ebenfalls gelöscht.

Um eine Verbindung mit einer DB-Instance oder einem Multi-AZ-DB-Cluster mit den Anmeldeinformationen in einem Secret herzustellen, können Sie das Secret von Secrets Manager abrufen. Weitere Informationen finden Sie im Benutzerhandbuch unter Abrufen von Geheimnissen aus AWS Secrets Manager und Herstellen einer Verbindung zu einer SQL-Datenbank mit Anmeldeinformationen in einem AWS Secrets Manager Geheimnis herstellen.AWS Secrets Manager

Vorteile der Verwaltung von Hauptbenutzerpasswörtern mit Secrets Manager

Die Verwaltung von RDS-Hauptbenutzerpasswörtern mit Secrets Manager bietet die folgenden Vorteile:

- RDS generiert automatisch Datenbankanmeldeinformationen.
- RDS speichert und verwaltet automatisch Datenbankanmeldedaten in AWS Secrets Manager.
- RDS rotiert die Datenbankanmeldeinformationen regelmäßig, ohne dass Anwendungsänderungen erforderlich sind.
- Secrets Manager schützt Datenbankanmeldeinformationen vor menschlichem Zugriff und der Klartextansicht.
- Secrets Manager ermöglicht das Abrufen von Datenbankanmeldeinformationen in Secrets für Datenbankverbindungen.
- Secrets Manager ermöglicht eine detaillierte Steuerung des Zugriffs auf Datenbankanmeldeinformationen in Secrets mithilfe von IAM.
- Optional können Sie die Datenbankverschlüsselung von der Anmeldeinformationsverschlüsselung mit unterschiedlichen KMS-Schlüsseln trennen.
- Sie können die manuelle Verwaltung und Rotation der Datenbankanmeldeinformationen vermeiden.
- Sie können Datenbankanmeldedaten einfach mit AWS CloudTrail Amazon überwachen CloudWatch.

Weitere Informationen zu den Vorteilen von Secrets Manager finden Sie im Benutzerhandbuch für AWS Secrets Manager.

Erforderliche Berechtigungen für die Integration von Secrets Manager

Benutzer müssen über die erforderlichen Berechtigungen verfügen, um Operationen im Zusammenhang mit der Integration von Secrets Manager auszuführen. Sie können IAM-

Richtlinien erstellen, die die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die benötigt werden. Sie können diese Richtlinien dann den IAM-Berechtigungssätzen oder -Rollen zuordnen, die diese Berechtigungen benötigen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.

Für Erstellungs-, Änderungs- oder Wiederherstellungsoperationen muss der Benutzer, der angibt, dass Amazon RDS das Hauptbenutzerpasswort in Secrets Manager verwaltet, über entsprechende Berechtigungen für folgende Operationen verfügen:

- `kms:DescribeKey`
- `secretsmanager:CreateSecret`
- `secretsmanager:TagResource`

Die `kms:DescribeKey` Erlaubnis ist erforderlich, um auf Ihren vom Kunden verwalteten Schlüssel zuzugreifen `MasterUserSecretKmsKeyId` und ihn zu beschreiben`aws/secretsmanager`.

Für Erstellungs-, Änderungs- oder Wiederherstellungsoperationen muss der Benutzer, der den benutzerdefinierten Schlüssel zum Entschlüsseln des Secrets in Secrets Manager angibt, über entsprechende Berechtigungen für folgende Operationen verfügen:

- `kms:Decrypt`
- `kms:GenerateDataKey`
- `kms:CreateGrant`

Für Änderungsoperationen muss der Benutzer, der das Hauptbenutzerpasswort in Secrets Manager rotiert, über entsprechende Berechtigungen für die folgende Operation verfügen:

- `secretsmanager:RotateSecret`

Durchsetzung der Verwaltung des Masterbenutzerkennworts durch RDS in AWS Secrets Manager

Sie können IAM-Bedingungsschlüssel verwenden, um die RDS-Verwaltung des Hauptbenutzerpassworts in AWS Secrets Manager zu erzwingen. Die folgende Richtlinie erlaubt es Benutzern nicht, DB-Instances oder DB-Cluster zu erstellen oder wiederherzustellen oder Tenant-

Datenbanken zu erstellen oder zu ändern, es sei denn, das Master-Benutzerkennwort wird von RDS in Secrets Manager verwaltet.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": ["rds:CreateDBInstance", "rds:CreateDBCluster",
 "rds:RestoreDBInstanceFromS3", "rds:RestoreDBClusterFromS3",
                       "rds:RestoreDBInstanceFromDBSnapshot",
 "rds:RestoreDBInstanceToPointInTime", "rds:CreateTenantDatabase",
                       "rds:ModifyTenantDatabase"],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "rds:ManageMasterUserPassword": false
                }
            }
        }
    ]
}
```

> **Note**
>
> Diese Richtlinie erzwingt die Passwortverwaltung bereits AWS Secrets Manager bei der Erstellung. Sie können die Secrets-Manager-Integration jedoch nach wie vor deaktivieren und ein Hauptpasswort manuell festlegen, indem Sie die Instance ändern.
> Um dies zu verhindern, nehmen Sie `rds:ModifyDBInstance`, `rds:ModifyDBCluster` in den Aktionsblock der Richtlinie auf. Beachten Sie, dass der Benutzer dadurch keine weiteren Änderungen an vorhandenen Instances vornehmen kann, für die die Secrets-Manager-Integration nicht aktiviert ist.

Weitere Informationen zum Verwenden der Bedingungsschlüssels in IAM-Richtlinien finden Sie unter Schlüssel zu den Versicherungsbedingungen für Amazon RDS und Beispielrichtlinien: Verwenden von Bedingungsschlüsseln.

Verwaltung des Hauptbenutzerpassworts für eine DB-Instance mit Secrets Manager

Sie können die RDS-Verwaltung des Hauptbenutzerpassworts in Secrets Manager konfigurieren, wenn Sie die folgenden Aktionen ausführen:

- Erstellen einer Amazon RDS-DB-Instance
- Ändern einer Amazon RDS DB-Instance
- Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance
- Wiederherstellung auf einer DB-Instance(Nur RDS für Oracle)
- Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS(Nur RDS für Oracle)

Sie können die oben genannten Operationen mit der RDS-Konsole AWS CLI, der oder der RDS-API ausführen.

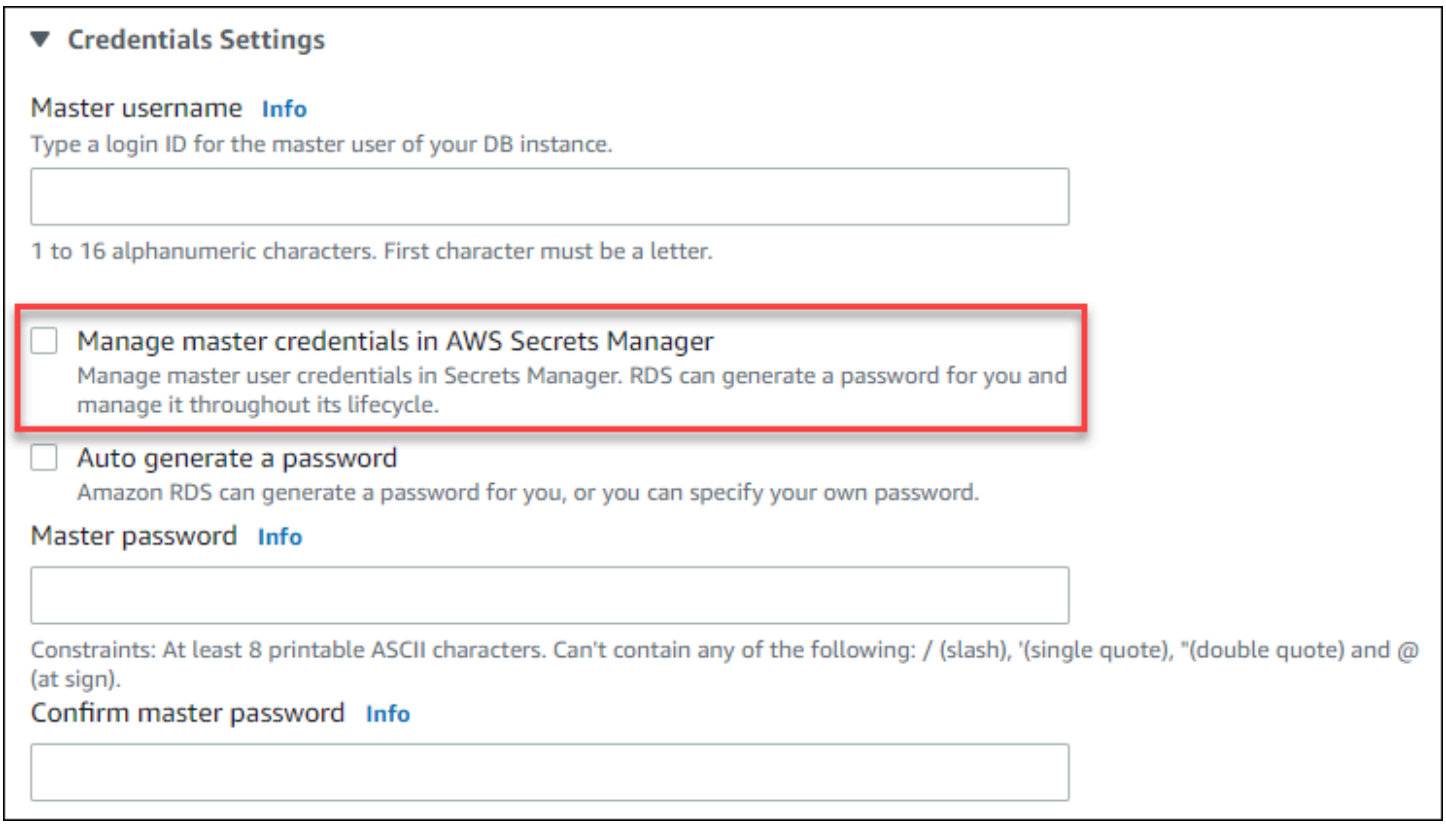
Konsole

Folgen Sie den Anweisungen zum Erstellen oder Ändern einer DB-Instance mit der RDS-Konsole:

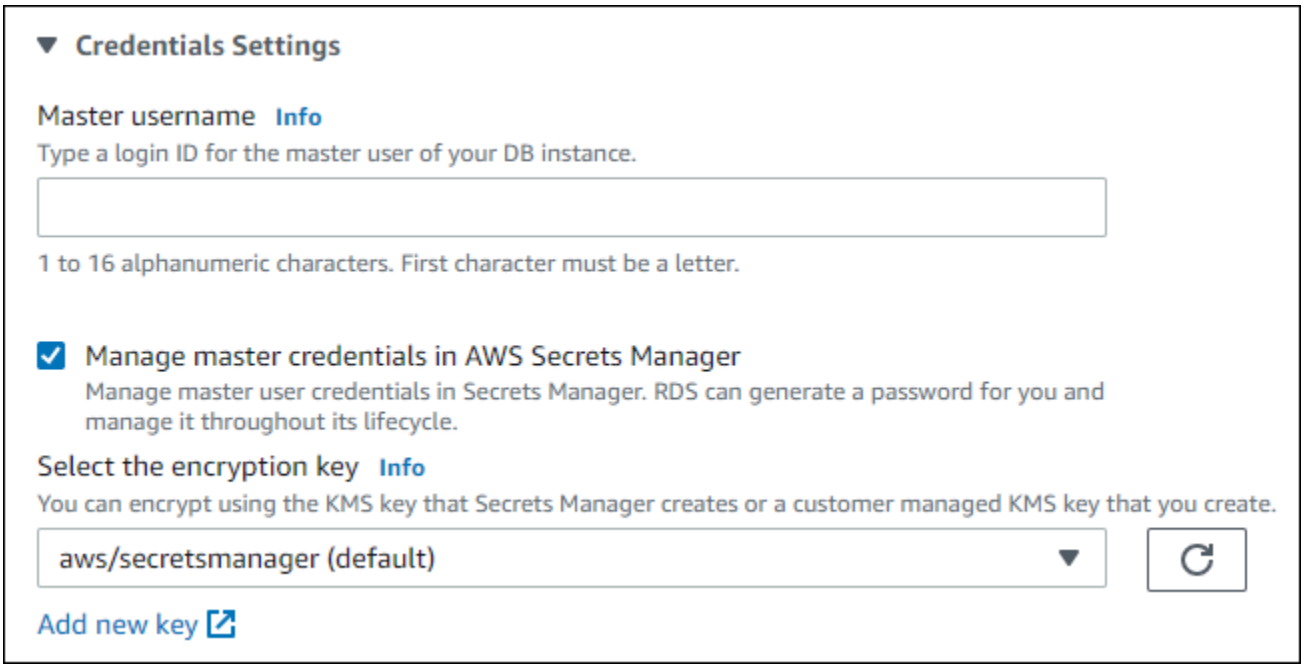
- Erstellen einer DB-Instance
- Ändern einer Amazon RDS DB-Instance
- Importieren von Daten aus Amazon S3 in eine neue MySQL-DB-Instance

Wenn Sie die RDS-Konsole verwenden, um eine dieser Operationen auszuführen, können Sie angeben, dass das Hauptbenutzerpasswort von RDS in Secrets Manager verwaltet wird. Wenn Sie eine DB-Instance erstellen oder wiederherstellen, wählen Sie AWS Secrets Manager in den Anmeldeinformationseinstellungen die Option Master-Anmeldeinformationen verwalten aus. Wenn Sie eine DB-Instance ändern, wählen Sie AWS Secrets Manager in den Einstellungen die Option Master-Anmeldeinformationen verwalten aus.

Die folgende Abbildung zeigt ein Beispiel für die Einstellung Hauptanmeldeinformationen in AWS Secrets Manager verwalten beim Erstellen oder Wiederherstellen einer DB-Instance.



Wenn Sie diese Option auswählen, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.



Sie können wählen, ob Sie das Secret mit einem von Secrets Manager bereitgestellten KMS-Schlüssel oder mit einem von Ihnen erstellten kundenverwalteten Schlüssel verschlüsseln möchten.

Nachdem RDS die Datenbankanmeldedaten für eine DB-Instance verwaltet hat, können Sie den KMS-Schlüssel, der zur Verschlüsselung des Geheimnisses verwendet wird, nicht mehr ändern.

Sie können andere Einstellungen auswählen, die Ihren Anforderungen entsprechen. Weitere Informationen zu den verfügbaren Einstellungen beim Erstellen einer DB-Instance finden Sie unterEinstellungen für DB-Instances. Weitere Informationen zu den verfügbaren Einstellungen, wenn Sie eine DB-Instance ändern, finden Sie unterEinstellungen für DB-Instances.

AWS CLI

Um das Masterbenutzerkennwort mit RDS in Secrets Manager zu verwalten, geben Sie die `--manage-master-user-password` Option in einem der folgenden AWS CLI Befehle an:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-s3
- restore-db-instance-from-db-snapshot (nur RDS für Oracle)
- restore-db-instance-to- point-in-time (RDS nur für Oracle)

Wenn Sie die `--manage-master-user-password`-Option angeben, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Sie können auch einen kundenverwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Standardschlüssel verwenden, der von Secrets Manager bereitgestellt wird. Verwenden Sie die `--master-user-secret-kms-key-id`-Option, um einen kundenverwalteten Schlüssel anzugeben. Die AWS KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Um einen KMS-Schlüssel in einem anderen zu verwenden AWS-Konto, geben Sie den Schlüssel-ARN oder den Alias-ARN an. Nachdem RDS die Datenbankanmeldeinformationen für eine DB-Instance verwaltet hat, können Sie den KMS-Schlüssel, der zum Verschlüsseln des Secrets verwendet wird, nicht mehr ändern.

Sie können andere Einstellungen auswählen, die Ihren Anforderungen entsprechen. Weitere Informationen zu den verfügbaren Einstellungen beim Erstellen einer DB-Instance finden Sie unter Einstellungen für DB-Instances. Weitere Informationen zu den verfügbaren Einstellungen beim Ändern einer DB-Instance finden Sie unter Einstellungen für DB-Instances.

Im folgenden Beispiel wird eine DB-Instance erstellt und angegeben, dass RDS das Masterbenutzerkennwort in Secrets Manager verwaltet. Das Secret wird mit dem KMS-Schlüssel verschlüsselt, der von Secrets Manager bereitgestellt wird.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.39 \
    --db-instance-class db.r5b.large \
    --allocated-storage 200 \
    --master-username testUser \
    --manage-master-user-password
```

Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
    --engine-version 8.0.39 ^
    --db-instance-class db.r5b.large ^
    --allocated-storage 200 ^
    --master-username testUser ^
    --manage-master-user-password
```

RDS-API

Wenn Sie angeben möchten, dass RDS das Hauptbenutzerpasswort in Secrets Manager verwaltet, legen Sie den `ManageMasterUserPassword`-Parameter in einer der folgenden RDS-API-Operationen auf `true` fest:

- CreateDBInstance
- Modifizieren SieDBInstance
- DBInstanceAus S3 wiederherstellen
- Wiederherstellen DBInstance FromSnapshot (nur RDS für Oracle)
- Wiederherstellen DBInstance ToPointInTime (nur RDS für Oracle)

Wenn Sie den `ManageMasterUserPassword`-Parameter in einer dieser Operationen auf `true` festlegen, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Sie können auch einen kundenverwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Standardschlüssel verwenden, der von Secrets Manager bereitgestellt wird. Verwenden Sie den `MasterUserSecretKmsKeyId`-Parameter, um einen kundenverwalteten Schlüssel anzugeben. Die AWS KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Geben Sie den Schlüssel-ARN oder Alias-ARN an, um einen KMS-Schlüssel in einem anderen AWS-Konto zu verwenden. Nachdem RDS die Datenbankanmeldeinformationen für eine DB-Instance verwaltet hat, können Sie den KMS-Schlüssel, der zum Verschlüsseln des Secrets verwendet wird, nicht mehr ändern.

Verwaltung des Master-Benutzerpassworts für eine RDS for Oracle-Mandantendatenbank mit Secrets Manager

Sie können die RDS-Verwaltung des Hauptbenutzerpassworts in Secrets Manager konfigurieren, wenn Sie die folgenden Aktionen ausführen:

- Hinzufügen einer RDS-für-Oracle-Tenant-Datenbank zu Ihrer CDB-Instance
- Ändern einer Tenant-Datenbank von RDS für Oracle

Sie können die RDS-Konsole AWS CLI, die oder die RDS-API verwenden, um die oben genannten Aktionen auszuführen.

Konsole

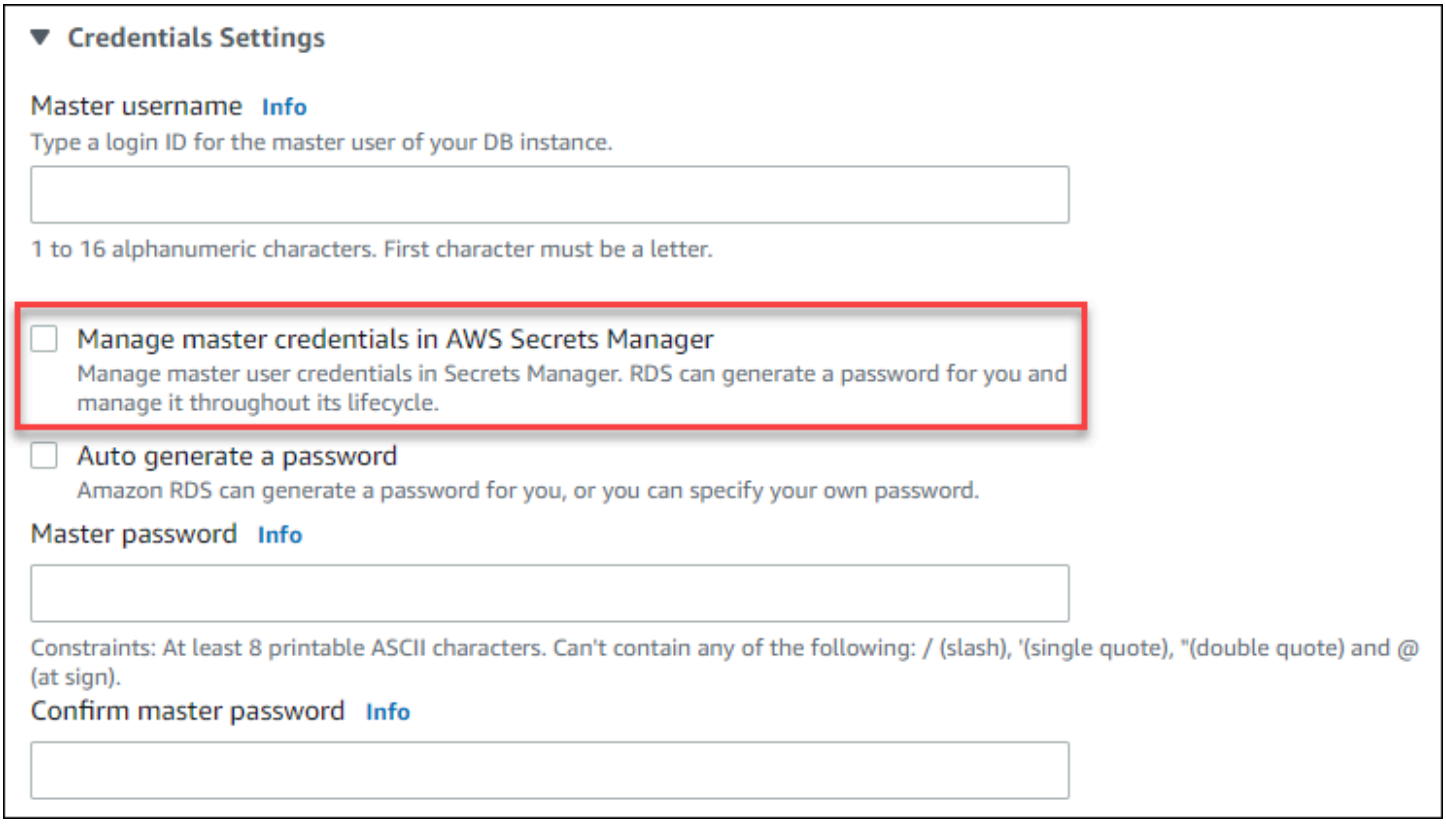
Folgen Sie den Anweisungen zum Erstellen oder Ändern einer RDS for Oracle-Mandantendatenbank mit der RDS-Konsole:

- Hinzufügen einer RDS-für-Oracle-Tenant-Datenbank zu Ihrer CDB-Instance
- Ändern einer Tenant-Datenbank von RDS für Oracle

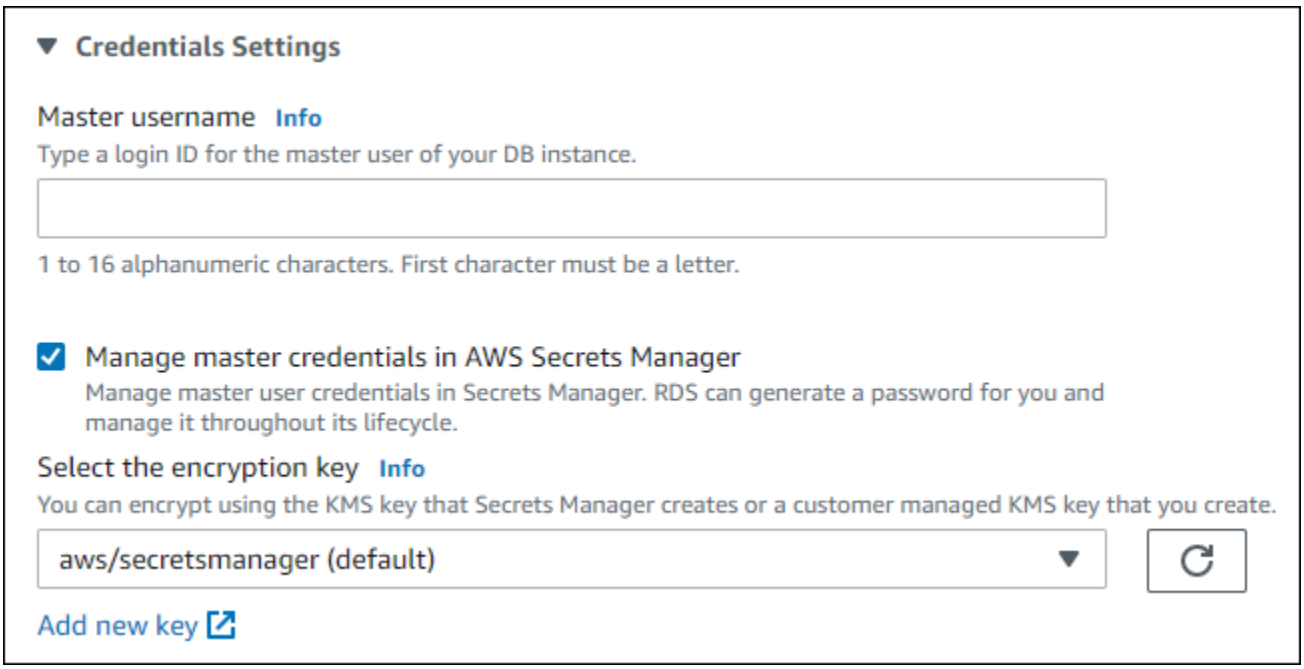
Wenn Sie die RDS-Konsole verwenden, um einen der oben genannten Vorgänge auszuführen, können Sie angeben, dass RDS das Master-Passwort in Secrets Manager verwaltet.
Wenn Sie eine Mandantendatenbank erstellen, wählen Sie AWS Secrets Manager in den Anmeldeinformationseinstellungen die Option Masteranmeldedaten verwalten aus. Wenn Sie eine

Tenant-Datenbank ändern, wählen Sie AWS Secrets Manager in den Einstellungen die Option Master-Anmeldeinformationen verwalten aus.

Die folgende Abbildung zeigt ein Beispiel für die AWS Secrets Manager Einstellung Masteranmeldedaten verwalten, wenn Sie eine Mandantendatenbank erstellen.



Wenn Sie diese Option auswählen, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.



Sie können wählen, ob Sie das Secret mit einem von Secrets Manager bereitgestellten KMS-Schlüssel oder mit einem von Ihnen erstellten kundenverwalteten Schlüssel verschlüsseln möchten. Nachdem RDS die Datenbankanmeldedaten für eine Tenant-Datenbank verwaltet hat, können Sie den KMS-Schlüssel, der zur Verschlüsselung des Geheimnisses verwendet wird, nicht mehr ändern.

Sie können andere Einstellungen auswählen, die Ihren Anforderungen entsprechen. Weitere Informationen zu den verfügbaren Einstellungen beim Erstellen einer Tenant-Datenbank finden Sie unterEinstellungen für DB-Instances. Weitere Informationen zu den verfügbaren Einstellungen beim Ändern einer Tenant-Datenbank finden Sie unterEinstellungen für DB-Instances.

AWS CLI

Um das Hauptbenutzerpasswort mit RDS in Secrets Manager zu verwalten, geben Sie die `--manage-master-user-password`-Option in einem der folgenden AWS CLI -Befehle an:

- create-tenant-database
- modify-tenant-database

Wenn Sie die `--manage-master-user-password` Option in den vorherigen Befehlen angeben, generiert RDS das Masterbenutzerkennwort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Sie können auch einen kundenverwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Standardschlüssel verwenden, der von Secrets Manager bereitgestellt wird. Verwenden Sie die `--master-user-secret-kms-key-id`-Option, um einen kundenverwalteten Schlüssel anzugeben. Die AWS KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Um einen KMS-Schlüssel in einem anderen zu verwenden AWS-Konto, geben Sie den Schlüssel-ARN oder den Alias-ARN an. Nachdem RDS die Datenbankanmeldeinformationen für eine Tenant-Datenbank verwaltet hat, können Sie den KMS-Schlüssel, der zur Verschlüsselung des Geheimnisses verwendet wird, nicht mehr ändern.

Sie können andere Einstellungen auswählen, die Ihren Anforderungen entsprechen. Weitere Informationen zu den verfügbaren Einstellungen beim Erstellen einer Tenant-Datenbank finden Sie unter create-tenant-database. Weitere Informationen zu den verfügbaren Einstellungen beim Ändern einer Tenant-Datenbank finden Sie unter modify-tenant-database.

Im folgenden Beispiel wird eine Mandantendatenbank RDS für Oracle erstellt und angegeben, dass RDS das Masterbenutzerkennwort in Secrets Manager verwaltet. Das Secret wird mit dem KMS-Schlüssel verschlüsselt, der von Secrets Manager bereitgestellt wird.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name mypdb2 \
    --master-username mypdb2-admin \
    --character-set-name UTF-16 \
    --manage-master-user-password
```

Windows:

```
aws rds create-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name mypdb2 ^
    --master-username mypdb2-admin ^
    --character-set-name UTF-16 ^
    --manage-master-user-password
```

RDS-API

Wenn Sie angeben möchten, dass RDS das Hauptbenutzerpasswort in Secrets Manager verwaltet, legen Sie den `ManageMasterUserPassword`-Parameter in einer der folgenden RDS-API-Operationen auf `true` fest:

- CreateTenantDatabase
- ModifyTenantDatabase

Wenn Sie den `ManageMasterUserPassword`-Parameter in einer dieser Operationen auf `true` festlegen, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Sie können auch einen kundenverwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Standardschlüssel verwenden, der von Secrets Manager bereitgestellt wird. Verwenden Sie den `MasterUserSecretKmsKeyId`-Parameter, um einen kundenverwalteten Schlüssel anzugeben. Die AWS -KMS-Schlüsselkennung ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Alias-Name für den KMS-Schlüssel. Um einen KMS-Schlüssel in einem anderen zu verwenden AWS-Konto, geben Sie den Schlüssel-ARN oder den Alias-ARN an. Nachdem RDS die Datenbankanmeldeinformationen für eine Tenant-Datenbank verwaltet hat, können Sie den KMS-Schlüssel, der zur Verschlüsselung des Geheimnisses verwendet wird, nicht mehr ändern.

Verwaltung des Hauptbenutzerpassworts für einen Multi-AZ-DB-Cluster mit Secrets Manager

Sie können die RDS-Verwaltung des Hauptbenutzerpassworts in Secrets Manager konfigurieren, wenn Sie die folgenden Aktionen ausführen:

- Einen Multi-AZ-DB-Cluster für Amazon RDS erstellen
- Ändern eines Multi-AZ-DB-Clusters für Amazon RDS

Sie können die RDS-Konsole AWS CLI, die oder die RDS-API verwenden, um diese Aktionen auszuführen.

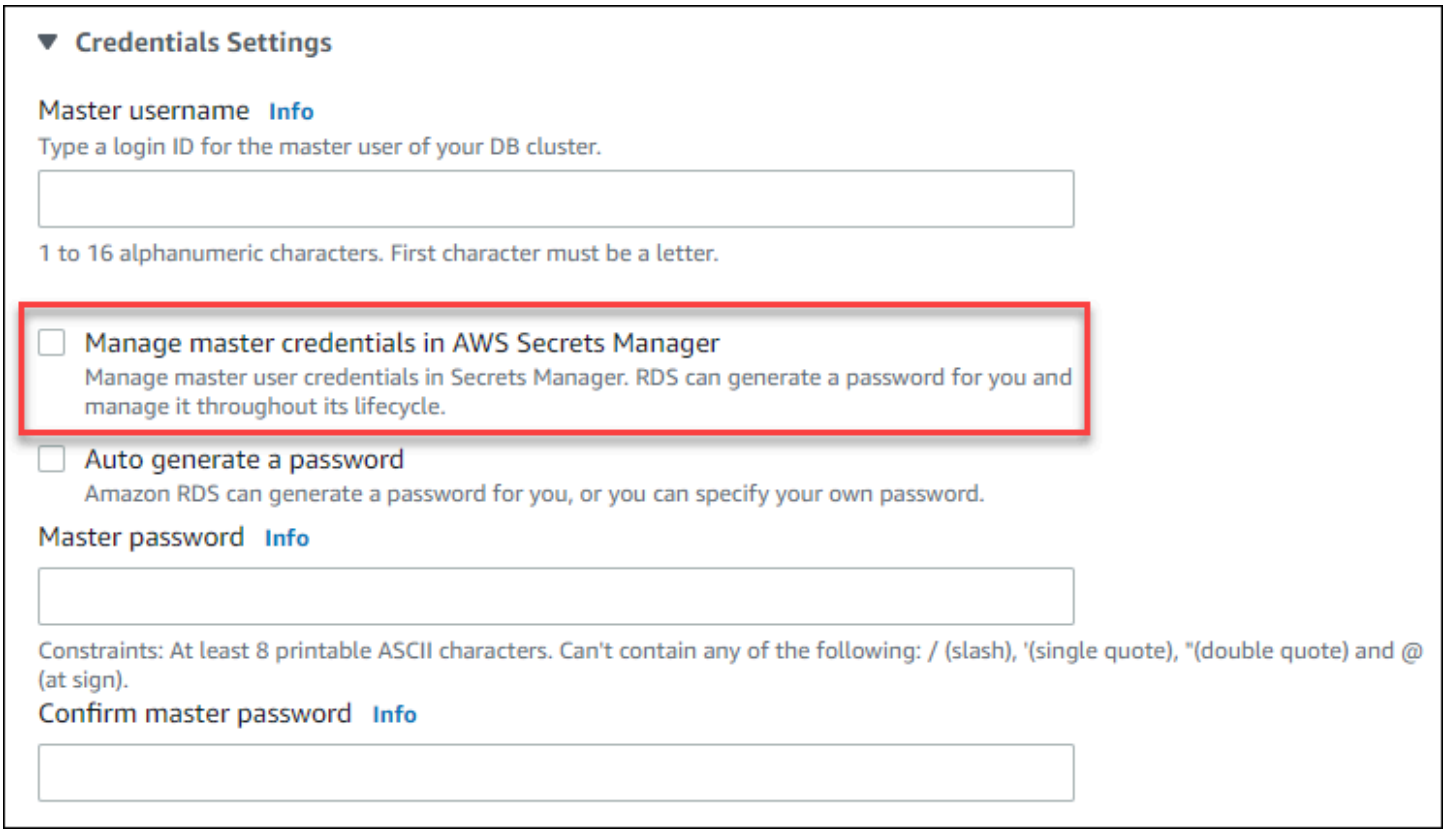
Konsole

Folgen Sie den Anweisungen zum Erstellen oder Ändern eines Multi-AZ DB-Clusters mit der RDS-Konsole:

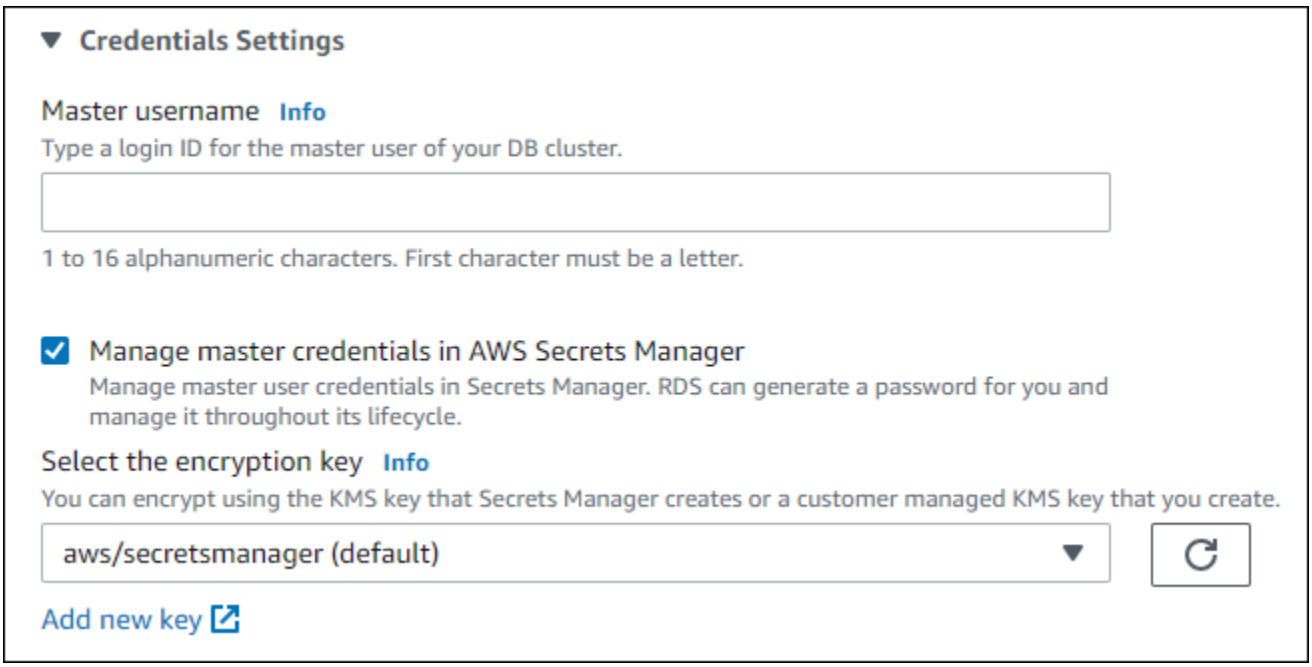
- Erstellen eines DB-Clusters
- Ändern eines Multi-AZ-DB-Clusters für Amazon RDS

Wenn Sie die RDS-Konsole verwenden, um eine dieser Operationen auszuführen, können Sie angeben, dass das Hauptbenutzerpasswort von RDS in Secrets Manager verwaltet wird. Wählen Sie dazu beim Erstellen eines DB-Clusters Hauptanmeldeinformationen in AWS Secrets Manager verwalten unter Anmeldeinformationseinstellungen aus. Wenn Sie einen DB-Cluster ändern, wählen Sie Hauptanmeldeinformationen in AWS Secrets Manager verwalten unter Einstellungen aus.

Die folgende Abbildung zeigt ein Beispiel für die Einstellung Hauptanmeldeinformationen in AWS Secrets Manager verwalten beim Erstellen eines DB-Clusters.



Wenn Sie diese Option auswählen, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.



Sie können wählen, ob Sie das Secret mit einem von Secrets Manager bereitgestellten KMS-Schlüssel oder mit einem von Ihnen erstellten kundenverwalteten Schlüssel verschlüsseln möchten. Nachdem RDS die Datenbankanmeldeinformationen für einen DB-Cluster verwaltet hat, können Sie den KMS-Schlüssel, der zum Verschlüsseln des Secrets verwendet wird, nicht mehr ändern.

Sie können andere Einstellungen auswählen, die Ihren Anforderungen entsprechen.

Weitere Informationen zu den verfügbaren Einstellungen beim Erstellen eines Multi-AZ-DB-Clusters finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern. Weitere Informationen zu den verfügbaren Einstellungen beim Ändern eines Multi-AZ-DB-Clusters finden Sie unter Einstellungen zum Ändern von Multi-AZ-DB-Clustern.

AWS CLI

Wenn Sie angeben möchten, dass RDS das Hauptbenutzerpasswort in Secrets Manager verwalten soll, geben Sie die `--manage-master-user-password`-Option in einem der folgenden Befehle an:

- create-db-cluster
- modify-db-cluster

Wenn Sie die `--manage-master-user-password`-Option angeben, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Sie können auch einen kundenverwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Standardschlüssel verwenden, der von Secrets Manager bereitgestellt wird. Verwenden Sie die `--master-user-secret-kms-key-id`-Option, um einen kundenverwalteten Schlüssel anzugeben. Die AWS KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Um einen KMS-Schlüssel in einem anderen zu verwenden AWS-Konto, geben Sie den Schlüssel-ARN oder den Alias-ARN an. Nachdem RDS die Datenbankanmeldeinformationen für einen DB-Cluster verwaltet hat, können Sie den KMS-Schlüssel, der zum Verschlüsseln des Secrets verwendet wird, nicht mehr ändern.

Sie können andere Einstellungen auswählen, die Ihren Anforderungen entsprechen.

Weitere Informationen zu den verfügbaren Einstellungen beim Erstellen eines Multi-AZ-DB-Clusters finden Sie unter Einstellungen zum Erstellen von Multi-AZ-DB-Clustern. Weitere Informationen zu den verfügbaren Einstellungen beim Ändern eines Multi-AZ-DB-Clusters finden Sie unter Einstellungen zum Ändern von Multi-AZ-DB-Clustern.

In diesem Beispiel wird ein Multi-AZ-DB-Cluster erstellt und angegeben, dass RDS das Passwort in Secrets Manager verwaltet. Das Secret wird mit dem KMS-Schlüssel verschlüsselt, der von Secrets Manager bereitgestellt wird.

Example

Für LinuxmacOS, oderUnix:

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.39  \
   --backup-retention-period 1  \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.r6gd.xlarge \
   --master-username testUser \
   --manage-master-user-password
```

Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
```

```
    --engine mysql ^
    --engine-version 8.0.39 ^
    --backup-retention-period 1 ^
    --allocated-storage 4000 ^
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.r6gd.xlarge ^
    --master-username testUser ^
    --manage-master-user-password
```

RDS-API

Wenn Sie angeben möchten, dass RDS das Hauptbenutzerpasswort in Secrets Manager verwaltet, legen Sie den `ManageMasterUserPassword`-Parameter in einer der folgenden Operationen auf `true` fest:

- CreateDBCluster
- Modifizieren SieDBCluster

Wenn Sie den `ManageMasterUserPassword`-Parameter in einer dieser Operationen auf `true` festlegen, generiert RDS das Hauptbenutzerpasswort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Sie können auch einen kundenverwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Standardschlüssel verwenden, der von Secrets Manager bereitgestellt wird. Verwenden Sie den `MasterUserSecretKmsKeyId`-Parameter, um einen kundenverwalteten Schlüssel anzugeben. Die AWS KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Geben Sie den Schlüssel-ARN oder Alias-ARN an, um einen KMS-Schlüssel in einem anderen AWS-Konto zu verwenden. Nachdem RDS die Datenbankanmeldeinformationen für einen DB-Cluster verwaltet hat, können Sie den KMS-Schlüssel, der zum Verschlüsseln des Secrets verwendet wird, nicht mehr ändern.

Rotieren des Hauptbenutzerpasswort-Secrets für eine DB-Instance

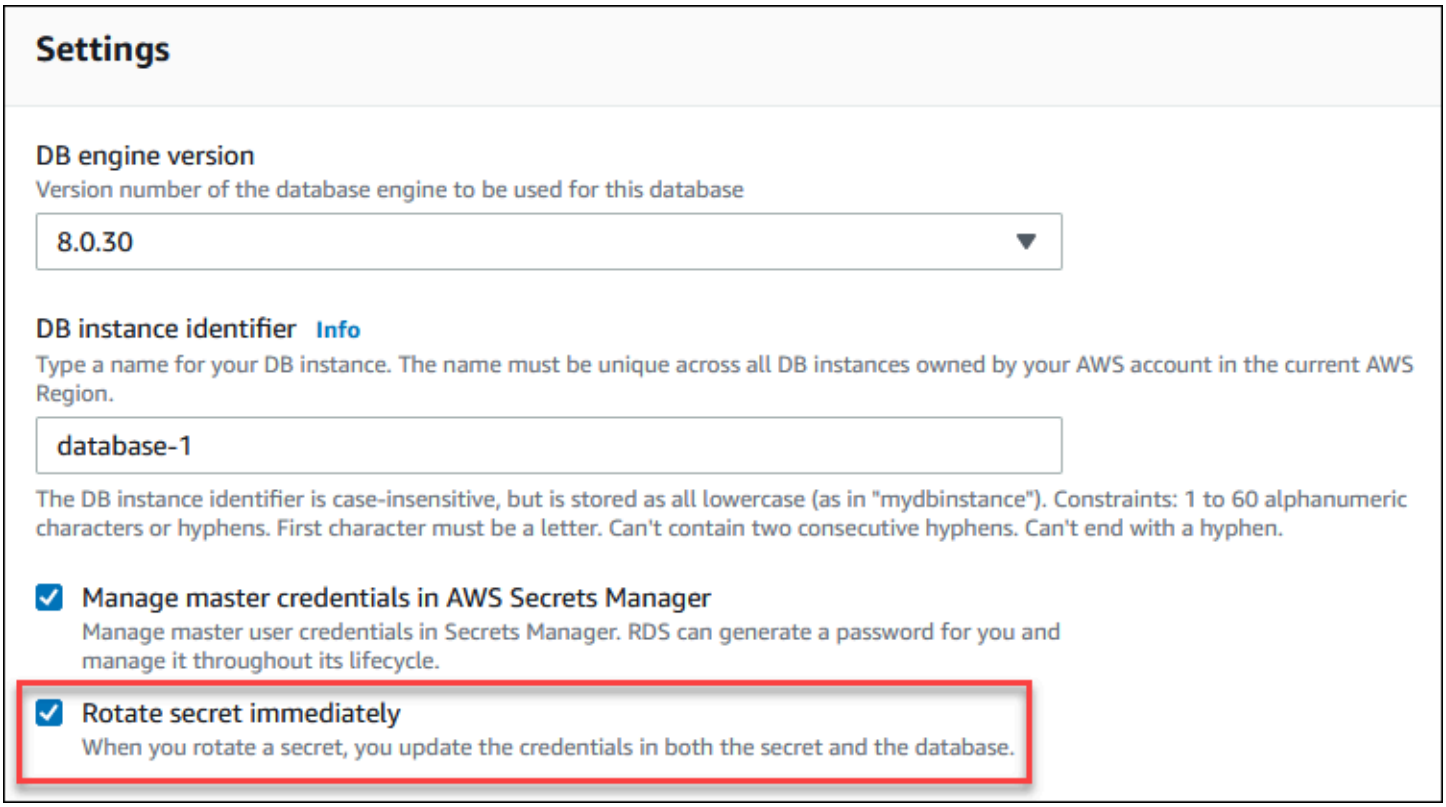
Wenn RDS ein Hauptbenutzerpasswort-Secret rotiert, generiert Secrets Manager eine neue geheime Version für das vorhandene Secret. Die neue Secret-Version enthält das neue Hauptbenutzerpasswort. Amazon RDS ändert das Hauptbenutzerpasswort für die DB-Instance, sodass es dem Passwort für die neue Secret-Version entspricht.

Sie können ein Secret sofort rotieren, anstatt auf eine geplante Rotation zu warten. Ändern Sie die DB-Instance, um ein Hauptbenutzerpasswort-Secret in Secrets Manager zu rotieren. Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Sie können ein geheimes Masterbenutzerpasswort sofort mit der RDS-Konsole AWS CLI, der oder der RDS-API rotieren. Das neue Passwort ist immer 28 Zeichen lang und enthält mindestens einen Groß- und Kleinbuchstaben, eine Zahl und ein Satzzeichen.

Konsole

Wenn Sie das Hauptbenutzerpasswort-Secret mithilfe der RDS-Konsole rotieren möchten, ändern Sie die DB-Instance und wählen Sie die Option Rotate secret immediately (Sofortige Secret-Drehung) unter Settings (Einstellungen) aus.



Folgen Sie den Anweisungen zum Ändern einer DB-Instance mit der RDS-Konsole in Ändern einer Amazon RDS DB-Instance. Sie müssen auf der Bestätigungsseite die Option Apply immediately (Sofort anwenden) auswählen.

AWS CLI

Verwenden Sie den modify-db-instanceBefehl und geben Sie die Option an AWS CLI, um das geheime Masterbenutzerpasswort mithilfe von zu ändern. `--rotate-master-user-password` Sie müssen die `--apply-immediately`-Option angeben, wenn Sie das Hauptpasswort rotieren.

In diesem Beispiel wird ein Hauptbenutzerpasswort-Secret rotiert.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --rotate-master-user-password \
    --apply-immediately
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --rotate-master-user-password ^
    --apply-immediately
```

RDS-API

Sie können das geheime Masterbenutzerkennwort wechseln, indem Sie den DBInstance Vorgang Ändern verwenden und den `RotateMasterUserPassword` Parameter auf setzen`true`. Sie müssen den `ApplyImmediately`-Parameter auf `true` festlegen, wenn Sie das Hauptpasswort rotieren.

Rotieren des Hauptbenutzerpasswort-Secrets für einen Multi-AZ-DB-Cluster

Wenn RDS ein Hauptbenutzerpasswort-Secret rotiert, generiert Secrets Manager eine neue Secret-Version für das vorhandene Secret. Die neue Secret-Version enthält das neue Hauptbenutzerpasswort. Amazon RDS ändert das Hauptbenutzerpasswort für den Multi-AZ-DB-Cluster so, dass es mit dem Passwort für die neue Secret-Version übereinstimmt.

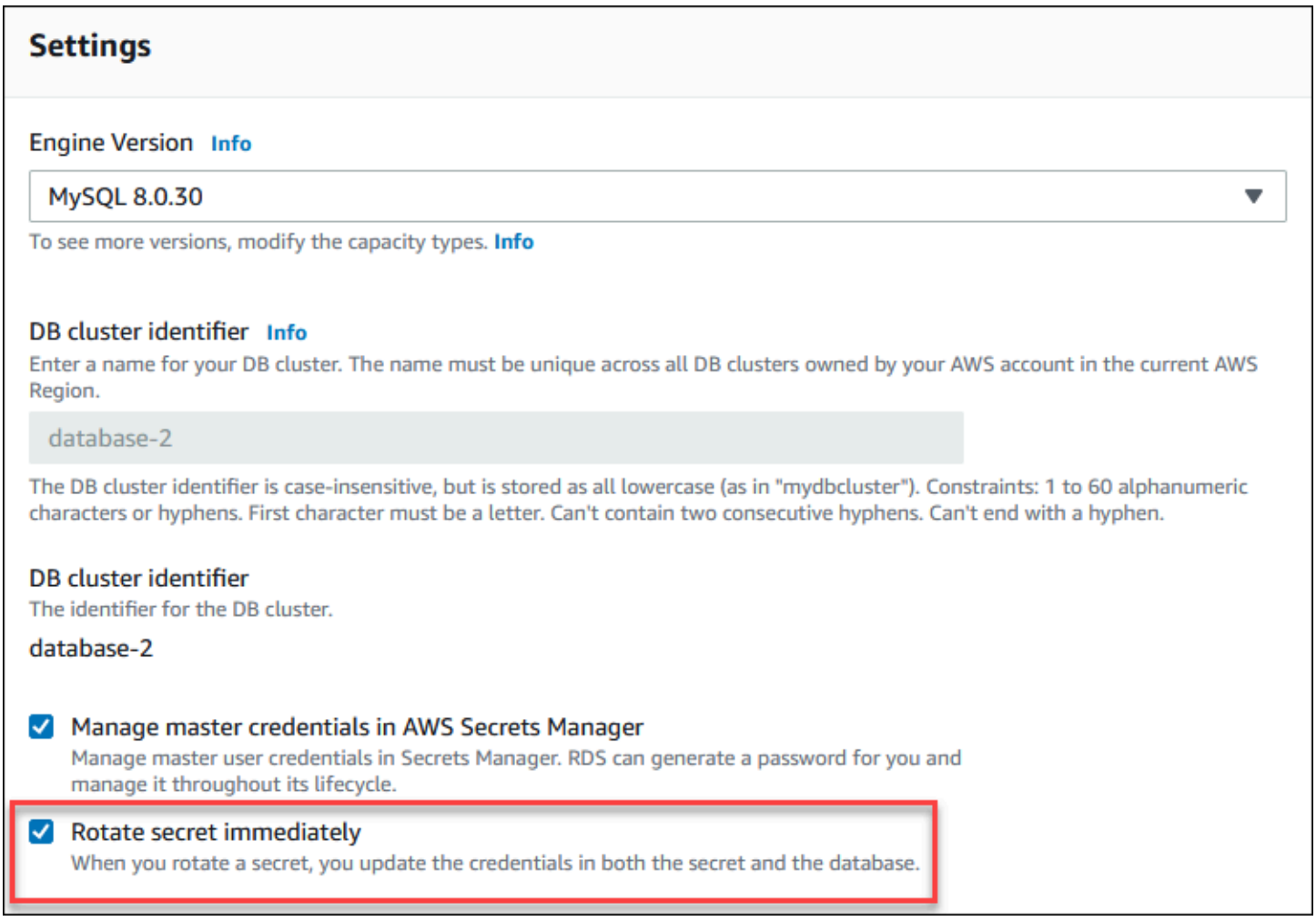
Sie können ein Secret sofort rotieren, anstatt auf eine geplante Rotation zu warten. Wenn Sie das Hauptbenutzerpasswort-Secret in Secrets Manager rotieren möchten, ändern Sie den Multi-AZ DB-

Cluster. Informationen über das Ändern eines Multi-AZ-DB-Clusters finden Sie unter Ändern eines Multi-AZ-DB-Clusters für Amazon RDS.

Sie können das geheime Masterbenutzer-Passwort sofort mit der RDS-Konsole AWS CLI, der oder der RDS-API rotieren. Das neue Passwort ist immer 28 Zeichen lang und enthält mindestens einen Groß- und Kleinbuchstaben, eine Zahl und ein Satzzeichen.

Konsole

Wenn Sie das Hauptbenutzerpasswort-Secret mithilfe der RDS-Konsole rotieren möchten, ändern Sie den Multi-AZ DB-Cluster und wählen Sie die Option Rotate secret immediately (Sofortige Secret-Drehung) unter Settings (Einstellungen) aus.



Folgen Sie den Anweisungen zum Ändern eines Multi-AZ DB-Clusters mit der RDS-Konsole in Ändern eines Multi-AZ-DB-Clusters für Amazon RDS. Sie müssen auf der Bestätigungsseite die Option Apply immediately (Sofort anwenden) auswählen.

AWS CLI

Verwenden Sie den modify-db-clusterBefehl und geben Sie die Option an AWS CLI, um das geheime Masterbenutzerpasswort mithilfe von zu ändern. `--rotate-master-user-password` Sie müssen die `--apply-immediately`-Option angeben, wenn Sie das Hauptpasswort rotieren.

In diesem Beispiel wird ein Hauptbenutzerpasswort-Secret rotiert.

Example

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --rotate-master-user-password \
    --apply-immediately
```

Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --rotate-master-user-password ^
    --apply-immediately
```

RDS-API

Sie können das geheime Masterbenutzerkennwort wechseln, indem Sie den DBCluster Vorgang Ändern verwenden und den `RotateMasterUserPassword` Parameter auf setzen`true`. Sie müssen den `ApplyImmediately`-Parameter auf `true` festlegen, wenn Sie das Hauptpasswort rotieren.

Anzeigen der Details zu einem Secret für eine DB-Instance

Sie können Ihre Secrets mit der Konsole (https://console.aws.amazon.com/secretsmanager/) oder dem AWS CLI (get-secret-valueSecrets Manager Manager-Befehl) abrufen.

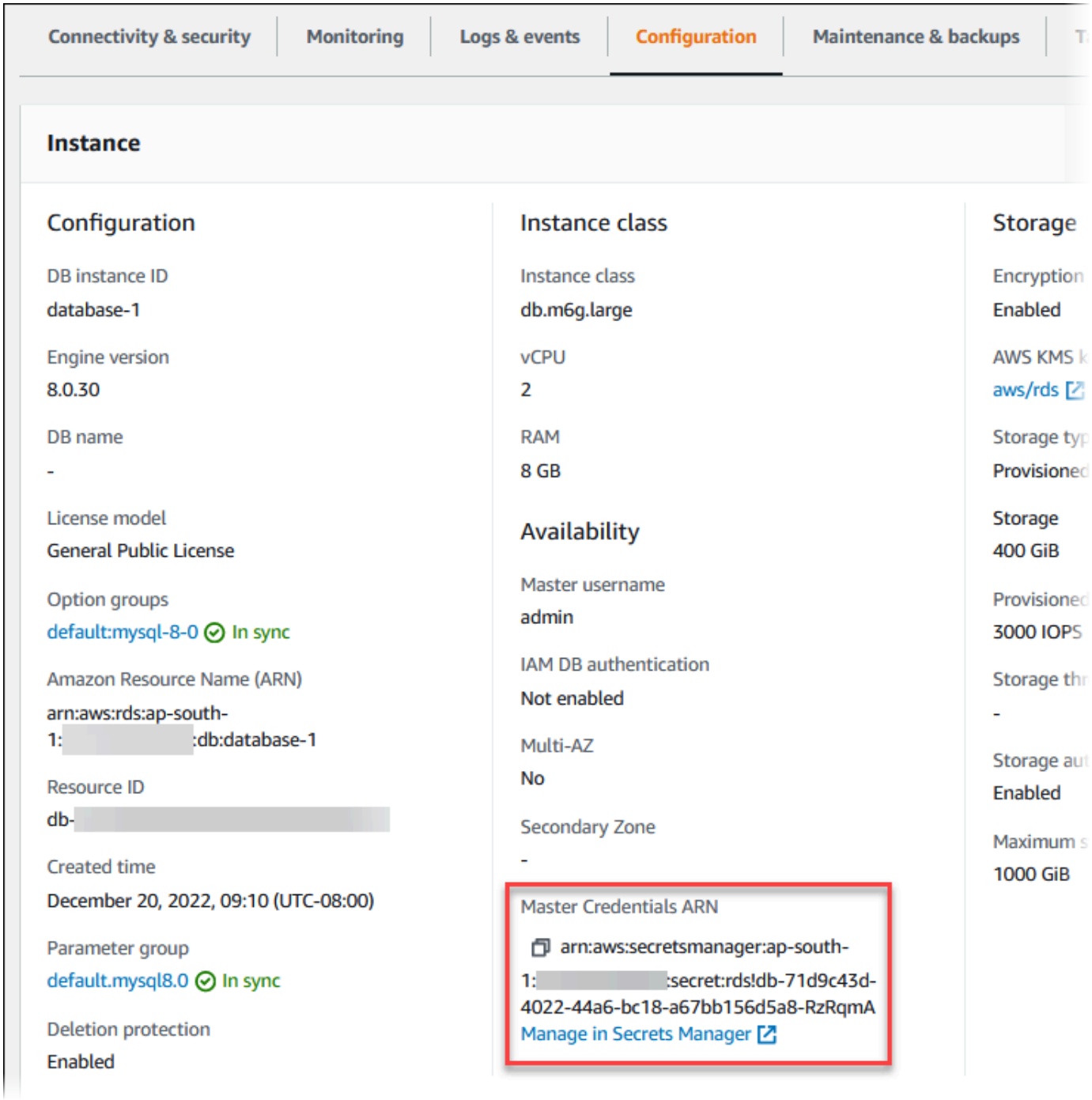
Sie finden den Amazon-Ressourcennamen (ARN) eines von RDS verwalteten Secrets im Secrets Manager mit der RDS-Konsole AWS CLI, der oder der RDS-API.

Konsole

So zeigen Sie die Details zu einem von RDS verwalteten Secret in Secrets Manager an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der entsprechenden DB-Instance aus, um deren Details anzuzeigen.
4. Wählen Sie die Registerkarte Konfiguration aus.

   Unter Master Credentials ARN (ARN der Hauptanmeldeinformationen) können Sie den geheimen ARN einsehen.



Sie können dem Link Manage in Secrets Manager (In Secrets Manager verwalten) folgen, um das Secret in der Secrets-Manager-Konsole anzuzeigen und zu verwalten.

AWS CLI

Sie können den describe-db-instancesRDS-CLI-Befehl verwenden, um die folgenden Informationen zu einem von RDS verwalteten Geheimnis in Secrets Manager zu finden:

- `SecretArn` – Der ARN des Secrets
- `SecretStatus` – Der Status des Secrets

  Mögliche Werte für den Status sind u. a. folgende:

  - `creating` – Das Secret wird erstellt.
  - `active` – Das Secret ist für den normalen Gebrauch und die Rotation verfügbar.
  - `rotating` – Das Secret wird rotiert.
  - `impaired` – Das Secret kann für den Zugriff auf Datenbankanmeldeinformationen verwendet werden, es kann jedoch nicht rotiert werden. Ein Secret kann diesen Status haben, wenn beispielsweise die Berechtigungen geändert werden, sodass RDS nicht mehr auf das Secret oder den KMS-Schlüssel für das Secret zugreifen kann.

    Wenn ein Secret diesen Status hat, können Sie die Bedingung korrigieren, die den Status verursacht hat. Wenn Sie die Bedingung korrigieren, die den Status verursacht hat, behält der Status bis zur nächsten Rotation den Wert `impaired`. Alternativ können Sie die DB-Instance ändern, um die automatische Verwaltung von Datenbankanmeldeinformationen zu deaktivieren, und dann die DB-Instance erneut ändern, um die automatische Verwaltung von Datenbankanmeldeinformationen zu aktivieren. Verwenden Sie die `--manage-master-user-password` Option im modify-db-instanceBefehl, um die DB-Instance zu ändern.
- `KmsKeyId` – Der ARN des KMS-Schlüssels, der verwendet wird, um das Secret zu verschlüsseln

Geben Sie die `--db-instance-identifier`-Option an, um die Ausgabe für eine bestimmte DB-Instance anzuzeigen. Dieses Beispiel zeigt die Ausgabe für ein Secret, das von einer DB-Instance verwendet wird.

Example

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Das folgende Beispiel zeigt die Ausgabe für ein Secret:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
```

```
        }
```

Wenn Sie über den geheimen ARN verfügen, können Sie mit dem get-secret-valueSecrets Manager-CLI-Befehl Details zum Secret Manager anzeigen.

Dieses Beispiel zeigt die Details für das Secret in der vorherigen Beispielausgabe.

Example

Für LinuxmacOS, oderUnix:

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

RDS-API

Sie können den ARN, den Status und den KMS-Schlüssel für ein von RDS verwaltetes Geheimnis in Secrets Manager anzeigen, indem Sie den DBInstances Vorgang Describe verwenden und den `DBInstanceIdentifier` Parameter auf eine DB-Instance-ID setzen. Details zum Secret sind in der Ausgabe enthalten

Wenn Sie über den geheimen ARN verfügen, können Sie mithilfe des GetSecretValueSecrets Manager-Vorgangs Details zu dem Secret anzeigen.

Anzeigen der Details zu einem Secret für einen Multi-AZ-DB-Cluster

Sie können Ihre Secrets mit der Konsole (https://console.aws.amazon.com/secretsmanager/) oder dem AWS CLI (get-secret-valueSecrets Manager Manager-Befehl) abrufen.

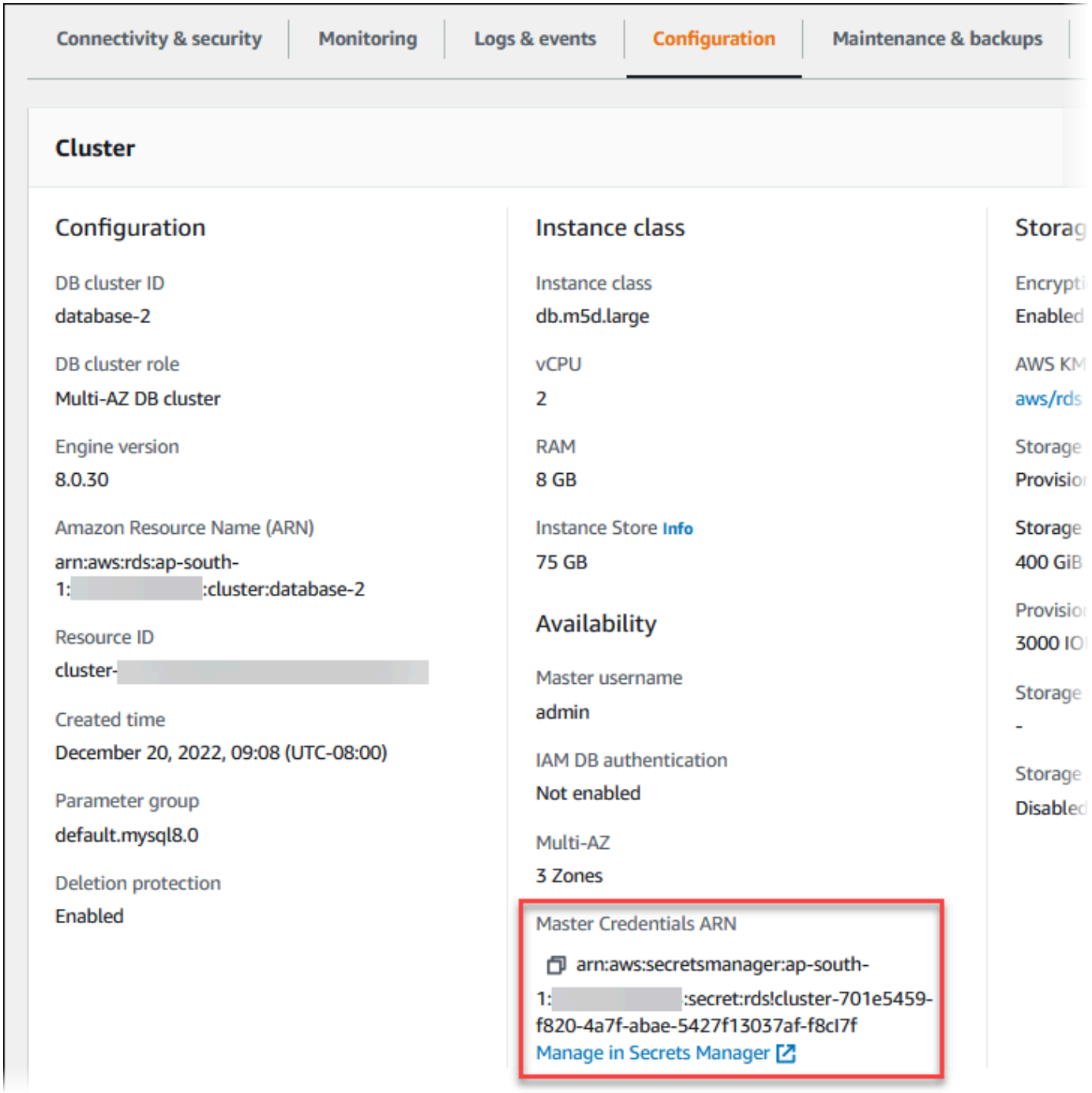
Sie finden den Amazon-Ressourcennamen (ARN) eines von RDS verwalteten Secrets im Secrets Manager mit der RDS-Konsole AWS CLI, der oder der RDS-API.

Konsole

So zeigen Sie die Details zu einem von RDS verwalteten Secret in Secrets Manager an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen des Multi-AZ-DB-Clusters aus, um dessen Details anzuzeigen.
4. Wählen Sie die Registerkarte Konfiguration aus.

   Unter Master Credentials ARN (ARN der Hauptanmeldeinformationen) können Sie den geheimen ARN einsehen.



Sie können dem Link Manage in Secrets Manager (In Secrets Manager verwalten) folgen, um das Secret in der Secrets-Manager-Konsole anzuzeigen und zu verwalten.

AWS CLI

Sie können den AWS CLI [describe-db-clusters](#)Befehl RDS verwenden, um die folgenden Informationen zu einem Secret zu finden, das von RDS in Secrets Manager verwaltet wird:

- `SecretArn` – Der ARN des Secrets
- `SecretStatus` – Der Status des Secrets

  Mögliche Werte für den Status sind u. a. folgende:

  - `creating` – Das Secret wird erstellt.
  - `active` – Das Secret ist für den normalen Gebrauch und die Rotation verfügbar.
  - `rotating` – Das Secret wird rotiert.
  - `impaired` – Das Secret kann für den Zugriff auf Datenbankanmeldeinformationen verwendet werden, es kann jedoch nicht rotiert werden. Ein Secret kann diesen Status haben, wenn beispielsweise die Berechtigungen geändert werden, sodass RDS nicht mehr auf das Secret oder den KMS-Schlüssel für das Secret zugreifen kann.

    Wenn ein Secret diesen Status hat, können Sie die Bedingung korrigieren, die den Status verursacht hat. Wenn Sie die Bedingung korrigieren, die den Status verursacht hat, behält der Status bis zur nächsten Rotation den Wert `impaired`. Alternativ können Sie den DB-Cluster ändern, um die automatische Verwaltung von Datenbankanmeldeinformationen zu deaktivieren, und dann den DB-Cluster erneut ändern, um die automatische Verwaltung von Datenbankanmeldeinformationen zu aktivieren. Verwenden Sie die `--manage-master-user-password` Option im modify-db-clusterBefehl, um den DB-Cluster zu ändern.
- `KmsKeyId` – Der ARN des KMS-Schlüssels, der verwendet wird, um das Secret zu verschlüsseln

Geben Sie die `--db-cluster-identifier`-Option an, um die Ausgabe für einen bestimmten DB-Cluster anzuzeigen. Dieses Beispiel zeigt die Ausgabe für ein Secret, das von einem DB-Cluster verwendet wird.

Example

```
aws rds describe-db-clusters --db-cluster-identifier mydbcluster
```

Das folgende Beispiel zeigt die Ausgabe für ein Secret:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
```

```
        }
```

Wenn Sie über den geheimen ARN verfügen, können Sie mit dem get-secret-valueSecrets Manager-CLI-Befehl Details zum Secret Manager anzeigen.

Dieses Beispiel zeigt die Details für das Secret in der vorherigen Beispielausgabe.

Example

Für LinuxmacOS, oderUnix:

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

RDS-API

Sie können den ARN, den Status und den KMS-Schlüssel für ein von RDS verwaltetes Geheimnis im Secrets Manager anzeigen, indem Sie den Vorgang DBClustersDescribe RDS verwenden und den `DBClusterIdentifier` Parameter auf eine DB-Cluster-ID setzen. Details zum Secret sind in der Ausgabe enthalten

Wenn Sie über den geheimen ARN verfügen, können Sie mithilfe des GetSecretValueSecrets Manager-Vorgangs Details zu dem Secret anzeigen.

Details zu einem geheimen Schlüssel für eine Tenant-Datenbank anzeigen

Sie können Ihre Secrets mit der Konsole (https://console.aws.amazon.com/secretsmanager/) oder dem AWS CLI (get-secret-valueSecrets Manager Manager-Befehl) abrufen.

Den Amazon-Ressourcennamen (ARN) eines von Amazon RDS verwalteten Geheimnisses finden Sie in AWS Secrets Manager der Amazon RDS-Konsole AWS CLI, der oder der Amazon RDS-API.

Konsole

Um die Details zu einem von Amazon RDS verwalteten Geheimnis AWS Secrets Manager für eine Tenant-Datenbank anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance, die die Tenant-Datenbank enthält, um deren Details anzuzeigen.
4. Wählen Sie die Registerkarte Konfiguration aus.

   Suchen Sie im Abschnitt Tenant-Datenbanken die Tenant-Datenbank und sehen Sie sich ihren ARN für Master-Anmeldeinformationen an.

   Sie können dem Link Manage in Secrets Manager (In Secrets Manager verwalten) folgen, um das Secret in der Secrets-Manager-Konsole anzuzeigen und zu verwalten.

AWS CLI

Sie können den describe-tenant-databasesAmazon AWS CLI RDS-Befehl verwenden, um die folgenden Informationen über ein von Amazon RDS verwaltetes Geheimnis in AWS Secrets Manager einer Mandantendatenbank zu finden:

- `SecretArn` – Der ARN des Secrets
- `SecretStatus` – Der Status des Secrets

  Mögliche Werte für den Status sind u. a. folgende:

  - `creating` – Das Secret wird erstellt.
  - `active` – Das Secret ist für den normalen Gebrauch und die Rotation verfügbar.
  - `rotating` – Das Secret wird rotiert.
  - `impaired` – Das Secret kann für den Zugriff auf Datenbankanmeldeinformationen verwendet werden, es kann jedoch nicht rotiert werden. Ein Geheimnis kann diesen Status haben, wenn beispielsweise die Berechtigungen geändert werden, sodass Amazon RDS nicht mehr auf das Geheimnis oder den KMS-Schlüssel für das Geheimnis zugreifen kann.

Wenn ein Secret diesen Status hat, können Sie die Bedingung korrigieren, die den Status verursacht hat. Wenn Sie die Bedingung korrigieren, die den Status verursacht hat, behält der Status bis zur nächsten Rotation den Wert `impaired`. Alternativ können Sie die Tenant-Datenbank ändern, um die automatische Verwaltung von Datenbankanmeldedaten zu deaktivieren, und dann die Tenant-Datenbank erneut ändern, um die automatische Verwaltung von Datenbankanmeldedaten zu aktivieren. Verwenden Sie die `--manage-master-user-password` Option im modify-tenant-databaseBefehl, um die Tenant-Datenbank zu ändern.

- `KmsKeyId` – Der ARN des KMS-Schlüssels, der verwendet wird, um das Secret zu verschlüsseln

Geben Sie die `--db-instance-identifier` Option an, um die Ausgabe für Tenant-Datenbanken in einer bestimmten DB-Instance anzuzeigen. Sie können auch die `--tenant-db-name` Option angeben, um die Ausgabe für eine bestimmte Tenant-Datenbank anzuzeigen. Dieses Beispiel zeigt die Ausgabe für ein Geheimnis, das von einer Tenant-Datenbank verwendet wird.

Example

```
aws rds describe-tenant-databases \
    --db-instance-identifier database-3 \
    --query "TenantDatabases[0].MasterUserSecret"
```

Das folgende Beispiel zeigt die Ausgabe für ein Secret:

```
{
    "SecretArn": "arn:aws:secretsmanager:us-east-2:123456789012:secret:rds!db-ABC123",
    "SecretStatus": "active",
    "KmsKeyId": "arn:aws:kms:us-east-2:123456789012:key/aa11bb22-####-####-####-
fedcba123456"
}
```

Wenn Sie den geheimen ARN haben, können Sie mit dem get-secret-valueSecrets AWS CLI Manager-Befehl Details zum Secret anzeigen.

Dieses Beispiel zeigt die Details für das Secret in der vorherigen Beispielausgabe.

Example

Für LinuxmacOS, oderUnix:

```
aws secretsmanager get-secret-value \
```

```
    --secret-id 'arn:aws:secretsmanager:us-east-2:123456789012:secret:rds!db-ABC123'
```

Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:us-east-2:123456789012:secret:rds!db-ABC123'
```

Amazon RDS-API

Sie können den ARN, den Status und den KMS-Schlüssel für ein von Amazon RDS verwaltetes Geheimnis anzeigen, AWS Secrets Manager indem Sie den DescribeTenantDatabasesVorgang verwenden und den `DBInstanceIdentifier` Parameter auf eine DB-Instance-ID setzen. Sie können den `TenantDBName` Parameter auch auf einen bestimmten Namen der Mandantendatenbank festlegen. Details zum Secret sind in der Ausgabe enthalten

Wenn Sie über den geheimen ARN verfügen, können Sie mithilfe des GetSecretValueSecrets Manager-Vorgangs Details zu dem Secret anzeigen.

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Verfügbarkeit von Versionen und Regionen für die Integration von Secrets Manager in Amazon RDS finden Sie unter Unterstützte Regionen und DB-Engines für die Secrets Manager Manager-Integration mit Amazon RDS.

Datenschutz in Amazon RDS

Das AWS Modell der gilt für den Datenschutz in Amazon Relational Database Service. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM)

einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon RDS oder anderen AWS-Services über die Konsole AWS CLI, API oder arbeiten AWS SDKs. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Themen

- Datenschutz durch Verschlüsselung
- Richtlinie für den Datenverkehr zwischen Netzwerken

Datenschutz durch Verschlüsselung

Sie können die Verschlüsselung für Datenbankressourcen aktivieren. Sie können auch Verbindungen zu verschlüsseln.

Themen

- Verschlüsseln von Amazon RDS-Ressourcen
- AWS KMS key Verwaltung
-
- Rotieren Ihrer SSL/TLS-Zertifikate

Verschlüsseln von Amazon RDS-Ressourcen

Amazon RDS kann Ihre Amazon RDS-DB-Instances verschlüsseln. Daten, die im Ruhezustand verschlüsselt werden, umfassen den zugehörigen Speicherplatz von DB-Instances sowie deren automatisierte Backups, Lesereplikate und Snapshots.

Amazon RDS-verschlüsselte DB-Instances verwenden den standardmäßig in der Branche verwendeten AES-256-Verschlüsselungsalgorithmus, um Ihre Daten auf dem Server zu verschlüsseln, der Ihre Amazon RDS-DB-Instances hostet. Sobald Sie die Daten verschlüsselt haben, übernimmt Amazon RDS die Authentifizierung des Zugriffs und die Entschlüsselung Ihrer Daten auf transparente Art und Weise und mit minimaler Auswirkung auf die Leistung. Sie müssen Ihre Datenbank-Client-Anwendungen nicht ändern, um Verschlüsselung anzuwenden.

> **Note**
>
> Bei verschlüsselten und unverschlüsselten werden Daten, die zwischen der Quelle und den Read Replicas übertragen werden, verschlüsselt, auch wenn sie regionsübergreifend repliziert werden. AWS

Themen

- Übersicht über die Verschlüsselung von Amazon RDS-Ressourcen
- Verschlüsseln einer DB-Instance
- Bestimmen, ob die Verschlüsselung für eine DB-Instance aktiviert ist
- Verfügbarkeit der Amazon RDS-Verschlüsselung
- Verschlüsselung während der Übertragung
- Einschränkungen von Amazon RDS-verschlüsselten DB-Instances

Übersicht über die Verschlüsselung von Amazon RDS-Ressourcen

Amazon RDS-verschlüsselte DB-Instances bieten zusätzlichen Datenschutz, indem Sie Ihre Daten vor unautorisiertem Zugriff auf den zugehörigen Speicherplatz sichern. Sie können die Amazon RDS-Verschlüsselung verwenden, um den Datenschutz für Ihre in der Cloud bereitgestellten Anwendungen zu erhöhen und die Compliance-Anforderungen bei der Verschlüsselung von Daten im Ruhezustand zu erfüllen. Für eine Amazon-RDS-verschlüsselte DB-Instance werden alle Protokolle, Backups und Snapshots verschlüsselt. Weitere Informationen zur Verfügbarkeit und zu den Einschränkungen der Verschlüsselung finden Sie unter und. Verfügbarkeit der Amazon RDS-Verschlüsselung Einschränkungen von Amazon RDS-verschlüsselten DB-Instances

Amazon RDS verwendet einen AWS Key Management Service Schlüssel, um diese Ressourcen zu verschlüsseln. AWS KMS kombiniert sichere, hochverfügbare Hardware und Software, um ein für die Cloud skaliertes Schlüsselverwaltungssystem bereitzustellen. Sie können einen Von AWS verwalteter Schlüssel verwenden oder vom Kunden verwaltete Schlüssel erstellen.

Wenn Sie eine verschlüsselte DB-Instanz erstellen, können Sie einen vom Kunden verwalteten Schlüssel oder den Von AWS verwalteter Schlüssel für Amazon RDS wählen, um Ihre DB-Instanz zu verschlüsseln. Wenn Sie die Schlüssel-ID für einen vom Kunden verwalteten Schlüssel nicht angeben, verwendet Amazon RDS die Von AWS verwalteter Schlüssel für Ihre neue DB-Instance. Amazon RDS erstellt eine Von AWS verwalteter Schlüssel für Amazon RDS für Ihr AWS Konto. Ihr AWS Konto hat Von AWS verwalteter Schlüssel für Amazon RDS für jede AWS Region ein anderes.

Um die vom Kunden verwalteten Schlüssel zu verwalten, die zum Verschlüsseln und Entschlüsseln Ihrer Amazon RDS-Ressourcen verwendet werden, verwenden Sie die AWS Key Management Service ()AWS KMS.

Mit AWS KMS dieser können Sie vom Kunden verwaltete Schlüssel erstellen und die Richtlinien definieren, die die Verwendung dieser vom Kunden verwalteten Schlüssel steuern. AWS KMS unterstützt CloudTrail, sodass Sie die Verwendung von KMS-Schlüsseln überprüfen können, um sicherzustellen, dass vom Kunden verwaltete Schlüssel ordnungsgemäß verwendet werden. Sie können Ihre vom Kunden verwalteten Schlüssel mit Amazon Aurora und unterstützten AWS Diensten wie Amazon S3, Amazon EBS und Amazon Redshift verwenden. Eine Liste der Dienste, die integriert sind, finden Sie unter AWS KMSAWS Serviceintegration. Einige Überlegungen zur Verwendung von KMS-Schlüsseln:

- Sobald Sie eine verschlüsselte DB-Instanz erstellt haben, können Sie den von dieser DB-Instanz verwendeten KMS-Schlüssel nicht mehr ändern. Stellen Sie daher sicher, dass Sie Ihre KMS-Schlüsselanforderungen bestimmen, bevor Sie Ihre verschlüsselte DB-Instanz erstellen.

Wenn Sie den Verschlüsselungsschlüssel für Ihre DB-Instance ändern müssen, erstellen Sie einen manuellen Snapshot Ihrer Instance und aktivieren Sie die Verschlüsselung beim Kopieren des Snapshots. Weitere Informationen finden Sie im Artikel Re:POST Knowledge.

- Wenn Sie einen verschlüsselten Snapshot kopieren, können Sie zum Verschlüsseln des Ziel-Snapshots einen anderen KMS-Schlüssel verwenden als den, der zum Verschlüsseln des Quell-Snapshots verwendet wurde.
- Eine Read Replica einer Amazon RDS-verschlüsselten Instance muss mit demselben KMS-Schlüssel wie die primäre DB-Instance verschlüsselt werden, wenn sich beide in derselben AWS Region befinden.
- Wenn sich die primäre DB-Instance und die Read Replica in unterschiedlichen AWS Regionen befinden, verschlüsseln Sie die Read Replica mit dem KMS-Schlüssel für diese Region. AWS
- Sie können keinen Snapshot teilen, der mit dem Von AWS verwalteter Schlüssel des AWS Kontos verschlüsselt wurde, das den Snapshot geteilt hat.
- Amazon RDS unterstützt auch die Verschlüsselung einer Oracle- oder SQL Server-DB-Instance mit Transparent Data Encryption (TDE). TDE kann mit RDS-Verschlüsselung im Ruhezustand verwendet werden, jedoch kann sich die gleichzeitige Verwendung von TDE und RDS-Verschlüsselung im Ruhezustand geringfügig auf die Leistung Ihrer Datenbank auswirken. Sie müssen verschiedene Schlüssel für jede Verschlüsselungsmethode verwalten. Weitere Informationen zu TDE finden Sie unter Oracle Transparent Data Encryption oder Support für transparente Datenverschlüsselung im SQL Server.

Important

Amazon RDS verliert den Zugriff auf den KMS-Schlüssel für eine DB-Instance, wenn Sie den KMS-Schlüssel deaktivieren. Wenn Sie den Zugriff auf einen KMS-Schlüssel verlieren, geht die verschlüsselte DB-Instance in Instances, in denen Backups aktiviert sind, 2 Stunden nach der Erkennung in den `inaccessible-encryption-credentials-recoverable` Status über. Die DB-Instance bleibt sieben Tage lang in diesem Zustand. Während dieser Zeit wird die Instance gestoppt. API-Aufrufe, die während dieser Zeit an die DB-Instance getätigt wurden, sind möglicherweise nicht erfolgreich. Um die DB-Instance wiederherzustellen, aktivieren Sie den KMS-Schlüssel und starten Sie diese DB-Instance neu. Aktivieren Sie den KMS-Schlüssel über die AWS Management Console AWS CLI, oder RDS-API. Starten Sie die DB-Instance mit dem AWS CLI Befehl start-db-instanceoder neu AWS Management Console.

Der `inaccessible-encryption-credentials-recoverable` Status gilt nur für DB-Instances, die gestoppt werden können. Dieser wiederherstellbare Status gilt nicht für Instances, die nicht gestoppt werden können, wie Read Replicas und Instances mit Read Replicas. Weitere Informationen finden Sie unter Einschränkungen beim Stoppen Ihrer DB-Instance.

Wenn die DB-Instance nicht innerhalb von sieben Tagen wiederhergestellt wird, wechselt sie in den Terminalstatus. `inaccessible-encryption-credentials` In diesem Zustand ist die DB-Instance nicht mehr nutzbar und Sie können die DB-Instance nur aus einem Backup wiederherstellen. Wir empfehlen nachdrücklich, dass Sie zu jeder Zeit Backups für verschlüsselte DB-Instances aktivieren, um sich gegen den Datenverlust von verschlüsselten Daten in Ihren Datenbanken abzusichern.

Während der Erstellung einer DB-Instance prüft Amazon RDS, ob der aufrufende Principal Zugriff auf den KMS-Schlüssel hat, und generiert aus dem KMS-Schlüssel einen Grant, den es für die gesamte Lebensdauer der DB-Instance verwendet. Das Widerrufen des Zugriffs des aufrufenden Prinzipals auf den KMS-Schlüssel hat keine Auswirkungen auf eine laufende Datenbank. Wenn KMS-Schlüssel in kontoübergreifenden Szenarien verwendet werden, z. B. beim Kopieren eines Snapshots in ein anderes Konto, muss der KMS-Schlüssel mit dem anderen Konto geteilt werden. Wenn Sie aus dem Snapshot eine DB-Instance erstellen, ohne einen anderen KMS-Schlüssel anzugeben, verwendet die neue Instance den KMS-Schlüssel aus dem Quellkonto. Wenn Sie den Zugriff auf den Schlüssel widerrufen, nachdem Sie die DB-Instance erstellt haben, hat dies keine Auswirkungen auf die Instance. Die Deaktivierung des Schlüssels wirkt sich jedoch auf alle DB-Instances aus, die mit diesem Schlüssel verschlüsselt wurden. Um dies zu verhindern, geben Sie während des Snapshot-Kopiervorgangs einen anderen Schlüssel an.

DB-Instances mit deaktivierten Backups bleiben verfügbar, bis die Volumes während einer Instance-Änderung oder Wiederherstellung vom Host getrennt werden. RDS versetzt die Instances je nach `inaccessible-encryption-credentials-recoverable` Bedarf in einen `inaccessible-encryption-credentials` Status oder einen Status.

Weitere Informationen über KMS-Schlüssel finden Sie unter AWS KMS keys im Entwicklerhandbuch zu AWS Key Management Service und in AWS KMS key Verwaltung.

Verschlüsseln einer DB-Instance

Wählen Sie Enable encryption (Verschlüsselung aktivieren) in der Amazon RDS-Konsole aus, um eine neue DB-Instance zu verschlüsseln. Hinweise zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance.

Wenn Sie den create-db-instance AWS CLI Befehl verwenden, um eine verschlüsselte DB-Instance zu erstellen, legen Sie den `--storage-encrypted` Parameter fest. Wenn Sie den Vorgang Create DBInstance API verwenden, setzen Sie den `StorageEncrypted` Parameter auf true.

Wenn Sie den AWS CLI `create-db-instance` Befehl verwenden, um eine verschlüsselte DB-Instance mit einem vom Kunden verwalteten Schlüssel zu erstellen, legen Sie den `--kms-key-id` Parameter auf eine beliebige Schlüssel-ID für den KMS-Schlüssel fest. Wenn Sie den Vorgang Amazon RDS API `CreateDBInstance` verwenden, setzen Sie den Parameter `KmsKeyId` auf einen beliebigen Schlüsselbezeichner für den KMS-Schlüssel. Um einen vom Kunden verwalteten Schlüssel in einem anderen AWS Konto zu verwenden, geben Sie den Schlüssel-ARN oder den Alias-ARN an.

Bestimmen, ob die Verschlüsselung für eine DB-Instance aktiviert ist

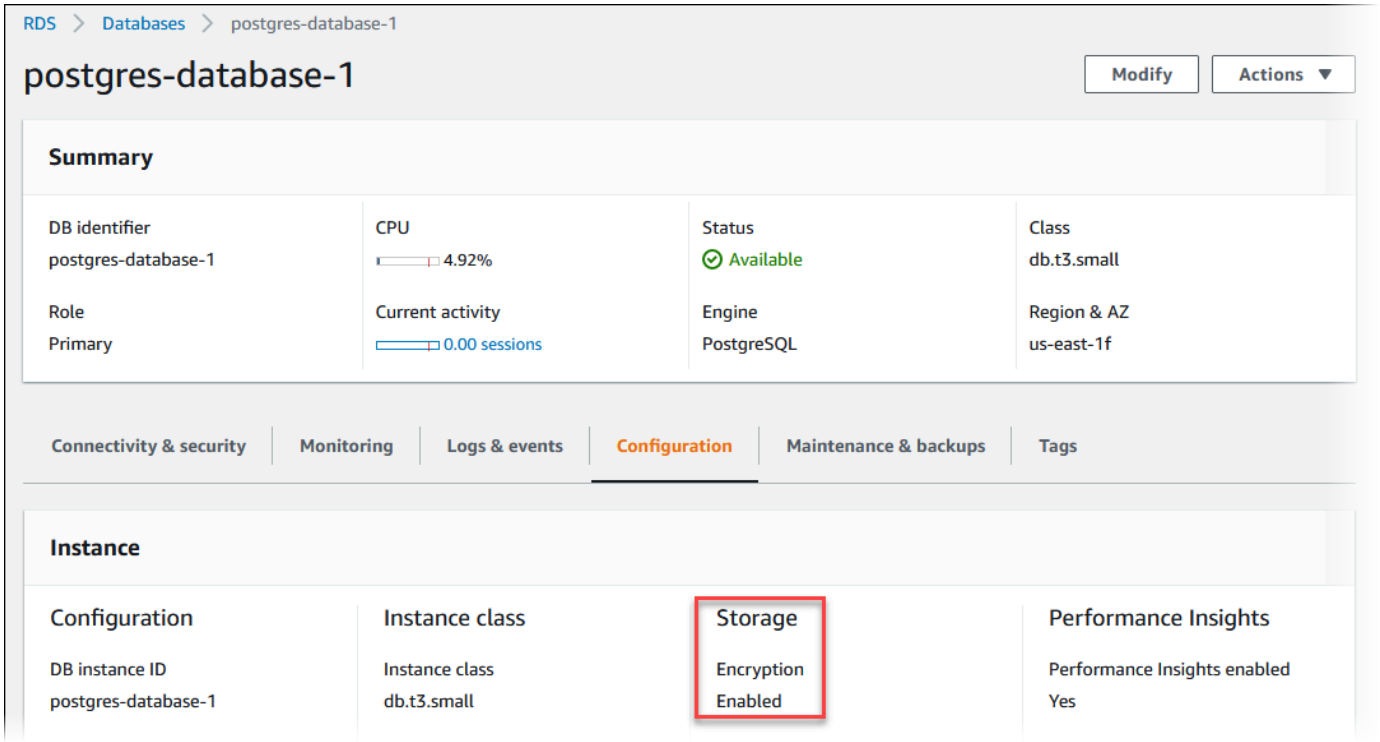
Sie können die AWS Management Console, oder RDS-API verwenden AWS CLI, um festzustellen, ob die Verschlüsselung im Ruhezustand für eine DB-Instance aktiviert ist.

Konsole

So ermitteln Sie, ob die Verschlüsselung im Ruhezustand für eine DB-Instance aktiviert ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie den Namen der DB-Instance aus, die Sie überprüfen möchten, um die Details anzuzeigen.
4. Wählen Sie die Registerkarte Konfiguration aus und überprüfen Sie den Wert für die Verschlüsselung unter Speicher.

   Es zeigt entweder Aktiviert oder Nicht aktiviert.



AWS CLI

Rufen Sie den describe-db-instancesBefehl mit der folgenden Option auf, um festzustellen AWS CLI, ob die Verschlüsselung im Ruhezustand für eine DB-Instance aktiviert ist:

- `--db-instance-identifier` – der Name der DB-Instance

Im folgenden Beispiel wird eine Abfrage verwendet, um entweder `TRUE` oder `FALSE` bezüglich der Verschlüsselung im Ruhezustand für die `mydb` DB-Instance zurückzugeben.

Example

```
aws rds describe-db-instances --db-instance-identifier mydb --query "*[].
{StorageEncrypted:StorageEncrypted}" --output text
```

RDS-API

Um mithilfe der Amazon RDS-API zu ermitteln, ob die Verschlüsselung im Ruhezustand für eine DB-Instance aktiviert ist, rufen Sie den DBInstances Vorgang Describe mit dem folgenden Parameter auf:

- `DBInstanceIdentifier` – der Name der DB-Instance.

Verfügbarkeit der Amazon RDS-Verschlüsselung

Die Amazon RDS-Verschlüsselung ist aktuell für alle Datenbank-Engines und Speichertypen verfügbar.

Amazon RDS-Verschlüsselung ist für die meisten DB-Instance-Klassen verfügbar. In der folgenden Tabelle sind die DB-Instance-Klassen aufgeführt, die Amazon-RDS-Verschlüsselung nicht unterstützen:

Instance-Typ	Instance class
Allzweck (M1)	db.m1.small db.m1.medium db.m1.large db.m1.xlarge
Arbeitsspeicheroptimiert (M2)	db.m2.xlarge db.m2.2xlarge db.m2.4xlarge
Burstable (T2)	db.t2.micro

Verschlüsselung während der Übertragung

Verschlüsselung auf der physischen Ebene

Alle Daten, die AWS-Regionen über das AWS globale Netzwerk übertragen werden, werden auf der physischen Ebene automatisch verschlüsselt, bevor sie AWS gesicherte Einrichtungen verlassen. Der gesamte Datenverkehr zwischen beiden AZs ist verschlüsselt. Zusätzliche Verschlüsselungsebenen, einschließlich der in diesem Abschnitt aufgeführten, können zusätzlichen Schutz bieten.

Verschlüsselung durch Amazon VPC-Peering und regionsübergreifendes Transit Gateway Gateway-Peering

Der gesamte regionsübergreifende Datenverkehr, der Amazon-VPC- und Transit-Gateway-Peering verwendet, wird automatisch massenverschlüsselt, wenn er eine Region verlässt. Auf der physischen Ebene wird automatisch eine zusätzliche Verschlüsselungsebene für den gesamten Datenverkehr bereitgestellt, bevor er gesicherte Einrichtungen verlässt. AWS

Verschlüsselung zwischen Instanzen

AWS bietet sichere und private Konnektivität zwischen DB-Instances aller Typen. Darüber hinaus verwenden einige Instance-Typen die Offload-Funktionen der zugrunde liegenden Nitro-System-Hardware, um den Datenverkehr während der Übertragung zwischen Instances automatisch zu verschlüsseln. Diese Verschlüsselung verwendet AEAD-Algorithmen (Authenticated Encryption with Associated Data) mit 256-Bit-Verschlüsselung. Es gibt keine Auswirkungen auf die Netzwerkleistung. Um diese zusätzliche Verschlüsselung des Datenverkehrs während der Übertragung zwischen Instances zu unterstützen, müssen die folgenden Anforderungen erfüllt sein:

- Die Instances verwenden die folgenden Instance-Typen:
  - Allgemeiner Zweck: M6i, M6id, M6in, M6idn, M7g
  - Speicheroptimiert: R6i, R6id, R6in, R6idn, R7G, X2idn, X2iEDN, X2IEZn
- Die Instanzen AWS-Region befinden sich in derselben.
- Die Instances befinden sich in derselben VPC oder werden per Peering betrieben VPCs, und der Datenverkehr wird nicht über ein virtuelles Netzwerkgerät oder einen virtuellen Netzwerkdienst wie einen Load Balancer oder ein Transit-Gateway geleitet.

Einschränkungen von Amazon RDS-verschlüsselten DB-Instances

Folgende Einschränkungen bestehen für Amazon RDS-verschlüsselte DB-Instances:

- Sie können eine Amazon-RDS-DB-Instance nur beim Erstellen verschlüsseln, nicht nachdem die DB-Instance bereits erstellt ist.

  Da es jedoch möglich ist, die Kopie eines unverschlüsselten Snapshots zu verschlüsseln, können Sie quasi eine Verschlüsselung zu einer unverschlüsselten DB-Instance hinzufügen. Dies lässt sich durchführen, indem Sie einen Snapshot von Ihrer DB-Instance erstellen und dann eine verschlüsselte Kopie dieses Snapshots erstellen. Anschließend können Sie Ihre DB-Instance aus dem verschlüsselten Snapshot wiederherstellen und verfügen so über eine verschlüsselte Kopie

Ihrer ursprünglichen DB-Instance. Weitere Informationen finden Sie unter Kopieren eines DB-Snapshots für Amazon RDS.

- Sie können die Verschlüsselung für eine verschlüsselte DB-Instance nicht deaktivieren.
- Sie können keinen verschlüsselten Snapshot einer/eines unverschlüsselten DB-Instance erstellen.
- Ein Snapshot eines verschlüsselten DB-Instance muss mit demselben KMS-Schlüssel verschlüsselt werden wie der DB-Instance.
- Es ist nicht möglich, ein verschlüsseltes Lesereplikat einer unverschlüsselten DB-Instance oder ein unverschlüsseltes Lesereplikat einer verschlüsselten DB-Instance zu erstellen.
- Verschlüsselte Lesereplikate müssen mit demselben KMS-Schlüssel verschlüsselt werden wie die Quell-DB-Instanz, wenn sich beide in derselben AWS Region befinden.
- Sie können ein unverschlüsseltes Backup oder einen solchen Snapshot nicht als verschlüsselte DB-Instance wiederherstellen.
- Um einen verschlüsselten Snapshot von einer AWS Region in eine andere zu kopieren, müssen Sie den KMS-Schlüssel in der Zielregion angeben. AWS Dies liegt daran, dass KMS-Schlüssel für die AWS Region spezifisch sind, in der sie erstellt wurden.

  Der Quell-Snapshot bleibt den gesamten Kopiervorgang über verschlüsselt. Amazon RDS verwendet Envelope-Verschlüsselung, um Daten während des Kopiervorgangs zu schützen. Weitere Informationen zur Envelope-Verschlüsselung finden Sie unter Envelope-Verschlüsselung im AWS Key Management Service -Entwicklerhandbuch.
- Sie können eine(n) verschlüsselte(n) DB-Instance nicht entschlüsseln. Sie können jedoch Daten aus einer/einem verschlüsselten DB-Instance exportieren und die Daten in eine(n) unverschlüsselte(n) DB-Instance importieren.

AWS KMS key Verwaltung

Amazon RDS integriert sich automatisch mit AWS Key Management Service (AWS KMS) für die Schlüsselverwaltung. Amazon RDS verwendet Umschlagverschlüsselung. Weitere Informationen zur Envelope-Verschlüsselung finden Sie unter Envelope-Verschlüsselung im AWS Key Management Service -Entwicklerhandbuch.

Sie können zwei Arten von AWS KMS Schlüsseln verwenden, um Ihre zu verschlüsseln.

- Wenn Sie die volle Kontrolle über einen KMS-Schlüssel haben möchten, müssen Sie einen vom Kunden verwalteten Schlüssel erstellen. Weitere Informationen über kundenverwaltete Schlüssel finden Sie unter Kundenverwaltete Schlüssel im AWS Key Management Service Developer Guide.

- Von AWS verwaltete Schlüsselsind KMS-Schlüssel in Ihrem Konto, die in Ihrem Namen von einem integrierten AWS Dienst erstellt, verwaltet und verwendet werden. AWS KMS Standardmäßig wird RDS Von AWS verwalteter Schlüssel (`aws/rds`) für die Verschlüsselung verwendet. Sie können den RDS nicht verwalten, rotieren oder löschen Von AWS verwalteter Schlüssel. Weitere Informationen zu Von AWS verwaltete Schlüssel finden Sie Von AWS verwaltete Schlüsselim AWS Key Management Service Entwicklerhandbuch.

Um KMS-Schlüssel zu verwalten, die für Amazon RDS verwendet werden, verwenden Sie AWS Key Management Service (AWS KMS) in der AWS KMS Konsole AWS CLI, die oder die AWS KMS API. Verwenden Sie zur Anzeige von Audit-Protokollen für jede Aktion, die mit einem AWS -verwalteten oder kundenverwalteten Schlüssel durchgeführt wurde AWS CloudTrail. Weitere Informationen zum Rotieren der Schlüssel finden Sie unter Rotieren von AWS KMS -Schlüsseln.

Autorisieren der Verwendung eines kundenverwalteten Schlüssels

Wenn RDS einen vom Kunden verwalteten Schlüssel für kryptografische Operationen verwendet, handelt es im Namen des Benutzers, der die RDS erstellt oder ändert.

Um eine RDS mit einem vom Kunden verwalteten Schlüssel zu erstellen, muss ein Benutzer berechtigt sein, die folgenden Operationen mit dem vom Kunden verwalteten Schlüssel aufzurufen:

- `kms:CreateGrant`
- `kms:DescribeKey`

Sie können diese erforderlichen Berechtigungen in einer Schlüsselrichtlinie oder in einer IAM-Richtlinie angeben, wenn die Schlüsselrichtlinie dies zulässt.

> **Important**
>
> Wenn Sie explizite Deny-Statements für alle Ressourcen (*) in AWS KMS wichtigen Richtlinien mit verwalteten Services wie Amazon RDS verwenden, müssen Sie eine Bedingung angeben, um das Konto, das die Ressource besitzt, zuzulassen. Ohne diese Bedingung können Operationen fehlschlagen, auch wenn die Ablehnungsregel Ausnahmen für Ihren IAM-Benutzer enthält.

> **Tip**
>
> Um den Grundsatz der Erteilung der geringsten erforderlichen Berechtigungen zu befolgen, lassen Sie den vollständigen Zugriff auf `kms:CreateGrant` nicht zu. Verwenden Sie stattdessen den ViaService Bedingungsschlüssel kms:, damit der Benutzer nur dann Berechtigungen für den KMS-Schlüssel erstellen kann, wenn der Zuschuss im Namen des Benutzers von einem AWS Dienst erstellt wird.

Sie können die IAM-Richtlinie auf verschiedene Weise strikter gestalten. Wenn Sie beispielsweise zulassen möchten, dass der vom Kunden verwaltete Schlüssel nur für Anfragen verwendet wird, die ihren Ursprung in RDS haben, verwenden Sie den kms:ViaService Bedingungsschlüssel mit dem `rds.<region>.amazonaws.com` Wert. Sie können auch die Schlüssel oder Werte im Amazon-RDS-Verschlüsselungskontext als Bedingung für die Verwendung des vom Kunden verwalteten Schlüssels für die Verschlüsselung verwenden.

Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung eines KMS-Schlüssels erlauben im AWS Key Management Service -Entwicklerhandbuch und unter Schlüsselrichtlinien in AWS KMS.

Amazon-RDS-Verschlüsselungskontext

Wenn RDS Ihren KMS-Schlüssel verwendet oder wenn Amazon EBS den KMS-Schlüssel im Namen von RDS verwendet, spezifiziert der Service einen Verschlüsselungskontext. Der Verschlüsselungskontext besteht aus zusätzlichen authentifizierten Daten (AAD), die zur Sicherstellung der AWS KMS Datenintegrität verwendet werden. Wenn für eine Verschlüsselungsoperation ein Verschlüsselungskontext angegeben wird, muss der Service denselben Verschlüsselungskontext auch für die Entschlüsselungsoperation angeben. Andernfalls schlägt die Entschlüsselung fehl. Der Verschlüsselungskontext wird zudem in Ihre AWS CloudTrail-Protokolle geschrieben, sodass Sie jederzeit nachvollziehen können, warum ein bestimmter KMS-Schlüssel verwendet wurde. Ihre CloudTrail Protokolle können viele Einträge enthalten, die die Verwendung eines KMS-Schlüssels beschreiben, aber der Verschlüsselungskontext in jedem Protokolleintrag kann Ihnen helfen, den Grund für diese bestimmte Verwendung zu ermitteln.

Amazon RDS verwendet mindestens immer die der DB-Instance für den Verschlüsselungskontext, wie im folgenden Beispiel im JSON-Format:

```
{ "aws:rds:db-id": "db-CQYSMDPBRZ7BPMH7Y3RTDG5QY" }
```

Dieser Verschlüsselungskontext kann Ihnen helfen, den zu identifizieren, für den Ihr KMS-Schlüssel verwendet wurde.

Wenn Ihr KMS-Schlüssel für einen bestimmten und ein bestimmtes Amazon EBS-Volume verwendet wird, werden sowohl die als auch die Amazon EBS-Volume-ID für den Verschlüsselungskontext verwendet, wie im folgenden Beispiel im JSON-Format:

```
{
  "aws:rds:db-id": "db-BRG7VYS3SVIFQW7234EJQOM5RQ",
  "aws:ebs:id": "vol-ad8c6542"
}
```

Sie können Secure Socket Layer (SSL) oder Transport Layer Security (TLS) von Ihrer Anwendung aus verwenden, um eine Verbindung zu einer Datenbank zu verschlüsseln, auf der Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle oder PostgreSQL ausgeführt wird.

Optional kann Ihre SSL/TLS Verbindung die Serveridentität überprüfen, indem das in Ihrer Datenbank installierte Serverzertifikat validiert wird. Gehen Sie wie folgt vor, um eine Überprüfung der Serveridentität vorzuschreiben:

1. Wählen Sie die Zertifizierungsstelle (Certificate Authority, CA) aus, die das DB-Serverzertifikat für Ihre Datenbank zertifiziert. Weitere Informationen zu Zertifizierungsstellen finden Sie unterZertifizierungsstellen.
2. Laden Sie ein Zertifikatspaket herunter, das verwendet werden soll, wenn Sie eine Verbindung zur Datenbank herstellen. Informationen zum Herunterladen eines Zertifikatspakets finden Sie unter Zertifikatspakete von AWS-Region.

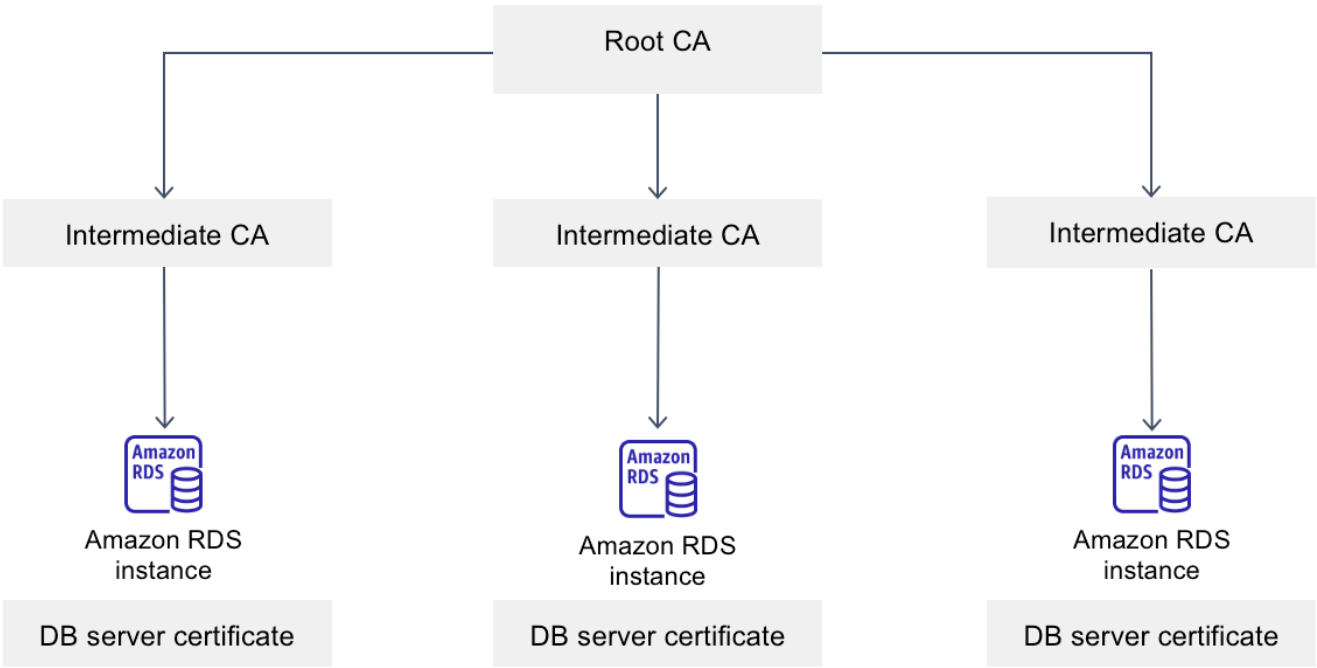
   > **Note**
   >
   > Alle Zertifikate stehen nur über SSL/TLS-Verbindungen zum Download zur Verfügung.

3. Connect Sie mithilfe des Prozesses Ihrer DB-Engine zur Implementierung von SSL/TLS Verbindungen eine Verbindung zur Datenbank her. Jede DB-Engine hat ihren eigenen Prozess für die Implementierung SSL/TLS. To learn how to implement SSL/TLS für Ihre Datenbank. Folgen Sie dem Link, der Ihrer DB-Engine entspricht:
   - Verwendung SSL/TLS mit einer Amazon RDS for Db2-DB-Instance
   - SSL/TLSUnterstützung für MariaDB-DB-Instances auf Amazon RDS
   - Verwendung SSL mit einer Microsoft SQL Server-DB-Instance

- SSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS
- Verwenden von SSL mit einer DB-Instance von RDS für Oracle
- Verwenden von SSL mit einer PostgreSQL-DB-Instance

Zertifizierungsstellen

Die Zertifizierungsstelle (CA) ist das Zertifikat, das die Stamm-CA an der Spitze der Zertifikatskette identifiziert. Die CA signiert das DB-Serverzertifikat. Dies ist das Serverzertifikat, das auf jeder DB-Instance installiert ist. Das DB-Serverzertifikat identifiziert die DB-Instance als vertrauenswürdigen Server.



Amazon RDS bietet Folgendes CAs , um das DB-Serverzertifikat für eine Datenbank zu signieren.

Zertifizierungsstelle (Certificate authority, CA)	Beschreibung	Allgemeiner Name (CN)
rds-ca-rsa2048-g1	Verwendet in den meisten Fällen eine Zertifizierungsstelle mit einem RSA 2048-Algorithmus für private Schlüssel und	Amazon RDS *region-id entifier* Root CA RSA2 048 G1

Zertifizierungsstelle (Certificate authority, CA)	Beschreibung	Allgemeiner Name (CN)
	SHA256 einem Signaturalgorithmus. AWS-Regionen In der AWS GovCloud (US) Regions verwendet diese Zertifizierungsstelle eine Zertifizierungsstelle mit dem RSA 2048-Algorithmus für private Schlüssel und dem Signaturalgorithmus. SHA384 Diese CA unterstützt die automatische Rotation von Serverzertifikaten.	
rds-ca-rsa4096-g1	Verwendet eine Zertifizierungsstelle mit RSA 4096-Algorithmus und Signatura lgorithmus für private Schlüssel. SHA384 Diese CA unterstützt die automatische Rotation von Serverzertifikaten.	Amazon RDS *region-id entifier* Root CA RSA4 096 G1
rds-ca-ecc384-g1	Verwendet eine Zertifizierungsstelle mit ECC 384-Algorithmus für private Schlüssel und Signaturalgorithmu s. SHA384 Diese CA unterstützt die automatische Rotation von Serverzer tifikaten.	Amazon RDS *region-id entifier* Root CA ECC384 G1

> **Note**
>
> Wenn Sie das verwenden AWS CLI, können Sie die Gültigkeiten der oben aufgeführten Zertifizierungsstellen mithilfe von describe-certificates überprüfen.

Diese CA-Zertifikate sind im regionalen und globalen Zertifikat-Bundle enthalten. Wenn Sie die Zertifizierungsstelle rds-ca-rsa 2048-g1, rds-ca-rsa 4096-g1 oder rds-ca-ecc 384-g1 mit einer

Datenbank verwenden, verwaltet RDS das DB-Serverzertifikat in der Datenbank. RDS rotiert das DB-Serverzertifikat automatisch, bevor es abläuft.

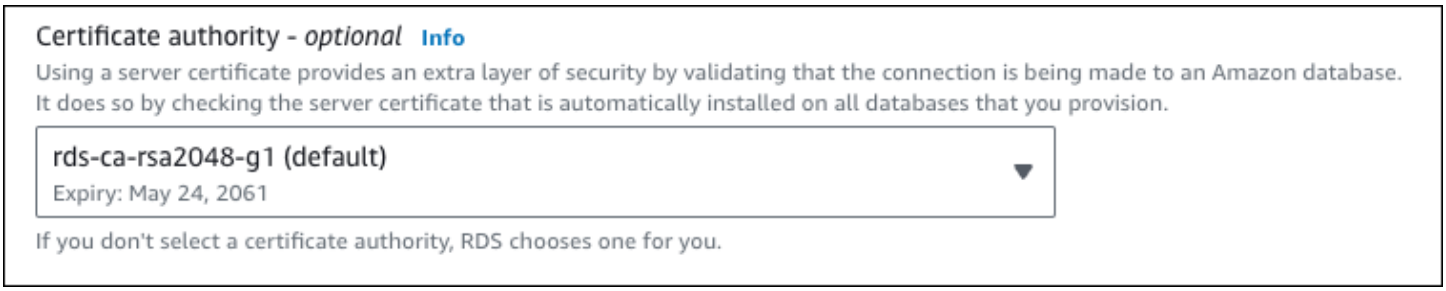
Einstellung der CA für Ihre Datenbank

Sie können die CA für eine Datenbank einstellen, wenn Sie die folgenden Aufgaben ausführen:

- Erstellen Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster — Sie können die CA festlegen, wenn Sie eine DB-Instance oder einen DB-Cluster erstellen. Anweisungen finden Sie unter the section called “Erstellen einer DB-Instance” oderthe section called “Erstellen eines Multi-AZ-DB-Clusters”.
- Ändern Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster — Sie können die CA für eine DB-Instance oder einen DB-Cluster festlegen, indem Sie sie ändern. Anweisungen finden Sie unter the section called “Ändern einer DB-Instance” oderthe section called “Ändern eines Multi-AZ-DB-Clusters”.

> **Note**
>
> Die Standard-CA ist auf rds-ca-rsa 2048-g1 festgelegt. Sie können die Standard-CA für Sie überschreiben, AWS-Konto indem Sie den Befehl modify-certificates verwenden.

Die verfügbaren Optionen CAs hängen von der DB-Engine und der DB-Engine-Version ab. Wenn Sie die verwenden AWS Management Console, können Sie die Zertifizierungsstelle mithilfe der Einstellung Zertifizierungsstelle auswählen, wie in der folgenden Abbildung dargestellt.



In der Konsole werden nur die angezeigt CAs , die für die DB-Engine und die DB-Engine-Version verfügbar sind. Wenn Sie die verwenden AWS CLI, können Sie die CA für eine DB-Instance mit dem modify-db-instanceBefehl create-db-instanceor festlegen. Sie können die CA für einen Multi-AZ-DB-Cluster mit dem modify-db-clusterBefehl create-db-clusteror festlegen.

Wenn Sie den verwenden AWS CLI, können Sie mit dem Befehl describe-certificates die CAs für Ihr Konto verfügbaren Werte anzeigen. Dieser Befehl zeigt in der Ausgabe auch das Ablaufdatum für jede CA in `ValidTill` an. Mithilfe des Befehls können Sie herausfinden CAs , welche für eine bestimmte DB-Engine und DB-Engine-Version verfügbar sind. describe-db-engine-versions

Das folgende Beispiel zeigt die für die Standardversion der RDS for PostgreSQL DB-Engine CAs verfügbare Version.

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Ihre Ausgabe sieht Folgendem ähnlich. Die verfügbaren CAs sind unter aufgeführt. `SupportedCACertificateIdentifiers` Die Ausgabe zeigt auch, ob die Version der DB-Engine das Rotieren des Zertifikats ohne Neustart in `SupportsCertificateRotationWithoutRestart` unterstützt.

```
{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "MajorEngineVersion": "13",
            "EngineVersion": "13.4",
            "DBParameterGroupFamily": "postgres13",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 13.4-R1",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [
                "Lambda"
            ],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "SupportsBabelfish": false,
            "SupportsCertificateRotationWithoutRestart": true,
            "SupportedCACertificateIdentifiers": [
                "rds-ca-rsa2048-g1",
                "rds-ca-ecc384-g1",
                "rds-ca-rsa4096-g1"
            ]
        }
    ]
```

```
}
```

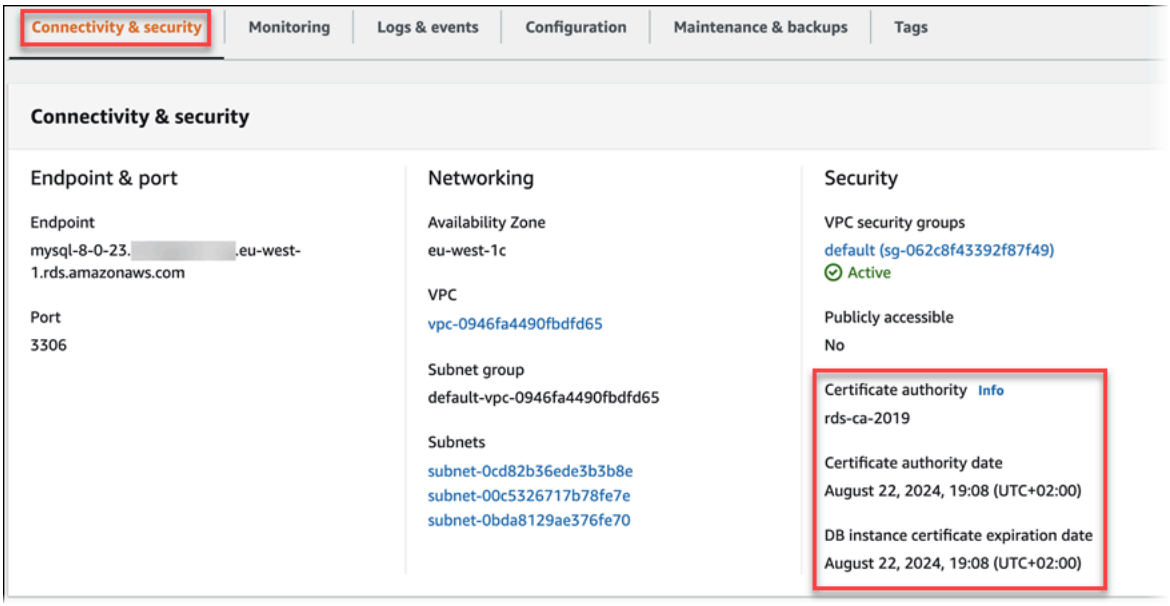
Gültigkeiten von DB-Serverzertifikaten

Die Gültigkeit des DB-Serverzertifikats hängt von der DB-Engine und der Version der DB-Engine ab. Wenn die Version der DB-Engine das Rotieren des Zertifikats ohne Neustart unterstützt, beträgt die Gültigkeit des DB-Serverzertifikats 1 Jahr. Andernfalls beträgt die Gültigkeit 3 Jahre.

Weitere Hinweise zur Rotation von DB-Serverzertifikaten finden Sie unterAutomatische Rotation von Serverzertifikaten.

Die CA für Ihre DB-Instance anzeigen

Sie können die Details zur CA für eine Datenbank einsehen, indem Sie die Registerkarte Konnektivität und Sicherheit in der Konsole aufrufen, wie in der folgenden Abbildung dargestellt.



Wenn Sie die verwenden AWS CLI, können Sie die Details zur CA für eine DB-Instance mithilfe des describe-db-instancesBefehls anzeigen. Sie können die Details zur CA für einen Multi-AZ-DB-Cluster mithilfe des describe-db-clustersBefehls anzeigen.

Wenn Sie mit SSL oder TLS eine Verbindung zu Ihrer Datenbank herstellen, benötigt die Datenbank-Instance ein Vertrauenszertifikat von Amazon RDS. Wählen Sie den entsprechenden Link in der folgenden Tabelle aus, um das Paket herunterzuladen, das dem entspricht, AWS-Region wo Sie Ihre Datenbank hosten.

Zertifikatspakete von AWS-Region

Die Zertifikatspakete für alle Regionen AWS-Regionen und GovCloud (USA) enthalten die folgenden Root-CA-Zertifikate:

- `rds-ca-rsa2048-g1`
- `rds-ca-rsa4096-g1`
- `rds-ca-ecc384-g1`

Die `rds-ca-ecc384-g1` Zertifikate `rds-ca-rsa4096-g1` und sind in den folgenden Regionen nicht verfügbar:

- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Melbourne)
- Kanada West (Calgary)
- Europa (Zürich)
- Europa (Spain)
- Israel (Tel Aviv)

Ihr Anwendungs-Truststore muss nur das Root-CA-Zertifikat registrieren.

> **Note**
>
> Amazon RDS Proxy Zertifikate von AWS Certificate Manager (ACM). Wenn Sie RDS Proxy verwenden, müssen Sie keine Amazon RDS-Zertifikate herunterladen oder Anwendungen aktualisieren, die RDS-Proxy-Verbindungen verwenden. Weitere Informationen finden Sie unter TLS/SSLmit RDS Proxy verwenden.

Um ein Zertifikatspaket für ein herunterzuladen AWS-Region, wählen Sie in der folgenden Tabelle den Link für AWS-Region das Zertifikat aus, das Ihre Datenbank hostet.

AWS Region	Zertifikat-Paket (PEM)	Zertifikatspaket (PKCS7)
Irgendein kommerzieller AWS-Region	global-bundle.pem	global-bundle.p7b

AWS Region	Zertifikat-Paket (PEM)	Zertifikatspaket (PKCS7)
USA Ost (Nord-Virginia)	us-east-1-bundle.pem	us-east-1-bundle.p7b
US East (Ohio)	us-east-2-bundle.pem	us-east-2-bundle.p7b
USA West (Nordkalifornien)	us-west-1-bundle.pem	us-west-1-bundle.p7b
USA West (Oregon)	us-west-2-bundle.pem	us-west-2-bundle.p7b
Africa (Cape Town)	af-south-1-bundle.pem	af-south-1-bundle.p7b
Asia Pacific (Hong Kong)	ap-east-1-bundle.pem	ap-east-1-bundle.p7b
Asien-Pazifik (Hyderabad)	ap-south-2-bundle.pem	ap-south-2-bundle.p7b
Asien-Pazifik (Jakarta)	ap-southeast-3-bundle.pem	ap-southeast-3-bundle.p7b
Asien-Pazifik (Malaysia)	ap-southeast-5-bundle.pem	ap-southeast-5-bundle.p7b
Asien-Pazifik (Melbourne)	ap-southeast-4-bundle.pem	ap-southeast-4-bundle.p7b
Asien-Pazifik (Mumbai)	ap-south-1-bundle.pem	ap-south-1-bundle.p7b
Asia Pacific (Osaka)	ap-northeast-3-bundle.pem	ap-northeast-3-bundle.p7b
Asien-Pazifik (Thailand)	ap-southeast-7-bundle.pem	ap-southeast-7-bundle.p7b
Asien-Pazifik (Tokio)	ap-northeast-1-bundle.pem	ap-northeast-1-bundle.p7b
Asia Pacific (Seoul)	ap-northeast-2-bundle.pem	ap-northeast-2-bundle.p7b
Asien-Pazifik (Singapur)	ap-southeast-1-bundle.pem	ap-southeast-1-bundle.p7b
Asien-Pazifik (Sydney)	ap-southeast-2-bundle.pem	ap-southeast-2-bundle.p7b
Canada (Central)	ca-central-1-bundle.pem	ca-central-1-bundle.p7b
Kanada West (Calgary)	ca-west-1-bundle.pem	ca-west-1-bundle.p7b
Europa (Frankfurt)	eu-central-1-bundle.pem	eu-central-1-bundle.p7b

AWS Region	Zertifikat-Paket (PEM)	Zertifikatspaket (PKCS7)
Europa (Irland)	eu-west-1-bundle.pem	eu-west-1-bundle.p7b
Europe (London)	eu-west-2-bundle.pem	eu-west-2-bundle.p7b
Europe (Milan)	eu-south-1-bundle.pem	eu-south-1-bundle.p7b
Europe (Paris)	eu-west-3-bundle.pem	eu-west-3-bundle.p7b
Europa (Spain)	eu-south-2-bundle.pem	eu-south-2-bundle.p7b
Europa (Stockholm)	eu-north-1-bundle.pem	eu-north-1-bundle.p7b
Europa (Zürich)	eu-central-2-bundle.pem	eu-central-2-bundle.p7b
Israel (Tel Aviv)	il-central-1-bundle.pem	il-central-1-bundle.p7b
Mexiko (Zentral)	mx-central-1-bundle.pem	mx-central-1-bundle.p7b
Naher Osten (Bahrain)	me-south-1-bundle.pem	me-south-1-bundle.p7b
Naher Osten (VAE)	me-central-1-bundle.pem	me-central-1-bundle.p7b
Südamerika (São Paulo)	sa-east-1-bundle.pem	sa-east-1-bundle.p7b
Irgendein s AWS GovCloud (US) Region	global-bundle.pem	global-bundle.p7b
AWS GovCloud (US-Ost)	us-gov-east-1-bundle.pem	us-gov-east-1-Bundle.p7b
AWS GovCloud (US-West)	us-gov-west-1-bundle.pem	us-gov-west-1-Bundle.p7b

Den Inhalt Ihres CA-Zertifikats anzeigen

Verwenden Sie den folgenden Befehl, um den Inhalt Ihres CA-Zertifikatspakets zu überprüfen:

```
keytool -printcert -v -file global-bundle.pem
```

Rotieren Ihrer SSL/TLS-Zertifikate

Die Zertifikate der Amazon RDS Certificate Authority rds-ca-2019 sind im August 2024 abgelaufen. Wenn Sie Secure Sockets Layer (SSL) oder Transport Layer Security (TLS) mit Zertifikatsüberprüfung verwenden oder dies planen, um eine Verbindung zu Ihren RDS-DB-Instances oder Multi-AZ-DB-Clustern herzustellen, sollten Sie die Verwendung eines der neuen CA-Zertifikate rds-ca-rsa 2048-g1, 4096-g1 oder 384-g1 in Betracht ziehen. rds-ca-rsa rds-ca-ecc Wenn Sie derzeit keine Zertifikatsverifizierung verwendenSSL/TLS with certificate verification, you might still have an expired CA certificate and must update them to a new CA certificate if you plan to use SSL/TLS, um eine Verbindung zu Ihren RDS-Datenbanken herzustellen.

Amazon RDS bietet neue CA-Zertifikate als bewährte AWS Sicherheitsmethode. Informationen zu den neuen Zertifikaten und den unterstützten AWS Regionen finden Sie unter.

Verwenden Sie die folgenden Methoden, um das CA-Zertifikat für Ihre Datenbank zu aktualisieren:

- Aktualisierung Ihres CA-Zertifikats durch Änderung Ihrer DB-Instance oder Ihres Clusters
- Aktualisieren des CA-Zertifikats durch Anwenden der Wartung

Bevor Sie Ihre DB-Instances oder Multi-AZ-DB-Cluster für die Verwendung des neuen CA-Zertifikats aktualisieren, stellen Sie sicher, dass Sie Ihre Clients oder Anwendungen aktualisieren, die eine Verbindung zu Ihren RDS-Datenbanken herstellen.

Überlegungen zur Rotation von Zertifikaten

Beachten Sie die folgenden Situationen, bevor Sie Ihr Zertifikat rotieren:

- Amazon RDS Proxy Zertifikate von AWS Certificate Manager (ACM). Wenn Sie RDS Proxy verwenden, müssen Sie bei der Rotation Ihres SSL/TLS-Zertifikats keine Anwendungen aktualisieren, die RDS-Proxy-Verbindungen verwenden. Weitere Informationen finden Sie unter TLS/SSLmit RDS Proxy verwenden.
- Wenn Sie eine Go-Anwendung der Version 1.15 mit einer DB-Instance oder einem Multi-AZ-DB-Cluster verwenden, das vor dem 28. Juli 2020 erstellt oder auf das rds-ca-2019-Zertifikat aktualisiert wurde, müssen Sie das Zertifikat erneut aktualisieren.

  Verwenden Sie den `modify-db-instance` Befehl für eine DB-Instance oder den Befehl für einen Multi-AZ-DB-Cluster unter Verwendung der neuen `modify-db-cluster` CA-Zertifikats-ID. Mithilfe des `describe-db-engine-versions` Befehls können Sie herausfinden CAs , welche für eine bestimmte DB-Engine und DB-Engine-Version verfügbar sind.

Wenn Sie Ihre Datenbank nach dem 28. Juli 2020 erstellt oder ihr Zertifikat aktualisiert haben, sind keine Maßnahmen erforderlich. Weitere Informationen finden Sie in GitHub Go-Ausgabe #39568.

Aktualisierung Ihres CA-Zertifikats durch Änderung Ihrer DB-Instance oder Ihres Clusters

Im folgenden Beispiel wird Ihr CA-Zertifikat von rds-ca-2019 auf 2048-g1 aktualisiert. rds-ca-rsa Sie können ein anderes Zertifikat wählen. Weitere Informationen finden Sie unterZertifizierungsstellen.

Aktualisieren Sie Ihren Application Trust Store, um Ausfallzeiten im Zusammenhang mit der Aktualisierung Ihres CA-Zertifikats zu reduzieren. Weitere Informationen zu Neustarts im Zusammenhang mit der Rotation von CA-Zertifikaten finden Sie unterAutomatische Rotation von Serverzertifikaten.

So aktualisieren Sie Ihr CA-Zertifikat, indem Sie Ihre DB-Instance oder Ihren Cluster ändern

1. Laden Sie das neue SSL/TLS-Zertifikat herunter, wie unter beschrieben.
2. Aktualisieren Sie Ihre Anwendungen zur Verwendung der neuen SSL/TLS-Zertifikate.

   Die Methoden zur Aktualisierung von Anträgen auf neue SSL/TLS certificates depend on your specific applications. Work with your application developers to update the SSL/TLS Zertifikate für Ihre Anwendungen.

   Informationen zur Prüfung auf SSL/TLS-Verbindungen und die Aktualisierung von Anwendungen für jede DB_Engine finden Sie in den folgenden Themen:

   - Aktualisierung von Anwendungen für die Verbindung zu MariaDB-Instanzen mithilfe neuer /-Zertifikate SSL TLS
   - Aktualisierung von Anwendungen für die Verbindung mit Microsoft SQL Server-DB-Instances mithilfe neuer SSL TLS /-Zertifikate
   - Aktualisierung von Anwendungen für die Verbindung zu My SQL DB-Instances mithilfe neuer SSL TLS /-Zertifikate
   - Aktualisieren von Anwendungen, um Verbindungen mit Oracle-DB-Instances mithilfe neuer SSL/TLS-Zertifikate herzustellen
   - Aktualisierung von Anwendungen für die Verbindung zu SQL Postgre-DB-Instances mithilfe neuer SSL TLS /-Zertifikate

Ein Beispielskript, das einen Trust Store für ein Linux-Betriebssystem aktualisiert, finden Sie unterBeispielskript für den Import von Zertifikaten in Ihren Trust Store.

> **Note**
>
> Das Zertifikat-Bundle enthält Zertifikate für die neue und die alte CA, damit Sie Ihre Anwendung sicher aktualisieren und die Verbindung während der Übergangsphase aufrecht erhalten können. Wenn Sie das verwenden, um eine Datenbank AWS Database Migration Service zu einer DB-Instance oder einem zu migrieren, empfehlen wir die Verwendung des Zertifikatspakets, um die Konnektivität während der Migration sicherzustellen.

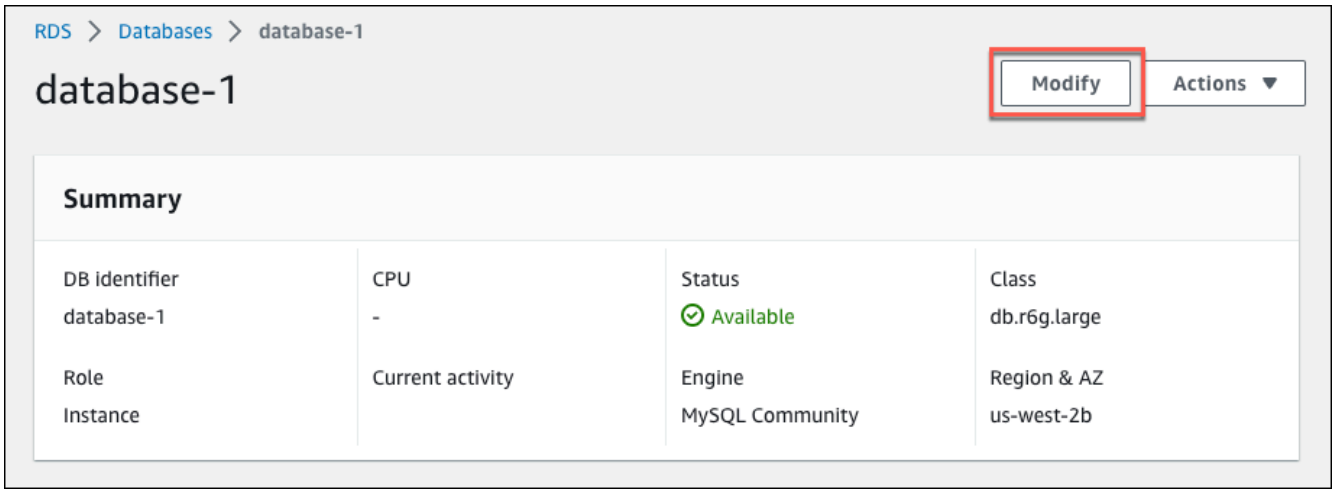
3. Ändern Sie die DB-Instance oder den Multi-AZ-DB-Cluster, um die CA von rds-ca-2019 auf 2048-g1 zu ändern. rds-ca-rsa Um zu überprüfen, ob Ihre Datenbank zum Aktualisieren der CA-Zertifikate neu gestartet werden muss, verwenden Sie den Befehl und überprüfen Sie das Flag. describe-db-engine-versions`SupportsCertificateRotationWithoutRestart`

   > **Important**
   >
   > Wenn nach dem Ablauf des Zertifikats Verbindungsprobleme auftreten, verwenden Sie die Option „Sofort anwenden“, indem Sie Sofort anwenden in der Konsole angeben oder die Option `--apply-immediately` mit der AWS CLI festlegen. Die Ausführung dieser Operation ist standardmäßig während Ihres nächsten Wartungsfensters eingeplant.
   > Für RDS for Oracle DB-Instances empfehlen wir, dass Sie Ihre Oracle-DB neu starten, um Verbindungsfehler zu vermeiden.
   > Bei RDS for SQL Server Multi-AZ-Instances mit AlwaysOn aktivierter Spiegelungsoption wird ein Failover erwartet, wenn die Instanz nach der Zertifikatsrotation neu gestartet wird.
   > Verwenden Sie den CLI-Befehl modify-certificates, um eine Überschreibung für Ihre festzulegen, die sich von der Standard-RDS-CA unterscheidet.

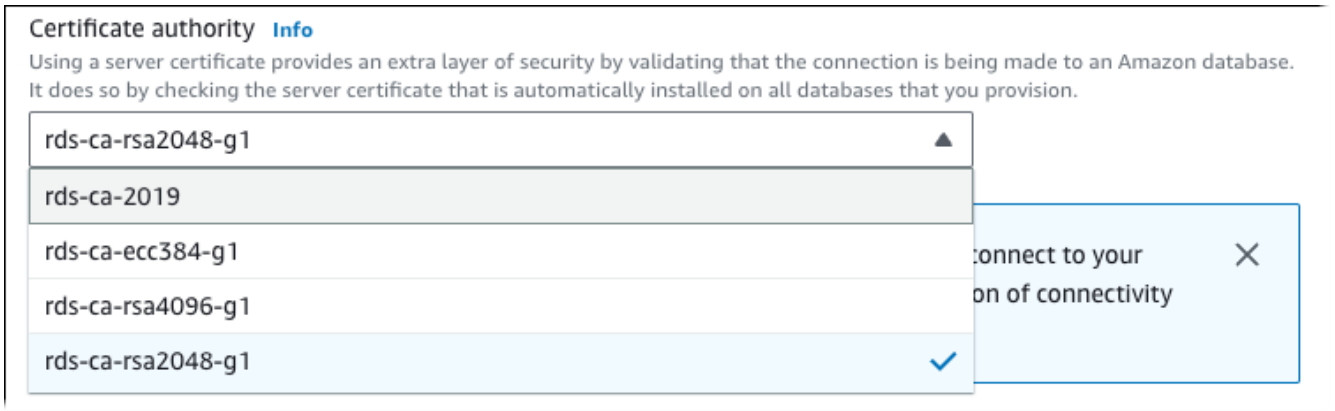
Sie können den AWS Management Console oder verwenden, AWS CLI um das CA-Zertifikat für eine DB-Instance oder einen Multi-AZ-DB-Cluster von rds-ca-2019 auf rds-ca-rsa2048-g1 zu ändern.

Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken und dann die DB-Instance oder den Multi-AZ-DB-Cluster aus, den Sie ändern möchten.
3. Wählen Sie Ändern aus.



4. Wählen Sie im Abschnitt Konnektivität die Option rds-ca-rsa2048-g1 aus.



5. Klicken Sie auf Weiter und überprüfen Sie die Zusammenfassung aller Änderungen.
6. Wählen Sie Sofort anwenden aus, um die Änderungen sofort anzuwenden.
7. Überprüfen Sie auf der Bestätigungsseite Ihre Änderungen. Wenn sie korrekt sind, wählen Sie Modify DB Instance oder Modify cluster, um Ihre Änderungen zu speichern.

> **Important**
>
> Wenn Sie diese Operation planen, stellen Sie sicher, dass Sie zuvor Ihren clientseitigen Vertrauensspeicher aktualisiert haben.

Oder klicken Sie auf Zurück, um Ihre Änderungen zu bearbeiten, oder auf Abbrechen, um Ihre Änderungen zu verwerfen.

AWS CLI

AWS CLI Um die CA für eine DB-Instance oder einen Multi-AZ-DB-Cluster von rds-ca-2019 auf rds-ca-rsa2048-g1 zu ändern, rufen Sie den Befehl or auf. modify-db-instancemodify-db-cluster Geben Sie die DB-Instance- oder Cluster-ID und die Option an. `--ca-certificate-identifier`

Verwenden Sie den `--apply-immediately` Parameter, um das Update sofort anzuwenden. Die Ausführung dieser Operation ist standardmäßig während Ihres nächsten Wartungsfensters eingeplant.

> **Important**
>
> Wenn Sie diese Operation planen, stellen Sie sicher, dass Sie zuvor Ihren clientseitigen Vertrauensspeicher aktualisiert haben.

Example

DB-Instance

Im folgenden Beispiel wird die Änderung vorgenommen, `mydbinstance` indem das CA-Zertifikat auf `rds-ca-rsa2048-g1` festgelegt wird.

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

> **Note**
>
> Wenn Ihre Instance neu gestartet werden muss, können Sie den modify-db-instanceCLI-Befehl verwenden und die `--no-certificate-rotation-restart` Option angeben.

Example

Multi-AZ-DB-Cluster

Im folgenden Beispiel wird die Änderung vorgenommen, `mydbcluster` indem das CA-Zertifikat auf festgelegt wird. `rds-ca-rsa2048-g1`

Für LinuxmacOS, oderUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

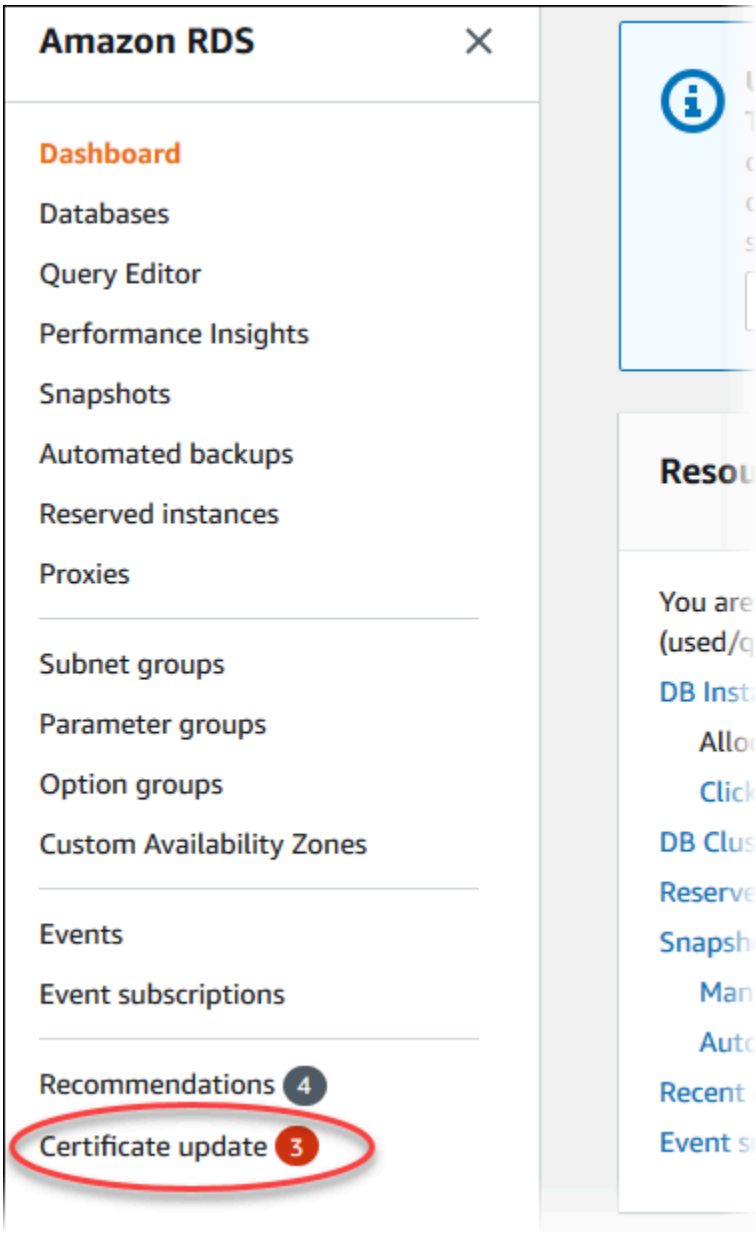
Aktualisieren des CA-Zertifikats durch Anwenden der Wartung

Führen Sie die folgenden Schritte aus, um Ihr CA-Zertifikat zu aktualisieren, indem Sie die Wartung durchführen.

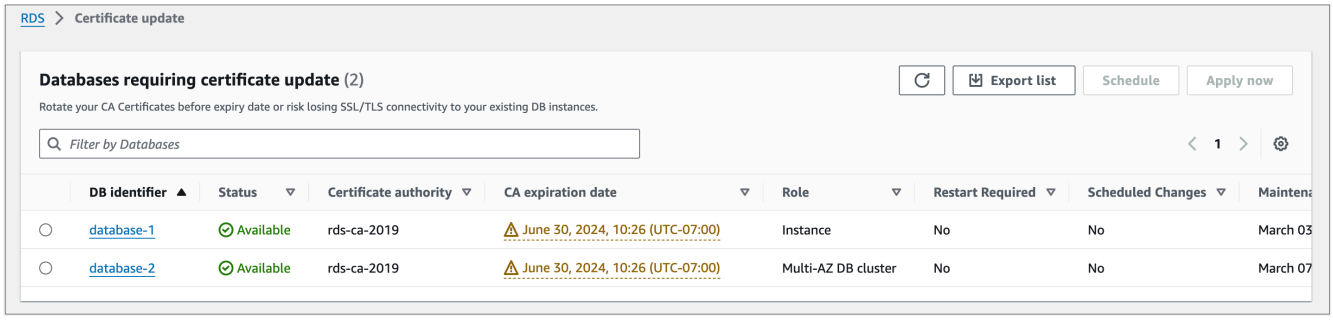
Console

Um Ihr CA-Zertifikat zu aktualisieren, indem Sie Wartungsarbeiten durchführen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich die Option Certificate update aus.



Die Seite Datenbanken, die eine Zertifikataktualisierung erfordern wird angezeigt.



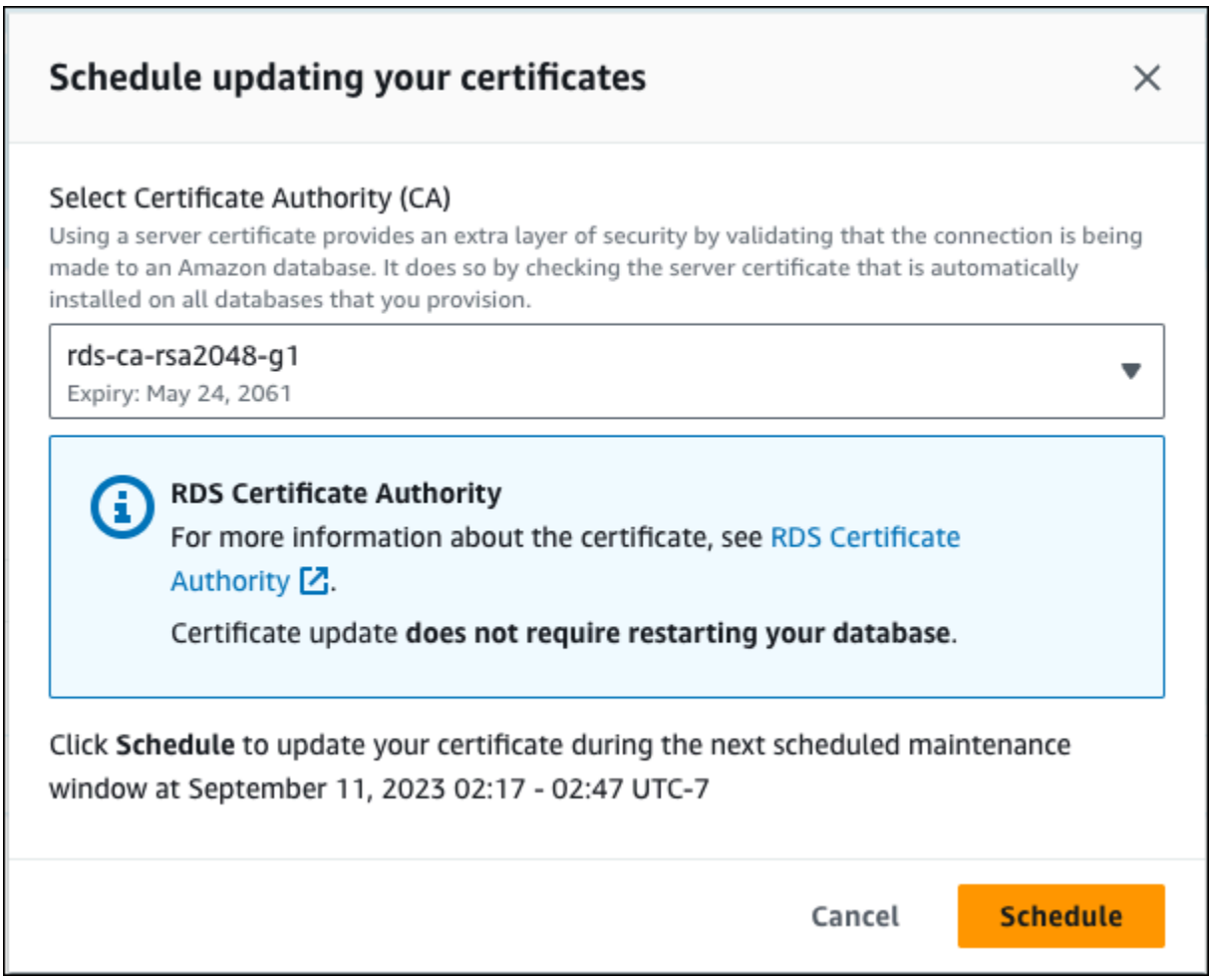
> **Note**
>
> Auf dieser Seite werden nur die aktuellen DB-Instances und -Cluster angezeigt AWS-Region. Wenn Sie Datenbanken in mehr als einer haben AWS-Region, überprüfen Sie diese Seite auf jeder Seite, AWS-Region um alle DB-Instances mit alten SSL/TLS-Zertifikaten zu sehen.

3. Wählen Sie die DB-Instance oder den Multi-AZ-DB-Cluster aus, den Sie aktualisieren möchten.

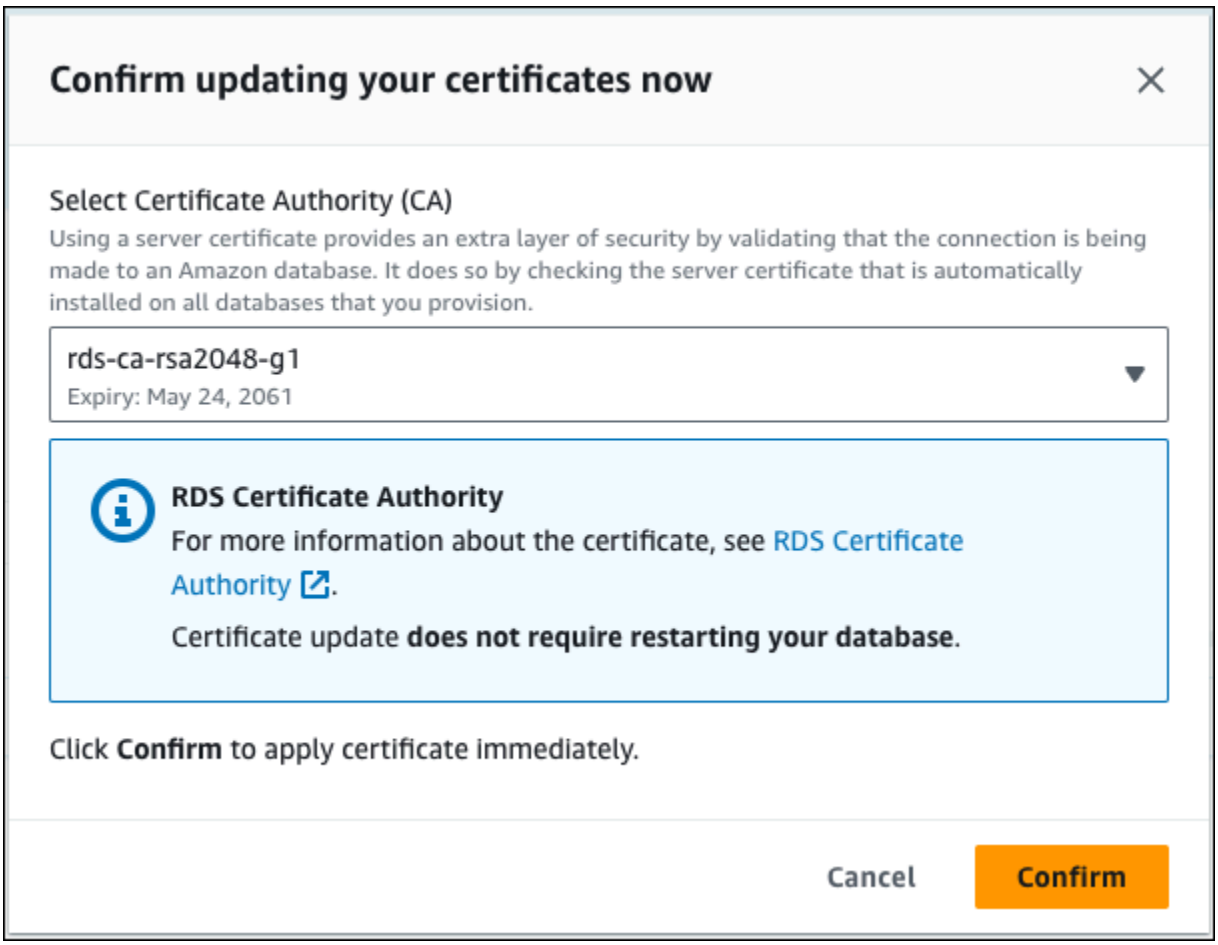
   Sie können die Zertifikatrotation für das nächste Wartungsfenster planen, indem Sie Zeitplan wählen. Wenden Sie die Rotation sofort an, indem Sie Jetzt anwenden wählen.

> **Important**
>
> Wenn nach Ablauf des Zertifikats Verbindungsprobleme auftreten, verwenden Sie die Option Jetzt anwenden.

4. a. Wenn Sie Zeitplan wählen, werden Sie aufgefordert, die Rotation der CA-Zertifikate zu bestätigen. In dieser Aufforderung wird auch das geplante Fenster für das Update angegeben.



b. Wenn Sie Jetzt anwenden wählen, werden Sie aufgefordert, die Rotation der CA-Zertifikate zu bestätigen.



> **Important**
>
> Bevor Sie die Rotation des CA-Zertifikats in der Datenbank planen, aktualisieren Sie alle Clientanwendungen, die SSL/TLS und das Serverzertifikat verwenden, um eine Verbindung herzustellen. Diese Updates sind spezifisch für Ihre DB-Engine. Nachdem Sie diese Clientanwendungen aktualisiert haben, können Sie die Rotation des CA-Zertifikats bestätigen.

Aktivieren Sie das Kontrollkästchen und klicken Sie dann auf Confirm (Bestätigen), um fortzufahren.

5. Wiederholen Sie die Schritte 3 und 4 für jede DB-Instance und jeden Cluster, den Sie aktualisieren möchten.

Automatische Rotation von Serverzertifikaten

Wenn Ihre Stammzertifizierungsstelle die automatische Rotation von Serverzertifikaten unterstützt, übernimmt RDS automatisch die Rotation des DB-Serverzertifikats. RDS verwendet dieselbe Stamm-CA für diese automatische Rotation, sodass Sie kein neues CA-Paket herunterladen müssen. SieheZertifizierungsstellen.

Die Rotation und Gültigkeit Ihres DB-Serverzertifikats hängen von Ihrer DB-Engine ab:

- Wenn Ihre DB-Engine die Rotation ohne Neustart unterstützt, rotiert RDS das DB-Serverzertifikat automatisch, ohne dass Sie etwas unternehmen müssen. RDS versucht, Ihr DB-Serverzertifikat in Ihrem bevorzugten Wartungsfenster nach der Halbwertszeit des DB-Serverzertifikats zu rotieren. Das neue DB-Serverzertifikat ist 12 Monate lang gültig.
- Wenn Ihre DB-Engine die Rotation ohne Neustart nicht unterstützt, benachrichtigt RDS Sie mindestens 6 Monate vor Ablauf des DB-Serverzertifikats über ein Wartungsereignis. Das neue DB-Serverzertifikat ist 36 Monate lang gültig.

Verwenden Sie den describe-db-engine-versionsBefehl und überprüfen Sie das `SupportsCertificateRotationWithoutRestart` Kennzeichen, um festzustellen, ob die DB-Engine-Version die Rotation des Zertifikats ohne Neustart unterstützt. Weitere Informationen finden Sie unter Einstellung der CA für Ihre Datenbank.

Beispielskript für den Import von Zertifikaten in Ihren Trust Store

Nachfolgend sind Beispiel-Shell-Skripte aufgeführt, die das Zertifikatspaket in einen Vertrauensspeicher importieren.

Jedes Beispiel-Shell-Skript verwendet keytool, das Teil des Java Development Kits (JDK) ist. Weitere Informationen über die Installation von JDK finden Sie im JDK-Installationshandbuch.

Linux

Nachfolgend finden Sie ein Beispiel-Shell-Skript, das das Zertifikatpaket in einen Trust Store auf einem Linux-Betriebssystem importiert.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
```

```
mkdir -p "${mydir}"
fi truststore=${mydir}/rds-truststore.jks storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem">
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
 {split_after=1}{print > "rds-ca-" n+1 ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do alias=$(openssl x509 -noout -text -in $CERT | perl -ne
 'next unless /Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias |
 cut -d " " -f3- | while read alias
do expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -
alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

macOS

Es folgt ein Beispiel für ein Shell-Skript, das das Zertifikatspaket in einen Vertrauensspeicher unter macOS importiert.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
mkdir -p "${mydir}"
fi truststore=${mydir}/rds-truststore.jks storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem">
 ${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-
```

```
for CERT in rds-ca-*; do alias=$(openssl x509 -noout -text -in $CERT | perl -ne
 'next unless /Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias |
 cut -d " " -f3- | while read alias
do expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -
alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Richtlinie für den Datenverkehr zwischen Netzwerken

Verbindungen sind sowohl zwischen Amazon RDS und lokalen Anwendungen als auch zwischen Amazon RDS und anderen AWS Ressourcen innerhalb derselben AWS Region geschützt.

Datenverkehr zwischen Service und On-Premises-Clients und -Anwendungen

Sie haben zwei Verbindungsoptionen zwischen Ihrem privaten Netzwerk und AWS:

- Eine AWS Site-to-Site VPN-Verbindung. Weitere Informationen finden Sie unter Was ist AWS Site-to-Site VPN?
- Eine AWS Direct Connect Verbindung. Weitere Informationen finden Sie unter Was ist AWS Direct Connect?

Sie erhalten über das Netzwerk Zugriff auf Amazon RDS , indem Sie AWS-veröffentlichte API-Operationen verwenden. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Identitäts- und Zugriffsmanagement für Amazon RDS

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um RDS Amazon-Ressourcen zu nutzen. IAMist eine AWS-Service , die Sie ohne zusätzliche Kosten verwenden können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- So funktioniert RDSAmazon mit IAM
- Beispiele für identitätsbasierte Amazon-RDS-Richtlinien
- AWS verwaltete Richtlinien für Amazon RDS
- Amazon RDS-Updates für AWS verwaltete Richtlinien
- Vermeidung des dienstübergreifenden Confused-Deputy-Problems
- IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL
- Fehlerbehebung für Amazon RDS-Identität und -Zugriff

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon RDS Amazon .

Servicebenutzer — Wenn Sie den Amazon RDS für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Amazon RDS verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf eine Funktion in Amazon RDS nicht zugreifen können, finden Sie weitere Informationen unterFehlerbehebung für Amazon RDS-Identität und -Zugriff.

Service-Administrator — Wenn Sie in Ihrem Unternehmen für die Amazon RDS verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon RDS . Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon RDS Ihre Mitarbeiter zugreifen sollen. Sie müssen anschließend bei Ihrem -Administrator entsprechende Änderungen für die Berechtigungen Ihrer Service-Benutzer anfordern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM zu verstehen. Weitere Informationen darüber, wie Ihr Unternehmen Amazon RDS nutzen IAM kann, finden Sie unterSo funktioniert RDSAmazon mit IAM.

Administrator — Wenn Sie ein Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon RDS zu verwalten. Beispiele für identitätsbasierte Amazon RDS , die Sie in verwenden könnenIAM, finden Sie unter. Beispiele für identitätsbasierte Amazon-RDS-Richtlinien

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAMIdentity Center-) Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAMBenutzerhandbuch unter AWS Signature Version 4 für API Anfragen.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS Empfiehlt beispielsweise, die Multi-

Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center Benutzerhandbuch und AWS Multi-Faktor-Authentifizierung IAM im IAM Benutzerhandbuch.

AWS Konto (Root-Benutzer)

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter Aufgaben, für die Root-Benutzeranmeldedaten erforderlich sind. IAM

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAMBenutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach

Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind.

Ein IAM-Gruppe ist eine Identität, die eine Sammlung von IAM.Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie im Benutzerhandbuch unter Anwendungsfälle für IAM IAM Benutzer.

Sie können sich mit der IAM-Datenbankauthentifizierung bei Ihrem DB-Instance- authentifizieren.

Die IAM-Datenbankauthentifizierung funktioniert mit den folgenden DB-Engines:

- RDS for MariaDB
- RDSfür mein SQL
- RDSfür Postgre SQL

Weitere Informationen zur Authentifizierung bei Ihrem mithilfe von finden Sie IAM unterIAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL.

IAM-Rollen

Eine IAMRolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist mit einem Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwendenURL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie unter Verwenden von IAM Rollen im IAMBenutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Temporäre Benutzerberechtigungen — Ein Benutzer kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle für einen externen Identitätsanbieter (Federation). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Kontoübergreifender Zugriff: Sie können eine IAM-Rolle verwenden, um jemandem (einem vertrauenswürdigen Prinzipal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie im Benutzerhandbuch unter Unterschiede zwischen IAM Rollen und ressourcenbasierten Richtlinien. IAM
- Serviceübergreifender Zugriff — Einige verwenden Funktionen in anderen. AWS-Services AWS-Services Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Zugriffssitzungen weiterleiten — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Wenn Sie einige Dienste verwenden, führen Sie möglicherweise eine Aktion aus, die dann eine weitere Aktion in einem anderen Dienst auslöst. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgeschaltete Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

  - Servicerolle — Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.
  - Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API Anfragen stellen AWS CLI . Das ist empfehlenswerter, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2 Instance eine AWS Rolle zuzuweisen und sie all ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Verwenden einer IAM Rolle, um Berechtigungen für Anwendungen zu erteilen, die auf EC2 Amazon-Instances ausgeführt werden.

Informationen darüber, ob IAM Rollen verwendet werden sollten, finden Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle (anstelle eines IAM Benutzers) erstellt werden.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff, AWS indem Sie Richtlinien erstellen und diese an IAM Identitäten oder AWS Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn eine Entität (Root-Benutzer, Benutzer oder IAM Rolle) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter Überblick über JSON Richtlinien.

Ein Administrator kann mithilfe von Richtlinien angeben, wer Zugriff auf AWS Ressourcen hat und welche Aktionen er mit diesen Ressourcen ausführen kann. Jede IAM Entität (Berechtigungssatz oder Rolle) beginnt ohne Berechtigungen. Anders ausgedrückt, können

Benutzer standardmäßig keine Aktionen ausführen und nicht einmal ihr Passwort ändern. Um einem Benutzer die Berechtigung für eine Aktion zu erteilen, muss ein Administrator einem Benutzer eine Berechtigungsrichtlinie zuweisen. Alternativ kann der Administrator den Benutzer zu einer Gruppe hinzufügen, die über die gewünschten Berechtigungen verfügt. Wenn ein Administrator einer Gruppe Berechtigungen erteilt, erhalten alle Benutzer in dieser Gruppe diese Berechtigungen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen Berechtigungssatz oder eine Rolle. Diese Richtlinien steuern, welche Aktionen diese Identität für welche Ressourcen und unter welchen Bedingungen ausführen kann. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter Erstellen von IAM Richtlinien im Benutzerhandbuch. IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Berechtigungssatz oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Berechtigungssätzen und Rollen in Ihrem AWS Konto zuordnen können. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAMBenutzerhandbuch unter Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien.

Informationen zu AWS verwalteten Richtlinien, die speziell für Amazon RDS Amazon , finden Sie unterAWS verwaltete Richtlinien für Amazon RDS.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität

(Berechtigungssatz oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die resultierenden Berechtigungen sind eine Schnittmenge der identitätsbasierten Richtlinien der Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Berechtigungssatz oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter Berechtigungsgrenzen für IAM Entitäten.

- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS Konten, die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter So SCPs arbeiten Sie im AWS Organizations Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Berechtigungssatzes oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im Benutzerhandbuch zu IAM.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter Bewertungslogik für Richtlinien.

So funktioniert RDSAmazon mit IAM

Bevor Sie Amazon Aurora verwendenIAM, um den Zugriff auf RDSAmazon zu verwalten, sollten Sie sich darüber im Klaren sein, welche IAM Funktionen für Amazon RDS verfügbar sind.

In der folgenden Tabelle sind IAM Funktionen aufgeführt, die Sie mit Amazon RDS Amazon verwenden können:

IAM-Feature	Amazon RDS Amazon
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
Attributbasierte Zugriffskontrolle (ABAC) (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Zugriffssitzungen weiterleiten	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Ja

Einen umfassenden Überblick darüber, wie Amazon RDS Amazon und andere AWS Dienste zusammenarbeitenIAM, finden Sie IAM im IAMBenutzerhandbuch unter AWS Dienste, die mit funktionieren.

Themen

- Identitätsbasierte Richtlinien von Amazon RDS
-
- Politische Maßnahmen für Amazon RDS
- Politische Ressourcen für Amazon RDS

- Schlüssel zu den Versicherungsbedingungen für Amazon RDS
- Zugriffskontrolllisten (ACLs) in Amazon RDS
-
- Temporäre Anmeldeinformationen mit Amazon RDS verwenden
- Zugriffssitzungen für Amazon RDS weiterleiten
- Servicerollen für Amazon RDS
-

Identitätsbasierte Richtlinien von Amazon RDS

Unterstützt identitätsbasierte Richtlinien: Ja.

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie im Benutzerhandbuch unter Definieren benutzerdefinierter IAM Berechtigungen mit vom Kunden verwalteten Richtlinien. IAM

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie in der Referenz zu IAM JSON Richtlinienelementen im IAMBenutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien von Amazon RDS finden Sie unter. Beispiele für identitätsbasierte Amazon-RDS-Richtlinien

Unterstützt ressourcenbasierte Richtlinien: Nein.

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche

Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie IAMim IAMBenutzerhandbuch unter Kontenübergreifender Ressourcenzugriff.

Politische Maßnahmen für Amazon RDS

Unterstützt politische Maßnahmen: Ja.

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur eine Genehmigung erforderlich ist und für die es keinen entsprechenden Vorgang gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in Amazon RDS verwenden das folgende Präfix vor der Aktion:`rds:`. Um beispielsweise jemandem die Erlaubnis zu erteilen, DB-Instances mit dem RDS `DescribeDBInstances` API Amazon-Vorgang zu beschreiben, nehmen Sie die `rds:DescribeDBInstances` Aktion in seine Richtlinie auf. Richtlinienanweisungen müssen entweder ein – `Action`oder ein `NotAction`-Element enthalten. Amazon RDS definiert eigene Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service ausführen können.

Um mehrere -Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie folgendermaßen durch Kommas.

```
"Action": [
      "rds:action1",
      "rds:action2"
```

Sie können auch Platzhalter (*) verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "rds:Describe*"
```

Eine Liste der Amazon RDS finden Sie unter Von Amazon definierte Aktionen RDS in der Service Authorization Reference.

Politische Ressourcen für Amazon RDS

Unterstützt politische Ressourcen: Ja.

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem Amazon-Ressourcennamen (ARN) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Die DB-Instance-Ressource hat den folgenden Amazon-Ressourcennamen (ARN).

```
arn:${Partition}:rds:${Region}:${Account}:{ResourceType}/${Resource}
```

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

Um beispielsweise die `dbtest` DB-Instance in Ihrem Kontoauszug anzugeben, verwenden Sie Folgendes. ARN

```
"Resource": "arn:aws:rds:us-west-2:123456789012:db:dbtest"
```

Wenn Sie alle DB-Instances angeben möchten, die einem bestimmten Konto angehören, verwenden Sie den Platzhalter (*).

```
"Resource": "arn:aws:rds:us-east-1:123456789012:db:*"
```

Einige RDS API Operationen, z. B. die zum Erstellen von Ressourcen, können für eine bestimmte Ressource nicht ausgeführt werden. Verwenden Sie in diesen Fällen den Platzhalter (*).

```
"Resource": "*"
```

Viele RDS API Amazon-Operationen beinhalten mehrere Ressourcen. `CreateDBInstance` erstellt beispielsweise eine DB-Instance. Sie können festlegen, dass ein -Benutzer beim Erstellen einer DB-Instance eine bestimmte Sicherheitsgruppe und Parametergruppe verwenden muss. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie ARNs durch Kommas.

```
"Resource": [
      "resource1",
      "resource2"
```

Eine Liste der Amazon RDS und ihrer Eigenschaften ARNs finden Sie unter Von Amazon definierte Ressourcen RDS in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter Von Amazon definierte Aktionen RDS.

Schlüssel zu den Versicherungsbedingungen für Amazon RDS

Unterstützt servicespezifische Richtlinien-Bedingungsschlüssel: Ja.

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinienelemente: Variablen und Tags.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontext-Schlüssel für AWS globale Bedingungen im IAMBenutzerhandbuch.

Amazon RDS definiert seinen eigenen Satz von Bedingungsschlüsseln und unterstützt auch die Verwendung einiger globaler Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontext-Schlüssel für AWS globale Bedingungen im IAMBenutzerhandbuch.

Alle RDS API Operationen unterstützen den `aws:RequestedRegion` Bedingungsschlüssel.

Eine Liste der Amazon RDS finden Sie unter Condition Keys for Amazon RDS in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon definierte Aktionen RDS.

Zugriffskontrolllisten (ACLs) in Amazon RDS

Unterstützt Zugriffskontrolllisten (ACLs): Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) Berechtigungen für den Zugriff auf eine Ressource haben. ACLsähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Unterstützt attributebasierte Zugriffssteuerungs-Tags (ABAC) in Richtlinien: Ja

Die attributebasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen auf der Grundlage von Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt vonABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABACist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen dazu finden Sie ABAC unter Definieren von Berechtigungen mit ABAC Autorisierung im IAMBenutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter Verwenden der attributebasierten Zugriffskontrolle (ABAC). IAM

Weitere Informationen zum Taggen von Amazon RDS finden Sie unterFestlegen von Bedingungen: Verwenden von benutzerdefinierten Tags. Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter Erteilen von Berechtigungen für Aktionen in einer Ressource mit einem bestimmten Tag und zwei verschiedenen Tag-Werten.

Temporäre Anmeldeinformationen mit Amazon RDS verwenden

Unterstützt temporäre Anmeldeinformationen: Ja.

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen darüber, AWS-Services wie

Sie mit temporären Anmeldeinformationen arbeiten können AWS-Services , finden Sie IAM im IAMBenutzerhandbuch unter Informationen zum Arbeiten mit.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Kennwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie im Benutzerhandbuch unter Von einem Benutzer zu einer IAM Rolle (Konsole) wechseln. IAM

Mit dem AWS CLI oder können Sie manuell temporäre Anmeldeinformationen erstellen AWS API. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen unter IAM.

Zugriffssitzungen für Amazon RDS weiterleiten

Unterstützt Forward-Access-Sitzungen: Ja.

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Wenn Sie einige Dienste verwenden, führen Sie möglicherweise eine Aktion aus, die dann eine weitere Aktion in einem anderen Dienst auslöst. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgeschaltete Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für Amazon RDS

Unterstützt Servicerollen: Ja.

Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Amazon RDS beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon RDS Sie dazu anleitet.

Unterstützt serviceverknüpfte Rollen: Ja.

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zur Verwendung von serviceverknüpften Amazon RDS finden Sie unterVerwenden von serviceverknüpften Rollen für Amazon RDS.

Beispiele für identitätsbasierte Amazon-RDS-Richtlinien

Standardmäßig besitzen Berechtigungssätze und Rollen keine Berechtigungen zum Erstellen oder Ändern von Amazon-RDS--Ressourcen. Sie können auch keine Aufgaben mit der AWS Management Console AWS CLI, oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Berechtigungssätzen und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den Berechtigungssätzen oder Rollen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Amazon RDS-Konsole
- Erforderliche Berechtigungen für die Verwendung der Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Berechtigungsrichtlinien zum Erstellen, Ändern und Löschen von Ressourcen in Amazon RDS

- Beispielrichtlinien: Verwenden von Bedingungsschlüsseln
- Festlegen von Bedingungen: Verwenden von benutzerdefinierten Tags
- Erteilen Sie die Erlaubnis, Amazon RDS während der Erstellung zu taggen

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien können festlegen, ob jemand Amazon-RDS-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder daraus löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue

und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.

- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon RDS-Konsole

Um auf die Amazon RDS-Konsole zuzugreifen, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon RDS Amazon in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die Sie ausführen möchten.

Um sicherzustellen, dass diese Entitäten weiterhin die Amazon RDS verwenden können, fügen Sie den Entitäten auch die folgende AWS verwaltete Richtlinie hinzu.

```
AmazonRDSReadOnlyAccess
```

Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Erforderliche Berechtigungen für die Verwendung der Konsole

Damit ein Benutzer mit der Konsole arbeiten kann, muss dieser Benutzer über einen Minimumsatz an Berechtigungen verfügen. Diese Berechtigungen ermöglichen es dem Benutzer, die Amazon RDS

Amazon für sein AWS Konto zu beschreiben und andere verwandte Informationen bereitzustellen, einschließlich EC2 Amazon-Sicherheits- und Netzwerkinformationen.

Wenn Sie eine IAM-Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Benutzer mit dieser IAM-Richtlinie. Um sicherzustellen, dass diese Benutzer die Konsole weiterhin verwenden können, fügen Sie dem Benutzer auch die verwaltete Richtlinie `AmazonRDSReadOnlyAccess` an. Einzelheiten dazu finden Sie unter Verwalten des Zugriffs mit Richtlinien.

Für Benutzer, die nur Aufrufe an die AWS CLI oder Amazon-RDS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen.

Die folgende Richtlinie gewährt vollen Zugriff auf alle Amazon RDS Amazon für das AWS Root-Konto:

```
AmazonRDSFullAccess
```

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
```

```
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Berechtigungsrichtlinien zum Erstellen, Ändern und Löschen von Ressourcen in Amazon RDS

Die folgenden Abschnitte enthalten Beispiele für Berechtigungsrichtlinien, die den Zugriff auf Ressourcen gewähren und einschränken:

Erlauben Sie einem Benutzer, DB-Instances in einem AWS Konto zu erstellen

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die es dem Konto mit der ID ermöglicht, DB-Instances für Ihr AWS Konto `123456789012` zu erstellen. Die Richtlinie setzt voraus, dass der Name der DB-Instance mit beginn `test`. Die neue DB-Instance muss auch die MySQL-Datenbank-Engine und die DB-Instance-Klasse `db.t2.micro` verwenden. Zusätzlich muss die neue DB-Instance eine Optionsgruppe und eine DB-Parametergruppe verwenden, die mit `default` beginnt und die Subnetzgruppe `default` verwendet.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowCreateDBInstanceOnly",
```

```
            "Effect": "Allow",
            "Action": [
               "rds:CreateDBInstance"
            ],
            "Resource": [
               "arn:aws:rds:*:123456789012:db:test*",
               "arn:aws:rds:*:123456789012:og:default*",
               "arn:aws:rds:*:123456789012:pg:default*",
               "arn:aws:rds:*:123456789012:subgrp:default"
            ],
            "Condition": {
               "StringEquals": {
                  "rds:DatabaseEngine": "mysql",
                  "rds:DatabaseClass": "db.t2.micro"
               }
            }
         }
      ]
}
```

Die Richtlinie ist ein einzelnes Statement, das die folgenden Berechtigungen für den -Benutzer bestimmt:

- Die Richtlinie ermöglicht es dem Konto, mithilfe des Vorgangs Create DBInstance API eine DB-Instance zu erstellen (dies gilt auch für den create-db-instance AWS CLI Befehl und den AWS Management Console).
- Das Element `Resource` gibt an, dass der Benutzer auf oder mit Ressourcen Aktionen ausführen kann. Sie geben Ressourcen über einen Amazon Resources Name (ARN) an. Dieser ARN umfasst den Namen des Dienstes, zu dem die Ressource gehört (`rds`), die AWS Region (*gibt in diesem Beispiel eine beliebige Region an), die AWS Kontonummer (123456789012ist in diesem Beispiel die Kontonummer) und den Ressourcentyp. Weitere Informationen zum Erstellen finden ARNs Sie unterAmazon-Ressourcennamen (ARNs) in Amazon RDS.

  Das `Resource`-Element im Beispiel gibt für den Benutzer die folgenden richtlinienbezogenen Einschränkungen für die Ressourcen an:

  - Die DB-Instance-Kennung für die neue DB-Instance muss mit `test` beginnen (zum Beispiel `testCustomerData1`, `test-region2-data`).
  - Die Optionsgruppe für die neue DB-Instance muss mit beginne `default`.
  - Die DB-Parametergruppe für die neue DB-Instance muss mit beginne `default`.

  - Die Subnetzgruppe für die neue DB-Instance muss mit `default` beginnen.
- Das `Condition`-Element gibt an, dass die DB-Engine MySQL sein muss und die DB-Instance-Klasse `db.t2.micro` sein muss. Das `Condition`-Element bestimmt die Bedingungen, wann eine Richtlinie wirksam sein soll. Sie können zusätzliche Berechtigungen oder Einschränkungen hinzufügen, indem Sie das `Condition`-Element verwenden. Weitere Informationen zur Angabe von Bedingungen finden Sie unter Schlüssel zu den Versicherungsbedingungen für Amazon RDS. Dieses Beispiel zeigt die Bedingungen `rds:DatabaseEngine` und `rds:DatabaseClass`. Informationen zu den gültigen Bedingungswerten für `rds:DatabaseEngine` finden Sie in der Liste unter dem `Engine` Parameter in Create DBInstance. Informationen zu den gültigen Bedingungswerten für `rds:DatabaseClass` finden Sie unter Unterstützte DB-Engines für DB-Instance-Klassen .

Das Element `Principal` ist in der Richtlinie nicht angegeben, da in identitätsbasierten Richtlinien die Angabe des Prinzipals als Empfänger der Berechtigung nicht erforderlich ist. Wenn Sie einem Benutzer eine Richtlinie zuweisen, ist der Benutzer automatisch der Prinzipal. Wird die Berechtigungsrichtlinie einer IAM-Rolle zugewiesen, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Um eine Liste von Amazon-RDS-Aktionen finden Sie unter Von Amazon RDS definierte Aktionen in der Service-Autorisierungs-Referenz

Einem Benutzer eine beliebige Beschreibungsaktion für eine beliebige RDS-Ressource erlauben

Die folgende Berechtigungsrichtlinie gewährt Berechtigungen für einen Benutzer, alle Aktionen auszuführen, die mit beginne `Describe`. Diese Aktionen zeigen Informationen zu einer RDS-Ressource, z. B. eine DB-Instance. Das Platzhalterzeichen (*) im `Resource`-Element zeigt an, dass die Aktionen für alle Amazon RDS-Ressourcen erlaubt sind, die dem Konto gehören.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowRDSDescribe",
         "Effect": "Allow",
         "Action": "rds:Describe*",
         "Resource": "*"
```

```
      }
   ]
}
```

Einem Benutzer erlauben, eine DB-Instance zu erstellen, die spezifische DB-Parametergruppe und Subnetzgruppe verwendet.

Die folgenden Berechtigungsrichtlinien erteilen einem Benutzer die Erlaubnis, ausschließlich eine DB-Instance mit der DB-Parametergruppe `mydbpg` und der DB-Subnetzgruppe `mydbsubnetgroup` zu erstellen.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": [
            "arn:aws:rds:*:*:pg:mydbpg",
            "arn:aws:rds:*:*:subgrp:mydbsubnetgroup"
         ]
      }
   ]
}
```

Erteilen von Berechtigungen für Aktionen in einer Ressource mit einem bestimmten Tag und zwei verschiedenen Tag-Werten

Sie können in Ihrer identitätsbasierten Richtlinie Bedingungen für die Steuerung des Zugriffs auf Amazon RDS-Ressourcen auf der Basis von Tags verwenden. Die folgende Richtlinie erteilt die Berechtigung, die API-Operation `CreateDBSnapshot` auf DB-Instances durchzuführen, bei denen das Tag `stage` entweder auf `development` oder `test` festgelegt ist.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Die folgende Richtlinie erteilt die Berechtigung, die API-Operation `ModifyDBInstance` auf DB-Instances durchzuführen, bei denen das Tag `stage` entweder auf `development` oder `test` festgelegt ist.

JSON

```
{
   "Version":"2012-10-17",
```

```
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Verhindern, dass ein Benutzer eine DB-Instance löscht

Die folgenden Berechtigungsrichtlinien erteilen Berechtigungen, um einen Benutzer davon abzuhalten, eine bestimmte DB-Instance zu löschen. Beispielsweise möchten Sie jedem Benutzer, der kein Administrator ist, verbieten, Ihre Produktions-DB-Instances zu löschen.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "DenyDelete1",
         "Effect": "Deny",
         "Action": "rds:DeleteDBInstance",
         "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"
      }
   ]
}
```

Verweigern des gesamten Zugriffs auf eine Ressource

Sie können den Zugriff auf eine Ressource explizit verweigern. Verweigerungsrichtlinien haben Vorrang vor Zulassungsrichtlinien. Die folgende Richtlinie verweigert einem Benutzer explizit die Möglichkeit, eine Ressource zu verwalten:

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": "rds:*",
         "Resource": "arn:aws:rds:us-east-1:123456789012:db:mydb"
      }
   ]
}
```

Beispielrichtlinien: Verwenden von Bedingungsschlüsseln

Im Folgenden finden Sie Beispiele für die Verwendung von Bedingungsschlüsseln in den IAM-Berechtigungsrichtlinien von Amazon RDS.

Beispiel 1: Erteilen der Berechtigung zum Erstellen einer DB-Instance mit einer bestimmten DB-Engine ohne Multi-AZ-Bereitstellung

Die folgende Richtlinie nutzt einen RDS-Bedingungsschlüssel und legt fest, dass ein Benutzer nur DB-Instances erstellen darf, die die MySQL-Datenbank-Engine und keine Multi-AZ-Bereitstellung verwenden. Das Element `Condition` gibt die Voraussetzung an, dass es sich um eine MySQL-Datenbank-Engine handeln muss.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowMySQLCreate",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql"
            },
            "Bool": {
               "rds:MultiAz": false
            }
         }
      }
   ]
}
```

Beispiel 2: Explizites Verweigern der Berechtigung, DB-Instances für bestimmte DB-Instance-Klassen sowie DB-Instances, die bereitgestellte IOPS verwenden, zu erstellen

Die folgende Richtlinie verweigert explizit die Berechtigung, DB-Instances für die DB-Instance-Klassen `r3.8xlarge` und `m4.10xlarge` zu erstellen, die zu den größten und teuersten DB-Instance-Klassen gehören. Diese Richtlinie hält Benutzer ebenfalls davon ab, DB-Instances zu erstellen, die bereitgestellte IOPS verwenden, durch die zusätzliche Kosten entstehen.

Eine explizit verweigerte Berechtigung überschreibt alle anderen erteilten Berechtigungen. So wird sichergestellt, dass Identitäten nicht aus Versehen eine Berechtigung erhalten, die Sie nie erteilen wollten.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "DenyLargeCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseClass": [
                  "db.r3.8xlarge",
                  "db.m4.10xlarge"
               ]
            }
         }
      },
      {
         "Sid": "DenyPIOPSCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "NumericNotEquals": {
               "rds:Piops": "0"
            }
         }
      }
   ]
}
```

Beispiel 3: Einschränken des Satzes von Tag-Schlüsseln und Werten, mit dem eine Ressource mit einem Tag versehen werden kann

Die folgende Richtlinie verwendet einen RDS-Bedingungsschlüssel und erlaubt es, ein Tag mit dem Schlüssel `stage` einer Ressource mit den Werten `test`, `qa` und `production` hinzuzufügen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTagEdits",
      "Effect": "Allow",
      "Action": [
        "rds:AddTagsToResource",
        "rds:RemoveTagsFromResource"
      ],
      "Resource": "arn:aws:rds:us-east-1:123456789012:db:db-123456",
      "Condition": {
        "StringEquals": {
          "rds:req-tag/stage": [
            "test",
            "qa",
            "production"
          ]
        }
      }
    }
  ]
}
```

Festlegen von Bedingungen: Verwenden von benutzerdefinierten Tags

Amazon RDS bietet Unterstützung für das Festlegen von Bedingungen in einer IAM-Richtlinie mithilfe von benutzerdefinierten Tags.

Angenommen, Sie fügen Ihren DB-Instances ein Tag namens `environment` mit Werten wie `beta`, `staging`, `production` und so weiter hinzu. Wenn dies der Fall ist, können Sie eine Richtlinie

erstellen, die bestimmte Benutzer basierend auf dem Tag-Wert `environment` auf DB-Instances beschränkt.

> **Note**
>
> Bei benutzerdefinierten Tag-Kennungen muss auf Groß- und Kleinschreibung geachtet werden.

In der folgenden Tabelle werden die RDS-Tag-Kennungen aufgeführt, die Sie in einem `Condition`-Element verwenden können.

RDS-Tag-ID	Gilt für
`db-tag`	DB-Instances einschließlich Lesereplikaten
`snapshot-tag`	DB-Snapshots
`ri-tag`	Reservierte DB-Instances
`og-tag`	DB-Optionsgruppen
`pg-tag`	DB-Parametergruppen
`subgrp-tag`	DB-Subnetzgruppen
`es-tag`	Ereignisabonnements
`cluster-tag`	DB-Cluster
`cluster-pg-tag`	DB-Cluster-Parametergruppen
`cluster-snapshot-tag`	DB-Cluster-Snapshots

Die Syntax für eine benutzerdefinierte Tag-Bedingung sieht wie folgt aus:

```
"Condition":{"StringEquals":{"rds:rds-tag-identifier/tag-name":
["value"]} }
```

Beispielsweise gilt das folgende `Condition`-Element für DB-Instances mit dem Tag `environment` und dem Tag-Wert `production`.

```
"Condition":{"StringEquals":{"rds:db-tag/environment": ["production"]} }
```

Weitere Informationen über die Erstellung von Tags finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.

Important

Wenn Sie den Zugriff auf Ihre RDS-Ressourcen mithilfe von Tags verwalten, empfehlen wir, den Zugriff auf die Tags Ihrer RDS-Ressourcen zu sichern. Sie können den Zugriff auf Tags verwalten, indem Sie Richtlinien für die Aktionen `AddTagsToResource` und `RemoveTagsFromResource` erstellen. Beispielsweise verweigert die folgende Richtlinie den Benutzern das Hinzufügen und Entfernen von Tags für alle Ressourcen. Sie können anschließend Richtlinien erstellen, die es bestimmten Benutzern ermöglichen, Tags hinzuzufügen oder zu entfernen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyTagUpdates",
         "Effect":"Deny",
         "Action":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"*"
      }
   ]
}
```

Um eine Liste von Amazon-RDS-Aktionen finden Sie unter Von Amazon RDS definierte Aktionen in der Service-Autorisierungs-Referenz

Beispielrichtlinien: Verwenden von benutzerdefinierten Tags

Im Folgenden finden Sie Beispiele für die Verwendung von benutzerdefinierten Tags in den IAM-Berechtigungsrichtlinien in Amazon RDS. Weitere Informationen zum Hinzufügen von Tags zu einer Amazon RDS-Ressource finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.

> **Note**
>
> Alle Beispiele verwenden die Region US-West-2 und enthalten ein fiktives Konto. IDs

Beispiel 1: Erteilen von Berechtigungen für Aktionen in einer Ressource mit einem bestimmten Tag und zwei verschiedenen Tag-Werten

Die folgende Richtlinie erteilt die Berechtigung, die API-Operation `CreateDBSnapshot` auf DB-Instances durchzuführen, bei denen das Tag `stage` entweder auf `development` oder `test` festgelegt ist.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
```

```
                "development",
                "test"
            ]
         }
      }
   ]
}
```

Die folgende Richtlinie erteilt die Berechtigung, die API-Operation `ModifyDBInstance` auf DB-Instances durchzuführen, bei denen das Tag `stage` entweder auf `development` oder `test` festgelegt ist.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
            ],
          "Resource": [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
            ]
       },
       {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
            ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
               "rds:db-tag/stage":[
                  "development",
```

```
                "test"
                ]
            }
        }
    }
  ]
}
```

Bespiel 2: Explizites Verweigern der Berechtigung zum Erstellen einer DB-Instance, die bestimmte DB-Parametergruppen verwendet

Die folgende Richtlinie verweigert explizit die Berechtigung, eine DB-Instance zu erstellen, die DB-Parametergruppen mit bestimmten Tag-Werten verwendet. Sie können diese Richtlinie anwenden, wenn stets eine bestimmte kundenseitig erstellte DB-Parametergruppe beim Erstellen von DB-Instances verwendet werden muss. Die Richtlinien mit `Deny` werden meist genutzt, um den von einer allgemeineren Richtlinie erteilten Zugriff einzuschränken.

Eine explizit verweigerte Berechtigung überschreibt alle anderen erteilten Berechtigungen. So wird sichergestellt, dass Identitäten nicht aus Versehen eine Berechtigung erhalten, die Sie nie erteilen wollten.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyProductionCreate",
         "Effect":"Deny",
         "Action":"rds:CreateDBInstance",
         "Resource":"arn:aws:rds:*:123456789012:pg:*",
         "Condition":{
            "StringEquals":{
               "rds:pg-tag/usage":"prod"
            }
         }
      }
   ]
}
```

Beispiel 3: Erteilen von Berechtigungen für Aktionen auf einer DB-Instance mit einem Instance-Namen, der den Benutzernamen als Präfix enthält

Die folgende Richtlinie erteilt die Berechtigung, eine beliebige API (mit Ausnahme von `AddTagsToResource` oder `RemoveTagsFromResource`) auf einer DB-Instance aufzurufen, deren Instance-Name den Benutzernamen als Präfix aufweist und das Tag `stage` mit dem Wert `devo` oder kein Tag `stage` enthält.

Die Zeile `Resource` in der Richtlinie kennzeichnet eine Ressource durch den Amazon-Ressourcennamen (ARN). Weitere Informationen zur Verwendung von Amazon RDS finden Sie unterAmazon-Ressourcennamen (ARNs) in Amazon RDS.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowFullDevAccessNoTags",
         "Effect":"Allow",
         "NotAction":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:${aws:username}*",
         "Condition":{
            "StringEqualsIfExists":{
               "rds:db-tag/stage":"devo"
            }
         }
      }
   ]
}
```

Erteilen Sie die Erlaubnis, Amazon RDS während der Erstellung zu taggen

Bei einigen RDS-API-Vorgängen können Sie Tags angeben, wenn Sie die Ressource erstellen. Sie können Resource-Tags (Markierungen) verwenden, um eine attributbasierte Steuerung (ABAC) zu implementieren. Weitere Informationen finden Sie unter Wozu dient ABAC? AWS und Steuern des Zugriffs auf AWS Ressourcen mithilfe von Tags.

Damit Benutzer Ressourcen bei der Erstellung taggen können, müssen sie über Berechtigungen verfügen, um die Aktion zu verwenden, mit der die Ressource erstellt wird, z. `rds:CreateDBInstance`B. Wenn in der Aktion „Erstellen“ Tags angegeben werden, führt RDS eine zusätzliche Autorisierung für die `rds:AddTagsToResource` Aktion durch, um zu überprüfen, ob Benutzer berechtigt sind, Tags zu erstellen. Daher benötigen die Benutzer außerdem die expliziten Berechtigungen zum Verwenden der `rds:AddTagsToResource`-Aktion.

In der IAM-Richtliniendefinition für die `rds:AddTagsToResource` Aktion können Sie den `aws:RequestTag` Bedingungsschlüssel verwenden, um Tags in einer Anfrage zur Kennzeichnung einer Ressource vorzuschreiben.

Die folgende Richtlinie ermöglicht es Benutzern beispielsweise, DB-Instances zu erstellen und Tags während der DB-Instance-Erstellung anzuwenden, jedoch nur mit bestimmten Tag-Schlüsseln (`environment`oder`project`):

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "rds:CreateDBInstance"
           ],
           "Resource": "*"
       },
       {
           "Effect": "Allow",
           "Action": [
               "rds:AddTagsToResource"
           ],
           "Resource": "*",
           "Condition": {
               "StringEquals": {
                   "aws:RequestTag/environment": ["production", "development"],
                   "aws:RequestTag/project": ["dataanalytics", "webapp"]
               },
               "ForAllValues:StringEquals": {
                   "aws:TagKeys": ["environment", "project"]
               }
```

```
            }
        }
    ]
}
```

Diese Richtlinie lehnt jede Anfrage zum Erstellen einer DB-Instance ab, die andere Tags als die `project` Tags `environment` oder enthält oder keines dieser Tags spezifiziert. Darüber hinaus müssen Benutzer Werte für die Tags angeben, die den zulässigen Werten in der Richtlinie entsprechen.

Die folgende Richtlinie ermöglicht es Benutzern, DB-Cluster zu erstellen und bei der Erstellung alle Tags außer dem `environment=prod` Tag anzuwenden:

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "rds:CreateDBCluster"
           ],
           "Resource": "*"
       },
       {
           "Effect": "Allow",
           "Action": [
               "rds:AddTagsToResource"
           ],
           "Resource": "*",
           "Condition": {
               "StringNotEquals": {
                   "aws:RequestTag/environment": "prod"
               }
           }
       }
   ]
}
```

Unterstützte RDS-API-Aktionen für das Tagging bei der Erstellung

Die folgenden RDS-API-Aktionen unterstützen das Tagging, wenn Sie eine Ressource erstellen. Für diese Aktionen können Sie beim Erstellen der Ressource Tags angeben:

- `CreateBlueGreenDeployment`
- `CreateCustomDBEngineVersion`
- `CreateDBCluster`
- `CreateDBClusterEndpoint`
- `CreateDBClusterParameterGroup`
- `CreateDBClusterSnapshot`
- `CreateDBInstance`
- `CreateDBInstanceReadReplica`
- `CreateDBParameterGroup`
- `CreateDBProxy`
- `CreateDBProxyEndpoint`
- `CreateDBSecurityGroup`
- `CreateDBShardGroup`
- `CreateDBSnapshot`
- `CreateDBSubnetGroup`
- `CreateEventSubscription`
- `CreateGlobalCluster`
- `CreateIntegration`
- `CreateOptionGroup`
- `CreateTenantDatabase`
- `CopyDBClusterParameterGroup`
- `CopyDBClusterSnapshot`
- `CopyDBParameterGroup`
- `CopyDBSnapshot`
- `CopyOptionGroup`
- `RestoreDBClusterFromS3`
- `RestoreDBClusterFromSnapshot`

- `RestoreDBClusterToPointInTime`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`
- `PurchaseReservedDBInstancesOffering`

Wenn Sie die API AWS CLI oder verwenden, um eine Ressource mit Tags zu erstellen, wird der `Tags` Parameter verwendet, um während der Erstellung Tags auf Ressourcen anzuwenden.

Wenn bei diesen API-Aktionen das Taggen fehlschlägt, wird die Ressource nicht erstellt und die Anfrage schlägt mit einem Fehler fehl. Dadurch wird sichergestellt, dass Ressourcen entweder mit Tags oder gar nicht erstellt werden, wodurch verhindert wird, dass Ressourcen ohne die vorgesehenen Tags erstellt werden.

AWS verwaltete Richtlinien für Amazon RDS

Um Berechtigungen zu Berechtigungssätzen und Rollen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS-Services verwalten und aktualisieren Sie AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Dienste fügen einer AWS verwalteten Richtlinie gelegentlich zusätzliche Berechtigungen hinzu, um neue Funktionen zu unterstützen. Diese Art von Update betrifft alle Identitäten (Berechtigungssätze und Rollen), denen die Richtlinie angehängt ist. Es ist sehr wahrscheinlich, dass Dienste eine AWS verwaltete Richtlinie aktualisieren, wenn eine neue Funktion eingeführt wird oder wenn neue Operationen verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die `ReadOnlyAccess` AWS verwaltete Richtlinie bietet beispielsweise nur Lesezugriff auf alle Ressourcen AWS-Services . Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS -Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

Themen

- AWS verwaltete Richtlinie: Amazon RDSRead OnlyAccess
- AWS verwaltete Richtlinie: Amazon RDSFull Access
- AWS verwaltete Richtlinie: Amazon RDSData FullAccess
- AWS verwaltete Richtlinie: Amazon RDSEnhanced MonitoringRole
- AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsReadOnly
- AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsFullAccess
- AWS verwaltete Richtlinie: Amazon RDSDirectory ServiceAccess
- AWS verwaltete Richtlinie: Amazon RDSService RolePolicy

- AWS verwaltete Richtlinie: Amazon RDSCustom ServiceRolePolicy
- AWS verwaltete Richtlinie: Amazon RDSCustom Instance ProfileRolePolicy
- AWS verwaltete Richtlinie: Amazon RDSPreview ServiceRolePolicy
- AWS verwaltete Richtlinie: Amazon RDSBeta ServiceRolePolicy

AWS verwaltete Richtlinie: Amazon RDSRead OnlyAccess

Diese Richtlinie ermöglicht den schreibgeschützten Zugriff auf Amazon RDS über die. AWS Management Console

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `rds` – Ermöglicht es Prinzipalen, Amazon-RDS-Ressourcen zu beschreiben und die Tags für Amazon-RDS-Ressourcen aufzulisten.
- `cloudwatch`— Ermöglicht Prinzipalen das Abrufen von CloudWatch Amazon-Metrikstatistiken.
- `ec2` – Ermöglicht es Prinzipalen, Availability Zones und Netzwerkressourcen zu beschreiben.
- `logs`— Ermöglicht Prinzipalen, Logs, CloudWatch Log-Streams von Log-Gruppen zu beschreiben und CloudWatch Log-Log-Ereignisse abzurufen.
- `devops-guru`— Ermöglicht Prinzipalen die Beschreibung von Ressourcen, die von Amazon DevOps Guru abgedeckt werden, was entweder durch CloudFormation Stacknamen oder Ressourcen-Tags spezifiziert wird.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSRead OnlyAccess im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSFull Access

Diese Richtlinie bietet vollen Zugriff auf Amazon RDS über die AWS Management Console.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `rds` – Ermöglicht Prinzipalen Vollzugriff auf alle Amazon RDS.

- `application-autoscaling` – Ermöglicht es Prinzipalen, Ziele und Richtlinien zur Skalierung der automatischen Anwendungsskalierung zu beschreiben und zu verwalten.
- `cloudwatch`— Ermöglicht es Schulleitern, CloudWatch metrische Statistiken abzurufen und Alarme zu verwalten CloudWatch .
- `ec2` – Ermöglicht es Prinzipalen, Availability Zones und Netzwerkressourcen zu beschreiben.
- `logs`— Ermöglicht Prinzipalen die Beschreibung von Logs, CloudWatch Log-Streams von Log-Gruppen und das Abrufen von CloudWatch Log-Log-Ereignissen.
- `outposts`— Ermöglicht Prinzipalen das Abrufen von AWS Outposts Instanztypen.
- `pi` – Ermöglicht es Prinzipalen, Performance-Insights-Metriken abzurufen.
- `sns` – Ermöglicht es Prinzipalen, Amazon Simple Notification Service (Amazon SNS)-Abonnements und -Themen zu abonnieren und Amazon-SNS-Nachrichten zu veröffentlichen.
- `devops-guru`— Ermöglicht Prinzipalen die Beschreibung von Ressourcen, die von Amazon DevOps Guru abgedeckt werden, was entweder durch CloudFormation Stacknamen oder Ressourcen-Tags spezifiziert wird.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie unter Amazon RDSFull Access im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSData FullAccess

Diese Richtlinie ermöglicht den vollen Zugriff auf die Nutzung der Daten-API und des Abfrage-Editors für Aurora Serverless Cluster in einem bestimmten Bereich AWS-Konto. Diese Richtlinie ermöglicht es AWS-Konto , den Wert eines Geheimnisses von abzurufen AWS Secrets Manager.

Sie können die `AmazonRDSDataFullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `dbqms` – Ermöglicht es Prinzipalen, auf Abfragen zuzugreifen, Abfragen zu erstellen, zu löschen, zu beschreiben und zu aktualisieren. Der Database Query Metadata Service (`dbqms`) ist ein reiner Dienst für interne Daten. Es bietet Ihre letzten und gespeicherten Abfragen für den Abfrage-Editor auf dem AWS Management Console für mehrere AWS-Services, einschließlich Amazon RDS.
- `rds-data` – Ermöglicht es Prinzipalen, SQL-Anweisungen in Aurora Serverless-Datenbanken auszuführen.

- `secretsmanager`— Ermöglicht es Prinzipalen, den Wert eines Geheimnisses von AWS Secrets Manager abzurufen.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSData FullAccess im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSEnhanced MonitoringRole

Diese Richtlinie bietet Zugriff auf Amazon CloudWatch Logs for Amazon RDS Enhanced Monitoring.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `logs`— Ermöglicht es Prinzipalen, CloudWatch Protokollgruppen und Aufbewahrungsrichtlinien zu erstellen und Log-Log-Streams von CloudWatch Protokollgruppen zu erstellen und zu beschreiben. Es ermöglicht Prinzipalen auch, Logs und CloudWatch Log-Ereignisse zu speichern und abzurufen.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSEnhanced MonitoringRole im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsReadOnly

Diese Richtlinie bietet schreibgeschützten Zugriff auf Amazon RDS Performance Insights für Amazon-RDS-DB-Instances und Amazon-Aurora-DB-Cluster.

Die Richtlinie enthält jetzt `Sid` (Anweisungs-ID) als Bezeichner für die Richtlinienanweisung.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `rds` – Ermöglicht es Prinzipalen, Amazon-RDS-DB-Instances und Amazon-Aurora-DB-Cluster zu beschreiben.
- `pi` – Ermöglicht es Prinzipalen, Aufrufe an die Amazon-RDS-Performance-Insights-API zu tätigen und auf Performance-Insights-Metriken zuzugreifen

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSPerformance InsightsReadOnly im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsFullAccess

Diese Richtlinie bietet Vollzugriff auf Erkenntnisse zur Amazon-RDS-Leistung für DB-Instances von Amazon RDS und DB-Clustern von Amazon Aurora.

Die Richtlinie enthält jetzt `Sid` (Anweisungs-ID) als Bezeichner für die Richtlinienanweisung.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `rds` – Ermöglicht es Prinzipalen, Amazon-RDS-DB-Instances und Amazon-Aurora-DB-Cluster zu beschreiben.
- `pi` – Ermöglicht es Prinzipalen, die API von Erkenntnissen zur Amazon-RDS-Leistung aufzurufen und Leistungsanalyseberichte zu erstellen, anzusehen und zu löschen.
- `cloudwatch`— Ermöglicht Prinzipalen, alle CloudWatch Amazon-Metriken aufzulisten und Metrikdaten und Statistiken abzurufen.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSPerformance InsightsFullAccess im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSDirectory ServiceAccess

Diese Richtlinie ermöglicht es Amazon RDS, Aufrufe an AWS Directory Service zu tätigen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgende Berechtigung:

- `ds`— Ermöglicht es Prinzipalen, AWS Directory Service Verzeichnisse zu beschreiben und die Autorisierung von AWS Directory Service Verzeichnissen zu kontrollieren.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSDirectory ServiceAccess im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSService RolePolicy

Sie können die `AmazonRDSServiceRolePolicy`-Richtlinie Ihren IAM-Entitäten nicht anfügen. Diese Richtlinie ist mit einer servicegebundenen Rolle verknüpft, die es Amazon RDS ermöglicht,

Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Berechtigungen von serviceverknüpften Rollen für Amazon RDS.

AWS verwaltete Richtlinie: Amazon RDSCustom ServiceRolePolicy

Sie können die `AmazonRDSCustomServiceRolePolicy`-Richtlinie Ihren IAM-Entitäten nicht anfügen. Diese Richtlinie ist mit einer servicebezogenen Rolle verknüpft, die es Amazon RDS ermöglicht, AWS Dienste im Namen Ihrer RDS-DB-Ressourcen aufzurufen.

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `ec2`- Ermöglicht RDS Custom, Backup-Operationen auf der DB-Instance durchzuführen, die point-in-time Wiederherstellungsfunktionen bietet.
- `secretsmanager`- Ermöglicht RDS Custom die Verwaltung von DB-Instance-spezifischen Geheimnissen, die von RDS Custom erstellt wurden.
- `cloudwatch`- Ermöglicht RDS Custom das Hochladen von Metriken und Protokollen der DB-Instance CloudWatch über den CloudWatch Agenten.
- `events`, `sqs` - Ermöglicht RDS Custom das Senden und Empfangen von Statusinformationen über die DB-Instance.
- `cloudtrail`- Ermöglicht RDS Custom, Änderungsereignisse über die DB-Instance zu empfangen
- `servicequotas`- Ermöglicht RDS Custom das Lesen von Dienstkontingenten, die sich auf die DB-Instance beziehen
- `ssm`- Ermöglicht RDS Custom, die der DB-Instance zugrunde liegende EC2 Instance zu verwalten.
- `rds`- Ermöglicht RDS Custom, RDS-Ressourcen für Ihre DB-Instance zu verwalten
- `iam`- Ermöglicht RDS Custom, das Instance-Profil zu validieren und der zugrunde liegenden Instance einer EC2 DB-Instance zuzuordnen.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSCustom ServiceRolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSCustom Instance ProfileRolePolicy

Sie sollten `AmazonRDSCustomInstanceProfileRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Es sollte nur an eine Instance-Profilrolle angehängt werden, die verwendet wird, um Ihrer Amazon RDS Custom DB-Instance Berechtigungen zur Ausführung verschiedener Automatisierungsaktionen und Datenbankverwaltungsaufgaben zu erteilen. Übergeben Sie das

Instance-Profil während der Erstellung der benutzerdefinierten RDS-Instance als `custom-iam-instance-profile` Parameter, und RDS Custom ordnet dieses Instance-Profil Ihrer DB-Instance zu.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `ssm`,`ssmmessages`, `ec2messages` - Ermöglicht RDS Custom die Kommunikation, Ausführung der Automatisierung und Wartung von Agenten auf der DB-Instance über Systems Manager.
- `ec2`, `s3` - Ermöglicht RDS Custom die Durchführung von Backup-Vorgängen auf der DB-Instance, die point-in-time Wiederherstellungsfunktionen bietet.
- `secretsmanager`- Ermöglicht RDS Custom die Verwaltung von DB-Instance-spezifischen Geheimnissen, die von RDS Custom erstellt wurden.
- `cloudwatch`, `logs` - Ermöglicht RDS Custom das Hochladen von DB-Instance-Metriken und Protokollen CloudWatch über den CloudWatch Agenten.
- `events`, `sqs` - Ermöglicht RDS Custom das Senden und Empfangen von Statusinformationen über die DB-Instance.
- `kms`- Ermöglicht RDS Custom die Verwendung eines instanzspezifischen KMS-Schlüssels zur Verschlüsselung von Geheimnissen und S3-Objekten, die von RDS Custom verwaltet werden.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie unter Amazon RDSCustom Instance ProfileRolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSPreview ServiceRolePolicy

Sie sollten `AmazonRDSPreviewServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer servicebezogenen Rolle verknüpft, die es Amazon RDS ermöglicht, AWS Dienste im Namen Ihrer RDS-DB-Ressourcen aufzurufen. Weitere Informationen finden Sie unter Servicebezogene Rolle für Amazon RDS Preview.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `ec2`- Ermöglicht Principals, Availability Zones und Netzwerkressourcen zu beschreiben.
- `secretsmanager`— Ermöglicht es Prinzipalen, den Wert eines Geheimnisses von zu ermitteln. AWS Secrets Manager

- `cloudwatch`, `logs` - Ermöglicht Amazon RDS das Hochladen von DB-Instance-Metriken und Protokollen CloudWatch über den CloudWatch Agenten.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSPreview ServiceRolePolicy im AWS Managed Policy Reference Guide.

AWS verwaltete Richtlinie: Amazon RDSBeta ServiceRolePolicy

Sie sollten `AmazonRDSBetaServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer servicebezogenen Rolle verknüpft, die es Amazon RDS ermöglicht, AWS Dienste im Namen Ihrer RDS-DB-Ressourcen aufzurufen. Weitere Informationen finden Sie unter Servicebezogene Rollenberechtigungen für Amazon RDS Beta.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `ec2`- Ermöglicht Amazon RDS, Backup-Operationen auf der DB-Instance durchzuführen, die point-in-time Wiederherstellungsfunktionen bietet.
- `secretsmanager`- Ermöglicht Amazon RDS die Verwaltung von DB-Instance-spezifischen Geheimnissen, die von Amazon RDS erstellt wurden.
- `cloudwatch`, `logs` - Ermöglicht Amazon RDS das Hochladen von DB-Instance-Metriken und Protokollen CloudWatch über den CloudWatch Agenten.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSBeta ServiceRolePolicy im AWS Managed Policy Reference Guide.

Amazon RDS-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon RDS an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Um automatische Warnungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Amazon-RDS-Dokumentverlauf-Seite.

Änderung	Beschreibung	Datum
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat die Berechtigungen für die `AmazonRDSCustomServiceRolePolicy` `AWSServiceRoleForRDSCustom` serviceverknüpfte Rolle aktualisiert. Das Update `ec2:CopySnapshot` entfernt eine Anweisung und fügt zwei neue Anweisungen für Quell- und Ziel-Snapshot-Berechtigungen hinzu. Diese Updates entsprechen einer Änderung des CopySnapshot EBS-Autorisierungsverhaltens, wobei die effektiven Berechtigungen unverändert bleiben. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	7. August 2025
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle	25. März 2025

Änderung	Beschreibung	Datum
	`AWSServiceRoleForRDSCustom` hinzugefügt. Diese Berechtigungen ermöglichen RDS Custom die Verwaltung von EC2 Schlüsselpaaren und ermöglichen RDS Custom die Integration mit Amazon SQS. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	
AWS verwaltete Richtlinie: Amazon RDSCustom InstanceProfileRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der verwalteten Richtlinie neue Berechtigungen hinzugefügt`AmazonRDSCustomInstanceProfileRolePolicy`, um die Verwendung von RDS Custom Managed Secrets auf einer RDS Custom-Instance zu ermöglichen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSCustom InstanceProfileRolePolicy.	20. März 2025

Änderung	Beschreibung	Datum
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neuen Berechtigungen ermöglichen es RDS Custom, Secrets Manager aufzulisten und wiederherzustellen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	6. März 2025
AWS verwaltete Richtlinie: Amazon RDSPreviewServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSPreviewServiceRolePolicy` `AWSServiceRoleForRDSPreview` serviceverknüpften Rolle die `sns:Publish` Berechtigung entzogen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSPreview ServiceRolePolicy.	7. August 2024

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinie: Amazon RDSBeta ServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSBetaServiceRolePolicy` `AWSServiceRoleForRDSBeta` serviceverknüpften Rolle die `sns:Publish` Berechtigung entzogen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSBeta ServiceRolePolicy.	7. August 2024
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Die Berechtigungen ermöglichen es RDS Custom, mit Amazon RDS-Diensten in einem anderen zu kommunizieren AWS-Region und EC2 Bilder zu kopieren. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	18. Juli 2024

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinie: Amazon RDSService RolePolicy – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSServiceRolePolicy` `AWSServiceRoleForRDS` serviceverknüpften Rolle die `sns:Publish` Berechtigung entzogen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSService RolePolicy.	2. Juli 2024
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neue Berechtigung ermöglicht es RDS Custom, einer benutzerdefinierten RDS-Instance eine Servicerolle als Instanzprofil zuzuordnen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	19. April 2024

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinien für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSCustomServiceRolePolicy` `AWSServiceRoleForRDSCustom` serviceverknüpften Rolle eine neue Berechtigung hinzugefügt, die es RDS Custom for SQL Server ermöglicht, den zugrunde liegenden Datenbank-Host-Instance-Typ zu ändern. RDS hat außerdem die `ec2:DescribeInstanceTypes` Berechtigung hinzugefügt, Informationen zum Instance-Typ für den Datenbank-Host abzurufen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinien für Amazon RDS.	8. April 2024
AWS verwaltete Richtlinien für Amazon RDS – Neue Richtlinie	Amazon RDS hat eine neue verwaltete Richtlinie hinzugefügt`AmazonRDSCustomInstanceProfileRolePolicy` , mit der RDS Custom Automatisierungsaktionen und Datenbankverwaltungsaufgaben über ein EC2 Instance-Profil ausführen kann. Weitere Informationen finden Sie unter AWS verwaltete Richtlinien für Amazon RDS.	27. Februar 2024

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSServiceRolePolicy` `AWSServiceRoleForRDS` serviceverknüpften Rolle eine neue Erklärung IDs hinzugefügt. Weitere Informationen finden Sie unter Berechtigungen von serviceverknüpften Rollen für Amazon RDS.	19. Januar 2024
AWS verwaltete Richtlinien für Amazon RDS – Aktualisierung auf bestehende Richtlinien	Die von `AmazonRDS` `PerformanceInsightsReadOnly` und `AmazonRDSPerformanceInsightsFullAccess` verwalteten Richtlinien enthalten jetzt `Sid` (Statement-ID) als Bezeichner in der Richtlinienerklärung. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsReadOnly und AWS verwaltete Richtlinie: Amazon RDSPerformance InsightsFullAccess	23. Oktober 2023

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neuen Berechtigungen ermöglichen es RDS Custom for Oracle, EventBridge verwaltete Regeln zu erstellen, zu ändern und zu löschen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	20. September 2023

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinien für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der verwalteten `AmazonRDSFullAccess` -Richtlinie neue Berechtigungen hinzugefügt. Mit den Berechtigungen können Sie den Leistungsanalysebericht für einen bestimmten Zeitraum erstellen, anzeigen und löschen. Weitere Informationen zur Konfiguration von Zugriffsrichtlinien für Performance Insights finden Sie unter Konfigurieren von Zugriffsrichtlinien für Performance Insights.	17. August 2023

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinien für Amazon RDS – Neue Richtlinie und Aktualisierung der bestehenden Richtlinie	Amazon RDS hat der verwalteten `AmazonRDSPerformanceInsightsReadOnly` -Richtlinie neue Berechtigungen und eine neue verwaltete Richtlinie mit dem Namen `AmazonRDSPerformanceInsightsFullAccess` hinzugefügt. Diese Berechtigungen ermöglichen es Ihnen, Performance Insights für einen bestimmten Zeitraum zu analysieren, sich die Analyseergebnisse zusammen mit den Empfehlungen anzusehen und die Berichte zu löschen. Weitere Informationen zur Konfiguration von Zugriffsrichtlinien für Performance Insights finden Sie unter Konfigurieren von Zugriffsrichtlinien für Performance Insights.	16. August 2023

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neuen Berechtigungen erlauben RDS Custom für Oracle, DB-Snapshots zu verwenden. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	23. Juni 2023
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neuen Berechtigungen erlauben RDS Custom für Oracle, DB-Snapshots zu verwenden. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	23. Juni 2023

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neuen Berechtigungen erlauben RDS Custom, Netzwerkschnittstellen zu erstellen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	30. Mai 2023
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. Diese neuen Berechtigungen ermöglichen es RDS Custom, Amazon EBS aufzurufen, um das Speicherkontingent zu überprüfen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	18. April 2023

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS Custom hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` zur Integration in Amazon SQS hinzugefügt. RDS Custom erfordert die Integration mit Amazon SQS, um SQS-Warteschlangen im Kundenkonto zu erstellen und zu verwalten. Die SQS-Warteschlangennamen folgen dem Format `do-not-delete-rds-custom-[identifier]` und sind mit `Amazon RDS Custom` getaggt. Die Berechtigung für `ec2:CreateSnapshot` wurde ebenfalls hinzugefügt, damit RDS Custom Backups für Volumes erstellen kann, die der Instance angefügt sind. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	06. April 2023

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinien für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat einen neuen CloudWatch Amazon-Namespace `ListMetrics` zu `AmazonRDSFullAccess` und `AmazonRDSReadOnlyAccess` hinzugefügt. Dieser Namespace ist erforderlich, damit Amazon RDS spezifische Metriken zur Ressourcennutzung auflisten kann. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre CloudWatch Ressourcen.	4. April 2023

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinien für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat den Richtlinien eine neue Berechtigung hinzugefügt `AmazonRDSFullAccess` und Richtlinien `AmazonRDSReadOnlyAccess` verwaltet, um die Anzeige von Amazon DevOps Guru-Ergebnissen in der RDS-Konsole zu ermöglichen. Diese Genehmigung ist erforderlich, um die Anzeige von DevOps Guru-Ergebnissen zu ermöglichen. Weitere Informationen finden Sie unter Amazon RDS-Updates für AWS verwaltete Richtlinien.	30. März 2023

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSServiceRolePolicy` `AWSServiceRoleForRDS` serviceverknüpften Rolle neue Berechtigungen für die Integration mit AWS Secrets Manager hinzugefügt. RDS erfordert die Integration mit Secrets Manager für die Verwaltung von Hauptbenutzerpasswörtern in Secrets Manager. Das Secret verwendet eine reservierte Namenskonvention und schränkt Kundenaktualisierungen ein. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS, und AWS Secrets Manager.	22. Dezember 2022

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur `AmazonRDSCustomServiceRolePolicy` der serviceverknüpftem Rolle `AWSServiceRoleForRDSCustom` hinzugefügt. RDS Custom unterstützt DB-Cluster. Diese neuen Berechtigungen in der Richtlinie ermöglichen es RDS Custom, AWS-Services im Namen Ihrer DB-Cluster Anrufe zu tätigen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom.	9. November 2022

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue Berechtigungen zur serviceverknüpfte Rolle `AWSServiceRoleForRDS` zur Integration in AWS Secrets Manager hinzugefügt. Die Integration in Secrets Manager ist erforderlich, damit SQL Server Reporting Services (SSRS)-E-Mail auf RDS funktioniert. SSRS-E-Mail erstellt im Namen des Kunden ein Secret. Das Secret verwendet eine reservierte Namenskonvention und schränkt Kundenaktualisierungen ein. Weitere Informationen finden Sie unter Verwenden von SSRS E-Mail zum Senden von Berichten.	26. August 2022

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat einen neuen CloudWatch Amazon-Namespace zu `AmazonRDSPreviewServiceRolePolicy` for `PutMetricData` hinzugefügt. Dieser Namespace ist erforderlich, damit Amazon RDS Metriken zur Ressourcennutzung veröffentlichen kann. Weitere Informationen finden Sie unter Bedingungsschlüssel verwenden, um den Zugriff auf CloudWatch Namespaces zu beschränken im CloudWatch Amazon-Benutzerhandbuch.	7. Juni 2022

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat einen neuen CloudWatch Amazon-Namespace zu `AmazonRDSBetaServiceRolePolicy` for `PutMetricData` hinzugefügt. Dieser Namespace ist erforderlich, damit Amazon RDS Metriken zur Ressourcennutzung veröffentlichen kann. Weitere Informationen finden Sie unter Bedingungsschlüssel verwenden, um den Zugriff auf CloudWatch Namespaces zu beschränken im CloudWatch Amazon-Benutzerhandbuch.	7. Juni 2022

Änderung	Beschreibung	Datum
Berechtigungen von servicever knüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat einen neuen CloudWatch Amazon-Namespace zu `AWSServiceRoleForRDS` for `PutMetricData` hinzugefügt. Dieser Namespace ist erforderlich, damit Amazon RDS Metriken zur Ressourcennutzung veröffentlichen kann. Weitere Informationen finden Sie unter Bedingungsschlüssel verwenden, um den Zugriff auf CloudWatch Namespaces zu beschränken im CloudWatch Amazon-Benutzerhandbuch.	22. April 2022

Änderung	Beschreibung	Datum
Berechtigungen von serviceverknüpften Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AWSServiceRoleForRDS` serviceverknüpften Rolle neue Berechtigungen hinzugefügt, um Berechtigungen für kundeneigene IP-Pools und lokale Gateway-Routentabellen (LGW-) zu verwalten. RTBs Diese Berechtigungen sind erforderlich, damit RDS on Outposts eine Multi-AZ-Replikation im lokalen Netzwerk der Outposts durchführen kann. Weitere Informationen finden Sie unter Arbeiten mit Multi-AZ-Bereitstellungen für Amazon RDS in AWS Outposts.	19. April 2022

Änderung	Beschreibung	Datum
Identitätsbasierte Richtlinien – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat der `AmazonRDSFullAccess` verwalteten Richtlinie eine neue Berechtigung hinzugefügt, um Berechtigungen für LGW- RTBs zu beschreiben. Diese Berechtigung ist erforderlich, um Berechtigungen zu beschreiben, damit RDS on Outposts eine Multi-AZ-Replikation im lokalen Netzwerk der Outposts durchführen kann. Weitere Informationen finden Sie unter Arbeiten mit Multi-AZ-Bereitstellungen für Amazon RDS in AWS Outposts.	19. April 2022

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinien für Amazon RDS – Neue Richtlinie	Amazon RDS hat eine neue verwaltete Richtlinie hinzugefügt `AmazonRDSPerformanceInsightsReadOnly`, die es Amazon RDS ermöglicht, AWS Dienste im Namen Ihrer DB-Instances aufzurufen. Weitere Informationen zur Konfiguration von Zugriffsrichtlinien für Performance Insights finden Sie unter Konfigurieren von Zugriffsrichtlinien für Performance Insights.	10. März 2022
Berechtigungen von servicegebundenen Rollen für Amazon RDS – Aktualisierung auf eine bestehende Richtlinie	Amazon RDS hat neue CloudWatch Amazon-Namespaces zu for hinzugefügt. `AWSServiceRoleForRDS` `PutMetricData` Diese Namespaces sind erforderlich, damit Amazon DocumentDB (mit MongoDB-Kompatibilität) und Amazon Neptune Metriken veröffentlichen können. CloudWatch Weitere Informationen finden Sie unter Bedingungsschlüssel verwenden, um den Zugriff auf CloudWatch Namespaces zu beschränken im CloudWatch Amazon-Benutzerhandbuch.	4. März 2022

Änderung	Beschreibung	Datum
Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom – Neue Richtlinie	Amazon RDS hat eine neue serviceverknüpfte Rolle namens `AWSServiceRoleForRDSCustom` hinzugefügt, um RDS Custom zu erlauben, AWS-Services im Namen Ihrer DB-Instances aufzurufen.	26. Oktober 2021
Amazon RDS hat mit der Verfolgung von Änderungen begonnen	Amazon RDS begann, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	26. Oktober 2021

Vermeidung des dienstübergreifenden Confused-Deputy-Problems

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine Entität, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine Entität mit größeren Rechten zwingen kann, die Aktion auszuführen. In AWS kann ein dienstübergreifendes Identitätswechsels zu einem Problem mit dem verwirrten Stellvertreter führen.

Ein serviceübergreifender Identitätswechsel kann auftreten, wenn ein Service (der Anruf-Service) einen anderen Service anruft (den aufgerufenen Service). Der aufrufende Dienst kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, AWS bietet Tools, mit denen Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde. Weitere Informationen finden Sie unter Das Problem des verwirrten Stellvertreters im IAM-Benutzerhandbuch.

Wir empfehlen die Verwendung der globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` in ressourcenbasierten Richtlinien, um die Berechtigungen, die Amazon RDS einem anderen Service erteilt, auf eine bestimmte Ressource zu beschränken.

In einigen Fällen enthält der `aws:SourceArn`-Wert nicht die Konto-ID, z. B. wenn Sie den Amazon-Ressourcennamen (ARN) für einen Amazon-S3-Bucket verwenden. Stellen Sie in diesen Fällen sicher, dass Sie beide globalen Kontextschlüssel für die Bedingung verwenden, um Berechtigungen einzuschränken. In einigen Fällen verwenden Sie beide globalen Bedingungskontextschlüssel und der `aws:SourceArn`-Wert enthält die Konto-ID. Stellen Sie in diesen Fällen sicher, dass der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn` dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden. Wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten, verwenden Sie `aws:SourceArn`. Wenn Sie zulassen möchten, dass eine Ressource im angegebenen AWS Konto der dienstübergreifenden Nutzung zugeordnet wird, verwenden Sie. `aws:SourceAccount`

Stellen Sie sicher, dass der Wert von `aws:SourceArn` ein ARN für einen Amazon-RDS-Ressourcentyp ist. Weitere Informationen finden Sie unter Amazon-Ressourcennamen (ARNs) in Amazon RDS.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontextschlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. In einigen Fällen kennen Sie möglicherweise den vollständigen ARN der Ressource nicht oder Sie geben möglicherweise mehrere Ressourcen an. Verwenden Sie in diesen Fällen die

globalen `aws:SourceArn`-Kontextbedingungsschlüssel mit Platzhaltern (*) für die unbekannten Teile des ARN. Ein Beispiel ist `arn:aws:rds:*:123456789012:*`.

Das folgende Beispiel zeigt, wie Sie die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` für Amazon RDS verwenden können, um das Confused-Deputy-Problem zu verhindern.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Weitere Beispiele für Richtlinien, die die globalen Bedingungskontextschlüssel `aws:SourceArn` und `aws:SourceAccount` verwenden, finden Sie in den folgenden Abschnitten:

- Erteilen von Berechtigungen zum Veröffentlichen von Benachrichtigungen in einem Amazon-SNS-Thema
- Manuelles Erstellen einer IAM-Rolle für native Backups und Wiederherstellungen
- Einrichten einer Windows-Authentifizierung für SQL Server-DB-Instances
- Voraussetzungen für die Integration von RDS-for-SQL-Server mit S3
- Manuelles Erstellen einer IAM-Rolle für SQL Server Audit

- Konfigurieren von IAM-Berechtigungen für die Integration von RDS for Oracle in Amazon S3
- Einrichten des Zugriffs auf einen Amazon S3-Bucket (PostgreSQL-Import)
- Einrichten des Zugriffs auf einen Amazon S3-Bucket (PostgreSQL-Export)

IAM-Datenbankauthentifizierung für MariaDB, MySQL und PostgreSQL

Sie können sich bei Ihrem mithilfe der AWS Identity and Access Management (IAM-) Datenbankauthentifizierung authentifizieren. Die IAM-Datenbankauthentifizierung funktioniert mit MariaDB, MySQL und PostgreSQL. Mit dieser Authentifizierungsmethode benötigen Sie kein Passwort, um eine Verbindung mit einer DB-Instance herzustellen. Stattdessen verwenden Sie ein Authentifizierungstoken.

Ein Authentifizierungstoken ist eine eindeutige Zeichenfolge, die von Amazon RDS auf Anforderung erzeugt wird. Authentifizierungstoken werden mit AWS Signature Version 4 generiert. Jedes Token verfällt 15 Minuten nach seiner Erzeugung. Da die Authentifizierung mithilfe von IAM extern verwaltet wird, ist es nicht erforderlich, Benutzeranmeldeinformationen in der Datenbank zu speichern. Sie können weiterhin auch die Standard-Datenbank-Authentifizierung verwenden. Das Token wird nur zur Authentifizierung verwendet und wirkt sich nicht auf die Sitzung aus, nachdem es eingerichtet wurde.

Die IAM-Datenbank-Authentifizierung bietet die folgenden Vorteile:

- Der Netzwerkverkehr zur und von der Datenbank wird mit Secure Socket Layer (SSL) oder Transport Layer Security (TLS) verschlüsselt. Weitere Informationen zur Verwendung von Amazon RDS finden Sie unter.
- Sie können IAM verwenden, um den Zugriff auf Ihre Datenbankressourcen zentral zu verwalten, anstatt den Zugriff individuell auf jeder DB-Instance zu verwalten.
- Für Anwendungen, die auf Amazon ausgeführt werden EC2, können Sie für Ihre EC2 Instance spezifische Profilanmeldedaten anstelle eines Kennworts verwenden, um die Sicherheit zu erhöhen.

Erwägen Sie im Allgemeinen, die IAM-Datenbankauthentifizierung zu verwenden, wenn Ihre Anwendungen weniger als 200 Verbindungen pro Sekunde erstellen und Sie Benutzernamen und Passwörter nicht direkt in Ihrem Anwendungscode verwalten möchten.

Der Amazon Web Services (AWS) JDBC-Treiber unterstützt die IAM-Datenbankauthentifizierung. Weitere Informationen finden Sie unter AWS IAM-Authentifizierungs-Plug-In im Amazon Web Services (AWS) JDBC-Treiber-Repository. GitHub

Der Amazon Web Services (AWS) Python-Treiber unterstützt die IAM-Datenbankauthentifizierung. Weitere Informationen finden Sie unter AWS IAM-Authentifizierungs-Plug-In im GitHubPython-Treiber-Repository von Amazon Web Services (AWS).

Sehen Sie sich die folgenden Themen an, um mehr über den Prozess zur Einrichtung von IAM für die DB-Authentifizierung zu erfahren:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung
- Herstellen einer Verbindung zu Ihrem DB-Instance- mithilfe der IAM-Authentifizierung

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit und der Support von Funktionen variieren je nach Version der einzelnen Datenbank-Engines. Weitere Informationen zur Verfügbarkeit von Engine, Version und Region mit Amazon RDS- und IAM-Datenbankauthentifizierung finden Sie unterUnterstützte Regionen und DB-Engines für die IAM-Datenbankauthentifizierung in Amazon RDS.

CLI- und SDK-Unterstützung

Die IAM-Datenbankauthentifizierung ist für die AWS CLIund für die folgenden sprachspezifischen Sprachen verfügbar: AWS SDKs

- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java
- AWS SDK für JavaScript
- AWS SDK für PHP
- AWS SDK für Python (Boto3)
- AWS SDK für Ruby

Einschränkungen der IAM-Datenbank-Authentifizierung

Beachten Sie bei der Verwendung der IAM-Datenbankauthentifizierung die folgenden Einschränkungen:

- Die IAM-Datenbankauthentifizierung drosselt Verbindungen auf 200 Verbindungen pro Sekunde.

Verbindungen, die dasselbe Authentifizierungstoken verwenden, werden nicht gedrosselt. Wir empfehlen, dass Sie Authentifizierungstoken nach Möglichkeit wiederverwenden.

- Derzeit unterstützt die IAM-Datenbankauthentifizierung nicht alle globalen Bedingungskontextschlüssel.

  Weitere Informationen über globale Bedingungskontextschlüssel finden Sie unter Globale AWS - Bedingungskontextschlüssel im IAM-Benutzerhandbuch.
- Wenn die IAM-Rolle (`rds_iam`) einem Benutzer (einschließlich RDS-Hauptbenutzer) hinzugefügt wird, hat bei PostgreSQL die IAM-Authentifizierung Vorrang vor der Passwort-Authentifizierung, so dass sich der Benutzer als IAM-Benutzer anmelden muss.
- Für PostgreSQL unterstützt Amazon RDS nicht die gleichzeitige Aktivierung der Authentifizierungsmethoden IAM und Kerberos.
- Für PostgreSQL können Sie die IAM-Authentifizierung nicht verwenden, um eine Replikationsverbindung herzustellen.
- Sie können keinen benutzerdefinierten Route-53-DNS-Datensatz anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.
- CloudWatch und protokollieren Sie die IAM-Authentifizierung CloudTrail nicht. Diese Dienste verfolgen keine `generate-db-auth-token` API-Aufrufe, die die IAM-Rolle autorisieren, eine Datenbankverbindung zu aktivieren.
- Für eine zuverlässige Konnektivität benötigen Sie zwischen 300 und 1000 MiB zusätzlichen Speicher in Ihrer Datenbank. Um den für Ihren Workload benötigten Speicher zu ermitteln, vergleichen Sie die RES-Spalte für RDS-Prozesse in der Prozessliste für Enhanced Monitoring vor und nach der Aktivierung der IAM-DB-Authentifizierung. Siehe Betriebssystemmetriken in der RDS Konsole anzeigen.

  Wenn Sie eine Burstable-Klasseninstanz verwenden, vermeiden Sie, dass Ihnen der Speicherplatz ausgeht, indem Sie den von anderen Parametern wie Puffern und Cache belegten Speicher um den gleichen Betrag reduzieren.
- Für RDS for MySQL können Sie keine kennwortbasierte Authentifizierung für einen Datenbankbenutzer verwenden, den Sie mit IAM-Authentifizierung konfigurieren.
- Die IAM-DB-Authentifizierung wird für RDS on Outposts für keine Engine unterstützt.

Empfehlungen für die IAM-Datenbankauthentifizierung

Bei Verwendung der IAM-Datenbankauthentifizierung empfehlen wir Folgendes:

- Verwenden Sie die IAM-Datenbankauthentifizierung, wenn Ihre Anwendung weniger als 200 neue IAM-Datenbankauthentifizierungsverbindungen pro Sekunde benötigt.

  Die Datenbank-Engines, die mit Amazon RDS kompatibel sind, setzen den Authentifizierungsversuchen pro Sekunde keine Grenzen. Wenn Sie jedoch die IAM-Datenbank-Authentifizierung verwenden, muss Ihre Anwendung ein Authentifizierungstoken erzeugen. Ihre Anwendung verwendet dann dieses Token, um eine Verbindung mit der DB-Instance herzustellen. Wenn Sie die Höchstanzahl neuer Verbindungen pro Sekunde überschreiten, kann der zusätzliche Overhead der IAM-Datenbankauthentifizierung zur Verbindungsablehnung führen.

  Erwägen Sie die Verwendung von Verbindungspooling in Ihren Anwendungen, um den ständigen Verbindungsaufbau zu begrenzen. Dadurch lässt sich der Aufwand für die IAM-DB-Authentifizierung reduzieren und Ihre Anwendungen können bestehende Verbindungen wiederverwenden. Erwägen Sie alternativ die Verwendung von RDS-Proxy für diese Anwendungsfälle. RDS-Proxy ist mit zusätzlichen Kosten verbunden. Weitere Informationen finden Sie unter RDS-Proxy – Preise.
- Die Größe eines IAM-Datenbankauthentifizierungstokens hängt von vielen Faktoren ab, einschließlich der Anzahl der IAM-Tags, IAM-Servicerichtlinien, ARN-Längen sowie andere IAM- und Datenbankeigenschaften. Die Mindestgröße dieses Tokens beträgt im Allgemeinen etwa 1 KB, kann aber durchaus größer sein. Da dieses Token mithilfe der IAM-Authentifizierung als Passwort in der Verbindungszeichenfolge zur Datenbank verwendet wird, sollten Sie sicherstellen, dass Ihr Datenbanktreiber (z. B. ODBC) und and/or alle Tools dieses Token nicht aufgrund seiner Größe einschränken oder anderweitig kürzen. Ein verkürztes Token führt dazu, dass die von der Datenbank und IAM durchgeführte Authentifizierungsüberprüfung fehlschlägt.
- Wenn Sie beim Erstellen eines IAM-Datenbank-Authentifizierungstokens temporäre Anmeldeinformationen verwenden, müssen die temporären Anmeldeinformationen weiterhin gültig sein, wenn Sie das IAM-Datenbank-Authentifizierungstoken für eine Verbindungsanforderung verwenden.

Kontextschlüssel für globale Bedingungen werden nicht unterstützt AWS

Die IAM-Datenbankauthentifizierung unterstützt die folgende Teilmenge der AWS globalen Bedingungskontextschlüssel nicht.

- `aws:Referer`
- `aws:SourceIp`
- `aws:SourceVpc`

- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

Weitere Informationen finden Sie unter Globale AWS -Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung

Die IAM-Datenbank-Authentifizierung ist für DB-Instances standardmäßig deaktiviert. Sie können die IAM-Datenbankauthentifizierung mithilfe der AWS Management Console AWS CLI, oder der API aktivieren oder deaktivieren.

Sie können die IAM-Datenbankauthentifizierung aktivieren, wenn Sie eine der folgenden Aktionen ausführen:

- Informationen zum Erstellen einer neuen DB-Instance mit aktivierter IAM-Datenbankauthentifizierung finden Sie unter Erstellen einer Amazon RDS-DB-Instance.
- Informationen zum Ändern einer DB-Instance zur Aktivierung der IAM-Datenbankauthentifizierung finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Informationen zum Wiederherstellen einer DB-Instance aus einem Snapshot mit aktivierter IAM-Datenbankauthentifizierung finden Sie unter Wiederherstellung auf einer DB-Instance.
- Informationen zum Wiederherstellen eines DB-Instance-Clusters zu einem Zeitpunkt mit aktivierter IAM-Datenbankauthentifizierung finden Sie unter Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

IAM-Authentifizierung für PostgreSQL-DB-Instances erfordern 1 als SSL-Wert. Sie können die IAM-Authentifizierung für eine PostgreSQL-DB-Instance nicht aktivieren, wenn der SSL-Wert 0 ist. Sie können den SSL-Wert nicht auf 0 ändern, wenn die IAM-Authentifizierung für eine PostgreSQL-DB-Instance aktiviert ist.

Konsole

Jeder Erstellungs- oder Änderungsworkflow verfügt über einen Abschnitt Database authentification (Datenbankauthentifizierung), in dem Sie die IAM-Datenbankauthentifizierung aktivieren oder deaktivieren können. Wählen Sie in diesem Abschnitt Password and IAM database authentication

(Passwort- und IAM-Datenbankauthentifizierung), um die IAM-Datenbankauthentifizierung zu aktivieren.

So aktivieren bzw. deaktivieren die IAM-Datenbankauthentifizierung für eine vorhandene DB-Instance:

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
3. Wählen Sie die DB-Instance aus, die Sie ändern möchten.

   > **Note**
   >
   > Stellen Sie sicher, dass die DB-Instance mit der IAM-Authentifizierung kompatibel ist. Überprüfen Sie die Kompatibilitätsanforderungen in Verfügbarkeit von Regionen und Versionen.

4. Wählen Sie Modify aus.
5. Wählen Sie in diesem Abschnitt Datenbankauthentifizierung Password and IAM database authentication (Passwort- und IAM-Datenbankauthentifizierung), um die IAM-Datenbankauthentifizierung zu aktivieren. Wählen Sie Passwort-Authentifizierung oder Passwort- und Kerberos-Authentifizierung aus, um die IAM-Authentifizierung zu deaktivieren.
6. Sie können sich auch dafür entscheiden, die Veröffentlichung von IAM-DB-Authentifizierungsprotokollen in Logs zu CloudWatch aktivieren. Wählen Sie unter Protokollexporte die Option iam-db-auth-error Protokoll aus. Durch das Veröffentlichen Ihrer CloudWatch Protokolle in Logs wird Speicherplatz verbraucht, und für diesen Speicherplatz fallen Gebühren an. Achten Sie darauf, alle CloudWatch Logs zu löschen, die Sie nicht mehr benötigen.
7. Klicken Sie auf Continue.
8. Um die Änderungen sofort anzuwenden, wählen Sie im Abschnitt Scheduling of modifications (Planen von Änderungen) die Option Immediately (Sofort).
9. Wählen Sie Modify DB instance (DB-Instance ändern).

AWS CLI

Verwenden Sie den `create-db-instance`Befehl, um mithilfe von eine neue DB-Instance mit IAM-Authentifizierung zu erstellen. AWS CLI Geben Sie die Option `--enable-iam-database-authentication` wie im folgenden Beispiel an.

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m3.medium \
    --engine MySQL \
    --allocated-storage 20 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --enable-iam-database-authentication
```

Um eine vorhandene DB-Instance zu aktualisieren, um eine IAM-Authentifizierung zu ermöglichen oder zu verhindern, verwenden Sie den AWS CLI -Befehl `modify-db-instance`. Sie müssen entweder die Option `--enable-iam-database-authentication` oder `--no-enable-iam-database-authentication` angeben.

> **Note**
>
> Stellen Sie sicher, dass die DB-Instance mit der IAM-Authentifizierung kompatibel ist. Überprüfen Sie die Kompatibilitätsanforderungen in Verfügbarkeit von Regionen und Versionen.

Standardmäßig führt Amazon RDS die Änderung während des nächsten Wartungsfensters durch. Wenn Sie diese Einstellung übergehen und die IAM-DB-Authentifizierung schnellstmöglich aktivieren möchten, verwenden Sie den Parameter `--apply-immediately`.

Im folgenden Beispiel wird gezeigt, wie Sie die IAM-Authentifizierung für eine vorhandene DB-Instance sofort aktivieren können.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --apply-immediately \
    --enable-iam-database-authentication
```

Wenn Sie einen wiederherstellen, verwenden Sie einen der folgenden AWS CLI Befehle:

- `restore-db-instance-to-point-in-time`
- `restore-db-instance-from-db-snapshot`

Die Voreinstellung der IAM-Datenbank-Authentifizierung entspricht der Einstellung des Quell-Snapshot. Wählen Sie die entsprechende Option `--enable-iam-database-authentication` oder `--no-enable-iam-database-authentication` aus.

RDS-API

Verwenden Sie die API-Operation , um mithilfe der API eine neue DB-Instance mit IAM-Authentifizierung zu erstellen `CreateDBInstance`. Stellen Sie den Parameter `EnableIAMDatabaseAuthentication` auf `true` ein.

Um eine bestehende DB-Instance mit oder ohne IAM-Authentifizierung zu aktualisieren, verwenden Sie die API-Operation `ModifyDBInstance`. Setzen Sie den Parameter `EnableIAMDatabaseAuthentication` auf `true`, um die IAM-Authentifizierung zu aktivieren, oder auf `false`, um sie zu deaktivieren.

> **Note**
>
> Stellen Sie sicher, dass die DB-Instance mit der IAM-Authentifizierung kompatibel ist. Überprüfen Sie die Kompatibilitätsanforderungen in Verfügbarkeit von Regionen und Versionen.

Verwenden Sie eine der folgenden API-Operationen, wenn Sie einen DB-Instance- wiederherstellen.

- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

Die Voreinstellung der IAM-Datenbank-Authentifizierung entspricht der Einstellung des Quell-Snapshot. Zum Ändern dieser Einstellung setzen Sie den Parameter `EnableIAMDatabaseAuthentication` auf `true`, um die IAM-Authentifizierung zu aktivieren, oder auf `false`, um sie zu deaktivieren.

Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff

Sie müssen eine IAM-Richtlinie erstellen, um einem Benutzer oder einer Rolle zu erlauben, eine Verbindung mit Ihrer DB-Instance herzustellen. Anschließend fügen Sie die Richtlinie an einen Berechtigungssatz oder eine Rolle an.

> **Note**
>
> Weitere Informationen zu IAM-Richtlinien finden Sie unter Identitäts- und Zugriffsmanagement für Amazon RDS.

Die folgende Beispielrichtlinie erlaubt einem Benutzer, eine Verbindung mit einer DB-Instance mithilfe der IAM-Datenbank-Authentifizierung herzustellen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rds-db:connect"
            ],
            "Resource": [
                "arn:aws:rds-db:us-east-2:111122223333:dbuser:db-
ABCDEFGHIJKL01234/db_user"
            ]
        }
    ]
}
```

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "rds-db:connect"
            ],
            "Resource": [
                "arn:aws:rds-db:us-east-2:111122223333:dbuser:cluster-
ABCDEFGHIJKL01234/db_user"
            ]
        }
    ]
}
```

Important

Ein Benutzer mit Administratorberechtigungen kann auf DB-Instances zugreifen, ohne über eine IAM-Richtlinie explizite Berechtigungen zu erhalten. Wenn Sie den Administratorzugriff auf DB-Instance- einschränken möchten, können Sie eine IAM-Rolle mit geeigneten, weniger privilegierten Berechtigungen erstellen und diese dem Administrator zuweisen.

Note

Verwechseln Sie das Präfix `rds-db:` nicht mit anderen RDS-API-Operationspräfixen, die mit `rds:` beginnen. Das Präfix `rds-db:` und die Aktion `rds-db:connect` werden nur zur IAM-Datenbank-Authentifizierung verwendet. Sie sind in keinem anderen Kontext gültig.

Die Beispielrichtlinie enthält eine einzige Anweisung mit den folgenden Elementen:

- `Effect` – Geben Sie `Allow` an, um den Zugriff auf die DB-Instance zu gewähren. Wenn Sie den Zugriff nicht ausdrücklich erlauben, wird er automatisch verweigert.
- `Action` – Geben Sie `rds-db:connect` an, um Verbindungen mit der DB-Instance zu erlauben.
- `Resource` – Geben Sie einen Amazon-Ressourcennamen (ARN) an, der ein Datenbankkonto auf einer DB-Instance beschreibt. Das ARN-Format lautet folgendermaßen.

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

Ersetzen Sie in diesem Format Folgendes:

- *region*ist die AWS Region für den . In der Beispielrichtlinie ist die AWS Regionus-east-2.
- *account-id*ist die AWS Kontonummer für den . In der Beispielrichtlinie lautet die Kontonummer 1234567890. Der Benutzer muss demselben Konto wie das Konto für den DB-Instance- angehören.

  Erstellen Sie für einen kontoübergreifenden Zugriff eine IAM-Rolle mit der oben angegebenen Richtlinie in dem Konto für den DB-Instance- und erlauben Sie Ihrem anderen Konto, die Rolle zu übernehmen.
- *DbiResourceId* ist die Kennung der DB-Instance. Diese Kennung ist für eine AWS Region eindeutig und ändert sich nie. In dieser Beispielrichtlinie lautet die Kennung db-ABCDEFGHIJKL01234.

  Um eine in Amazon RDS zu finden, wählen Sie den aus, um dessen Details zu sehen. Wechseln Sie zur Registerkarte Konfiguration. Die Resource ID (Ressourcen-ID) wird im Abschnitt Configuration Details (Konfigurationsdetails) angezeigt.

  Alternativ können Sie den AWS CLI Befehl verwenden, um die Identifikatoren und Ressourcen IDs für Ihren gesamten in der aktuellen AWS Region aufzulisten, wie im Folgenden dargestellt.

```
aws rds describe-db-instances --query "DBInstances[*].
[DBInstanceIdentifier,DbiResourceId]"
```

  Wenn Sie Amazon Aurora verwenden, geben Sie eine DbClusterResourceId anstelle einer DbiResourceId an. Weitere Informationen finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff im Amazon Aurora Benutzerhandbuch.

  > **Note**
  >
  > Wenn Sie über den RDS-Proxy eine Verbindung zu einer Datenbank herstellen, geben Sie die Proxy-Ressourcen-ID an, z. B. prx-ABCDEFGHIJKL01234. Hinweise zur Verwendung der IAM-Datenbankauthentifizierung mit RDS-Proxy finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

- *db-user-name* ist der Name des mit der IAM-Authentifizierung zu verknüpfenden Datenbankkontos. In der Beispielrichtlinie lautet das Datenbankkonto `db_user`.

Sie können andere erstellen ARNs , um verschiedene Zugriffsmuster zu unterstützen. Die folgende Richtlinie erlaubt den Zugriff auf zwei verschiedene Datenbankkonten auf einer DB-Instance .

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
jane_doe",
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
mary_roe"
         ]
      }
   ]
}
```

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
```

```
            "arn:aws:rds-db:us-east-2:123456789012:dbuser:cluster-
ABCDEFGHIJKL01234/jane_doe",
            "arn:aws:rds-db:us-east-2:123456789012:dbuser:cluster-
ABCDEFGHIJKL01234/mary_roe"
         ]
      }
   ]
}
```

Die folgende Richtlinie verwendet das Zeichen „*“, um allen und Datenbankkonten für ein bestimmtes AWS Konto und eine bestimmte AWS Region zuzuordnen.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rds-db:connect"
            ],
            "Resource": [
                "arn:aws:rds-db:us-east-2:111122223333:dbuser:*/*"
            ]
        }
    ]
}
```

Die folgende Richtlinie entspricht allen für ein bestimmtes AWS Konto und eine bestimmte AWS Region. Allerdings gewährt die Richtlinie nur den Zugriff auf DB-Instances, die über ein `jane_doe`-Datenbankkonto verfügen.

JSON

```
{
   "Version": "2012-10-17",
   "Statement": [
```

```
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:*/jane_doe"
         ]
      }
   ]
}
```

Der Benutzer oder die Rolle hat nur auf jene Datenbanken Zugriff, auf die der Datenbankbenutzer zugreifen kann. Nehmen wir beispielsweise an, dass Ihre DB-Instance über eine Datenbank mit dem Namen dev und eine weitere Datenbank mit dem Namen test verfügt. Wenn der Datenbankbenutzer `jane_doe` nur auf dev zugreifen kann, haben auch alle anderen Benutzer oder Rollen, die auf diese DB-Instance mit dem Benutzer `jane_doe` zugreifen, lediglich die Berechtigung für den Zugriff auf dev. Diese Zugriffsbeschränkung gilt auch für andere Datenbankobjekte, wie z. B. Tabellen, Ansichten usw.

Ein Administrator muss IAM-Richtlinien erstellen, die Entitäten die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den Berechtigungssätzen oder Rollen anfügen, die diese Berechtigungen benötigen. Beispiele für Richtlinien finden Sie unter Beispiele für identitätsbasierte Amazon-RDS-Richtlinien.

Anfügen einer IAM-Richtlinie an einen Berechtigungssatz oder eine Rolle

Nachdem Sie eine IAM-Richtlinie erstellt haben, um die Datenbank-Authentifizierung zu erlauben, müssen Sie die Richtlinie an einen Berechtigungssatz oder eine Rolle anfügen. Eine praktische Anleitung zu diesem Thema finden Sie unter Erstellen und Anfügen Ihrer ersten vom Kunden verwalteten Richtlinie im IAM-Benutzerhandbuch.

Wenn Sie die praktischen Anleitung durchgehen, können Sie eine der in diesem Abschnitt aufgeführten Beispielrichtlinien als Ausgangsbasis verwenden und auf Ihre Bedürfnisse anpassen. Am Ende des Tutorials verfügen Sie über einen Berechtigungssatz mit einer angefügten Richtlinie, der zur Durchführung der Aktion `rds-db:connect` berechtigt ist.

> **Note**
>
> Sie können mehrere Berechtigungssätze oder Rollen mit demselben Datenbank-Benutzerkonto verknüpfen. Nehmen wir beispielsweise an, dass Ihre IAM-Richtlinie folgenden Ressourcen-ARN enthält.
>
> ```
> arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/
> jane_doe
> ```
>
> Wenn Sie die Richtlinie an Jane, Bob und Diego anfügen, sind diese Benutzer in der Lage, eine Verbindung mit der angegebenen DB-Instance mithilfe des Datenbankkontos `jane_doe` herzustellen.

Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

Mit der IAM-Datenbank-Authentifizierung müssen Sie den von Ihnen erstellten Benutzerkonten keine Datenbankpasswörter zuweisen. Wenn Sie einen mit dem Datenbankkonto verknüpften Benutzer entfernen, sollten Sie auch das Datenbankkonto mit der Anweisung `DROP USER` entfernen.

> **Note**
>
> Der für die IAM-Authentifizierung verwendete Benutzername muss mit der Schreibweise des Benutzernamens in der Datenbank übereinstimmen.

Themen

- Verwenden der IAM-Authentifizierung mit MariaDB und MySQL
- Verwenden der IAM-Authentifizierung mit PostgreSQL

Verwenden der IAM-Authentifizierung mit MariaDB und MySQL

Bei MariaDB und MySQL erfolgt die Authentifizierung über `AWSAuthenticationPlugin` ein von uns AWS bereitgestelltes Plugin, das nahtlos mit IAM zusammenarbeitet, um Ihre Benutzer zu authentifizieren. Stellen Sie als Hauptbenutzer oder als anderer Benutzer, der Benutzer erstellen und

Berechtigungen gewähren kann, eine Verbindung mit der DB-Instance her. Geben Sie die `CREATE USER`-Anweisung aus, wie im folgenden Beispiel dargestellt.

```
CREATE USER 'jane_doe' IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

Die Klausel `IDENTIFIED WITH` ermöglicht MariaDB und MySQL die Verwendung von `AWSAuthenticationPlugin`, um das Datenbankkonto (`jane_doe`) zu authentifizieren. Die `AS 'RDS'`-Klausel bezieht sich auf die Authentifizierungsmethode. Stellen Sie sicher, dass der angegebene Datenbankbenutzername mit einer Ressource in der IAM-Richtlinie für den IAM-Datenbankzugriff identisch ist. Weitere Informationen finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff.

> **Note**
>
> `ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded`
> Stellen Sie sicher, dass Sie eine unterstützte Konfiguration verwenden und die IAM-Datenbankauthentifizierung auf Ihrer DB-Instance aktiviert haben, um den Fehler zu beheben. Weitere Informationen erhalten Sie unter Verfügbarkeit von Regionen und Versionen und Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung.

Nachdem Sie ein Konto mithilfe von `AWSAuthenticationPlugin` erstellt haben, können Sie es in der gleichen Weise wie sonstige Datenbankkonten verwalten. Sie können beispielsweise Kontoberechtigungen mit den Anweisungen `GRANT` und `REVOKE` oder mehrere Kontoattribute mit der Anweisung `ALTER USER` ändern.

Der Datenbank-Netzwerkverkehr wird bei Verwendung von IAM mit SSL/TLS verschlüsselt. Ändern Sie mit dem folgenden Befehl das Benutzerkonto, um SSL-Verbindungen zuzulassen.

```
ALTER USER 'jane_doe'@'%' REQUIRE SSL;
```

Verwenden der IAM-Authentifizierung mit PostgreSQL

Wenn Sie die IAM-Authentifizierung mit PostgreSQL verwenden möchten, stellen Sie als Hauptbenutzer oder als anderer Benutzer, der Benutzer erstellen und Berechtigungen gewähren kann, eine Verbindung mit der DB-Instance her. Nachdem die Verbindung hergestellt wurde,

erstellen Sie Datenbankbenutzer und weisen ihnen die `rds_iam`-Rolle zu, wie im folgenden Beispiel dargestellt.

```
CREATE USER db_userx;
GRANT rds_iam TO db_userx;
```

Stellen Sie sicher, dass der angegebene Datenbankbenutzername mit einer Ressource in der IAM-Richtlinie für den IAM-Datenbankzugriff identisch ist. Weitere Informationen finden Sie unter Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff. Sie müssen die `rds_iam` Rolle zuweisen, um die IAM-Authentifizierung verwenden zu können. Sie können auch verschachtelte Mitgliedschaften oder indirekte Zuweisungen der Rolle verwenden.

Herstellen einer Verbindung zu Ihrem DB-Instance- mithilfe der IAM-Authentifizierung

Bei einer IAM-Datenbankauthentifizierung verwenden Sie ein Authentifizierungstoken, wenn Sie sich mit Ihrer DB-Instance verbinden. Ein Authentifizierungstoken ist eine Zeichenfolge, die Sie anstelle eines Passworts verwenden. Nachdem Sie ein Authentifizierungstoken erzeugt haben, ist es 15 Minuten lang gültig. Wenn Sie versuchen, sich mit einem verfallenen Token zu verbinden, wird die Verbindungsabfrage abgelehnt.

Jedes Authentifizierungstoken muss eine gültige Signatur unter Verwendung von AWS Signature Version 4 enthalten. (Weitere Informationen finden Sie unter Signaturvorgang für Signature Version 4 in der Allgemeine AWS-Referenz.) Das AWS CLI und ein AWS SDK, z. B. das AWS SDK für Java oder AWS SDK für Python (Boto3), können jedes von Ihnen erstellte Token automatisch signieren.

Sie können ein Authentifizierungstoken verwenden, wenn Sie von einem anderen AWS Service aus eine Verbindung zu Amazon RDS AWS Lambda herstellen, z. Durch Verwendung eines Tokens können Sie vermeiden, ein Passwort in Ihrem Code angeben zu müssen. Alternativ können Sie ein AWS SDK verwenden, um ein Authentifizierungstoken programmgesteuert zu erstellen und programmgesteuert zu signieren.

Nachdem Sie über ein signiertes IAM-Authentifizierungstoken verfügen, können Sie eine Verbindung mit einer Amazon RDS-DB-Instance herstellen. Im Folgenden erfahren Sie, wie Sie dies entweder mit einem Befehlszeilentool oder einem AWS SDK, wie dem oder, tun können. AWS SDK für Java AWS SDK für Python (Boto3)

Weitere Informationen finden Sie in den folgenden Blogeinträgen:

- IAM-Authentifikation zum Verbinden mit von SQL Workbench/J mit Aurora MySQL oder Amazon RDS for MySQL verwenden

- Verwenden der IAM-Authentifizierung zum Verbinden mit pgAdmin Amazon Aurora PostgreSQL oder Amazon RDS for PostgreSQL

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

Themen

- Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung mit den Treibern AWS
- Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung über die Befehlszeile: AWS CLI und MySQL-Client
- Herstellen einer Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung von der Befehlszeile aus: AWS CLI und psql-Client
- Herstellen einer Verbindung zu Ihrem DB-Instance- mithilfe der IAM-Authentifizierung und AWS SDK für .NET
- Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung und der AWS SDK für Go
- Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung und der AWS SDK für Java
- Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung und der AWS SDK für Python (Boto3)

Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung mit den Treibern AWS

Die AWS Treibersuite wurde so konzipiert, dass sie schnellere Switchover- und Failover-Zeiten sowie Authentifizierung mit AWS Secrets Manager, AWS Identity and Access Management (IAM) und Federated Identity unterstützt. Die AWS Treiber sind darauf angewiesen, den Status der zu kennen, um den neuen Writer zu ermitteln. Dieser Ansatz reduziert die Switchover- und Failover-Zeiten auf einstellige Sekunden, verglichen mit mehreren zehn Sekunden bei Open-Source-Treibern.

Weitere Informationen zu den AWS Treibern finden Sie im entsprechenden Sprachtreiber für Ihre RDS for MariaDB-, RDS for MySQL- oder RDS for PostgreSQL-DB-Instance.

> **Note**
>
> Die einzigen Funktionen, die für RDS for MariaDB unterstützt werden, sind Authentifizierung mit AWS Secrets Manager, AWS Identity and Access Management (IAM) und Federated Identity.

Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung über die Befehlszeile: AWS CLI und MySQL-Client

Sie können von der Befehlszeile aus eine Verbindung zu einem herstellen, indem Sie das `mysql` Befehlszeilentool AWS CLI und verwenden, wie im Folgenden beschrieben.

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

> **Note**
>
> Informationen zum Herstellen einer Verbindung mit Ihrer Datenbank mithilfe von SQL Workbench/J mit IAM-Authentifizierung finden Sie im Blogbeitrag IAM-Authentifikation zum Verbinden von SQL Workbench/J Aurora MySQL oder Amazon RDS for MySQL verwenden.

Themen

- Generieren eines IAM-Authentifizierungstokens
- Herstellen der Verbindung zu einem DB-Instance-

Generieren eines IAM-Authentifizierungstokens

Im folgenden Beispiel wird gezeigt, wie Sie ein signiertes Authentifizierungstoken mithilfe der erhalte AWS CLI.

```
aws rds generate-db-auth-token \
   --hostname rdsmysql.123456789012.us-west-2.rds.amazonaws.com \
   --port 3306 \
   --region us-west-2 \
   --username jane_doe
```

In diesem Beispiel lauten die Parameter folgendermaßen:

- `--hostname` – Der Hostname des DB-Instance-, auf den Sie zugreifen möchten.
- `--port` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `--region`— Die AWS Region, in der der läuft
- `--username` – Das Datenbankkonto, auf das Sie zugreifen möchten.

Die ersten Zeichen des Tokens sehen folgendermaßen aus.

```
rdsmysql.123456789012.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

Herstellen der Verbindung zu einem DB-Instance-

Das allgemeine Format zur Herstellung einer Verbindung wird nachfolgend dargestellt.

```
mysql --host=hostName --port=portNumber --ssl-ca=full_path_to_ssl_certificate --enable-
cleartext-plugin --user=userName --password=authToken
```

Dabei werden die folgenden Parameter verwendet:

- `--host` – Der Hostname des DB-Instance-, auf den Sie zugreifen möchten.
- `--port` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `--ssl-ca` – Die SSL-Zertifikatsdatei, die den öffentlichen Schlüssel enthält

Weitere Informationen zur SSL/TLS-Unterstützung für MariaDB finden Sie unter SSL/TLSUnterstützung für MariaDB-DB-Instances auf Amazon RDS.

Weitere Informationen zur SSL/TLS-Unterstützung für MySQL finden Sie unter SSL/TLSUnterstützung für My SQL DB-Instances auf Amazon RDS.

Zum Download eines SSL-Zertifikats siehe .

- `--enable-cleartext-plugin` – Ein Wert, der angibt, dass `AWSAuthenticationPlugin` für diese Verbindung zu verwenden ist.

  Wenn Sie einen MariaDB-Client verwenden, ist die `--enable-cleartext-plugin` Option nicht erforderlich.
- `--user` – Das Datenbankkonto, auf das Sie zugreifen möchten.
- `--password` – Ein signiertes IAM-Authentifizierungstoken.

Das Authentifizierungstoken besteht aus hunderten von Zeichen. Dessen Handhabung in der Befehlszeile kann unhandlich sein. Eine Möglichkeit zur Umgehung dieses Problems besteht darin, das Token in einer Umgebungsvariable zu speichern und dann bei der Verbindungsherstellung diese Variable zu verwenden. Im folgenden Beispiel ist eine Möglichkeit dargestellt, um dieses Problem zu umgehen. Im Beispiel ist */sample_dir/* der vollständige Pfad zur SSL-Zertifikatsdatei, die den öffentlichen Schlüssel enthält.

```
RDSHOST="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
TOKEN="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 3306 --region us-
west-2 --username jane_doe )"

mysql --host=$RDSHOST --port=3306 --ssl-ca=/sample_dir/global-bundle.pem --enable-
cleartext-plugin --user=jane_doe --password=$TOKEN
```

Wenn Sie die Verbindung mithilfe von `AWSAuthenticationPlugin` herstellen, wird die Verbindung mit SSL geschützt. Geben Sie an der `mysql>`-Eingabeaufforderung Folgendes ein, um dies zu überprüfen.

```
show status like 'Ssl%';
```

In den folgenden Linien in der Ausgabe finden Sie weitere Details.

```
+---------------+-------------+
| Variable_name | Value

                                                                    |

+---------------+-------------+
| ...           | ...
| Ssl_cipher    | AES256-SHA

                                                                    |

| ...           | ...
| Ssl_version   | TLSv1.1

                                                                    |

| ...           | ...
+-----------------------------+
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Herstellen einer Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung von der Befehlszeile aus: AWS CLI und psql-Client

Sie können über die Befehlszeile eine Verbindung mit einer Amazon-RDS-for-PostgreSQL-DB-Instance einem mit der AWS CLI und dem psql-Befehlszeilen-Tool herstellen, wie nachfolgend beschrieben.

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

> **Note**
>
> Informationen zum Herstellen einer Verbindung mit Ihrer Datenbank mithilfe von pgAdmin mit IAM-Authentifizierung finden Sie im Blogbeitrag Using IAM authentication to connect with pgAdmin Amazon Aurora PostgreSQL or Amazon RDS for PostgreSQL.

Themen

- Generieren eines IAM-Authentifizierungstokens
- Verbinden mit einem Amazon RDS-PostgreSQL-Instance und

Generieren eines IAM-Authentifizierungstokens

Das Authentifizierungstoken besteht aus hunderten von Zeichen, wodurch es in der Befehlszeile unhandlich wird. Eine Möglichkeit zur Umgehung dieses Problems besteht darin, das Token in einer Umgebungsvariable zu speichern und dann bei der Verbindungsherstellung diese Variable zu verwenden. Im folgenden Beispiel wird gezeigt, wie Sie die AWS CLI verwenden, um ein signiertes Authentifizierungstoken mithilfe des `generate-db-auth-token`-Befehls zu erzeugen und dieses anschließend in einer `PGPASSWORD`-Umgebungsvariablen zu speichern.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"
```

In diesem Beispiel lauten die Parameter für den `generate-db-auth-token`-Befehl folgendermaßen:

- `--hostname`– Der Hostname des DB-Instance-, auf den Sie zugreifen möchten.
- `--port` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `--region` – Die AWS-Region, in der die DB-Instance ausgeführt wird.
- `--username` – Das Datenbankkonto, auf das Sie zugreifen möchten.

Die ersten Zeichen des generierten Tokens sehen folgendermaßen aus.

```
rdspostgres.123456789012.us-west-2.rds.amazonaws.com:5432/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

Verbinden mit einem Amazon RDS-PostgreSQL-Instance und

Das allgemeine Format zur Herstellung einer Verbindung mithilfe von psql wird nachfolgend dargestellt.

```
psql "host=hostName port=portNumber sslmode=verify-full
 sslrootcert=full_path_to_ssl_certificate dbname=DBName user=userName
 password=authToken"
```

Dabei werden die folgenden Parameter verwendet:

- `host`– Der Hostname des DB-Instance-, auf den Sie zugreifen möchten.
- `port` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `sslmode` – Der zu verwendende SSL-Modus.

  Wenn Sie `sslmode=verify-full` verwenden, prüft die SSL-Verbindung den Endpunkt der DB-Instance gegen den Endpunkt des SSL-Zertifikats.
- `sslrootcert` – Die SSL-Zertifikatsdatei, die den öffentlichen Schlüssel enthält

  Weitere Informationen finden Sie unter Verwenden von SSL mit einer PostgreSQL-DB-Instance.

  Zum Download eines SSL-Zertifikats siehe .
- `dbname` – Die Datenbank, auf die Sie zugreifen möchten.
- `user` – Das Datenbankkonto, auf das Sie zugreifen möchten.
- `password` – Ein signiertes IAM-Authentifizierungstoken.

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

Das folgende Beispiel zeigt die Verwendung von psql für die Verbindung. Im Beispiel verwendet psql die Umgebungsvariable RDSHOST für den Host und die Umgebungsvariable PGPASSWORD für das generierte Token. Zudem ist */sample_dir/* der vollständige Pfad zur SSL-Zertifikatsdatei, die den öffentlichen Schlüssel enthält.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"

psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/global-
bundle.pem dbname=DBName user=jane_doe password=$PGPASSWORD"
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Herstellen einer Verbindung zu Ihrem DB-Instance- mithilfe der IAM-Authentifizierung und AWS SDK für .NET

Sie können sich mit dem AWS SDK für .NET wie nachfolgend beschrieben mit einer RDS-für-MariaDB-, RDS-für-MySQL- oder RDS-für-PostgreSQL-DB-Instance verbinden.

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

Beispiele

In den folgenden Beispielcodes wird gezeigt, wie Sie ein Authentifizierungstoken erzeugen und anschließend zum Verbinden mit einer DB-Instance verwenden.

Um dieses Codebeispiel auszuführen, benötigen Sie das AWS SDK für .NET, das sich auf der AWS-Seite befindet. Die `AWSSDK.CORE`- und die `AWSSDK.RDS`-Pakete sind erforderlich. Um eine Verbindung mit einer DB-Instance herzustellen, verwenden Sie den .NET-Datenbankkonnektor für die DB-Engine, z. B. MySqlConnector für MariaDB oder MySQL oder Npgsql für PostgreSQL.

Dieser Code stellt eine Verbindung mit einer MariaDB- oder MySQL-DB-Instance her. Ändern Sie die Werte der folgenden Parameter nach Bedarf.

- `server` – Der Endpunkt des DB-Instance-, auf den Sie zugreifen möchten.
- `user` – Das Datenbankkonto, auf das Sie zugreifen möchten.
- `database` – Die Datenbank, auf die Sie zugreifen möchten.
- `port` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `SslMode` – Der zu verwendende SSL-Modus.

  Wenn Sie `SslMode=Required` verwenden, prüft die SSL-Verbindung den Endpunkt der DB-Instance gegen den Endpunkt des SSL-Zertifikats.
- `SslCa` – Der vollständige Pfad zum SSL-Zertifikat für Amazon RDS

  Informationen zum Download eines Zertifikats finden Sie unter .

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

```
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using Amazon;

namespace ubuntu
{
  class Program
  {
    static void Main(string[] args)
    {
```

```
      var pwd =
 Amazon.RDS.Util.RDSAuthTokenGenerator.GenerateAuthToken(RegionEndpoint.USEast1,
 "mysqldb.123456789012.us-east-1.rds.amazonaws.com", 3306, "jane_doe");
      // for debug only Console.Write("{0}\n", pwd);  //this verifies the token is
 generated

      MySqlConnection conn = new MySqlConnection($"server=mysqldb.123456789012.us-
east-1.rds.amazonaws.com;user=jane_doe;database=mydB;port=3306;password={pwd};SslMode=Required;
      conn.Open();

      // Define a query
      MySqlCommand sampleCommand = new MySqlCommand("SHOW DATABASES;", conn);

      // Execute a query
      MySqlDataReader mysqlDataRdr = sampleCommand.ExecuteReader();

      // Read all rows and output the first column in each row
      while (mysqlDataRdr.Read())
        Console.WriteLine(mysqlDataRdr[0]);

      mysqlDataRdr.Close();
      // Close connection
      conn.Close();
    }
  }
}
```

Dieser Code stellt eine Verbindung zu einer PostgreSQL-DB-Instance her.

Ändern Sie die Werte der folgenden Parameter nach Bedarf.

- `Server` – Der Endpunkt des DB-Instance-, auf den Sie zugreifen möchten.
- `User ID` – Das Datenbankkonto, auf das Sie zugreifen möchten.
- `Database` – Die Datenbank, auf die Sie zugreifen möchten.
- `Port` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `SSL Mode` – Der zu verwendende SSL-Modus.

  Wenn Sie `SSL Mode=Required` verwenden, prüft die SSL-Verbindung den Endpunkt der DB-Instance gegen den Endpunkt des SSL-Zertifikats.
- `Root Certificate` – Der vollständige Pfad zum SSL-Zertifikat für Amazon RDS

Informationen zum Download eines Zertifikats finden Sie unter .

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

```
using System;
using Npgsql;
using Amazon.RDS.Util;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
 RDSAuthTokenGenerator.GenerateAuthToken("postgresmydb.123456789012.us-
east-1.rds.amazonaws.com", 5432, "jane_doe");
// for debug only Console.Write("{0}\n", pwd);  //this verifies the token is generated

            NpgsqlConnection conn = new
 NpgsqlConnection($"Server=postgresmydb.123456789012.us-east-1.rds.amazonaws.com;User
 Id=jane_doe;Password={pwd};Database=mydb;SSL Mode=Require;Root
 Certificate=full_path_to_ssl_certificate");
            conn.Open();

            // Define a query
                   NpgsqlCommand cmd = new NpgsqlCommand("select count(*) FROM
 pg_user", conn);

            // Execute a query
            NpgsqlDataReader dr = cmd.ExecuteReader();

            // Read all rows and output the first column in each row
            while (dr.Read())
                Console.Write("{0}\n", dr[0]);

            // Close connection
```

```
            conn.Close();
        }
    }
}
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung und der AWS SDK für Go

Sie können wie im Folgenden beschrieben eine Verbindung zu einer RDS for MariaDB-, MySQL- oder PostgreSQL-DB-Instance mit herstellen. AWS SDK für Go

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

Beispiele

Um diese Codebeispiele auszuführen, benötigen Sie den, der auf der AWS SDK für GoWebsite zu finden ist. AWS

Ändern Sie die Werte der folgenden Parameter nach Bedarf.

- `dbName` – Die Datenbank, auf die Sie zugreifen möchten.
- `dbUser` – Das Datenbankkonto, auf das Sie zugreifen möchten.
- `dbHost` – Der Endpunkt des DB-Instance-, auf den Sie zugreifen möchten.

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

- `dbPort` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `region`— Die AWS Region, in der der ausgeführt wird

Stellen Sie außerdem sicher, dass die importierten Bibliotheken im Beispielcode auf Ihrem System vorhanden sind.

> **Important**
>
> Die Beispiele in diesem Abschnitt verwenden den folgenden Code, um Anmeldeinformationen bereitzustellen, die von einer lokalen Umgebung aus auf eine Datenbank zugreifen:
> `creds := credentials.NewEnvCredentials()`
> Wenn Sie von einem AWS Service wie Amazon EC2 oder Amazon ECS auf eine Datenbank zugreifen, können Sie den Code durch den folgenden Code ersetzen:
> `sess := session.Must(session.NewSession())`
> `creds := sess.Config.Credentials`
> Wenn Sie diese Änderung vornehmen, stellen Sie sicher, dass Sie den folgenden Import hinzufügen:
> `"github.com/aws/aws-sdk-go/aws/session"`

Themen

- Verbindung mithilfe der IAM-Authentifizierung und V2 herstellen AWS SDK für Go
- Verbindung mithilfe der IAM-Authentifizierung und V1 herstellen. AWS SDK für Go

Verbindung mithilfe der IAM-Authentifizierung und V2 herstellen AWS SDK für Go

Sie können mithilfe der IAM-Authentifizierung und V2 eine Verbindung zu einem herstellen. AWS SDK für Go

In den folgenden Beispielcodes wird gezeigt, wie Sie ein Authentifizierungstoken erzeugen und anschließend zum Verbinden mit einer DB-Instance verwenden.

Dieser Code stellt eine Verbindung mit einer MariaDB- oder MySQL-DB-Instance her.

```
package main

import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
```

```
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/go-sql-driver/mysql"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 3306
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authenticationToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Dieser Code stellt eine Verbindung zu einer PostgreSQL-DB-Instance her.

```
package main
```

```
import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/lib/pq"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 5432
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authenticationToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
```

```
}
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Verbindung mithilfe der IAM-Authentifizierung und V1 herstellen. AWS SDK für Go

Sie können mithilfe der IAM-Authentifizierung und V1 eine Verbindung zu einem herstellen AWS SDK für Go

In den folgenden Beispielcodes wird gezeigt, wie Sie ein Authentifizierungstoken erzeugen und anschließend zum Verbinden mit einer DB-Instance verwenden.

Dieser Code stellt eine Verbindung mit einer MariaDB- oder MySQL-DB-Instance her.

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/go-sql-driver/mysql"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
```

```
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Dieser Code stellt eine Verbindung zu einer PostgreSQL-DB-Instance her.

```
package main

import (
 "database/sql"
 "fmt"

 "github.com/aws/aws-sdk-go/aws/credentials"
 "github.com/aws/aws-sdk-go/service/rds/rdsutils"
 _ "github.com/lib/pq"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )
```

```
    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung und der AWS SDK für Java

Sie können wie im Folgenden beschrieben eine Verbindung zu einer RDS for MariaDB-, MySQL- oder PostgreSQL-DB-Instance mit herstellen. AWS SDK für Java

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung
- Richten Sie das AWS SDK for Java ein

Beispiele zur Verwendung des SDK for Java 2.x finden Sie unter Amazon RDS-Beispiele mit SDK for Java 2.x. Sie können auch den AWS Advanced JDBC Wrapper verwenden, siehe AWS Advanced JDBC Wrapper Dokumentation.

Themen

- Generieren eines IAM-Authentifizierungstokens
- Manuelles Erzeugen eines IAM-Authentifizierungstokens
- Herstellen der Verbindung zu einem DB-Instance-

Generieren eines IAM-Authentifizierungstokens

Wenn Sie Programme mit dem schreiben AWS SDK für Java, können Sie mithilfe der Klasse ein signiertes Authentifizierungstoken erhalten. `RdsIamAuthTokenGenerator` Für die Verwendung dieser Klasse müssen Sie AWS Anmeldeinformationen angeben. Dazu erstellen Sie eine Instanz der `DefaultAWSCredentialsProviderChain` Klasse. `DefaultAWSCredentialsProviderChain`verwendet den ersten AWS Zugriffsschlüssel und den ersten geheimen Schlüssel, den es in der standardmäßigen Anbieterkette für Anmeldeinformationen findet. Weitere Informationen über AWS -Zugriffsschlüssel finden Sie unter Verwalten von Zugriffsschlüsseln für Benutzer.

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

Nachdem Sie eine Instance von `RdsIamAuthTokenGenerator` erstellt haben, können Sie das `getAuthToken`-Verfahren aufrufen, um ein signiertes Token zu erhalten. Geben Sie die AWS -Region, den Hostnamen, die Portnummer und den Benutzernamen an. Der folgende Beispielcode veranschaulicht diese Vorgehensweise.

```
package com.amazonaws.codesamples;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;

public class GenerateRDSAuthToken {

    public static void main(String[] args) {

     String region = "us-west-2";
     String hostname = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
     String port = "3306";
     String username = "jane_doe";

     System.out.println(generateAuthToken(region, hostname, port, username));
    }
```

```
    static String generateAuthToken(String region, String hostName, String port, String
 username) {

     RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
      .credentials(new DefaultAWSCredentialsProviderChain())
      .region(region)
      .build();

     String authToken = generator.getAuthToken(
      GetIamAuthTokenRequest.builder()
      .hostname(hostName)
      .port(Integer.parseInt(port))
      .userName(username)
      .build());

     return authToken;
    }

}
```

Manuelles Erzeugen eines IAM-Authentifizierungstokens

Der einfachste Weg in Java, um ein Authentifizierungstoken zu erzeugen, ist die Verwendung von `RdsIamAuthTokenGenerator`. Diese Klasse erstellt ein Authentifizierungstoken für Sie und signiert es dann mit der AWS Signaturversion 4. Weitere Informationen finden Sie unter Signaturprozess mit Signaturversion 4 im Allgemeine AWS-Referenz.

Sie können allerdings das Authentifizierungstoken auch manuell erstellen und signieren, wie im folgenden Beispielcode dargestellt.

```
package com.amazonaws.codesamples;

import com.amazonaws.SdkClientException;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.SigningAlgorithm;
import com.amazonaws.util.BinaryUtils;
import org.apache.commons.lang3.StringUtils;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
import java.util.SortedMap;
import java.util.TreeMap;

import static com.amazonaws.auth.internal.SignerConstants.AWS4_TERMINATOR;
import static com.amazonaws.util.StringUtils.UTF8;

public class CreateRDSAuthTokenManually {
    public static String httpMethod = "GET";
    public static String action = "connect";
    public static String canonicalURIParameter = "/";
    public static SortedMap<String, String> canonicalQueryParameters = new TreeMap();
    public static String payload = StringUtils.EMPTY;
    public static String signedHeader = "host";
    public static String algorithm = "AWS4-HMAC-SHA256";
    public static String serviceName = "rds-db";
    public static String requestWithoutSignature;

    public static void main(String[] args) throws Exception {

        String region = "us-west-2";
        String instanceName = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        Date now = new Date();
        String date = new SimpleDateFormat("yyyyMMdd").format(now);
        String dateTimeStamp = new
 SimpleDateFormat("yyyyMMdd'T'HHmmss'Z'").format(now);
       DefaultAWSCredentialsProviderChain creds = new
 DefaultAWSCredentialsProviderChain();
    String awsAccessKey = creds.getCredentials().getAWSAccessKeyId();
    String awsSecretKey = creds.getCredentials().getAWSSecretKey();
       String expiryMinutes = "900";

       System.out.println("Step 1:  Create a canonical request:");
       String canonicalString = createCanonicalString(username, awsAccessKey, date,
 dateTimeStamp, region, expiryMinutes, instanceName, port);
       System.out.println(canonicalString);
       System.out.println();

       System.out.println("Step 2:  Create a string to sign:");
       String stringToSign = createStringToSign(dateTimeStamp, canonicalString,
 awsAccessKey, date, region);
```

```
        System.out.println(stringToSign);
        System.out.println();

        System.out.println("Step 3:  Calculate the signature:");
        String signature = BinaryUtils.toHex(calculateSignature(stringToSign,
newSigningKey(awsSecretKey, date, region, serviceName)));
        System.out.println(signature);
        System.out.println();

        System.out.println("Step 4:  Add the signing info to the request");

        System.out.println(appendSignature(signature));
        System.out.println();

    }

    //Step 1: Create a canonical request date should be in format YYYYMMDD and dateTime
should be in format YYYYMMDDTHHMMSSZ
    public static String createCanonicalString(String user, String accessKey, String
date, String dateTime, String region, String expiryPeriod, String hostName, String
port) throws Exception {
        canonicalQueryParameters.put("Action", action);
        canonicalQueryParameters.put("DBUser", user);
        canonicalQueryParameters.put("X-Amz-Algorithm", "AWS4-HMAC-SHA256");
        canonicalQueryParameters.put("X-Amz-Credential", accessKey + "%2F" + date +
"%2F" + region + "%2F" + serviceName + "%2Faws4_request");
        canonicalQueryParameters.put("X-Amz-Date", dateTime);
        canonicalQueryParameters.put("X-Amz-Expires", expiryPeriod);
        canonicalQueryParameters.put("X-Amz-SignedHeaders", signedHeader);
        String canonicalQueryString = "";
        while(!canonicalQueryParameters.isEmpty()) {
            String currentQueryParameter = canonicalQueryParameters.firstKey();
            String currentQueryParameterValue =
canonicalQueryParameters.remove(currentQueryParameter);
            canonicalQueryString = canonicalQueryString + currentQueryParameter + "=" +
currentQueryParameterValue;
            if (!currentQueryParameter.equals("X-Amz-SignedHeaders")) {
                canonicalQueryString += "&";
            }
        }
        String canonicalHeaders = "host:" + hostName + ":" + port + '\n';
        requestWithoutSignature = hostName + ":" + port + "/?" + canonicalQueryString;

        String hashedPayload = BinaryUtils.toHex(hash(payload));
```

```
        return httpMethod + '\n' + canonicalURIParameter + '\n' + canonicalQueryString
 + '\n' + canonicalHeaders + '\n' + signedHeader + '\n' + hashedPayload;

    }

    //Step 2: Create a string to sign using sig v4
    public static String createStringToSign(String dateTime, String canonicalRequest,
 String accessKey, String date, String region) throws Exception {
        String credentialScope = date + "/" + region + "/" + serviceName + "/
aws4_request";
        return algorithm + '\n' + dateTime + '\n' + credentialScope + '\n' +
 BinaryUtils.toHex(hash(canonicalRequest));

    }

    //Step 3: Calculate signature
    /**
     * Step 3 of the &AWS; Signature version 4 calculation. It involves deriving
     * the signing key and computing the signature. Refer to
     * http://docs.aws.amazon
     * .com/general/latest/gr/sigv4-calculate-signature.html
     */
    public static byte[] calculateSignature(String stringToSign,
                                            byte[] signingKey) {
        return sign(stringToSign.getBytes(Charset.forName("UTF-8")), signingKey,
                SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(byte[] data, byte[] key,
                          SigningAlgorithm algorithm) throws SdkClientException {
        try {
            Mac mac = algorithm.getMac();
            mac.init(new SecretKeySpec(key, algorithm.toString()));
            return mac.doFinal(data);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    public static byte[] newSigningKey(String secretKey,
                                   String dateStamp, String regionName, String
 serviceName) {
```

```
        byte[] kSecret = ("AWS4" + secretKey).getBytes(Charset.forName("UTF-8"));
        byte[] kDate = sign(dateStamp, kSecret, SigningAlgorithm.HmacSHA256);
        byte[] kRegion = sign(regionName, kDate, SigningAlgorithm.HmacSHA256);
        byte[] kService = sign(serviceName, kRegion,
                SigningAlgorithm.HmacSHA256);
        return sign(AWS4_TERMINATOR, kService, SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(String stringData, byte[] key,
                      SigningAlgorithm algorithm) throws SdkClientException {
        try {
            byte[] data = stringData.getBytes(UTF8);
            return sign(data, key, algorithm);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    //Step 4: append the signature
    public static String appendSignature(String signature) {
        return requestWithoutSignature + "&X-Amz-Signature=" + signature;
    }

    public static byte[] hash(String s) throws Exception {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            md.update(s.getBytes(UTF8));
            return md.digest();
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to compute hash while signing request: "
                            + e.getMessage(), e);
        }
    }
}
```

Herstellen der Verbindung zu einem DB-Instance-

Im folgenden Codebeispiel wird gezeigt, wie Sie ein Authentifizierungstoken erzeugen und anschließend zum Verbinden mit einer Instance für die Ausführung mit MariaDB oder MySQL verwenden können.

Um dieses Codebeispiel auszuführen, benötigen Sie den AWS SDK für Java, auf der AWS Site zu finden. Außerdem benötigen Sie Folgendes:

- MySQL Connector/J. Dieses Codebeispiel wurde mit `mysql-connector-java-5.1.33-bin.jar` getestet.
- Ein Zwischenzertifikat für Amazon RDS , das für eine AWS Region spezifisch ist. (Weitere Informationen finden Sie unter .) Während der Laufzeit sucht der Class Loader das Zertifikat in demselben Verzeichnis, in dem sich dieser Java-Beispielcode befindet, damit er es finden kann.
- Ändern Sie die Werte der folgenden Parameter nach Bedarf.
  - `RDS_INSTANCE_HOSTNAME` – Der Hostname des DB-Instance-, auf den Sie zugreifen möchten.
  - `RDS_INSTANCE_PORT` – Die Nummer des Ports, der für die Verbindung zur PostgreSQL-DB-Instance verwendet wird.
  - `REGION_NAME`— Die AWS Region, in der der ausgeführt wird.
  - `DB_USER` – Das Datenbankkonto, auf das Sie zugreifen möchten.
  - `SSL_CERTIFICATE`— Ein regionsspezifisches SSL-Zertifikat für Amazon RDS . AWS

    Informationen zum Herunterladen eines Zertifikats für Ihre AWS -Region finden Sie unter . Speichern Sie das SSL-Zertifikat in demselben Verzeichnis wie diese Java-Programmdatei ab, damit der Class Loader das Zertifikat während der Laufzeit finden kann.

In diesem Codebeispiel werden AWS Anmeldeinformationen aus der standardmäßigen Anbieterkette für Anmeldeinformationen abgerufen.

> **Note**
>
> Geben Sie aus Sicherheitsgründen für `DEFAULT_KEY_STORE_PASSWORD` ein anderes Passwort als hier angegeben an.

```
package com.amazonaws.samples;

import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
```

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

import java.net.URL;

public class IAMDatabaseAuthenticationTester {
    //&AWS; Credentials of the IAM user with policy enabling IAM Database Authenticated
 access to the db by the db user.
    private static final DefaultAWSCredentialsProviderChain creds = new
 DefaultAWSCredentialsProviderChain();
    private static final String AWS_ACCESS_KEY =
 creds.getCredentials().getAWSAccessKeyId();
    private static final String AWS_SECRET_KEY =
 creds.getCredentials().getAWSSecretKey();

    //Configuration parameters for the generation of the IAM Database Authentication
 token
    private static final String RDS_INSTANCE_HOSTNAME = "rdsmysql.123456789012.us-
west-2.rds.amazonaws.com";
    private static final int RDS_INSTANCE_PORT = 3306;
    private static final String REGION_NAME = "us-west-2";
    private static final String DB_USER = "jane_doe";
    private static final String JDBC_URL = "jdbc:mysql://" + RDS_INSTANCE_HOSTNAME +
 ":" + RDS_INSTANCE_PORT;

    private static final String SSL_CERTIFICATE = "rds-ca-2019-us-west-2.pem";

    private static final String KEY_STORE_TYPE = "JKS";
    private static final String KEY_STORE_PROVIDER = "SUN";
    private static final String KEY_STORE_FILE_PREFIX = "sys-connect-via-ssl-test-
cacerts";
    private static final String KEY_STORE_FILE_SUFFIX = ".jks";
    private static final String DEFAULT_KEY_STORE_PASSWORD = "changeit";
```

```
    public static void main(String[] args) throws Exception {
        //get the connection
        Connection connection = getDBConnectionUsingIam();

        //verify the connection is successful
        Statement stmt= connection.createStatement();
        ResultSet rs=stmt.executeQuery("SELECT 'Success!' FROM DUAL;");
        while (rs.next()) {
             String id = rs.getString(1);
            System.out.println(id); //Should print "Success!"
        }

        //close the connection
        stmt.close();
        connection.close();

        clearSslProperties();

    }

    /**
     * This method returns a connection to the db instance authenticated using IAM
Database Authentication
     * @return
     * @throws Exception
     */
    private static Connection getDBConnectionUsingIam() throws Exception {
        setSslProperties();
        return DriverManager.getConnection(JDBC_URL, setMySqlConnectionProperties());
    }

    /**
     * This method sets the mysql connection properties which includes the IAM Database
Authentication token
     * as the password. It also specifies that SSL verification is required.
     * @return
     */
    private static Properties setMySqlConnectionProperties() {
        Properties mysqlConnectionProperties = new Properties();
        mysqlConnectionProperties.setProperty("verifyServerCertificate","true");
        mysqlConnectionProperties.setProperty("useSSL", "true");
        mysqlConnectionProperties.setProperty("user",DB_USER);
        mysqlConnectionProperties.setProperty("password",generateAuthToken());
        return mysqlConnectionProperties;
```

```
    }

    /**
     * This method generates the IAM Auth Token.
     * An example IAM Auth Token would look like follows:
     * btusi123.cmz7kenwo2ye.rds.cn-north-1.amazonaws.com.cn:3306/?
Action=connect&DBUser=iamtestuser&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20171003T010726Z&X-Amz-SignedHeaders=host&X-Amz-Expires=899&X-Amz-
Credential=AKIAPFXHGVDI5RNFO4AQ%2F20171003%2Fcn-north-1%2Frds-db%2Faws4_request&X-Amz-
Signature=f9f45ef96c1f770cdad11a53e33ffa4c3730bc03fdee820cfdf1322eed15483b
     * @return
     */
    private static String generateAuthToken() {
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(AWS_ACCESS_KEY,
 AWS_SECRET_KEY);

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
                .credentials(new
 AWSStaticCredentialsProvider(awsCredentials)).region(REGION_NAME).build();
        return generator.getAuthToken(GetIamAuthTokenRequest.builder()

 .hostname(RDS_INSTANCE_HOSTNAME).port(RDS_INSTANCE_PORT).userName(DB_USER).build());
    }

    /**
     * This method sets the SSL properties which specify the key store file, its type
 and password:
     * @throws Exception
     */
    private static void setSslProperties() throws Exception {
        System.setProperty("javax.net.ssl.trustStore", createKeyStoreFile());
        System.setProperty("javax.net.ssl.trustStoreType", KEY_STORE_TYPE);
        System.setProperty("javax.net.ssl.trustStorePassword",
 DEFAULT_KEY_STORE_PASSWORD);
    }

    /**
     * This method returns the path of the Key Store File needed for the SSL
 verification during the IAM Database Authentication to
     * the db instance.
     * @return
     * @throws Exception
     */
    private static String createKeyStoreFile() throws Exception {
```

```
        return createKeyStoreFile(createCertificate()).getPath();
    }

    /**
     *  This method generates the SSL certificate
     * @return
     * @throws Exception
     */
    private static X509Certificate createCertificate() throws Exception {
        CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
        URL url = new File(SSL_CERTIFICATE).toURI().toURL();
        if (url == null) {
            throw new Exception();
        }
        try (InputStream certInputStream = url.openStream()) {
            return (X509Certificate) certFactory.generateCertificate(certInputStream);
        }
    }

    /**
     * This method creates the Key Store File
     * @param rootX509Certificate - the SSL certificate to be stored in the KeyStore
     * @return
     * @throws Exception
     */
    private static File createKeyStoreFile(X509Certificate rootX509Certificate) throws
Exception {
        File keyStoreFile = File.createTempFile(KEY_STORE_FILE_PREFIX,
KEY_STORE_FILE_SUFFIX);
        try (FileOutputStream fos = new FileOutputStream(keyStoreFile.getPath())) {
            KeyStore ks = KeyStore.getInstance(KEY_STORE_TYPE, KEY_STORE_PROVIDER);
            ks.load(null);
            ks.setCertificateEntry("rootCaCertificate", rootX509Certificate);
            ks.store(fos, DEFAULT_KEY_STORE_PASSWORD.toCharArray());
        }
        return keyStoreFile;
    }

    /**
     * This method clears the SSL properties.
     * @throws Exception
     */
    private static void clearSslProperties() throws Exception {
           System.clearProperty("javax.net.ssl.trustStore");
```

```
            System.clearProperty("javax.net.ssl.trustStoreType");
            System.clearProperty("javax.net.ssl.trustStorePassword");
    }

}
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Herstellen einer Verbindung zu Ihrem mithilfe der IAM-Authentifizierung und der AWS SDK für Python (Boto3)

Sie können wie im Folgenden beschrieben eine Verbindung zu einer RDS for MariaDB-, MySQL- oder PostgreSQL-DB-Instance mit herstellen. AWS SDK für Python (Boto3)

Voraussetzungen

Die folgenden Voraussetzungen gelten für die Verbindung mit Ihrem DB-Instance- mithilfe der IAM-Authentifizierung:

- Aktivieren und Deaktivieren der IAM-Datenbank-Authentifizierung
- Erstellen und Verwenden einer IAM-Richtlinie für den IAM-Datenbankzugriff
- Erstellen eines Datenbankkontos mithilfe der IAM-Authentifizierung

Stellen Sie außerdem sicher, dass die importierten Bibliotheken im Beispielcode auf Ihrem System vorhanden sind.

Beispiele

Die Codebeispiele verwenden Profile für freigegebene Anmeldeinformationen. Informationen zur Angabe von Anmeldeinformationen finden Sie in der Dokumentation unter Anmeldeinformationen. AWS SDK für Python (Boto3)

In den folgenden Beispielcodes wird gezeigt, wie Sie ein Authentifizierungstoken erzeugen und anschließend zum Verbinden mit einer DB-Instance verwenden.

Um dieses Codebeispiel auszuführen, benötigen Sie den AWS SDK für Python (Boto3), auf der AWS Site zu finden.

Ändern Sie die Werte der folgenden Parameter nach Bedarf.

- `ENDPOINT` – Der Endpunkt des DB-Instance-, auf den Sie zugreifen möchten.
- `PORT` – Die Nummer des Ports, der für die Verbindung mit dem DB-Instance- verwendet wird.
- `USER` – Das Datenbankkonto, auf das Sie zugreifen möchten.
- `REGION`— Die AWS Region, in der der ausgeführt wird
- `DBNAME` – Die Datenbank, auf die Sie zugreifen möchten.
- `SSLCERTIFICATE` – Der vollständige Pfad zum SSL-Zertifikat für Amazon RDS

  Geben Sie für `ssl_ca` ein SSL-Zertifikat an. Zum Download eines SSL-Zertifikats siehe .

> **Note**
>
> Sie können keinen benutzerdefinierten Route-53-DNS-Eintrag anstelle des DB-Instance-Endpunkts verwenden, um das Authentifizierungstoken zu generieren.

Dieser Code stellt eine Verbindung mit einer MariaDB- oder MySQL-DB-Instance her.

Bevor Sie diesen Code ausführen, installieren Sie den PyMy SQL-Treiber, indem Sie den Anweisungen im Python-Paketindex folgen.

```
import pymysql
import sys
import boto3
import os

ENDPOINT="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='default')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)
```

```
try:
    conn =
 pymysql.connect(auth_plugin_map={'mysql_clear_password':None},host=ENDPOINT,
 user=USER, password=token, port=PORT, database=DBNAME, ssl_ca='SSLCERTIFICATE',
 ssl_verify_identity=True, ssl_verify_cert=True)
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Dieser Code stellt eine Verbindung zu einer PostgreSQL-DB-Instance her.

Bevor Sie diesen Code ausführen, installieren Sie `psycopg2`, indem Sie die Anweisungen in der Psycopg-Dokumentation befolgen.

```
import psycopg2
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
PORT="5432"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn = psycopg2.connect(host=ENDPOINT, port=PORT, database=DBNAME, user=USER,
 password=token, sslrootcert="SSLCERTIFICATE")
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
```

```
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Informationen dazu, wie Sie über einen Proxy eine Verbindung mit einer DB-Instance herstellen, finden Sie unter Herstellen einer Verbindung mit einem Proxy mithilfe der IAM-Authentifizierung.

Problembehandlung bei der IAM-DB-Authentifizierung

Im Folgenden finden Sie Ideen zur Fehlerbehebung für einige häufig auftretende Probleme mit der IAM-DB-Authentifizierung sowie Informationen zu CloudWatch Protokollen und Metriken für die IAM-DB-Authentifizierung.

Exportieren von Fehlerprotokollen für die IAM-DB-Authentifizierung in Logs CloudWatch

Fehlerprotokolle für die IAM-DB-Authentifizierung werden auf dem Datenbank-Host gespeichert, und Sie können diese Protokolle in Ihr Logs-Konto CloudWatch exportieren. Verwenden Sie die Protokolle und Behebungsmethoden auf dieser Seite, um Probleme mit der IAM-DB-Authentifizierung zu beheben.

Sie können CloudWatch Protokollexporte über die Konsole und die RDS-API in Logs aktivieren. AWS CLI Anweisungen für die Konsole finden Sie unterDatenbankprotokolle in Amazon CloudWatch Logs veröffentlichen.

Verwenden Sie den folgenden Befehl, um Ihre CloudWatch IAM-DB-Authentifizierungsfehlerprotokolle in Logs zu exportieren AWS CLI, wenn Sie einen aus dem erstellen:

```
aws rds create-db-instance --db-instance-identifier mydbinstance \
--region us-east-1 \
--db-instance-class db.t3.large \
--allocated-storage 50 \
--engine postgres \
--engine-version 16 \
--port 5432 \
--master-username master \
--master-user-password password \
--publicly-accessible \
--enable-iam-database-authentication \
--enable-cloudwatch-logs-exports=iam-db-auth-error
```

Verwenden Sie den folgenden Befehl, um Ihre CloudWatch IAM-DB-Authentifizierungsfehlerprotokolle nach Logs zu exportieren AWS CLI, wenn Sie ein aus dem ändern:

```
aws rds modify-db-instance --db-instance-identifier mydbinstance \
--region us-east-1 \
--cloudwatch-logs-export-configuration '{"EnableLogTypes":["iam-db-auth-error"]}'
```

Um zu überprüfen, ob Ihr CloudWatch IAM-DB-Authentifizierungsprotokolle nach Logs exportiert, überprüfen Sie, ob der `EnabledCloudwatchLogsExports` Parameter `iam-db-auth-error` in der Ausgabe des Befehls auf gesetzt ist. `describe-db-instances`

```
aws rds describe-db-instances --region us-east-1 --db-instance-identifier mydbinstance
            ...

             "EnabledCloudwatchLogsExports": [
                "iam-db-auth-error"
            ],
            ...
```

IAM-DB-Authentifizierungsmetriken CloudWatch

Amazon RDS liefert nahezu in Echtzeit Metriken zur IAM-DB-Authentifizierung für Ihr Amazon-Konto. CloudWatch In der folgenden Tabelle sind die IAM-DB-Authentifizierungsmetriken aufgeführt, die mit folgenden Methoden verfügbar sind: CloudWatch

Metrik	Beschreibung
`IamDbAuthConnectionRequests`	Gesamtzahl der Verbindungsanfragen, die mit der IAM-DB-Authentifizierung gestellt wurden.
`IamDbAuthConnectionSuccess`	Gesamtzahl der erfolgreichen IAM-DB-Au thentifizierungsanfragen.
`IamDbAuthConnectionFailure`	Gesamtzahl der fehlgeschlagenen IAM-DB-Au thentifizierungsanfragen.
`IamDbAuthConnectionFailureI nvalidToken`	Gesamtzahl der fehlgeschlagenen IAM-DB-Authentifizierungsanfragen aufgrund eines ungültigen Tokens.

Metrik	Beschreibung
`IamDbAuthConnectionFailureInsufficientPermissions`	Gesamtzahl der fehlgeschlagenen IAM-DB-Authentifizierungsanfragen aufgrund falscher Richtlinien oder Berechtigungen.
`IamDbAuthConnectionFailureThrottling`	Gesamtzahl der fehlgeschlagenen IAM-DB-Authentifizierungsanfragen aufgrund der Drosselung der IAM-DB-Authentifizierung.
`IamDbAuthConnectionFailureServerError`	Gesamtzahl der fehlgeschlagenen IAM-DB-Authentifizierungsanfragen aufgrund eines internen Serverfehlers in der IAM-DB-Authentifizierungsfunktion.

Häufige Probleme und Lösungen

Bei der Verwendung der IAM-DB-Authentifizierung können die folgenden Probleme auftreten. Verwenden Sie die Schritte zur Problembehebung in der Tabelle, um die Probleme zu lösen:

Fehler	Metrik (en)	Ursache	Lösung

`[ERROR] Failed to authenticate the connection request for user` *`db_user`* `because the provided token is malformed or otherwise invalid. (Status Code: 400, Error Code: InvalidToken)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureInvalidToken`
		Das IAM-DB-Authentifizierungstoken in der Verbindungsanforderung ist entweder kein gültiges SigV4A-Token oder es ist nicht richtig formatiert.	Überprüfen Sie Ihre Strategie zur Token-Generierung in Ihrer Anwendung. Stellen Sie in einigen Fällen sicher, dass Sie das Token mit einer gültigen Formatierung übergeben. Wenn Sie das Token kürzen (oder eine falsche Zeichenkettenformatierung), wird das Token ungültig.

Fehler	Metrik (en)	Ursache	Lösung

`[ERROR] Failed to authenticate the connection request for user db_user because the token age is longer than 15 minutes. (Status Code: 400, Error Code:ExpiredToken)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureInvalidToken`
		Das IAM-DB-Authentifizierungstoken ist abgelaufen. Tokens sind nur 15 Minuten gültig.	Überprüfen Sie Ihre Logik zur Token-Zwischenspeicherung in Ihrer Anwendung zur and/or Token-Wiederverwendung. Sie sollten Token, die älter als 15 Minuten sind, nicht wiederverwenden.

Fehler	Metrik (en)	Ursache	Lösung

`[ERROR] Failed to authorize the connection request for user` *`db_user`* `because the IAM policy assumed by the caller 'arn:aws:sts::123456789012:assumed-role/ <RoleName>/ <RoleSession>' is not authorized to perform `rds-db:connect` on the DB instance. (Status Code: 403, Error Code:NotAuthorized)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureInsufficientPermissions`
		Dies kann folgende Ursachen haben:
		• Die von der Anwendung angenommene IAM-Richtlinie autorisiert die Aktion nicht. `rds-db:connect`
		• Sie gehen von der falschen role/policy Methode aus *`db_user`*, um eine Verbindung zur Datenbank herzustellen.
		• Sie gehen von der richtigen Richtlinie für aus *`db_user`*, stellen aber keine Verbindung zur richtigen Datenbank her.
			Vergewissern Sie sich, dass die and/or IAM-Rollenrichtlinie, von der Sie ausgehen, in Ihrer Anwendung gilt. Stellen Sie sicher, dass Sie für die Generierung des Tokens dieselbe Richtlinie verwenden wie für die Verbindung mit der Datenbank.

Fehler	Metrik (en)	Ursache	Lösung

`[ERROR] Failed to authorize the connection request for user db_user due to IAM DB authentication throttling. (Status Code: 429, Error Code: ThrottlingException)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureThrottling`
		Sie stellen in kurzer Zeit zu viele Verbindungsanfragen an Ihre Datenbank. Das Limit für die Drosselung der IAM-DB-Authentifizierung liegt bei 200 Verbindungen pro Sekunde.	Reduzieren Sie die Geschwindigkeit, mit der neue Verbindungen hergestellt werden, mit der IAM-Authentifizierung. Erwägen Sie die Implementierung von Verbindungspooling mithilfe von RDS Proxy, um bestehende Verbindungen in Ihrer Anwendung wiederzuverwenden.

`[ERROR] Failed to authorize the connection request for user db_user due to an internal IAM DB authentication error. (Status Code: 500, Error Code: InternalError)`	`IamDbAuthConnectionFailure`
	
	`IamDbAuthConnectionFailureThrottling`
		Bei der Autorisierung der DB-Verbindung mit der IAM-DB-Authentifizierung ist ein interner Fehler aufgetreten.	Wenden Sie sich an, um das https://aws.amazon.com/premiumsupport/ Problem zu untersuchen.

Fehlerbehebung für Amazon RDS-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die beim Arbeiten mit Amazon RDS und IAM auftreten könnten.

Themen

- Ich bin nicht autorisiert, eine Aktion in Amazon RDS auszuführen

- Ich bin nicht zum Durchführen von iam:PassRole berechtigt
- Ich möchte Personen außerhalb meines AWS-Kontos Zugriff auf meine Amazon-RDS-Ressourcen erteilen

Ich bin nicht autorisiert, eine Aktion in Amazon RDS auszuführen

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der Benutzer `mateojackson` versucht, die Konsole zum Anzeigen von Details zu einem *`Widget`* zu verwenden, jedoch nicht über `rds:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 rds:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *`my-example-widget`* auf die Ressource `rds:`*`GetWidget`* zugreifen zu können.

Ich bin nicht zum Durchführen von iam:PassRole berechtigt

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der Aktion `iam:PassRole` autorisiert sind, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt. Bitten Sie diese Person um die Aktualisierung Ihrer Richtlinien, um eine Rolle an Amazon RDS übergeben zu können.

Einige AWS-Services gewähren Ihnen die Berechtigung zur Übergabe einer vorhandenen Rolle an diesen Service, statt eine neue Service-Rolle oder serviceverknüpfte Rolle erstellen zu müssen. Hierfür benötigen Sie Berechtigungen zur Übergabe der Rolle an den Service.

Der folgende Beispielfehler tritt auf, wenn ein Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon RDS auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Service-Rolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall bittet Mary ihren Administrator um die Aktualisierung ihrer Richtlinien, um die Aktion `iam:PassRole` ausführen zu können.

Ich möchte Personen außerhalb meines AWS-Kontos Zugriff auf meine Amazon-RDS-Ressourcen erteilen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen finden Sie hier:

- Informationen dazu, ob Amazon RDS diese Funktionen unterstützt, finden Sie unter So funktioniert RDSAmazon mit IAM.
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konten im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Drittkonten Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter Gewähren des Zugriffs auf AWS-Konten von externen Benutzern im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Protokollieren und Überwachen in Amazon RDS

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon RDS, und Ihren AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können. AWS bietet verschiedene Tools zur Überwachung Ihrer Amazon RDS Amazon und zur Reaktion auf potenzielle Vorfälle:

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Amazon-Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet, wird eine Benachrichtigung an ein Amazon SNS SNS-Thema oder eine AWS Auto Scaling Richtlinie gesendet. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

AWS CloudTrail Protokolle

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon RDS Amazon ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für Amazon RDS als Ereignisse, einschließlich Aufrufe von der Konsole und von Codeaufrufen an Amazon RDS-API-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon RDS Amazon gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln. Weitere Informationen finden Sie unter überwachen Amazon RDS API ruft an AWS CloudTrail.

Verbesserte Überwachung

Amazon RDS bietet Metriken in Echtzeit für das Betriebssystem (OS), auf dem Ihr läuft. Sie können die Metriken für Ihren mit der Konsole anzeigen oder die Enhanced Monitoring JSON-Ausgabe von Amazon CloudWatch Logs in einem Überwachungssystem Ihrer Wahl verwenden. Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).

Amazon RDS Performance Insights

Performance Insights erweitert die bestehenden Amazon RDS, um die Leistung Ihrer Datenbank zu veranschaulichen und Sie bei der Analyse aller Probleme zu unterstützen, die sich darauf auswirken. Mit dem Performance Insights-Dashboard können Sie die Datenbankauslastung visualisieren und die Auslastung nach Wartezeiten, SQL-Anweisungen, Hosts oder Benutzern filtern. Weitere Informationen finden Sie unter Überwachung mit Performance Insights auf Amazon RDS.

Datenbankprotokolle

Sie können Datenbankprotokolle mithilfe der, oder RDS-API anzeigen AWS Management Console AWS CLI, herunterladen und ansehen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS Amazon.

Amazon RDS

Amazon RDS bietet automatisierte Empfehlungen für Datenbankressourcen. Diese Empfehlungen dienen als Leitfaden für bewährte Verfahren, indem sie die Konfiguration, Nutzung und Leistungsdaten des analysieren. Weitere Informationen finden Sie unter Empfehlungen von Amazon RDS.

Amazon RDS

Amazon RDS verwendet den Amazon Simple Notification Service (Amazon SNS), um Benachrichtigungen bereitzustellen, wenn ein Amazon RDS Amazon eintritt. Diese Benachrichtigungen können in jedem von Amazon SNS für eine AWS Region unterstützten Benachrichtigungsformular erfolgen, z. B. in einer E-Mail, einer Textnachricht oder einem Anruf an einen HTTP-Endpunkt. Weitere Informationen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten.

AWS Trusted Advisor

Trusted Advisor stützt sich auf bewährte Verfahren, die bei der Betreuung von Hunderttausenden von AWS Kunden gelernt wurden. Trusted Advisor untersucht Ihre AWS Umgebung und gibt dann Empfehlungen, wenn Möglichkeiten bestehen, Geld zu sparen, die Systemverfügbarkeit und -leistung zu verbessern oder Sicherheitslücken zu schließen. Alle AWS Kunden haben Zugriff auf fünf Trusted Advisor Schecks. Kunden mit einem Business- oder Enterprise-Supportplan können alle Trusted Advisor Schecks einsehen.

Trusted Advisor hat die folgenden Amazon RDS Amazon Prüfungen:

- Amazon RDS Amazon im Leerlauf
- Amazon RDS
- Amazon RDS
- Amazon RDS Multi-AZ

Weitere Informationen zu diesen Prüfungen finden Sie unter Trusted Advisor – bewährte Methoden (Prüfungen).

Weitere Informationen zur Überwachung von Amazon RDS finden Sie unterMetriken in einer RDS Amazon-Instance überwachen.

Konformitätsvalidierung für Amazon RDS

Externe Prüfer bewerten die Sicherheit und Konformität von RDSAmazon im Rahmen mehrerer AWS Compliance-Programme. Dazu gehören SOC PCI RAMPHIPAA, Fed und andere.

Eine Liste der AWS Dienstleistungen im Rahmen bestimmter Compliance-Programme finden Sie unter AWS Dienstleistungen im Umfang der einzelnen Compliance-Programme. Allgemeine Informationen finden Sie unter AWS -Compliance-Programme.

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen in AWS Artifact.

Ihre Compliance-Verantwortung bei der Nutzung von RDSAmazon richtet sich nach der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften. AWS bietet die folgenden Ressourcen, um Sie bei der Einhaltung der Vorschriften zu unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Umgebungen beschrieben, auf denen auf Sicherheit und Compliance ausgerichtete Basisumgebungen eingerichtet werden. AWS
- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen damit HIPAA -konforme AWS Anwendungen erstellen können.
- AWS Ressourcen zur Einhaltung von Vorschriften — Diese Sammlung von Arbeitsmappen und Leitfäden, die möglicherweise auf Ihre Branche und Ihren Standort zutreffen.
- AWS Config— Mit diesem AWS Service wird bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.

Ausfallsicherheit in Amazon RDS

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die mit Netzwerken mit geringer Latenz, hohem Durchsatz und hochredundanten Vernetzungen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und -Availability Zones finden Sie unter Globale AWS-Infrastruktur.

Neben der globalen AWS-Infrastruktur stellt Amazon RDS Funktionen bereit, um Ihren Anforderungen an Ausfallsicherheit und Datensicherung gerecht zu werden.

Backup und Backup

Amazon RDS erstellt und speichert automatisierte Backups Ihrer DB-Instance. Amazon RDS erstellt einen Snapshot für das Speichervolumen der DB-Instance, damit die gesamte DB-Instance gesichert wird und nicht nur einzelne Datenbanken.

Amazon RDS erstellt während des Zeitfensters Ihrer Datenbank automatisierte Backups Ihrer DB-Instance. Amazon RDS speichert die automatisierten Backups Ihrer DB-Instance gemäß des Aufbewahrungszeitraums für Backups, den Sie angeben. Während dieses Vorhaltezeitraums kann Ihre Datenbank bei Bedarf auf einen gesicherten Zeitpunkt wiederhergestellt werden. Sie können Ihre DB-Instance auch manuell sichern, indem Sie einen DB-Snapshot erstellen.

Sie können eine DB-Instance erstellen, indem Sie sie aus diesem DB-Snapshot als Disaster Recovery-Lösung wiederherstellen, wenn die Quell-DB-Instance ausfällt.

Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

Replikation

Amazon RDS nutzt die integrierte Replikationsfunktionalität der MariaDB-, MySQL-, Oracle- und PostgreSQL-DB-Engines, um aus einer Quell-DB-Instance eine besondere Art von DB-Instance zu erstellen, die als Lesereplikat bezeichnet wird. In der Quell-DB-Instance ausgeführte Updates

werden asynchron in das Lesereplikat kopiert. Sie können die Arbeitslast für Ihre Quell-DB-Instance reduzieren, indem Sie Leseabfragen aus Ihren Anwendungen an das Lesereplikat weiterleiten. Mit Lesereplikaten können Sie die Kapazitätseinschränkungen einer einzelnen DB-Instance für leseintensive Datenbank-Workloads elastisch erweitern. Sie können ein Lesereplikat als Lösung zur Notfallwiederherstellung auf eine eigenständige Instance hochstufen, wenn die Quell-DB-Instance ausfällt. Für einige DB-Engines unterstützt Amazon RDS auch andere Replikationsoptionen.

Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.

Failover

Amazon RDS bietet durch Multi-AZ-Bereitstellungen hohe Verfügbarkeit und Failover-Unterstützung für DB-Instances. Amazon RDS verwendet mehrere verschiedene Technologien, um Failover-Unterstützung bereitzustellen. Multi-AZ-Bereitstellungen für Oracle-, PostgreSQL-, MySQL- und MariaDB-DB-Instances verwenden die Failover-Technologie von Amazon. SQL Server-DB-Instances verwenden die SQL Server-Datenbankspiegelung.

Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.

Infrastruktursicherheit in Amazon RDS

Als verwalteter Service ist Amazon Relational Database Service durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API Anrufe, um RDS über das Netzwerk auf Amazon zuzugreifen. Kunden müssen Folgendes unterstützen:

- Sicherheit auf Transportschicht (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit perfekter Vorwärtsgeheimhaltung (PFS) wie (Ephemeral Diffie-Hellman) oder DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem Prinzipal zugeordnet ist, signiert werden. IAM Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Darüber hinaus bietet Amazon RDS Funktionen zur Unterstützung der Infrastruktursicherheit.

Sicherheitsgruppen

Sicherheitsgruppen kontrollieren die Zugriffsaktivitäten von eingehendem und ausgehendem Datenverkehr in einer DB-Instance. Standardmäßig ist der Netzwerkzugriff auf eine DB-Instance deaktiviert. Sie können Regeln in einer Sicherheitsgruppe angeben, die den Zugriff aus einem IP-Adressbereich, über einen Port oder für eine Sicherheitsgruppe zulassen. Nach der Konfiguration von ingress-Regeln gelten diese für alle DB-Instances, die dieser Sicherheitsgruppe zugeordnet sind.

Weitere Informationen finden Sie unter Zugriffskontrolle mit Sicherheitsgruppen.

Public accessibility (Öffentliche Zugänglichkeit)

Wenn Sie eine DB-Instance in einer Virtual Private Cloud (VPC) starten, die auf dem VPC Amazon-Service basiert, können Sie den öffentlichen Zugriff für diese DB-Instance ein- oder ausschalten. Um festzulegen, ob die von Ihnen erstellte DB-Instance einen DNS Namen hat, der in eine öffentliche IP-

Adresse aufgelöst wird, verwenden Sie den Parameter Public Accessibility. Mit diesem Parameter können Sie festlegen, ob ein öffentlicher Zugriff auf die DB-Instance besteht. Sie können eine DB-Instance ändern und die öffentliche Zugänglichkeit im Parameter Öffentlicher Zugriff aktivieren und deaktivieren.

Weitere Informationen finden Sie unter Ausblenden einer DB-Instance in einer VPC vor dem Internet.

> **Note**
>
> Wenn sich Ihre DB-Instance in einer befindet, VPC aber nicht öffentlich zugänglich ist, können Sie auch eine AWS VPN Site-to-Site-Verbindung oder eine Verbindung verwenden, um von einem privaten AWS Direct Connect Netzwerk aus darauf zuzugreifen. Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.

Amazon-RDS-API und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und Amazon-RDS-API-Endpunkten herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen. Schnittstellenendpunkte werden von unterstütz AWS PrivateLink.

AWS PrivateLink ermöglicht Ihnen den privaten Zugriff auf Amazon RDS-API-Operationen ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct Connect Verbindung. DB-Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen für die Kommunikation mit Amazon-RDS-API-Endpunkten, um DB-Instances und DB-Cluster zu starten, zu ändern oder zu beenden. Ihre DB-Instances benötigen auch keine öffentlichen IP-Adressen, um beliebige der verfügbaren RDS-API-Operationen zu verwenden. Der Datenverkehr zwischen der VPC und Amazon RDS verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic Network-Schnittstellen in Ihren Subnetzen dargestellt. Weitere Informationen zu Elastic Network-Schnittstellen finden Sie unter Elastic Network-Schnittstellen im Amazon EC2 Benutzerhandbuch.

Weitere Informationen zu VPC-Endpunkten finden Sie unter Interface VPC Endpoints (AWS PrivateLink) im Amazon VPC-Benutzerhandbuch. Weitere Informationen zu RDS-API-Operationen finden Sie in der Amazon-RDS-API-Referenz.

Sie benötigen keinen Schnittstellen-VPC-Endpunkt, um eine Verbindung zu einer DB-Instance herzustellen. Weitere Informationen finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.

Überlegungen zu VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für Amazon RDS-API-Endpunkte einrichten, stellen Sie sicher, dass Sie die Eigenschaften und Einschränkungen des Schnittstellenendpunkts im Amazon VPC Benutzerhandbuch einsehen.

Alle RDS-API-Operationen, die für die Verwaltung von Amazon-RDS-Ressourcen relevant sind, sind mit AWS PrivateLinküber Ihre VPC verfügbar.

VPC-Endpunktrichtlinien werden für RDS-API-Endpunkte unterstützt. Standardmäßig ist der vollständige Zugriff auf RDS-API-Operationen über den Endpunkt zulässig. Weitere Informationen

finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon VPC Benutzerhandbuch.

Verfügbarkeit

Die Amazon RDS-API unterstützt derzeit VPC-Endpunkte in den folgenden Regionen: AWS

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- Kanada West (Calgary)
- China (Peking)
- China (Ningxia)
- Europa (Frankfurt)
- Europa (Zürich)
- Europa (Irland)
- Europe (London)
- Europe (Paris)
- Europe (Stockholm)
- Europa (Milan)
- Israel (Tel Aviv)

- Naher Osten (Bahrain)
- Südamerika (São Paulo)
- AWS GovCloud (US-Ost)
- AWS GovCloud (US-West)

Erstellen eines Schnittstellen-VPC-Endpunkts für die Amazon RDS-API

Sie können einen VPC-Endpunkt für die Amazon RDS-API entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface ()AWS CLI erstellen. Weitere Informationen finden Sie unter Erstellung eines Schnittstellenendpunkts im Amazon VPC Benutzerhandbuch.

Erstellen Sie einen VPC-Endpunkt für die Amazon RDS-API unter Verwendung des Servicenamens `com.amazonaws.region.rds`.

Mit Ausnahme von AWS Regionen in China können Sie, wenn Sie privates DNS für den Endpunkt aktivieren, API-Anfragen an Amazon RDS mit dem VPC-Endpunkt stellen, indem Sie beispielsweise `rds.us-east-1.amazonaws.com` dessen Standard-DNS-Namen für die AWS Region verwenden. Für die AWS Regionen China (Peking) und China (Ningxia) können Sie API-Anfragen mit dem VPC-Endpunkt jeweils mit `rds-api.cn-north-1.amazonaws.com.cn` und `rds-api.cn-northwest-1.amazonaws.com.cn` stellen.

Weitere Informationen finden Sie unter Zugriff auf einen Service über einen Schnittstellenendpunkt im Amazon VPC Benutzerhandbuch.

Erstellen einer VPC-Endpunktrichtlinie für die Amazon RDS-API

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf die Amazon RDS-API steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuerung des Zugriffs auf Services mit VPC-Endpunkten im Amazon VPC Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für Amazon RDS-API-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die Amazon RDS-API. Wenn diese Richtlinie an einen Endpunkt angefügt wird, gewährt sie Zugriff auf die aufgelisteten Amazon RDS-API-Aktionen für alle Prinzipale auf allen Ressourcen.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBInstance",
            "rds:ModifyDBInstance",
            "rds:CreateDBSnapshot"
         ],
         "Resource":"*"
      }
   ]
}
```

Beispiel: VPC-Endpunktrichtlinie, die jeglichen Zugriff von einem bestimmten Konto aus verweigert AWS

Die folgende VPC-Endpunktrichtlinie verweigert dem AWS Konto 123456789012 jeglichen Zugriff auf Ressourcen, die den Endpunkt verwenden. Die Richtlinie erlaubt alle Aktionen von anderen Konten.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
}
```

Bewährte Methoden für die Sicherheit in Amazon RDS

Verwenden Sie AWS Identity and Access Management (IAM-) Konten, um den Zugriff auf Amazon RDS-API-Operationen zu kontrollieren, insbesondere auf Operationen, die Amazon RDS-Ressourcen von Amazon erstellen, ändern oder löschen. Zu diesen Ressourcen gehören , Sicherheitsgruppen und Parametergruppen. Verwenden Sie IAM auch, um Aktionen zu steuern, mit denen allgemeine Verwaltungsaktionen wie das Sichern und Wiederherstellen von ausgeführt werden.

- Erstellen Sie einen individuellen Benutzer für jede Person, die Amazon RDS Amazon verwaltet, einschließlich Ihnen selbst. Verwenden Sie keine AWS Root-Anmeldeinformationen, um Amazon RDS Amazon zu verwalten.
- Gewähren Sie jedem Benutzer nur den Mindestsatz an Berechtigungen, die für die Ausführung seiner Aufgaben erforderlich sind.
- Verwenden Sie IAM-Gruppen, um Berechtigungen für mehrere Benutzer effektiv zu verwalten.
- Wechseln Sie regelmäßig die IAM-Anmeldeinformationen.
- Konfigurieren AWS Secrets Manager Sie so, dass die Secrets für Amazon RDS Amazon automatisch rotiert werden. Weitere Informationen finden Sie unter Rotieren von AWS Secrets Manager -Secrets im AWS Secrets Manager -Benutzerhandbuch. Sie können die Anmeldeinformationen auch AWS Secrets Manager programmgesteuert von abrufen. Weitere Informationen finden Sie unter Abrufen des Secret-Wertes im AWS Secrets Manager -Benutzerhandbuch.

Weitere Informationen zur Sicherheit von Amazon RDS finden Sie unterSicherheit in Amazon RDS. Weitere Informationen zu IAM finden Sie unter AWS Identity and Access Management. Informationen zu den bewährten Methoden für IAM finden Sie unter Bewährte Methoden für IAM.

AWS Security Hub verwendet Sicherheitskontrollen, um Ressourcenkonfigurationen und Sicherheitsstandards zu bewerten und Sie bei der Einhaltung verschiedener Compliance-Rahmenbedingungen zu unterstützen. Weitere Informationen zur Verwendung von Security Hub zur Bewertung von RDS-Ressourcen finden Sie unter Steuerelemente von Amazon Relational Database Service im AWS Security Hub -Benutzerhandbuch.

Sie können Ihre Nutzung von RDS in Bezug auf bewährte Sicherheitsmethoden mithilfe von Security Hub überwachen. Weitere Informationen finden Sie unter Was ist AWS Security Hub? .

Verwenden Sie die AWS Management Console AWS CLI, die oder die RDS-API, um das Passwort für Ihren Masterbenutzer zu ändern. Falls Sie zum Ändern des Hauptbenutzerpassworts ein anderes

Tool verwenden, beispielsweise einen SQL-Client, werden dem Benutzer unter Umständen ohne Absicht seine Berechtigungen entzogen.

Zugriffskontrolle mit Sicherheitsgruppen

VPC-Sicherheitsgruppen steuern den Zugriff, den der Datenverkehr innerhalb und außerhalb eines hat. Standardmäßig ist der Netzwerkzugriff für einen deaktiviert. Sie können Regeln in einer Sicherheitsgruppe angeben, die den Zugriff aus einem IP-Adressbereich, über einen Port oder für eine Sicherheitsgruppe zulassen. Nach der Konfiguration der Eingangsregeln gelten dieselben Regeln für alle , die dieser Sicherheitsgruppe zugeordnet sind. Sie können bis zu 20 Regeln in einer Sicherheitsgruppe angeben.

Überblick über VPC-Sicherheitsgruppen

Jede VPC-Sicherheitsgruppenregel ermöglicht es einer bestimmten Quelle, auf einen in einer VPC zuzugreifen, der dieser VPC-Sicherheitsgruppe zugeordnet ist. Die Quelle kann ein Adressenbereich (zum Beispiel: 203.0.113.0/24) oder eine andere VPC-Sicherheitsgruppe sein. Wenn Sie eine VPC-Sicherheitsgruppe als Quelle festlegen, erlauben Sie eingehenden Datenverkehr von allen Instances (typischerweise Anwendungsserver), die Quell-VPC-Sicherheitsgruppe verwenden. VPC-Sicherheitsgruppen können über Regeln verfügen, die den eingehenden und ausgehenden Datenverkehr regulieren. Die Regeln für ausgehenden Verkehr gelten nur, wenn der als Client fungiert. Zum Beispiel gelten Regeln für ausgehenden Datenverkehr für eine Oracle DB-Instance mit ausgehenden Datenbankverknüpfungen. Sie müssen die EC2Amazon-API oder die Sicherheitsgruppenoption auf der VPC-Konsole verwenden, um VPC-Sicherheitsgruppen zu erstellen.

Wenn Sie Regeln für Ihre VPC-Sicherheitsgruppe erstellen, die den Zugriff auf die in Ihrer VPC ermöglichen, müssen Sie einen Port für jeden Adressbereich angeben, für den die Regel Zugriff gewährt. Wenn Sie beispielsweise SSH-Zugriff auf Instances in der VPC aktivieren möchten, dann erstellen Sie eine Regel, die Zugriff auf TCP-Port 22 für den bestimmten Adressbereich zulässt.

Sie können mehrere VPC-Sicherheitsgruppen konfigurieren, die Zugriff auf verschiedenen Ports für verschiedenen Instances in Ihrer VPC zulassen. Beispielsweise können Sie eine VPC-Sicherheitsgruppe erstellen, die den Zugriff auf TCP-Port 80 für Webserver in Ihrer VPC ermöglicht. Anschließend können Sie eine weitere VPC-Sicherheitsgruppe erstellen, die den Zugriff auf den TCP-Port 3306 für RDS für MySQL-DB-Instances in Ihrer VPC ermöglicht.

Weitere Informationen zu VPC-Sicherheitsgruppen finden Sie unter Sicherheitsgruppen im Amazon Virtual Private Cloud-Benutzerhandbuch.

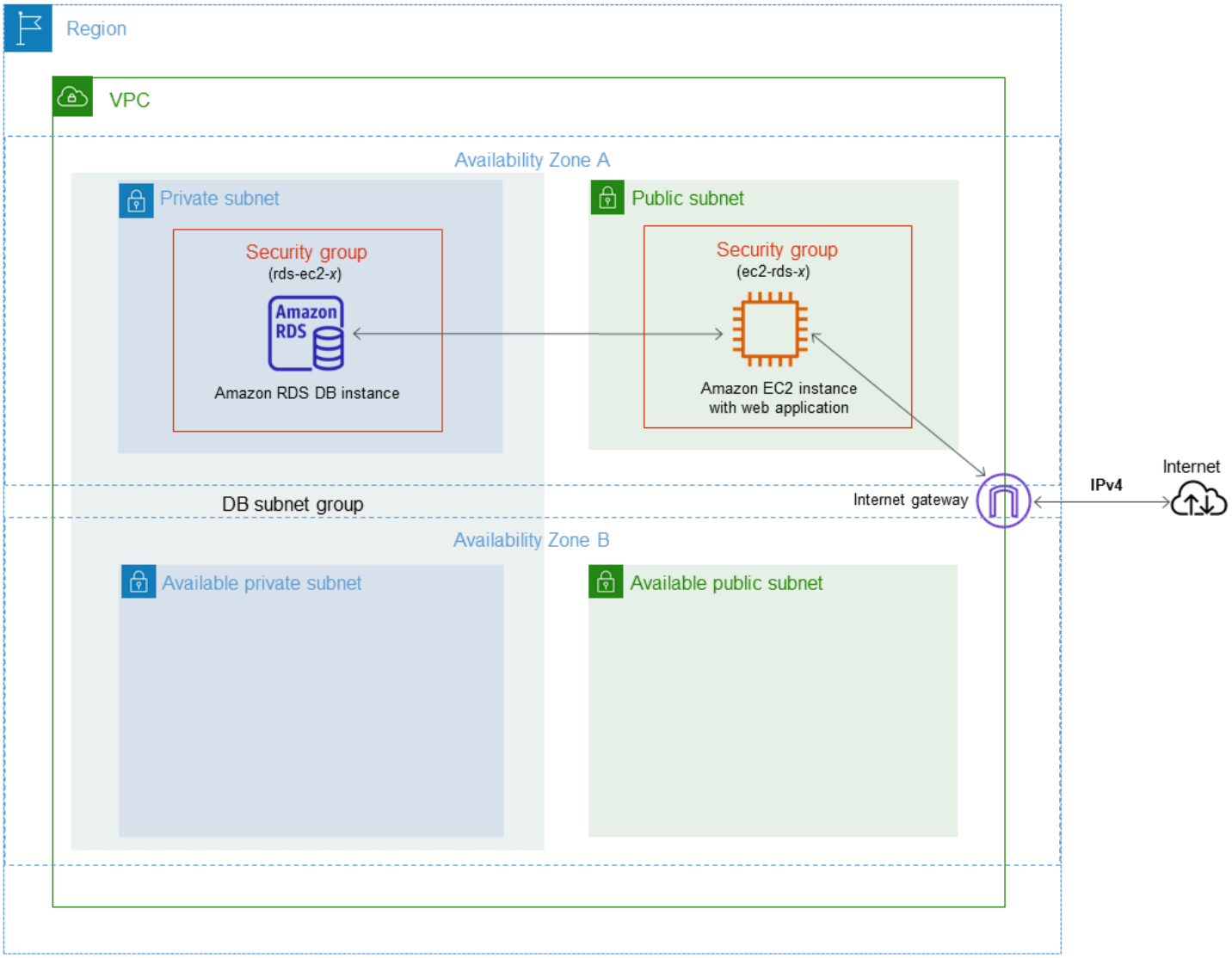
> **Note**
>
> Wenn sich Ihr in einer VPC befindet, aber nicht öffentlich zugänglich ist, können Sie auch eine AWS Site-to-Site VPN-Verbindung oder eine AWS Direct Connect Verbindung verwenden, um von einem privaten Netzwerk aus darauf zuzugreifen. Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.

Sicherheitsgruppenszenario

Ein in einer VPC wird häufig verwendet, um Daten mit einem Anwendungsserver gemeinsam zu nutzen, der in einer EC2 Amazon-Instance in derselben VPC ausgeführt wird, auf die von einer Client-Anwendung außerhalb der VPC zugegriffen wird. In diesem Szenario verwenden Sie die RDS- und VPC-Seiten der AWS Management Console oder die RDS- und EC2 API-Operationen, um die erforderlichen Instances und Sicherheitsgruppen zu erstellen:

1. Erstellen einer VPC-Sicherheitsgruppe (zum Beispiel `sg-0123ec2example`) und Definieren von eingehenden Regeln, welche die IP-Adressen der Client-Anwendung als Quelle verwenden. Diese Sicherheitsgruppe ermöglicht es Ihrer Client-Anwendung, eine Verbindung zu EC2 Instances in einer VPC herzustellen, die diese Sicherheitsgruppe verwendet.
2. Erstellen Sie eine EC2 Instanz für die Anwendung und fügen Sie die EC2 Instanz der VPC-Sicherheitsgruppe (`sg-0123ec2example`) hinzu, die Sie im vorherigen Schritt erstellt haben.
3. Erstellen Sie eine zweite VPC-Sicherheitsgruppe (zum Beispiel `sg-6789rdsexample`) und erstellen Sie eine neue Regel durch Festlegen der VPC-Sicherheitsregel, die Sie in Schritt 1 (`sg-0123ec2example`) als Quelle erstellt haben.
4. Erstellen Sie einen neuen und fügen Sie den der VPC-Sicherheitsgruppe (`sg-6789rdsexample`) hinzu, die Sie im vorherigen Schritt erstellt haben. Wenn Sie den erstellen, verwenden Sie dieselbe Portnummer wie die, die Sie für die VPC-Sicherheitsgruppe (`sg-6789rdsexample`) -Regel angegeben haben, die Sie in Schritt 3 erstellt haben.

Im folgenden Diagramm wird dieses Szenario veranschaulicht.



Ausführliche Anweisungen zur Konfiguration einer VPC für dieses Szenario finden Sie unterTutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur). Weitere Hinweise zur Verwendung einer VPC finden Sie unterAmazon VPC und Amazon RDS.

Erstellen einer VPC-Sicherheitsgruppe

Sie können unter Verwendung der VPC-Konsole eine VPC-Sicherheitsgruppe für eine DB-Instance erstellen. Weitere Informationen zum Erstellen einer Sicherheitsgruppe finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe und Sicherheitsgruppen im Amazon Virtual Private Cloud-Benutzerhandbuch.

Verknüpfen einer Sicherheitsgruppe mit einer DB-Instance

Sie können einer DB-Instance eine Sicherheitsgruppe zuordnen, indem Sie Modify auf der RDS-Konsole, die `ModifyDBInstance` Amazon RDS-API oder den `modify-db-instance` AWS CLI Befehl verwenden.

Das folgende CLI-Beispiel ordnet eine bestimmte VPC-Sicherheitsgruppe zu und entfernt DB-Sicherheitsgruppen aus der DB-Instance

```
aws rds modify-db-instance --db-instance-identifier dbName --vpc-security-group-ids sg-ID
```

Hinweise zum Ändern einer DB-Instance finden Sie unterÄndern einer Amazon RDS DB-Instance. Hinweise zu Sicherheitsgruppen bei der Wiederherstellung einer DB-Instance aus einem DB-Snapshot finden Sie unterÜberlegungen zu Sicherheitsgruppen.

> **Note**
>
> Die RDS-Konsole zeigt verschiedene Sicherheitsgruppenregelnamen für Ihre Datenbank an, wenn der Portwert auf einen anderen Wert als den Standardwert konfiguriert ist.

Für RDS for Oracle DB-Instances können zusätzliche Sicherheitsgruppen zugeordnet werden, indem die Einstellungen für die Sicherheitsgruppenoptionen für die Optionen Oracle Enterprise Manager Database Express (OEM), Oracle Management Agent for Enterprise Manager Cloud Control (OEM Agent) und Oracle Secure Sockets Layer aufgefüllt werden. In diesem Fall gelten sowohl die der DB-Instance zugewiesenen Sicherheitsgruppen als auch die Optionseinstellungen für die DB-Instance. Weitere Informationen zu diesen Optionsgruppen finden Sie unter Oracle Enterprise ManagerOracle Management Agent für Enterprise Cloud Control, undOracle Secure Sockets Layer.

Berechtigungen von Hauptbenutzerkonten

Wenn Sie einen neuen erstellen, erhält der Standard-Master-Benutzer, den Sie verwenden, bestimmte Rechte für diesen . Sie können den Master-Benutzernamen nicht ändern, nachdem der erstellt wurde.

> **Important**
>
> Wir empfehlen Ihnen, den Hauptbenutzer nicht direkt in Ihren Anwendungen zu verwenden. Bleiben Sie stattdessen bei der bewährten Methode, einen Datenbankbenutzer zu verwenden, der mit den Mindestberechtigungen erstellt wurde, die für Ihre Anwendung erforderlich sind.

> **Note**
>
> Wenn Sie versehentlich die Berechtigungen für den Master-Benutzer löschen, können Sie sie wiederherstellen, indem Sie den ändern und ein neues Masterbenutzer-Passwort festlegen. Weitere Informationen zum Ändern eines finden Sie unter Ändern einer Amazon RDS DB-Instance.

Die folgende Tabelle zeigt die Sonderrechte und Datenbankrollen, die der Hauptbenutzer in jeder der Datenbank-Engines erhält.

Datenban -Engine	Systemberechtigung	Datenbankrolle
RDS für Db2	Der Masterbenutzer ist der `masterdba` Gruppe zugewiesen und hat die`master_user_role` . SYSMON, DBADM mit DATAACCESS UNDACCCESSCT RL ,BINDADD, CONNECTCREATETAB ,CREATE_SE CURE_OBJECT ,EXPLAIN,IMPLICIT_ SCHEMA ,LOAD,SQLADM, WLMADM	DBA,DBA_RESTR ICTED , DEVELOPER ,ROLE_NULLID_PACKAG ES , ROLE_PROC EDURES ,ROLE_TABL ESPACES Weitere Informationen finden Sie unter Standardrollen von Amazon RDS für Db2.
RDS for MariaDB	SELECT,INSERT,UPDATE,DELETE, CREATE,DROP,RELOAD, PROCESS,REFERENCE S ,INDEX, ALTER,SHOW DATABASES ,CREATE TEMPORARY TABLES ,LOCK	—

Datenban-Engine	Systemberechtigung	Datenbankrolle
	`TABLES, EXECUTE,REPLICATION CLIENT,CREATE VIEW ,SHOW VIEW,CREATE ROUTINE, ALTER ROUTINE,CREATE USER, EVENT,TRIGGER,REPLICATION SLAVE` Ab RDS für MariaDB Version 11.4 erhält der Master-Benutzer auch das Recht. `SHOW CREATE ROUTINE`	
RDS für MySQL 8.0.36 und höher	`SELECT,INSERT,UPDATE, DELETE,CREATE,DROP, RELOAD,PROCESS, REFERENCES ,INDEX,ALTER, SHOW DATABASES ,CREATE TEMPORARY TABLES,LOCK TABLES,EXECUTE, REPLICATION SLAVE ,REPLICATION CLIENT , CREATE VIEW,SHOW VIEW,CREATE ROUTINE,ALTER ROUTINE,CREATE USER,EVENT,TRIGGER, CREATE ROLE,DROP ROLE, APPLICATION_PASSWORD_ADMIN , ROLE_ADMIN ,SET_USER_ID , XA_RECOVER_ADMIN`	`rds_superuser_role` Weitere Informationen zu finden Sie `rds_superuser_role` unter. Rollenbasiertes Privilegienmodell für for My RDS SQL
RDS für MySQL-Versionen unter 8.0.36	`SELECT,INSERT,UPDATE, DELETE,CREATE,DROP, RELOAD,PROCESS, REFERENCES ,INDEX,ALTER, SHOW DATABASES ,CREATE TEMPORARY TABLES,LOCK TABLES,EXECUTE, REPLICATION CLIENT ,CREATE VIEW, SHOW VIEW,CREATE ROUTINE,ALTER ROUTINE,CREATE USER,EVENT, TRIGGER,REPLICATION SLAVE`	—

Datenban-Engine	Systemberechtigung	Datenbankrolle
RDS for PostgreSQL	`CREATE ROLE,CREATE DB, PASSWORD VALID UNTIL INFINITY,CREATE EXTENSION ,ALTER EXTENSION ,DROP EXTENSION ,CREATE TABLESPACE ,ALTER <OBJECT> OWNER,CHECKPOINT , PG_CANCEL_BACKEND( ) , PG_TERMINATE_BACKEND() ,SELECT PG_STAT_REPLICATION ,EXECUTE PG_STAT_STATEMENTS_RESET() ,OWN POSTGRES_ FDW_HANDLER() ,OWN POSTGRES_FDW_VALIDATOR() ,OWN POSTGRES_FDW , EXECUTE PG_BUFFERCACHE_PAGES() ,SELECT PG_BUFFERCACHE`	`RDS_SUPERUSER` Weitere Hinweise zu RDS_SUPERUSER finden Sie unter. Grundlegendes zu PostgreSQL-Rollen und -Berechtigungen
RDS für Oracle	`ADMINISTER DATABASE TRIGGER ,ALTER DATABASE LINK,ALTER PUBLIC DATABASE LINK, AUDIT SYSTEM,CHANGE NOTIFICATION , DROP ANY DIRECTORY ,EXEMPT ACCESS POLICY,EXEMPT IDENTITY POLICY,EXEMPT REDACTION POLICY,FLASHBACK ANY TABLE, GRANT ANY OBJECT PRIVILEGE ,RESTRICTED SESSION ,SELECT ANY TABLE,UNLIMITED TABLESPACE`	`DBA` **Note** Die `DBA` Rolle ist von den folgenden Rechten ausgenommen: `ALTER DATABASE,ALTER SYSTEM, CREATE ANY DIRECTORY ,CREATE EXTERNAL JOB,CREATE PLUGGABLE DATABASE, GRANT ANY PRIVILEGE ,GRANT ANY ROLE,READ ANY FILE GROUP`

Datenban-Engine	Systemberechtigung	Datenbankrolle
Amazon RDS for Microsoft SQL Server	`ADMINISTER BULK OPERATIONS` ,`ALTER ANY CONNECTION` ,`ALTER ANY CREDENTIAL` , `ALTER ANY EVENT SESSION`,`ALTER ANY LINKED SERVER`,`ALTER ANY LOGIN`,`ALTER ANY SERVER AUDIT`,`ALTER ANY SERVER ROLE`, `ALTER SERVER STATE`,`ALTER TRACE`, `CONNECT SQL`,`CREATE ANY DATABASE`, `VIEW ANY DATABASE`,`VIEW ANY DEFINITION` ,`VIEW SERVER STATE`,`ALTER ON ROLE SQLAgentOperatorRole`	`DB_OWNER`(Rolle auf Datenbankebene),`PROCESSADMIN` (Rolle auf Serverebene),`SETUPADMIN` (Rolle auf Serverebene),`SQLAgentUserRole` (Rolle auf Datenbankebene)

Verwenden von serviceverknüpften Rollen für Amazon RDS

Amazon RDS verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Amazon RDS verknüpft ist. Servicebezogene Rollen sind von Amazon RDS Amazon vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle macht die Nutzung von Amazon RDS einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon RDS definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon RDS seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies schützt Ihre Amazon RDS-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Service-Linked Role (Serviceverknüpfte Rolle) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für Amazon RDS

Amazon RDS verwendet die serviceverknüpfte Rolle namens AWSService RoleFor RDS, um Amazon RDS zu ermöglichen, AWS Dienste im Namen Ihrer aufzurufen.

Die dienstbezogene AWSService RoleFor RDS-Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `rds.amazonaws.com`

Dieser dienstgebundenen Rolle ist eine Berechtigungsrichtlinie namens `AmazonRDSServiceRolePolicy` zugeordnet, die ihr Berechtigungen für den Betrieb in Ihrem Konto erteilt.

Weitere Informationen zu dieser Richtlinie, einschließlich des JSON-Richtliniendokuments, finden Sie bei Amazon RDSService RolePolicy im AWS Managed Policy Reference Guide.

Note

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Wenn Sie die folgende Fehlermeldung erhalten:
Unable to create the resource. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später noch einmal.
Stellen Sie sicher, dass Sie die folgenden Berechtigungen für Sie aktiviert sind:

```
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"rds.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon RDS

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine DB-Instance erstellen, erstellt Amazon RDS die serviceverknüpfte Rolle für Sie.

Important

Wenn Sie den Amazon RDS Amazon vor dem 1. Dezember 2017 genutzt haben, als er begann, serviceverknüpfte Rollen zu unterstützen, dann hat Amazon RDS die AWSService RoleFor RDS-Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter Eine neue Rolle wurde in meinem AWS Konto angezeigt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine DB-Instance erstellen, erstellt Amazon RDS wieder die serviceverknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für Amazon RDS

Amazon RDS erlaubt es Ihnen nicht, die mit dem AWSService RoleFor RDS-Dienst verknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon RDS

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch all Ihre DB-Instances löschen, bevor Sie die serviceverknüpfte Rolle löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus. Wählen Sie dann den Namen (nicht das Kontrollkästchen) der AWSService RoleFor RDS-Rolle aus.
3. Wählen Sie auf der Übersichtsseite für die gewählte Rolle die Registerkarte Letzter Zugriff aus.
4. Überprüfen Sie auf der Registerkarte Letzter Zugriff die letzten Aktivitäten für die mit dem Dienst verknüpfte Rolle.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob Amazon RDS die AWSService RoleFor RDS-Rolle verwendet, können Sie versuchen, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt die Löschung fehl und Sie können die AWS -Regionen anzeigen, in denen die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Wenn Sie die AWSService RoleFor RDS-Rolle entfernen möchten, müssen Sie zuerst alle Ihre löschen.

Löschen aller Ihrer Instances

Verwenden Sie eine dieser Verfahren, um Ihrer kompletten Instance zu löschen.

So löschen Sie eine Instance (Konsole)

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Datenbanken aus.
3. Wählen Sie die Instance aus, die Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Wenn Sie die Aufforderung Create final Snapshot? (Finalen Snapshot erstellen), wählen Sie Yes (Ja) oder No (Nein) aus.
6. Wenn Sie im vorherigen Schritt Yes (Ja) gewählt haben, geben Sie unter Final snapshot name (Endgültiger Snapshot-Name) den Namen Ihres endgültigen DB-Snapshots ein.
7. Wählen Sie Delete (Löschen).

So löschen Sie eine Instance (CLI)

Siehe `delete-db-instance` in der AWS CLI -Befehlsreferenz.

So löschen Sie eine Instance (API)

Weitere Informationen finden Sie im Amazon RDS API Reference unter `DeleteDBInstance`.

Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um die mit dem AWSService RoleFor RDS-Dienst verknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom

Amazon RDS Custom verwendet die benannte serviceverknüpfte Rolle`AWSServiceRoleForRDSCustom`, damit RDS Custom AWS Dienste im Namen Ihrer RDS-DB-Ressourcen aufrufen kann.

Die AWSService RoleFor RDSCustom servicebezogene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `custom.rds.amazonaws.com`

Dieser dienstgebundenen Rolle ist eine Berechtigungsrichtlinie namens `AmazonRDSCustomServiceRolePolicy` zugeordnet, die ihr Berechtigungen für den Betrieb in Ihrem Konto erteilt.

Das Erstellen, Bearbeiten oder Löschen der serviceverknüpften Rolle für RDS Custom funktioniert genauso wie bei Amazon RDS. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSCustom ServiceRolePolicy.

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Wenn Sie die folgende Fehlermeldung erhalten:
> Unable to create the resource. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später noch einmal.
> Stellen Sie sicher, dass Sie die folgenden Berechtigungen für Sie aktiviert sind:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
> AmazonRDSCustomServiceRolePolicy",
>     "Condition": {
> ```

```
        "StringLike": {
            "iam:AWSServiceName":"custom.rds.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Servicebezogene Rollenberechtigungen für Amazon RDS Beta

Amazon RDS verwendet die benannte serviceverknüpfte Rolle`AWSServiceRoleForRDSBeta`, um Amazon RDS zu ermöglichen, AWS Dienste im Namen Ihrer RDS-DB-Ressourcen aufzurufen.

Die AWSService RoleFor RDSBeta servicebezogene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `rds.amazonaws.com`

Dieser dienstgebundenen Rolle ist eine Berechtigungsrichtlinie namens `AmazonRDSBetaServiceRolePolicy` zugeordnet, die ihr Berechtigungen für den Betrieb in Ihrem Konto erteilt. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSBeta ServiceRolePolicy.

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Wenn Sie die folgende Fehlermeldung erhalten:
> Unable to create the resource. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später noch einmal.
> Stellen Sie sicher, dass Sie die folgenden Berechtigungen für Sie aktiviert sind:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
> ```

```
    "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
AmazonRDSBetaServiceRolePolicy",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"custom.rds.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Servicebezogene Rolle für Amazon RDS Preview

Amazon RDS verwendet die benannte serviceverknüpfte Rolle`AWSServiceRoleForRDSPreview`, um Amazon RDS zu ermöglichen, AWS Dienste im Namen Ihrer RDS-DB-Ressourcen aufzurufen.

Die AWSService RoleFor RDSPreview servicebezogene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `rds.amazonaws.com`

Dieser dienstgebundenen Rolle ist eine Berechtigungsrichtlinie namens `AmazonRDSPreviewServiceRolePolicy` zugeordnet, die ihr Berechtigungen für den Betrieb in Ihrem Konto erteilt. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSPreview ServiceRolePolicy.

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Wenn Sie die folgende Fehlermeldung erhalten:
> Unable to create the resource. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später noch einmal.
> Stellen Sie sicher, dass Sie die folgenden Berechtigungen für Sie aktiviert sind:

```
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
AmazonRDSPreviewServiceRolePolicy",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"custom.rds.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter Serviceverknüpfte Rollenberechtigung im IAM-Benutzerhandbuch.

Amazon VPC und Amazon RDS

Amazon Virtual Private Cloud (AmazonVPC) ermöglicht Ihnen den Start AWS Ressourcen, wie Amazon RDS DB-Instances, in eine virtuelle private Cloud (VPC).

Wenn Sie eine verwendenVPC, haben Sie die Kontrolle über Ihre virtuelle Netzwerkumgebung. Sie können Ihren eigenen IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen und Zugriffskontrolllisten konfigurieren. Für den Betrieb Ihres in einem fallen keine zusätzlichen Kosten anVPC.

Konten haben eine StandardeinstellungVPC. Alle neuen werden standardmäßig erstellt, VPC sofern Sie nichts anderes angeben.

Themen

- Arbeiten mit einer DB-Instance in einer VPC
- Aktualisierung der VPC für eine DB-Instance
- Szenarien für den Zugriff auf einen in einem VPC
- Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur)
- Tutorial: Erstellen einer VPC zur Verwendung mit einer DB–Instance (Dual-Stack-Modus)
- Verschieben einer DB-Instance von außerhalb einer VPC in eine VPC

Im Folgenden finden Sie eine Diskussion über VPC Funktionen, die für Amazon RDS DB-Instances und Amazon relevant sind. Weitere Informationen zu Amazon VPC finden Sie im Amazon-Handbuch „VPCErste Schritte“ und im VPCAmazon-Benutzerhandbuch.

Arbeiten mit einer DB-Instance in einer VPC

Ihr DB-Instance befindet sich in einer Virtual Private Cloud (VPC). Eine VPC ist ein virtuelles Netzwerk, das logisch von anderen virtuellen Netzwerken in der AWS Cloud isoliert ist. Amazon VPC ermöglicht es Ihnen, AWS Ressourcen wie einen Amazon RDS Amazon in einer VPC zu starten. EC2 Bei der VPC kann es sich um die mit Ihrem Konto verknüpfte Standard-VPC oder eine von Ihnen erstellte VPC handeln. Alle VPCs sind mit Ihrem AWS Konto verknüpft.

Die Standard-VPC besitzt drei Subnetze, mit denen Sie Ressourcen innerhalb der VPC separieren können. Zudem verfügt die Standard-VPC über ein Internet-Gateway, mit dem Sie den externen Zugriff auf Ressourcen in der VPC gewähren können.

Eine Liste von Szenarien mit Amazon RDS DB-Instance in einer VPC und außerhalb einer VPC finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.

Themen

- Arbeiten mit einer DB-Instance in einer VPC
- Arbeiten mit DB-Subnetzgruppen
- Gemeinsam genutzte Subnetze
- Amazon-RDS-IP-Adressierung
- Ausblenden einer DB-Instance in einer VPC vor dem Internet
- Erstellen einer DB-Instance in einer VPC

In den folgenden Tutorials lernen Sie, eine VPC zu erstellen, die Sie für ein gängiges Amazon-RDS-Szenario verwenden können:

- Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur)
- Tutorial: Erstellen einer VPC zur Verwendung mit einer DB–Instance (Dual-Stack-Modus)

Arbeiten mit einer DB-Instance in einer VPC

Hier sind einige Tipps für das Arbeiten mit einer DB-Instance in einer VPC:

- Ihre VPC muss mindestens zwei Subnetze besitzen. Diese Subnetze müssen sich in zwei verschiedenen Availability Zones in dem Bereich befinden AWS-Region , in dem Sie Ihren bereitstellen möchten. Ein Subnetz ist ein Segment des IP-Adressbereichs einer VPC, das Sie angeben können und das Sie zur Gruppierung von DB-Instance auf der Grundlage Ihrer Sicherheits- und Betriebsanforderungen verwenden können.

  Bei Multi-AZ-Bereitstellungen AWS-Region ermöglicht die Definition eines Subnetzes für zwei oder mehr Availability Zones in einer Amazon RDS, bei Bedarf ein neues Standby in einer anderen Availability Zone zu erstellen. Sie müssen dies sogar für Einzel-AZ-Bereitstellungen vornehmen, nur für den Fall, dass Sie sie zu einem späteren Zeitpunkt in Multi-AZ-Bereitstellungen umwandeln möchten.

  > **Note**
  >
  > Die DB-Subnetzgruppe für eine lokale Zone kann nur ein Subnetz haben.

- Wenn Sie möchten, dass Ihr DB-Instance in der VPC öffentlich zugänglich ist, stellen Sie sicher, dass Sie die VPC-Attribute DNS-Hostnamen und DNS-Auflösung aktivieren.
- Sie müssen für Ihre VPC eine DB-Sicherheitsgruppe erstellen. Sie erstellen eine DB-Subnetzgruppe, indem Sie die Subnetze angeben, die Sie erstellt haben. Amazon RDS wählt ein Subnetz und eine IP-Adresse innerhalb dieser Subnetzgruppe aus, die mit Ihrer DB-Instance verknüpft werden sollen. Die DB-Instance verwendet die Availability Zone, die das Subnetz enthält.
- Ihre VPC muss über eine VPC-Sicherheitsgruppe verfügen, die den Zugriff auf die DB-Instance zulässt.

  Weitere Informationen finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.
- Die CIDR-Blöcke in jedem Subnetz müssen groß genug sein, um freie IP-Adressen für Amazon RDS unterzubringen, die während der Wartungsarbeiten genutzt werden können, einschließlich Failover und Skalierung. Beispielsweise ist ein Bereich wie 10.0.0.0/24 und 10.0.1.0/24 normalerweise groß genug.
- Eine VPC kann über das Attribut instance tenancy mit dem Wert default oder dedicated verfügen. Bei allen Standardeinstellungen VPCs ist das Instance-Tenancy-Attribut auf Standard gesetzt, und eine Standard-VPC kann jede DB-Instance-Klasse unterstützen.

  Wenn Sie sich dafür entscheiden, Ihren in einer dedizierten VPC zu haben, in der das Instance-Tenancy-Attribut auf Dedicated gesetzt ist, muss die DB-Instance-Klasse Ihres einer der zugelassenen Amazon EC2 Dedicated Instance-Typen sein. Beispielsweise entspricht die EC2 dedizierte r5.large-Instance der DB-Instance-Klasse db.r5.large. Informationen über die Instance-Tenancy in einer VPC finden Sie unter Dedicated Instances im Amazon Elastic Compute Cloud-Benutzerhandbuch.

  Weitere Informationen zu den Instance-Typen, die sich in einer Dedicated Instance befinden können, finden Sie auf der EC2 Amazon-Preisseite unter Amazon EC2 Dedicated Instances.

  > **Note**
  >
  > Wenn Sie das Attribut instance tenancy für einen DB-Instance auf „dedicated“ setzen, garantiert dies nicht, dass der DB-Instance auf einem dedizierten Host läuft.
- Wenn einer DB-Instance eine Optionsgruppe zugewiesen ist, ist sie mit der VPC der DB-Instance verknüpft. Aufgrund dieser Verknüpfung können Sie die einer DB-Instance zugeordnete Optionsgruppe nicht verwenden, wenn Sie die DB-Instance in einer anderen VPC wiederherstellen.

- Wenn Sie eine DB-Instance in einer anderen VPC wiederherstellen, stellen Sie sicher, dass Sie entweder der DB-Instance die Standard-Optionsgruppe zuweisen, eine Optionsgruppe zuweisen, die mit dieser VPC verknüpft ist, oder eine neue Optionsgruppe erstellen und diese der DB-Instance zuweisen. Bei persistenten oder permanenten Optionen wie Oracle TDE müssen Sie eine neue Optionsgruppe erstellen, die die persistente oder permanente Option enthält, wenn Sie eine DB-Instance auf einer anderen VPC wiederherstellen.

Arbeiten mit DB-Subnetzgruppen

Subnetze sind Segmente eines IP-Adressbereichs der VPC, die Sie festlegen und mit denen Sie basierend auf Ihren Sicherheits- und Betriebsanforderungen Ressourcen gruppieren können. Eine DB-Subnetzgruppe ist eine Sammlung von Subnetzen (in der Regel private Subnetze), die Sie in einer VPC erstellen und anschließend Ihrer DB-Instances zuweisen. Mithilfe einer DB-Subnetzgruppe können Sie eine bestimmte VPC angeben, wenn Sie mithilfe der AWS CLI oder RDS-API erstellen. Wenn Sie die Konsole verwenden, können Sie die VPC und Subnetze auswählen, die Sie verwenden möchten.

Jede DB-Subnetzgruppe sollte über Subnetze in mindestens zwei Availability Zones in einer bestimmten AWS-Region verfügen. Beim Erstellen einer DB-Instance in einer VPC müssen Sie eine DB-Subnetzgruppe dafür auswählen. Aus der DB-Subnetzgruppe wählt Amazon RDS ein Subnetz und eine IP-Adresse innerhalb dieses Subnetzes aus, um es mit der DB-Instance zu verbinden. Die DB verwendet die Availability Zone, die das Subnetz enthält. Amazon RDS weist immer eine IP-Adresse aus einem Subnetz zu, das über freien IP-Adressraum verfügt.

Falls die primäre DB-Instance einer Multi-AZ-Bereitstellung ausfällt, kann Amazon RDS die entsprechende Standby-Instance hochstufen. Später wird eine neue Standby-Instance mithilfe einer IP-Adresse aus dem Subnetz in einer der anderen Availability Zones erstellt.

Die Subnetze in einer DB-Subnetzgruppe sind entweder öffentlich oder privat. Die Subnetze sind öffentlich oder privat, abhängig von der Konfiguration, die Sie für ihre Netzwerk-Zugriffskontrolllisten (Netzwerk ACLs) und Routing-Tabellen festgelegt haben. Damit öffentlich auf eine DB-Instance zugegriffen werden kann, müssen alle Subnetze in der entsprechenden DB-Subnetzgruppe öffentlich sein. Wenn ein Subnetz, das mit einer öffentlich zugänglichen DB-Instance verknüpft ist, von öffentlich in privat geändert wird, kann dies die Verfügbarkeit der DB-Instance beeinträchtigen.

Um eine DB-Subnetzgruppe zu erstellen, die den Dual-Stack-Modus unterstützt, stellen Sie sicher, dass jedem Subnetz, das Sie der DB-Subnetzgruppe hinzufügen, ein CIDR-Block der

Internetprotokollversion 6 (IPv6) zugeordnet ist. Weitere Informationen finden Sie unter Amazon-RDS-IP-Adressierung und Migration zu IPv6 im Amazon VPC-Benutzerhandbuch.

> **Note**
>
> Die DB-Subnetzgruppe für eine lokale Zone kann nur ein Subnetz haben.

Wenn Amazon RDS eine DB-Instance in einer VPC erstellt, wird Ihrer DB-Instance mithilfe einer IP-Adresse aus Ihrer DB-Subnetzgruppe eine Netzwerkschnittstelle zugewiesen. Wir empfehlen jedoch dringend, den DNS-Namen (Domain Name System) für die Verbindung zu Ihrem DB-Instance zu verwenden. Wir empfehlen dies, da sich die zugrunde liegende IP-Adresse während des Failovers ändert.

> **Note**
>
> Für jede DB-Instance, die Sie in einer VPC ausführen, stellen Sie sicher, mindestens eine Adresse in jedem Subnetz der DB-Subnetzgruppe für Wiederherstellungsmaßnahmen von Amazon RDS zu reservieren.

Gemeinsam genutzte Subnetze

Sie können eine(n) DB-Instance in einer gemeinsam genutzten VPC erstellen.

Einige Überlegungen, die Sie bei der Verwendung von Shared beachten sollten: VPCs

- Sie können eine(n) DB-Instance von einem gemeinsam genutzten VPC-Subnetz in ein nicht gemeinsam genutztes VPC-Subnetz verschieben und umgekehrt.
- Teilnehmer in einer gemeinsam genutzten VPC müssen eine Sicherheitsgruppe in der VPC erstellen, damit sie eine(n) DB-Instance erstellen können.
- Besitzer und Teilnehmer in einer gemeinsam genutzten VPC können mithilfe von SQL-Abfragen auf die Datenbank zugreifen. Allerdings kann nur der Ersteller einer Ressource beliebige API-Aufrufe für die Ressource tätigen.

Amazon-RDS-IP-Adressierung

IP-Adressen ermöglichen es Ressourcen in Ihrer VPC untereinander und mit Ressourcen im Internet zu kommunizieren. Amazon RDS unterstützt beide IPv4 und IPv6 Adressierungsprotokolle. Standardmäßig verwenden Amazon RDS, und Amazon VPC das IPv4 Adressierungsprotokoll. Sie können dieses Standardverhalten nicht deaktivieren. Achten Sie beim Erstellen einer VPC darauf, einen IPv4 CIDR-Block (einen Bereich von privaten IPv4 Adressen) anzugeben. Sie können Ihrer VPC und Ihren Subnetzen optional einen IPv6 CIDR-Block zuweisen und IPv6 Adressen aus diesem Block in Ihrem Subnetz zuweisen.

Die Support des IPv6 Protokolls erweitert die Anzahl der unterstützten IP-Adressen. Durch die Verwendung des IPv6 Protokolls stellen Sie sicher, dass Sie über ausreichend verfügbare Adressen für das future Wachstum des Internets verfügen. Neue und bestehende RDS-Ressourcen können verwendet IPv4 und innerhalb Ihrer VPC IPv6 adressiert werden. Das Konfigurieren, Sichern und Übersetzen des Netzwerkverkehrs zwischen den beiden Protokollen, die in verschiedenen Teilen einer Anwendung verwendet werden, können den Betriebsaufwand erhöhen. Sie können das IPv6 Protokoll für Amazon RDS-Ressourcen standardisieren, um Ihre Netzwerkkonfiguration zu vereinfachen.

Themen

- IPv4 Adressen
- IPv6 Adressen
- Dual-Stack-Modus

IPv4 Adressen

Wenn Sie eine VPC erstellen, müssen Sie einen IPv4 Adressbereich für die VPC in Form eines CIDR-Blocks angeben, z. B. `10.0.0.0/16` Eine DB-Subnetzgruppe definiert den Bereich der IP-Adressen in diesem CIDR-Block, den eine DB-Instance verwenden kann. Diese IP-Adressen können privat oder öffentlich sein.

Eine private IPv4 Adresse ist eine IP-Adresse, die über das Internet nicht erreichbar ist. Sie können private IPv4 Adressen für die Kommunikation zwischen Ihrem und anderen Ressourcen wie EC2 Amazon-Instances in derselben VPC verwenden. Jede DB-Instance hat eine private IP-Adresse für die Kommunikation in der VPC.

Eine öffentliche IP-Adresse ist eine IPv4 Adresse, die über das Internet erreichbar ist. Sie können öffentliche Adressen zur Kommunikation zwischen Ihrer DB-Instance und Ressourcen im Internet,

wie ein SQL-Client, verwenden. Anhand der folgenden Schritte können Sie kontrollieren, ob Ihre DB-Instance eine öffentliche IP-Adresse erhält:

Ein Tutorial, das Ihnen zeigt, wie Sie eine VPC mit nur privaten IPv4 Adressen erstellen, die Sie für ein gängiges Amazon RDS Amazon verwenden können, finden Sie unterTutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur).

IPv6 Adressen

Sie können Ihrer VPC und Ihren Subnetzen optional einen IPv6 CIDR-Block zuordnen und den Ressourcen in Ihrer VPC IPv6 Adressen aus diesem Block zuweisen. Jede IPv6 Adresse ist weltweit einzigartig.

Der IPv6 CIDR-Block für Ihre VPC wird automatisch aus dem Adresspool von IPv6 Amazon zugewiesen. Sie können den Bereich nicht selbst auswählen.

Wenn Sie eine Verbindung zu einer IPv6 Adresse herstellen, stellen Sie sicher, dass die folgenden Bedingungen erfüllt sind:

- Der Client ist so konfiguriert, dass der Datenverkehr zwischen Client und Datenbank zulässig IPv6 ist.
- Die von der DB-Instance verwendeten RDS-Sicherheitsgruppen sind korrekt konfiguriert, sodass der Datenverkehr vom Client zur Datenbank zulässig IPv6 ist.
- Der Client-Betriebssystemstapel ermöglicht den Datenverkehr über die IPv6 Adresse, und die Betriebssystemtreiber und Bibliotheken sind so konfiguriert, dass sie den richtigen Standard-DB-Instance-Endpunkt (entweder IPv4 oder IPv6) auswählen.

Weitere Informationen finden Sie IPv6 unter IP-Adressierung im Amazon VPC-Benutzerhandbuch.

Dual-Stack-Modus

Wenn ein sowohl über das als auch über das IPv4 IPv6 Adressierungsprotokoll kommunizieren kann, wird er im Dual-Stack-Modus ausgeführt. Ressourcen können also über, oder beides IPv4, IPv6 mit dem kommunizieren. RDS deaktiviert den Internet-Gateway-Zugriff für IPv6 Endpunkte von privaten DB-Instances im Dual-Stack-Modus. RDS tut dies, um sicherzustellen, dass Ihre IPv6 Endpunkte privat sind und nur von Ihrer VPC aus darauf zugegriffen werden kann.

Themen

- Dual-Stack-Modus und DB-Subnetzgruppen

- Arbeiten mit DB-Instances im Dual-Stack-Modus
- Ändern von IPv4 reinen zur Verwendung des Dual-Stack-Modus
- Verfügbarkeit von Regionen und Versionen
- Einschränkungen für Dual-Stack-Netzwerk-DB-Instances

Ein Tutorial, das Ihnen zeigt, wie Sie eine VPC mit beiden IPv4 IPv6 Adressen erstellen, die Sie für ein gängiges Amazon RDS Amazon verwenden können, finden Sie unterTutorial: Erstellen einer VPC zur Verwendung mit einer DB–Instance (Dual-Stack-Modus).

Dual-Stack-Modus und DB-Subnetzgruppen

Um den Dual-Stack-Modus zu verwenden, stellen Sie sicher, dass jedem Subnetz in der DB-Subnetzgruppe, das Sie dem zuordnen, ein IPv6 CIDR-Block zugeordnet ist. Sie können eine neue DB-Subnetzgruppe erstellen oder eine vorhandene DB-Subnetzgruppe ändern, um diese Anforderung zu erfüllen. Nachdem eine DB-Instance in den Dual-Stack-Modus gewechselt ist, können sich Clients normal damit verbinden. Stellen Sie sicher, dass die Client-Sicherheits-Firewalls und die Sicherheitsgruppen der RDS-DB-Instance korrekt konfiguriert sind, um den Datenverkehr zu ermöglichen. IPv6 Zum Herstellen einer Verbindung verwenden Clients den Endpunkt der DB-Instance. Clientanwendungen können angeben, welches Protokoll bevorzugt wird, wenn eine Verbindung mit einer Datenbank hergestellt wird. Im Dual-Stack-Modus erkennt der das vom Client bevorzugte Netzwerkprotokoll (entweder IPv4 oder IPv6) und verwendet dieses Protokoll für die Verbindung.

Wenn eine DB-Subnetzgruppe den Dual-Stack-Modus aufgrund der Löschung des Subnetzes oder der CIDR-Trennung nicht mehr unterstützt, besteht die Gefahr eines inkompatiblen Netzwerkstatus für DB-Instances, die mit der DB-Subnetzgruppe verknüpft sind. Sie können die DB-Subnetzgruppe auch nicht verwenden, wenn Sie eine neue DB-Instance im Dual-Stack-Modus erstellen.

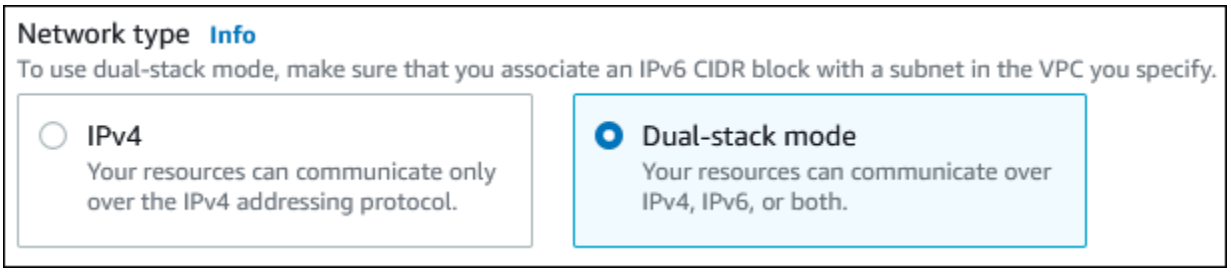
Um mithilfe von festzustellen, ob eine DB-Subnetzgruppe den Dual-Stack-Modus unterstützt AWS Management Console, sehen Sie sich den Netzwerktyp auf der Detailseite der DB-Subnetzgruppe an. Um mithilfe von zu ermitteln, ob eine DB-Subnetzgruppe den Dual-Stack-Modus unterstützt AWS CLI, führen Sie den describe-db-subnet-groupsBefehl aus und sehen Sie sich die Ausgabe an. `SupportedNetworkTypes`

Lesereplikate werden als unabhängige DB-Instances behandelt und können einen anderen Netzwerktyp haben als die primäre DB-Instance. Wenn Sie den Netzwerktyp der primären DB-Instance eines Lesereplikats ändern, ist das Lesereplikat nicht betroffen. Wenn Sie eine DB-Instance wiederherstellen, können Sie sie auf jedem unterstützten Netzwerktyp wiederherstellen.

Arbeiten mit DB-Instances im Dual-Stack-Modus

Wenn Sie einen erstellen oder ändern, können Sie den Dual-Stack-Modus angeben, damit Ihre Ressourcen mit Ihrem über IPv4 oder beides kommunizieren können. IPv6

Wenn Sie den verwenden AWS Management Console , um eine DB-Instance zu erstellen oder zu ändern, können Sie den Dual-Stack-Modus im Abschnitt Netzwerktyp angeben. Die folgende Abbildung zeigt den Abschnitt Network type (Netzwerktyp) in der Konsole.



Wenn Sie den verwenden AWS CLI , um einen zu erstellen oder zu ändern, stellen Sie die `--network-type` Option `DUAL` auf Dual-Stack-Modus ein. Wenn Sie die RDS API zum Erstellen oder Ändern einer DB-Instance verwenden, legen Sie den Parameter `NetworkType` auf `DUAL` fest, um den Dual-Stack-Modus zu verwenden. Wenn Sie den Netzwerktyp einer DB-Instance ändern, sind Ausfallzeiten möglich. Wenn der Dual-Stack-Modus von der angegebenen DB-Engine-Version oder der DB-Subnetzgruppe nicht unterstützt wird, wird der Fehler `NetworkTypeNotSupported` zurückgegeben.

Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance. Weitere Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Um festzustellen, ob sich ein DB-Instance im Dual-Stack-Modus befindet, sehen Sie sich den Netzwerktyp auf der Registerkarte Connectivity & security (Konnektivität & Sicherheit) für den DB-Instance an.

Ändern von IPv4 reinen zur Verwendung des Dual-Stack-Modus

Sie können einen verwendet. Dazu ändern Sie den Netzwerktyp der DB-Instance. Die Änderung kann zu Ausfallzeiten führen.

Es wird empfohlen, den Netzwerktyp Ihrer Amazon-RDS-DB-Instances während eines Wartungsfensters zu ändern. Das Festlegen des Netzwerktyps neuer Instances auf den Dual-Stack-Modus wird derzeit nicht unterstützt. Sie können den Netzwerktyp manuell festlegen, indem Sie den Befehl `modify-db-instance` verwenden.

Bevor Sie eine DB-Instance zur Verwendung des Dual-Stack-Modus ändern, stellen Sie sicher, dass ihre DB-Subnetzgruppe den Dual-Stack-Modus unterstützt. Wenn die mit der DB-Instance verknüpfte DB-Subnetzgruppe den Dual-Stack-Modus nicht unterstützt, geben Sie eine andere DB-Subnetzgruppe an, die DB-Instance unterstützt, wenn Sie sie ändern. Das Ändern der DB-Subnetzgruppe einer DB-Instance kann zu Ausfallzeiten führen.

Wenn Sie die DB-Subnetzgruppe einer DB-Instance ändern, bevor Sie die DB-Instance zur Verwendung des Dual-Stack-Modus ändern, stellen Sie sicher, dass die DB-Subnetzgruppe vor und nach der Änderung für die DB-Instance gültig ist.

Für RDS for PostgreSQL, RDS for MySQL, RDS for Oracle und RDS for MariaDB Single-AZ Instances empfehlen wir, dass Sie den modify-db-instanceBefehl nur mit dem `--network-type` Parameter auf ausführen, um das Netzwerk in den `DUAL` Dual-Stack-Modus zu ändern. Das Hinzufügen anderer Parameter zusammen mit dem Parameter `--network-type` in demselben API-Aufruf kann zu Ausfallzeiten führen. Wenn Sie mehrere Parameter ändern möchten, stellen Sie sicher, dass die Änderung des Netzwerktyps erfolgreich abgeschlossen wurde, bevor Sie eine weitere `modify-db-instance`-Anforderung mit anderen Parametern senden.

Änderungen des Netzwerktyps für RDS for PostgreSQL, RDS for MySQL, RDS for Oracle und RDS for MariaDB Multi-AZ DB-Instances führen zu einer kurzen Ausfallzeit und lösen einen Failover aus, wenn Sie nur den `--network-type` Parameter verwenden oder Parameter in einem Befehl kombinieren. modify-db-instance

Änderungen des Netzwerktyps auf Single-AZ- oder Multi-AZ-DB-Instances von RDS für SQL Server führen zu Ausfallzeiten, wenn Sie nur den Parameter `--network-type` verwenden oder wenn Sie Parameter in einem `modify-db-instance`-Befehl kombinieren. Änderungen des Netzwerktyps führen zu einem Failover in einer Multi-AZ-Instance von SQL Server.

Wenn Sie nach der Änderung keine Verbindung zum herstellen können, stellen Sie sicher, dass die Client- und Datenbanksicherheitsfirewalls und Routing-Tabellen korrekt konfiguriert sind, sodass der Datenverkehr zur Datenbank im ausgewählten Netzwerk zugelassen wird (entweder IPv4 oder IPv6). Möglicherweise müssen Sie auch Betriebssystemparameter, Bibliotheken oder Treiber ändern, um eine Verbindung über eine IPv6 Adresse herzustellen.

Wenn Sie eine DB-Instance auf den Dual-Stack-Modus umstellen, darf keine ausstehende Änderung von einer Single-AZ-Bereitstellung zu einer Multi-AZ-Bereitstellung oder von einer Multi-AZ-Bereitstellung zu einer Single-AZ-Bereitstellung vorhanden sein.

So ändern Sie einen , der IPv4 nur den Dual-Stack-Modus verwendet

1. Ändern Sie eine DB-Subnetzgruppe, um den Dual-Stack-Modus zu unterstützen, oder erstellen Sie eine DB-Subnetzgruppe, die den Dual-Stack-Modus unterstützt:

   a. Ordnen Sie Ihrer VPC einen IPv6 CIDR-Block zu.

      Anweisungen finden Sie unter Hinzufügen eines IPv6 CIDR-Blocks zu Ihrer VPC im Amazon VPC-Benutzerhandbuch.

   b. Hängen Sie den IPv6 CIDR-Block an alle Subnetze in Ihrer DB-Subnetzgruppe an.

      Anweisungen finden Sie unter Hinzufügen eines IPv6 CIDR-Blocks zu Ihrem Subnetz im Amazon VPC-Benutzerhandbuch.

   c. Vergewissern Sie sich, dass die DB-Subnetzgruppe den Dual-Stack-Modus unterstützt.

      Wenn Sie den verwenden AWS Management Console, wählen Sie die DB-Subnetzgruppe aus und stellen Sie sicher, dass der Wert Unterstützte Netzwerktypen auf Dual, gesetzt ist. IPv4

      Wenn Sie den verwenden AWS CLI, führen Sie den describe-db-subnet-groupsBefehl aus und stellen Sie sicher, dass der `SupportedNetworkType` Wert für die DB-Instance lautet`Dual, IPv4`.

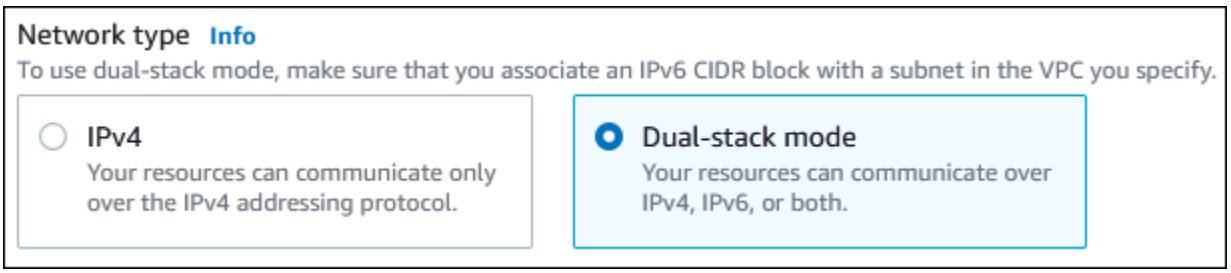
2. Ändern Sie die dem zugeordnete Sicherheitsgruppe, um IPv6 Verbindungen zur Datenbank zuzulassen, oder erstellen Sie eine neue Sicherheitsgruppe, die IPv6 Verbindungen zulässt.

   Eine Anleitung dazu finden Sie unter Sicherheitsgruppenregeln im Amazon-VPC-Benutzerhandbuch.

3. Ändern der DB-Instance zur Unterstützung des Dual-Stack-Modus. Setzen Sie hierzu den Netzwerk-Typ auf Dual-Stack-Modus.

   Wenn Sie die Konsole verwenden, stellen Sie sicher, dass die folgenden Einstellungen korrekt sind:

   - Network type (Netzwerktyp) – Dual-stack mode (Dual-Stack-Modus)



- DB-Subnet group (Subnetzgruppe) – Die DB-Subnetzgruppe, die Sie in einem vorherigen Schritt konfiguriert haben
- Security group (Sicherheitsgruppe) – Die Sicherheitsgruppe, die Sie in einem vorherigen Schritt konfiguriert haben

Wenn Sie den verwenden AWS CLI, stellen Sie sicher, dass die folgenden Einstellungen korrekt sind:

- `--network-type` – `dual`
- `--db-subnet-group-name` – Die DB-Subnetzgruppe, die Sie in einem vorherigen Schritt konfiguriert haben
- `--vpc-security-group-ids` – Die VPC-Sicherheitsgruppe, die Sie in einem vorherigen Schritt konfiguriert haben

Zum Beispiel:

```
aws rds modify-db-instance --db-instance-identifier my-instance --network-type
 "DUAL"
```

4. Vergewissern Sie sich, dass die DB-Instance den Dual-Stack-Modus unterstützt.

   Wenn Sie die Konsole verwenden, wählen Sie die Registerkarte Konnektivität und Sicherheit für die DB-Instance. Stellen Sie auf dieser Registerkarte sicher, dass der Wert des Netzwerk-TypsDual-Stack-Modus ist.

   Wenn Sie den verwenden AWS CLI, führen Sie den describe-db-instancesBefehl aus und stellen Sie sicher, dass der `NetworkType` Wert für die DB-Instance lautet`dual`.

   Führen Sie den `dig` Befehl auf dem aus, um die zugehörige IPv6 Adresse zu ermitteln.

   ```
   dig db-instance-endpoint AAAA
   ```

Verwenden Sie den Endpunkt der , nicht die IPv6 Adresse, um eine Verbindung zum herzustellen.

Verfügbarkeit von Regionen und Versionen

Die Verfügbarkeit von Funktionen und der Support variieren zwischen bestimmten Versionen der einzelnen Datenbank-Engines und in allen AWS-Regionen. Weitere Informationen zur Verfügbarkeit von Versionen und Regionen im Dual-Stack-Modus finden Sie unter Unterstützte Regionen und DB-Engines für den Dual-Stack-Modus in Amazon RDS.

Einschränkungen für Dual-Stack-Netzwerk-DB-Instances

Die folgenden Einschränkungen gelten für Dual-Stack-Netzwerk-DB-Instances:

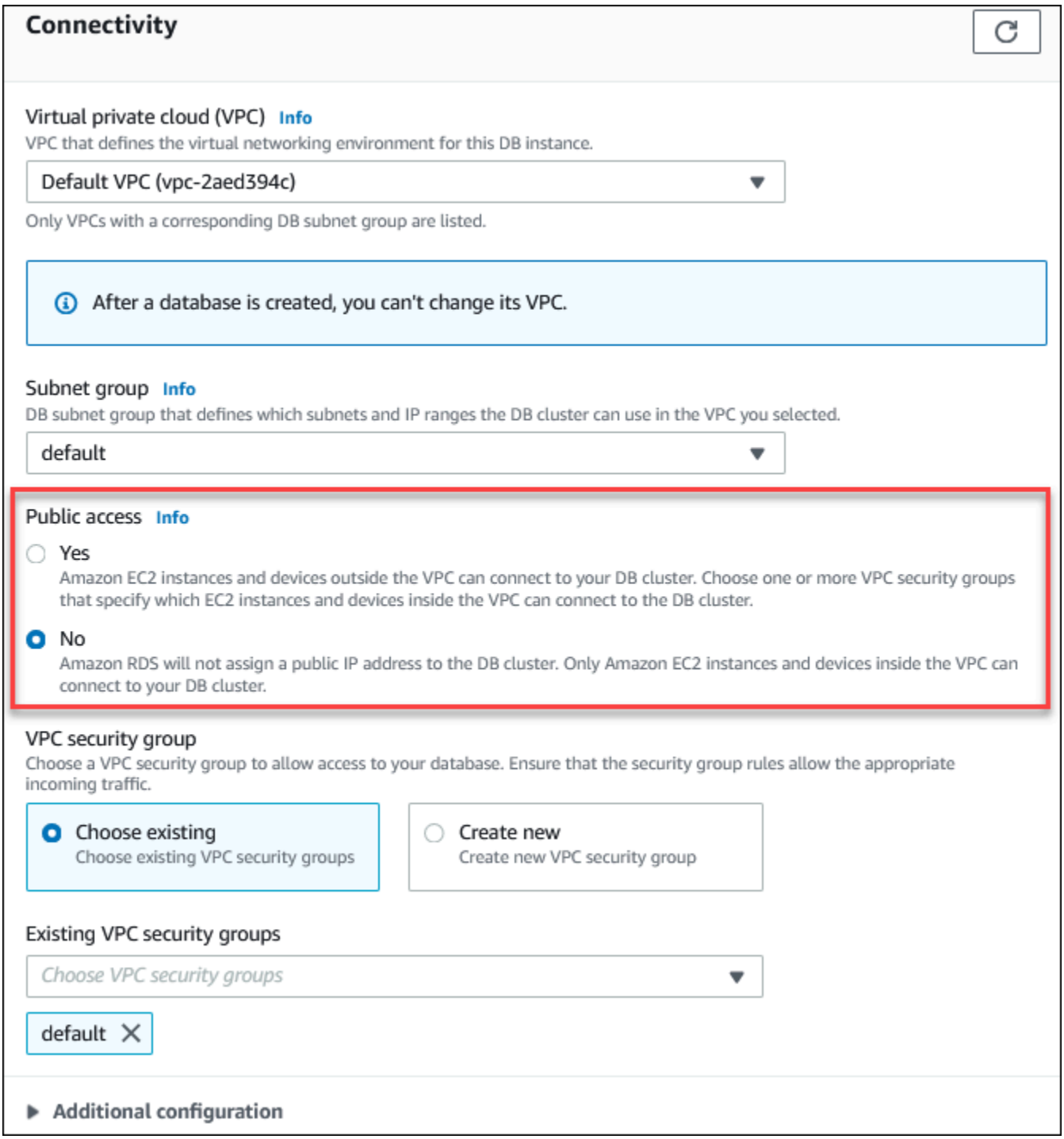
- können das IPv6 Protokoll nicht ausschließlich verwenden. Sie können IPv4 ausschließlich oder das IPv6 AND-Protokoll verwenden (Dual-Stack-Modus). IPv4
- Amazon RDS unterstützt keine systemeigenen IPv6 Subnetze.
- DB-Instances, die den Dual-Stack-Modus verwenden, müssen privat sein. Sie dürfen nicht öffentlich zugänglich sein.
- Bei RDS für SQL Server enthalten DB-Instances im Dual-Stack-Modus, die Listener-Endpunkte für Always AGs On-Verfügbarkeitsgruppen verwenden, nur Adressen. IPv4
- Sie können RDS Proxy nicht mit DB-Instances im Dual-Stack-Modus verwenden.
- Sie können den Dual-Stack-Modus nicht mit RDS auf DB-Instances verwenden. AWS Outposts
- Sie können den Dual-Stack-Modus nicht mit DB-Instances in einer lokalen Zone verwenden.

Ausblenden einer DB-Instance in einer VPC vor dem Internet

Ein gängiges RDS ist eine VPC, in der Sie über eine EC2 Amazon-Instance mit einer öffentlich zugänglichen Webanwendung und einen mit einer Datenbank verfügen, auf die nicht öffentlich zugegriffen werden kann. Sie können beispielsweise eine VPC mit einem öffentlichen Subnetz und einem privaten Subnetz erstellen. EC2 Instanzen, die als Webserver fungieren, können im öffentlichen Subnetz bereitgestellt werden. Die DB-Instances werden im privaten Subnetz bereitgestellt. Bei einer solchen Bereitstellung haben nur die Webserver Zugang zu den DB-Instances. Eine bildliche Darstellung dieses Szenarios finden Sie unter Ein in einer, auf die von einer EC2 Amazon-Instance VPC zugegriffen wird, in derselben VPC.

Wenn Sie eine DB-Instance innerhalb einer VPC starten, verfügt die DB-Instance über eine private IP-Adresse für den Datenverkehr innerhalb der VPC. Diese private IP-Adresse ist nicht öffentlich zugänglich. Mit der Option Public access (Öffentlicher Zugriff) können Sie festlegen, ob die DB-Instance neben der privaten IP-Adresse auch eine öffentliche IP-Adresse hat. Wenn die DB-Instance als öffentlich zugänglich bezeichnet wird, wird ihr DNS-Endpunkt innerhalb der VPC in die private IP-Adresse aufgelöst. Es wird in die öffentliche IP-Adresse von außerhalb der VPC aufgelöst. Zugriff auf die DB-Instance wird letztendlich von der Sicherheitsgruppe kontrolliert, die sie verwendet. Dieser öffentliche Zugang ist nicht erlaubt, wenn die dem DB-Instance zugewiesene Sicherheitsgruppe keine eingehenden Regeln enthält, die ihn erlauben. Damit ein DB-Instance öffentlich zugänglich ist, müssen die Subnetze in seiner DB-Subnetzgruppe außerdem über ein Internet-Gateway verfügen. Weitere Informationen finden Sie unter Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden

Sie können eine DB-Instance ändern und die öffentliche Zugänglichkeit mit der Option Public access (Öffentlicher Zugriff) aktivieren und deaktivieren. Die folgende Abbildung zeigt die Option Public access (Öffentlicher Zugriff) im Abschnitt Additional connectivity configuration (Zusätzliche Konnektivitätskonfiguration) . Um die Option festzulegen, öffnen Sie den Abschnitt Additional connectivity configuration (Zusätzliche Konnektivitätskonfiguration) im Abschnitt Connectivity (Konnektivität) .



Hinweise zum Ändern einer DB-Instance zum Festlegen der Option Public access (Öffentlicher Zugriff) finden Sie unter Ändern einer Amazon RDS DB-Instance.

Erstellen einer DB-Instance in einer VPC

In den folgenden Verfahren wird das Erstellen einer DB-Instance in einer VPC veranschaulicht. Um die Standard-VPC zu verwenden, können Sie mit Schritt 2 beginnen und die VPC- und DB-Subnetzgruppe verwenden, die bereits für Sie erstellt wurden. Wenn Sie eine zusätzliche VPC erstellen möchten, können Sie eine neue VPC erstellen.

> **Note**
>
> Wenn Sie möchten, dass Ihr DB-Instance in der VPC öffentlich zugänglich ist, müssen Sie die DNS-Informationen für die VPC aktualisieren, indem Sie die VPC-Attribute DNS-Hostnamen und DNS-Auflösung aktivieren. Weitere Informationen zum Aktualisieren der DNS-Informationen für eine VPC-Instance finden Sie unter Aktualisieren des DNS-Supports für Ihre VPC.

Gehen Sie wie folgt vor, um eine DB-Instance in einer VPC zu erstellen:

- Schritt 1: Erstellen einer VPC
- Schritt 2: Erstellen einer DB-Subnetzgruppe
- Schritt 3: Erstellen einer VPC-Sicherheitsgruppe
- Schritt 4: Erstellen einer DB-Instance in der VPC

Schritt 1: Erstellen einer VPC

Erstellen Sie eine VPC mit Subnetzen in mindestens zwei Availability Zones. Diese Subnetze verwenden Sie, wenn Sie eine DB-Subnetzgruppe erstellen. Wenn Sie eine Standard-VPC verwenden, wird automatisch ein Subnetz in jeder Availability Zone der AWS-Region für Sie erstellt.

Weitere Informationen finden Sie unter Erstellen einer VPC mit privaten und öffentlichen Subnetzen oder unter Create a VPC (VPC erstellen) im Amazon VPC Benutzerhandbuch.

Schritt 2: Erstellen einer DB-Subnetzgruppe

Eine DB-Subnetzgruppe ist eine Sammlung von Subnetzen (in der Regel private Subnetze), die Sie für eine VPC erstellen und anschließend Ihren DB-Instances zuweisen. Mit einer DB-Subnetzgruppe können Sie eine bestimmte VPC angeben, wenn Sie mithilfe der AWS CLI oder RDS-API erstellen. Wenn Sie die Konsole verwenden, können Sie einfach die VPC und Subnetze auswählen, die Sie verwenden möchten. Jede DB-Subnetzgruppe muss über mindestens ein Subnetz in mindestens

zwei Availability Zones der AWS-Region verfügen. Jede DB-Subnetzgruppe sollte über mindestens ein Subnetz für jede Availability Zone in einer bestimmten AWS-Region verfügen.

Bei Multi-AZ-Bereitstellungen AWS-Region ermöglicht die Definition eines Subnetzes für alle Availability Zones in einer Amazon RDS, bei Bedarf ein neues Standby-Replikat in einer anderen Availability Zone zu erstellen. Sie können diese bewährte Methode auch für Single-AZ-Bereitstellungen einsetzen, da Sie sie in Zukunft möglicherweise in Multi-AZ-Bereitstellungen konvertieren.

Damit öffentlich auf eine DB-Instance zugegriffen werden kann, müssen die Subnetze in der DB-Subnetzgruppe über ein Internet-Gateway verfügen. Weitere Informationen zu Internet-Gateways für Subnetze finden Sie unter Verbinden mit dem Internet durch einen Internet-Gateway im Amazon VPC Benutzerhandbuch.

> **Note**
>
> Die DB-Subnetzgruppe für eine lokale Zone kann nur ein Subnetz haben.

Beim Erstellen einer DB-Instance in einer VPC müssen Sie eine DB-Subnetzgruppe auswählen. Amazon RDS wählt ein Subnetz und eine IP-Adresse innerhalb dieses Subnetzes, um es mit Ihrem DB-Instance zu verbinden. Wenn keine DB-Subnetzgruppen existieren, erstellt Amazon RDS eine Standard-Subnetzgruppe, wenn Sie einen DB-Instance erstellen. Amazon RDS erstellt und ordnet Ihrer DB-Instance eine Elastic-Network-Schnittstelle mit dieser IP-Adresse zu. Die DB-Instance verwendet die Availability Zone, die das Subnetz enthält.

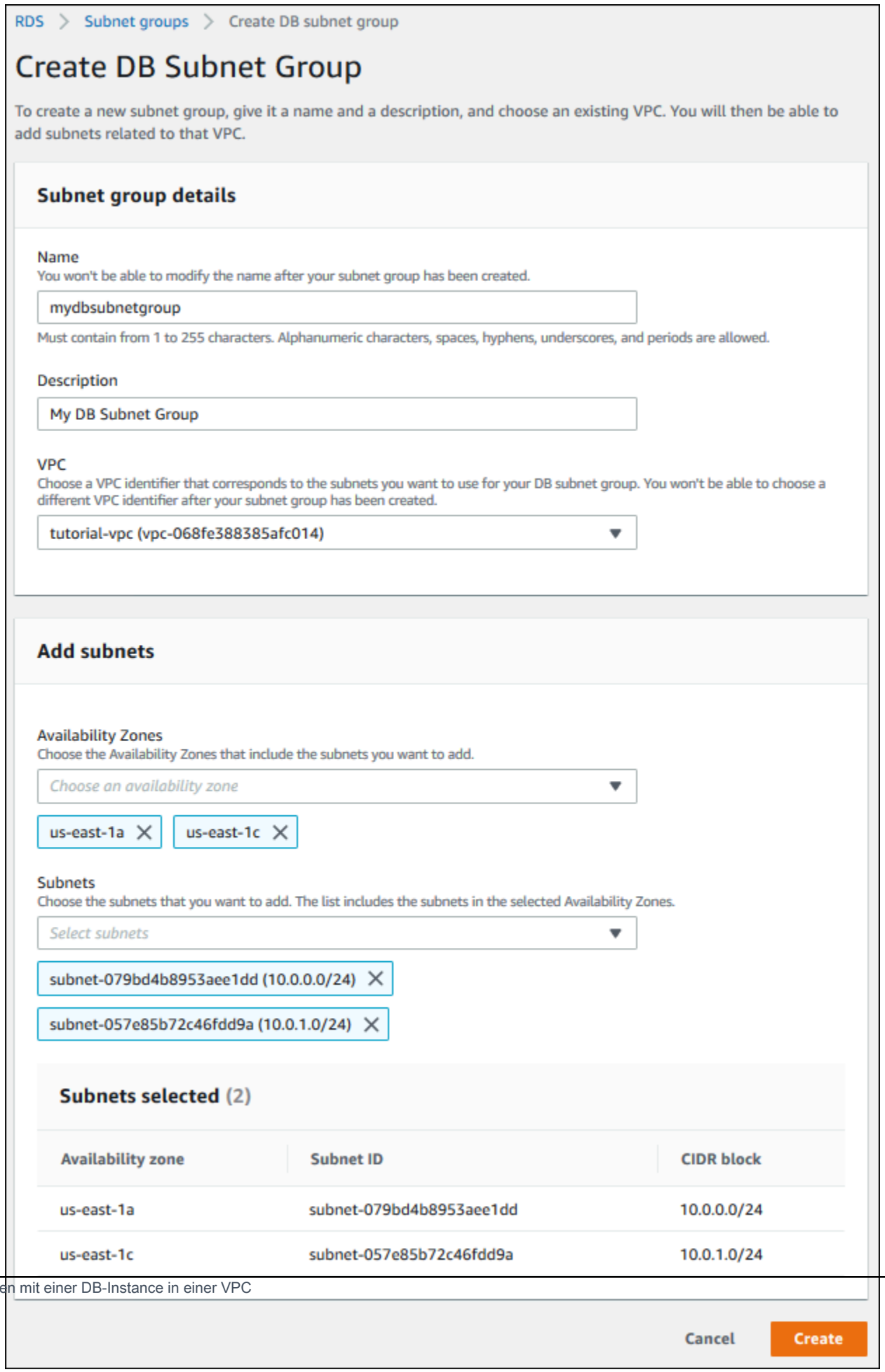
Bei Multi-AZ-Bereitstellungen AWS-Region ermöglicht die Definition eines Subnetzes für zwei oder mehr Availability Zones in einer Amazon RDS, bei Bedarf ein neues Standby in einer anderen Availability Zone einzurichten. Sie müssen dies sogar für Einzel-AZ-Bereitstellungen vornehmen, nur für den Fall, dass Sie sie zu einem späteren Zeitpunkt in Multi-AZ-Bereitstellungen umwandeln möchten.

In diesem Schritt erstellen Sie eine DB-Subnetzgruppe und fügen die Subnetze hinzu, die Sie für Ihre VPC erstellt haben.

Erstellen einer DB-Sicherheitsgruppe

1. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.

3. Wählen Sie DB-Subnetzgruppe erstellen aus.
4. Geben Sie im Feld Name den Namen Ihrer DB-Subnetzgruppe ein.
5. Geben Sie unter Beschreibung eine Beschreibung für Ihre DB-Subnetzgruppe ein.
6. Für VPC wählen Sie die Standard-VPC oder die zuvor erstellte VPC aus.
7. Wählen Sie im Abschnitt Subnetze hinzufügen die Availability Zones aus, die die Subnetze aus Availability Zones enthalten, und wählen Sie dann die Subnetze aus Subnetze aus.


RDS > Subnet groups > Create DB subnet group
Create DB Subnet Group
To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.
Subnet group details
Name
You won't be able to modify the name after your subnet group has been created.
mydbsubnetgroup
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.
Description
My DB Subnet Group
VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
tutorial-vpc (vpc-068fe388385afc014)
Add subnets
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a
us-east-1c
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-079bd4b8953aee1dd (10.0.0.0/24)
subnet-057e85b72c46fdd9a (10.0.1.0/24)
Subnets selected (2)
Availability zone
Subnet ID
CIDR block
us-east-1a
subnet-079bd4b8953aee1dd
10.0.0.0/24
us-east-1c
subnet-057e85b72c46fdd9a
10.0.1.0/24
Cancel
Create

> **Note**
>
> Wenn Sie eine lokale Zone aktiviert haben, können Sie auf der Seite Create DB subnet group (DB-Subnetzgruppe erstellen) eine Gruppe von Availability Zones auswählen. Wählen Sie in diesem Fall die Availability Zone group (Gruppe von Availability Zones), Availability Zones und Subnets (Subnetze) aus.

8. Wählen Sie Create (Erstellen) aus.

   Ihre neue DB-Subnetzgruppe wird in der Liste der DB-Subnetzgruppen in der RDS-Konsole angezeigt. Sie können die DB-Subnetzgruppe auswählen und unten im Detailbereich ausführliche Informationen einschließlich aller Subnetze für diese Gruppe anzeigen.

Schritt 3: Erstellen einer VPC-Sicherheitsgruppe

Vor der DB-Instance müssen Sie eine VPC-Sicherheitsgruppe erstellen, die Ihrer DB-Instance zugeordnet wird. Wenn Sie keine VPC-Sicherheitsgruppe erstellen, können Sie die Standardsicherheitsgruppe beim Erstellen einer DB-Instance verwenden. Eine Anleitung zum Erstellen einer Sicherheitsgruppe für Ihren DB-Instance finden Sie unter Erstellen einer VPC-Sicherheitsgruppe für eine private DB-Instance, oder unter Control traffic to resources using security groups (Kontrolle des Datenverkehrs zu Ressourcen mithilfe von Sicherheitsgruppen) im Amazon VPC Benutzerhandbuch.

Schritt 4: Erstellen einer DB-Instance in der VPC

In diesem Schritt erstellen Sie eine DB-Instance und verwenden den VPC-Namen, die DB-Subnetzgruppe und die VPC-Sicherheitsgruppe, die Sie in den vorherigen Schritten erstellt haben.

> **Note**
>
> Wenn Sie möchten, dass Ihr DBInstance in der VPC öffentlich zugänglich ist, müssen Sie die VPC-Attribute DNS-Hostnamen und DNS-Auflösung aktivieren. Weitere Infomationen finden Sie unter DNS-Attribute für Ihre VPC im Amazon VPC-Benutzerhandbuch.

Weitere Informationen zum Erstellen einer DB-Instance finden Sie unter Erstellen einer Amazon RDS-DB-Instance .

Wenn Sie im Abschnitt Konnektivität dazu aufgefordert werden, geben Sie den VPC-Namen, die DB-Subnetzgruppe und die VPC-Sicherheitsgruppe ein.

Aktualisierung der VPC für eine DB-Instance

Sie können die verwenden AWS Management Console , um Ihre DB-Instance auf eine andere zu verschiebenVPC.

Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance. Geben Sie im Abschnitt Connectivity (Konnektivität) der unten angezeigten Änderungsseite für DB-Subnet group (Subnetzgruppe) die neue DB-Subnetzgruppe ein. Die neue Subnetzgruppe muss eine Subnetzgruppe in einer neuen sein. VPC



Sie können die VPC für eine DB-Instance nicht ändern, wenn die folgenden Bedingungen zutreffen:

- Die DB-Instance befindet sich in mehreren Availability Zones. Sie können die DB-Instance in eine einzelne Availability Zone konvertieren, sie in eine neue VPC verschieben und sie dann wieder in eine Multi-AZ-DB-Instance konvertieren. Weitere Informationen finden Sie unter Konfiguration und Verwaltung einer Multi-AZ-Bereitstellung für Amazon RDS.
- Die DB-Instance hat mindestens ein Lesereplikat. Sie können die Read Replicas entfernen, die DB-Instance in eine neue VPC verschieben und dann die Read Replicas erneut hinzufügen. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.
- Die DB-Instance ist ein Lesereplikat. Sie können die Read Replica hochstufen und dann die eigenständige DB-Instance in eine neue verschieben. VPC Weitere Informationen finden Sie unter Hochstufen eines Lesereplikats zur eigenständigen DB-Instance.
- Die Subnetzgruppe im Ziel VPC hat keine Subnetze in der Availability Zone der DB-Instance. Sie können der DB-Subnetzgruppe Subnetze in der Availability Zone der DB-Instance hinzufügen und dann die DB-Instance in die neue verschieben. VPC Weitere Informationen finden Sie unter Arbeiten mit DB-Subnetzgruppen.

Szenarien für den Zugriff auf einen in einem VPC

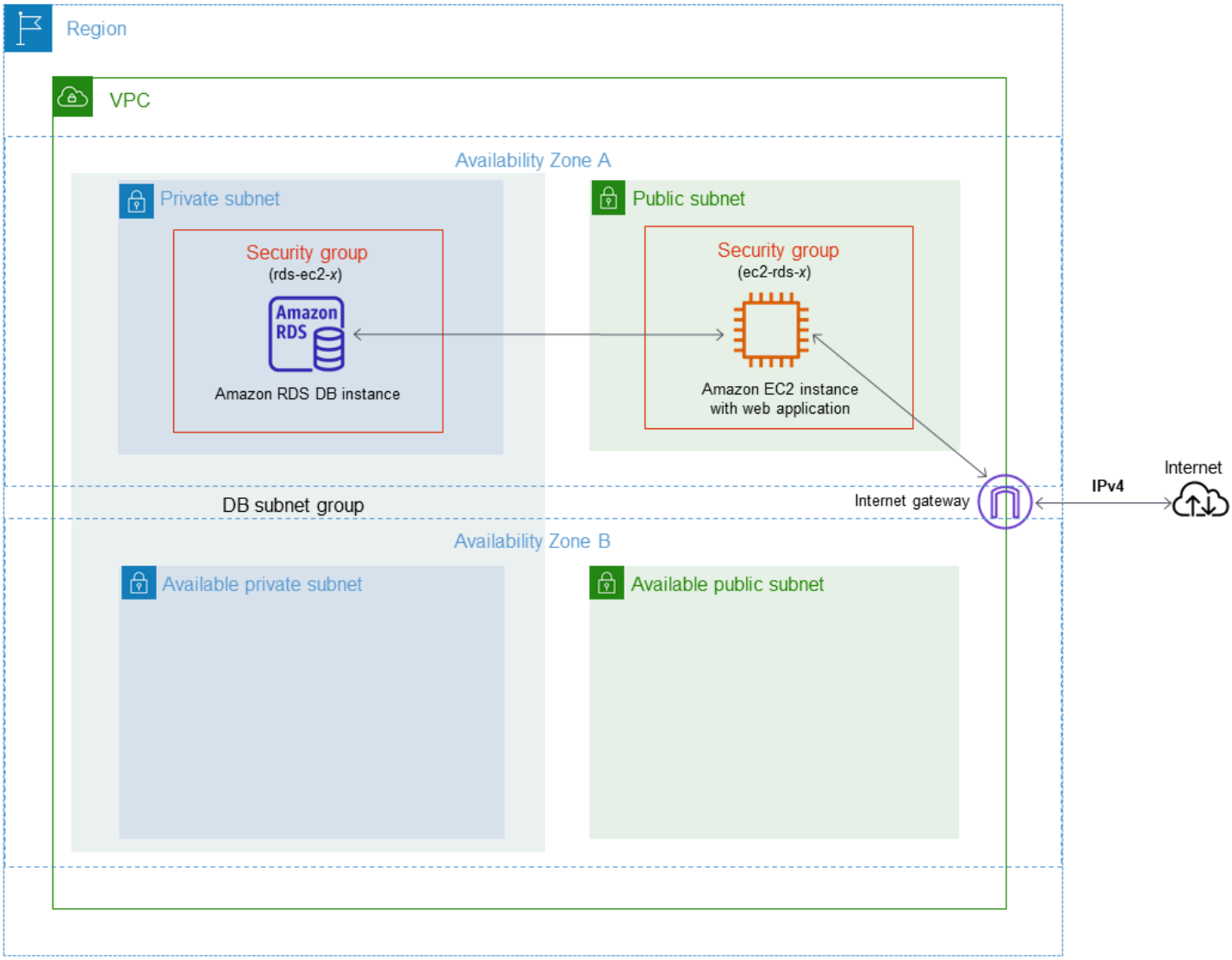
Amazon RDS unterstützt die folgenden Szenarien für den Zugriff auf einen in einemVPC:

- Eine EC2 Amazon-Instanz in derselben VPC
- Eine EC2 Instance in einer anderen VPC
- Eine Client-Anwendung über das Internet
- Ein Privates Netzwerk

Ein in einer, auf die von einer EC2 Amazon-Instance VPC zugegriffen wird, in derselben VPC

Ein in einem VPC wird häufig verwendet, um Daten mit einem Anwendungsserver gemeinsam zu nutzen, der in derselben EC2 Amazon-Instance ausgeführt wirdVPC.

Im folgenden Diagramm wird dieses Szenario veranschaulicht.



Der einfachste Weg, den Zugriff zwischen EC2 Instances und gleichzeitig zu verwalten, VPC besteht darin, wie folgt vorzugehen:

- Erstellen Sie eine VPC Sicherheitsgruppe, in der sich Ihre befinden sollen. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf DB-Instances zu beschränken. Sie können beispielsweise eine benutzerdefinierte Regel für diese Sicherheitsgruppe erstellen. Dies kann den TCP Zugriff über den Port ermöglichen, den Sie dem bei der Erstellung zugewiesen haben, und über eine IP-Adresse, die Sie für den Zugriff auf den für Entwicklungs- oder andere Zwecke verwenden.
- Erstellen Sie eine VPC Sicherheitsgruppe, in der sich Ihre EC2 Instances (Webserver und Clients) befinden sollen. Diese Sicherheitsgruppe kann bei Bedarf mithilfe der Routingtabelle von den Zugriff auf die EC2 Instance aus dem VPC Internet ermöglichen. Sie können beispielsweise Regeln

für diese Sicherheitsgruppe festlegen, um den TCP Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.

- Erstellen Sie in der Sicherheitsgruppe benutzerdefinierte Regeln für Ihre , die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Diese Regeln können jedem Mitglied der Sicherheitsgruppe den Zugang zu den DB-Instances ermöglichen.

Es gibt ein zusätzliches öffentliches und privates Subnetz in einer separaten Availability Zone. Eine RDS DB-Subnetzgruppe erfordert ein Subnetz in mindestens zwei Availability Zones. Das zusätzliche Subnetz erleichtert den Wechsel zu einer Multi-AZ-DB-Instance-Bereitstellung in der future.

Ein Tutorial, das Ihnen zeigt, wie Sie für dieses Szenario ein sowohl VPC mit öffentlichen als auch mit privaten Subnetzen erstellen, finden Sie unter. Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur)

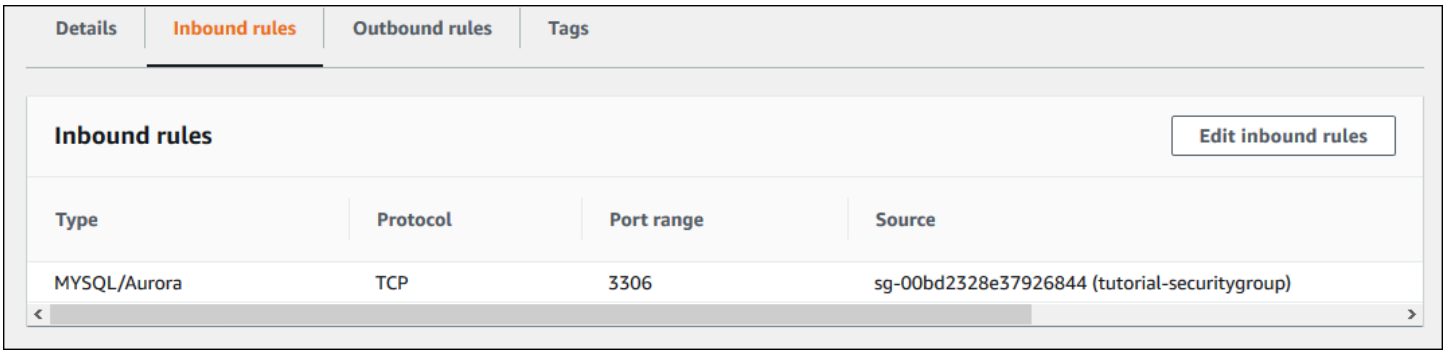
> **Tip**
>
> Sie können die Netzwerkkonnektivität zwischen einer EC2 Amazon-Instance und einem automatisch einrichten, wenn Sie den erstellen. Weitere Informationen finden Sie unter Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2 .

Gehen Sie wie folgt vor, um eine Regel in einer VPC Sicherheitsgruppe zu erstellen, die Verbindungen von einer anderen Sicherheitsgruppe aus zulässt:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die VPC Amazon-Konsole unter https://console.aws.amazon.com/vpc.
2. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
3. Erstellen oder wählen Sie eine Sicherheitsgruppe aus, der Sie den Zugriff zu Teilen einer anderen Sicherheitsgruppe erlauben möchten. Im vorhergehenden Szenario ist dies die Sicherheitsgruppe, die Sie für Ihre DB-Instances verwenden. Wählen Sie die Registerkarte Inbound Rules (Eingehende Regeln) und anschließend Edit Inbound Rules (Eingehende Regeln bearbeiten) aus.
4. Wählen Sie auf der Seite Edit inbound rules (Regeln für eingehenden Datenverkehr bearbeiten) die Option Add Rule (Regel hinzufügen).
5. Wählen Sie für Type den Eintrag aus, der dem Port entspricht, den Sie bei der Erstellung Ihres verwendet haben, z. B. MYSQL/Aurora.

6. Geben Sie im Feld Source (Quelle) die ID der Sicherheitsgruppe ein. Anschließend werden die übereinstimmenden Sicherheitsgruppen auflistet. Wählen Sie die Sicherheitsgruppe mit den Mitgliedern aus, die Zugriff auf die durch diese Sicherheitsgruppe geschützten Ressourcen erhalten sollen. Im vorherigen Szenario ist dies die Sicherheitsgruppe, die Sie für Ihre EC2 Instance verwenden.
7. Falls erforderlich, wiederholen Sie die Schritte für das TCP Protokoll, indem Sie eine Regel mit All TCP als Typ und Ihrer Sicherheitsgruppe im Feld Quelle erstellen. Wenn Sie das UDP Protokoll verwenden möchten, erstellen Sie eine Regel mit All UDP als Typ und Ihrer Sicherheitsgruppe im Feld Source.
8. Wählen Sie Save rules (Regeln speichern) aus.

Der folgende Bildschirm zeigt eine eingehende Regel mit einer Sicherheitsgruppe für ihre Quelle.

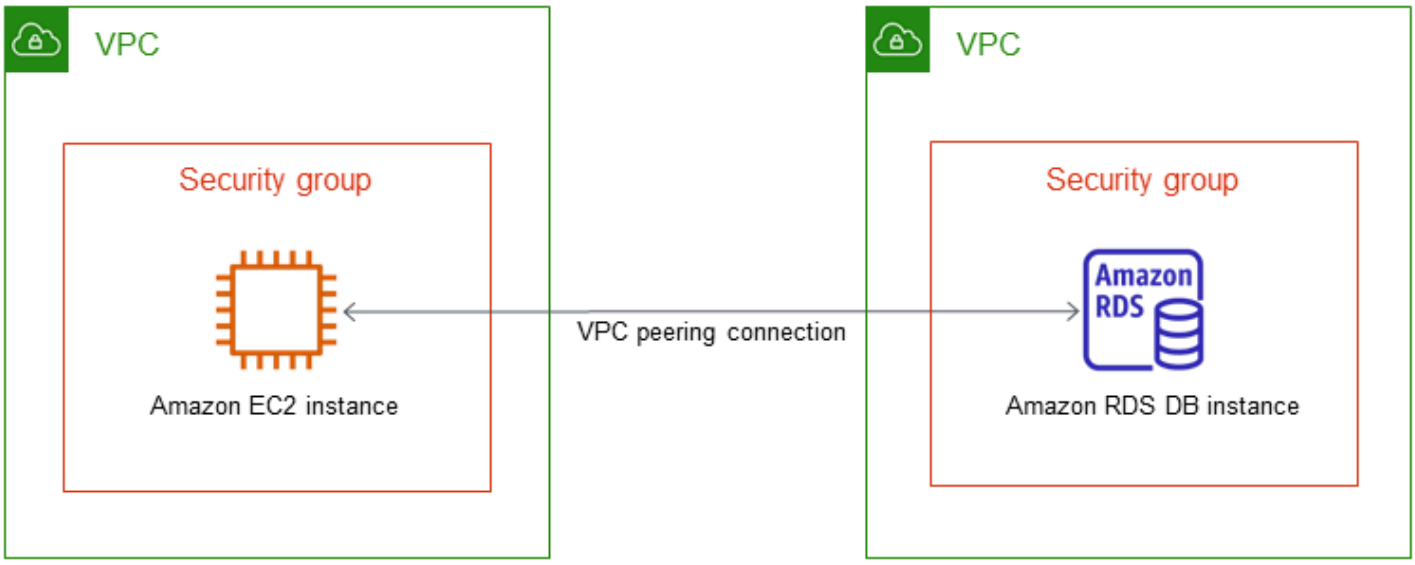


Weitere Informationen zum Herstellen einer Verbindung mit der von Ihrer EC2 Instance aus finden Sie unter Verbindung zu einer Amazon RDS DB-Instance herstellen.

Ein in einer, auf die von einer EC2 Instance in einer anderen VPC zugegriffen wird VPC

Wenn sich Ihr in einer anderen als VPC der EC2 Instance befindet, mit der Sie darauf zugreifen, können Sie VPC Peering verwenden, um auf den zuzugreifen.

Im folgenden Diagramm wird dieses Szenario veranschaulicht.

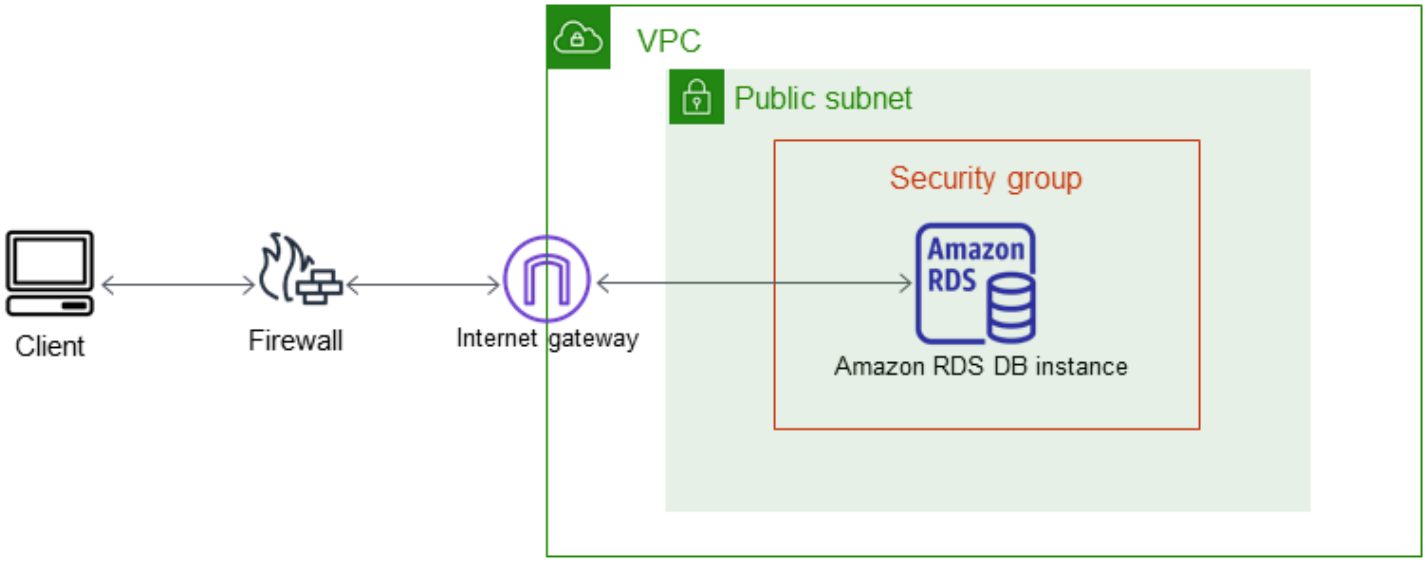


Eine VPC Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei PersonenVPCs, die es Ihnen ermöglicht, den Verkehr zwischen ihnen mithilfe privater IP-Adressen weiterzuleiten. Ressourcen in beiden VPC können miteinander kommunizieren, als ob sie sich im selben Netzwerk befinden würden. Sie können eine VPC Peering-Verbindung zwischen Ihrem eigenen KontoVPCs, mit einem Konto VPC in einem anderen oder mit einem AWS Konto VPC in einem anderen AWS-Region herstellen. Weitere Informationen über VPC Peering finden Sie unter VPCPeering im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Ein in einer Anwendung, auf die ein Client über das Internet VPC zugreift

Um von einer Client-Anwendung VPC aus über das Internet auf einer DB-Instance zuzugreifen, konfigurieren Sie eine VPC mit einem einzigen öffentlichen Subnetz und ein Internet-Gateway, um die Kommunikation über das Internet zu ermöglichen.

Im folgenden Diagramm wird dieses Szenario veranschaulicht.



Wir empfehlen die folgende Konfiguration:

- A VPC der Größe /16 (zum BeispielCIDR: 10.0.0.0/16). Diese Größe bietet 65.536 private IP-Adressen.
- Ein Subnetz der Größe /24 (zum Beispiel: 10.0.0.0/24). CIDR Diese Größe bietet 256 private IP-Adressen.
- Ein Amazon RDS Amazon , der dem VPC und dem Subnetz zugeordnet ist.
- Ein Internet-Gateway, das Sie mit VPC dem Internet und anderen AWS Produkten verbindet.
- Eine Sicherheitsgruppe, die der DB-Instance zugeordnet ist. Die Regeln für eingehenden Datenverkehr der Sicherheitsgruppe erlauben der Clientanwendung den Zugriff auf die DB-Instance .

Hinweise zum Erstellen eines in einem VPC finden Sie unterErstellen einer DB-Instance in einer VPC.

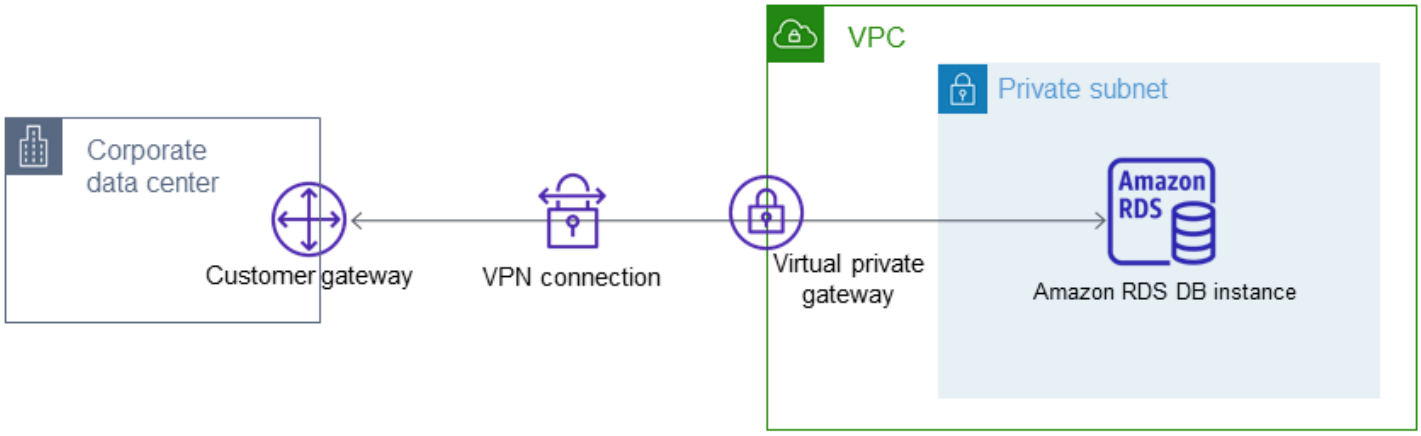
Ein in einem Netzwerk, auf das über ein privates Netzwerk VPC zugegriffen wird

Wenn Ihre DB-Instance nicht öffentlich zugänglich ist, haben Sie die folgenden Optionen, um von einem privaten Netzwerk aus darauf zuzugreifen:

- Eine AWS Site-to-Site VPN Verbindung. Weitere Informationen finden Sie unter Was ist AWS Site-to-Site VPN?

- Eine AWS Direct Connect Verbindung. Weitere Informationen finden Sie unter Was ist AWS Direct Connect?
- Eine AWS Client VPN Verbindung. Weitere Informationen finden Sie unter Was ist AWS Client VPN?

Das folgende Diagramm zeigt ein Szenario mit einer AWS Site-to-Site VPN Verbindung.

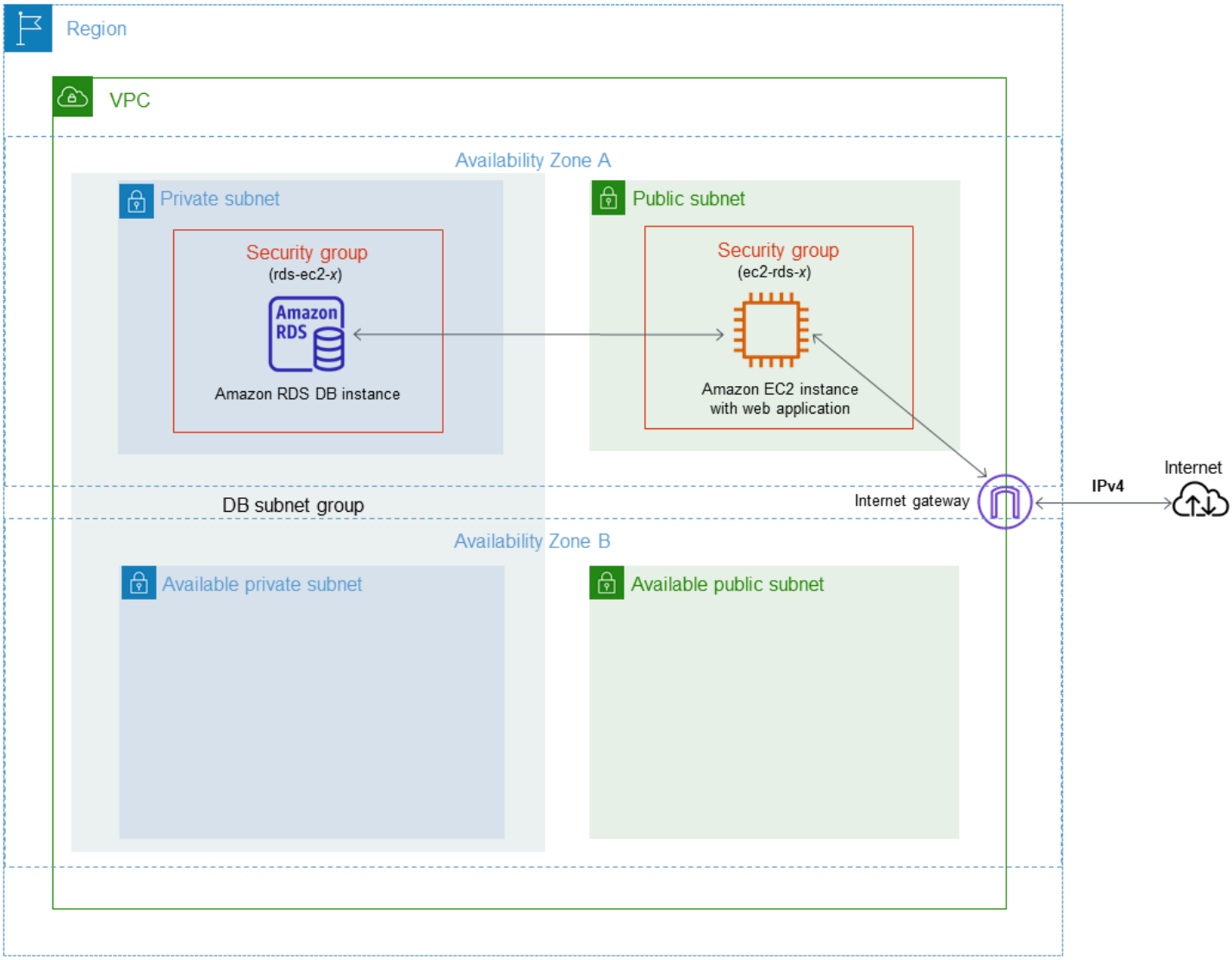


Weitere Informationen finden Sie unter Richtlinie für den Datenverkehr zwischen Netzwerken.

Tutorial: Eine VPC zur Verwendung mit einem erstellen (IPv4 nur)

Ein häufiges Szenario umfasst eine DB-Instance in einer Virtual Private Cloud (VPC), die auf dem Amazon-VPC-Service basiert. Diese VPC teilt Daten mit einem Webserver, der in derselben VPC ausgeführt wird. In diesem Tutorial erstellen Sie die VPC für dieses Szenario.

Im folgenden Diagramm wird dieses Szenario veranschaulicht. Weitere Informationen zu anderen Szenarien finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.



Ihre DB-Instance braucht nur für Ihren Webserver verfügbar zu sein und muss nicht vom öffentlichen Internet aus erreichbar sein. Daher erstellen Sie eine VPC sowohl mit öffentlichen als auch mit privaten Subnetzen. Der Webserver wird im öffentlichen Subnetz gehostet, damit er im öffentlichen Internet erreicht werden kann. Die DB-Instance wird in einem privaten Subnetz gehostet. Der Webserver kann eine Verbindung mit der DB-Instance herstellen, da das Hosten innerhalb derselben

VPC erfolgt. Die DB-Instance ist jedoch nicht für das öffentliche Internet verfügbar und bietet so mehr Sicherheit.

In diesem Tutorial wird ein zusätzliches öffentliches und ein privates Subnetz in einer separaten Availability Zone konfiguriert. Diese Subnetze werden im Tutorial nicht verwendet. Eine RDS-DB-Subnetzgruppe erfordert ein Subnetz in mindestens zwei Availability Zones. Das zusätzliche Subnetz erleichtert den späteren Wechsel zu einer Multi-AZ-Bereitstellung der DB-Instance.

In diesem Tutorial wird das Konfigurieren einer VPC für Amazon RDS-DB-Instances beschrieben. Ein Tutorial, das Ihnen veranschaulicht, wie Sie einen Webserver für dieses VPC-Szenario erstellen, finden Sie unter Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance. Weitere Informationen zu Amazon VPC finden Sie unter Amazon-VPC-Handbuch „Erste Schritte“ und Amazon-VPC-Benutzerhandbuch.

> **Tip**
>
> Sie können die Netzwerkkonnektivität zwischen einer EC2 Amazon-Instance und einem automatisch einrichten, wenn Sie den erstellen. Die Netzwerkkonfiguration ähnelt der in diesem Tutorial beschriebenen Konfiguration. Weitere Informationen finden Sie unter Konfigurieren Sie die automatische Netzwerkkonnektivität mit einer Instance EC2 .

Erstellen einer VPC mit privaten und öffentlichen Subnetzen

Verwenden sie die folgenden Vorgänge, um eine VPC sowohl mit öffentlichen als auch mit privaten Subnetzen zu erstellen.

So erstellen Sie eine VPC und Subnetze

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie in der oberen rechten Ecke von die Region aus AWS Management Console, in der Sie Ihre VPC erstellen möchten. In diesem Beispiel wird die Region USA West (Oregon) verwendet.
3. Wählen Sie links oben die Option VPC-Dashboard aus. Wählen Sie Create VPC (VPC erstellen) aus, um mit dem Erstellen einer VPC zu beginnen.
4. Wählen Sie unter VPC Settings (VPC-Einstellungen) für Resources to create (Zu erstellende Ressourcen) VPC and more (VPC und mehr) aus.
5. Legen Sie für VPC settings (VPC-Einstellungen) folgende Werte fest:

- Name tag auto-generation (Namens-Tag automatisch erstellen) – **`tutorial`**
- IPv4 CIDR-Block — **`10.0.0.0/16`**
- IPv6 CIDR-Block — Kein IPv6 CIDR-Block
- Tenancy – Default (Standard)
- Anzahl der Availability Zones (AZs) — 2
- Anpassen AZs — Behalten Sie die Standardwerte bei.
- Number of public subnet (Anzahl der öffentlichen Subnetze) – 2
- Number of private subnets (Anzahl der privaten Subnetze) – 2
- Customize subnets CIDR blocks (CIDR-Blöcke für Subnetze anpassen) – Übernehmen Sie die Standardwerte.
- NAT gateways ($) (NAT-Gateways ($)) – None (Keine)
- VPC endpoints (VPC-Endpunkte) – None (Keine)
- DNS options (DNS-Optionen) – Übernehmen Sie die Standardwerte.

> **Note**
>
> Amazon RDS erfordert mindestens zwei Subnetze in zwei verschiedenen Availability Zones, um Multi-AZ-Bereitstellungen von DB-Instances zu unterstützen. In diesem Tutorial wird eine Single-AZ-Bereitstellung erstellt, aber die Anforderung erleichtert die spätere Konvertierung in eine Multi-AZ-Bereitstellung von DB-Instances.

6. Wählen Sie VPC erstellen aus.

Erstellen einer VPC-Sicherheitsgruppe für einen öffentlichen Webserver

Als Nächstes erstellen Sie eine Sicherheitsgruppe für öffentlichen Zugriff. Um eine Verbindung zu öffentlichen EC2 Instances in Ihrer VPC herzustellen, fügen Sie Ihrer VPC-Sicherheitsgruppe Regeln für eingehenden Datenverkehr hinzu. Diese ermöglichen die Verbindung des Datenverkehrs aus dem Internet.

So erstellen Sie eine VPC-Sicherheitsgruppe

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

2. Wählen Sie VPC Dashboard (VPC-Dashboard), Security Groups (Sicherheitsgruppen) und anschließend Create security group (Sicherheitsgruppe erstellen).
3. Legen Sie auf der Seite Create security group (Sicherheitsgruppe erstellen) die folgenden Werte fest:

   - Security group name (Name der Sicherheitsgruppe: **`tutorial-securitygroup`**
   - Description (Beschreibung: **`Tutorial Security Group`**
   - VPC: Wählen Sie die VPC aus, die Sie zuvor erstellt haben, zum Beispiel: vpc- (tutorial-vpc) ***`identifier`***
4. Fügen Sie der Sicherheitsgruppe Regeln für den eingehenden Datenverkehr hinzu.

   a. Ermitteln Sie die IP-Adresse, die verwendet werden soll, um mithilfe von Secure Shell (SSH) eine Verbindung zu EC2 Instances in Ihrer VPC herzustellen. Um Ihre öffentliche IP-Adresse in einem anderen Browserfenster oder einer anderen Registerkarte zu ermitteln, können Sie den Dienst unter verwenden. https://checkip.amazonaws.com Ein Beispiel für eine IP-Adresse ist `203.0.113.25/32`.

      In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Suchen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

      > **Warning**
      >
      > Wenn Sie `0.0.0.0/0` für SSH-Zugriff verwenden, ermöglichen Sie für alle IP-Adressen den Zugriff auf Ihre öffentlichen Instances. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Für die Produktion wird nur eine bestimmte IP-Adresse bzw. ein bestimmter Adressbereich für den Zugriff auf Ihre Instances autorisiert.

   b. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.

   c. Legen Sie die folgenden Werte für Ihre neue Eingangsregel fest, um SSH-Zugriff auf Ihre EC2 Amazon-Instance zu ermöglichen. In diesem Fall können Sie eine Verbindung zu Ihrer EC2 Amazon-Instance herstellen, um den Webserver und andere Dienstprogramme zu installieren. Sie stellen auch eine Verbindung zu Ihrer EC2 Instance her, um Inhalte für Ihren Webserver hochzuladen.

      - Typ: **`SSH`**

- Quelle: Die IP-Adresse bzw. der IP-Bereich aus Schritt a, zum Beispiel: **`203.0.113.25/32`**.

d. Wählen Sie Add rule.

e. Stellen Sie die folgenden Werte für Ihre neue eingehende Regel ein, um HTTP-Zugriff auf Ihren Webserver zuzulassen:

- Typ: **`HTTP`**
- Quelle: **`0.0.0.0/0`**

5. Wählen Sie Create security group (Sicherheitsgruppe erstellen) aus, um die Sicherheitsgruppe zu erstellen.

   Notieren Sie sich die Sicherheitsgruppen-ID, da Sie sie später in diesem Tutorial benötigen.

Erstellen einer VPC-Sicherheitsgruppe für eine private DB-Instance

Erstellen Sie eine zweite Sicherheitsgruppe für privaten Zugriff, um Ihre DB-Instance privat zu halten. Um eine Verbindung zu privaten in Ihrer VPC herzustellen, fügen Sie Ihrer VPC-Sicherheitsgruppe Regeln für eingehenden Datenverkehr hinzu, die nur Datenverkehr von Ihrem Webserver zulassen.

So erstellen Sie eine VPC-Sicherheitsgruppe

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie VPC Dashboard (VPC-Dashboard), Security Groups (Sicherheitsgruppen) und anschließend Create security group (Sicherheitsgruppe erstellen).
3. Legen Sie auf der Seite Create security group (Sicherheitsgruppe erstellen) die folgenden Werte fest:

   - Security group name (Name der Sicherheitsgruppe: **`tutorial-db-securitygroup`**
   - Description (Beschreibung: **`Tutorial DB Instance Security Group`**
   - VPC: Wählen Sie die VPC aus, die Sie zuvor erstellt haben, zum Beispiel: vpc- (tutorial-vpc) ***`identifier`***

4. Fügen Sie der Sicherheitsgruppe Regeln für den eingehenden Datenverkehr hinzu.

   a. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.

   b. Legen Sie die folgenden Werte für Ihre neue eingehende Regel fest, um MySQL-Verkehr auf Port 3306 von Ihrer EC2 Amazon-Instance aus zuzulassen. In diesem Fall können Sie von

Ihrem Webserver eine Verbindung mit Ihrer DB-Instance herstellen. Auf diese Weise können Sie Daten aus Ihrer Webanwendung in Ihre Datenbank abrufen und dort speichern.

- Typ: **`MySQL/Aurora`**
- Source (Quelle): Die Kennung der Sicherheitsgruppe tutorial-securitygroup, die Sie zuvor in diesem Tutorial erstellt haben, z. B. sg-9edd5cfb.

5. Wählen Sie Create security group (Sicherheitsgruppe erstellen) aus, um die Sicherheitsgruppe zu erstellen.

Erstellen einer DB-Subnetzgruppe

Eine DB-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie in einer VPC erstellen und anschließend den DB-Instances zuweisen. Mithilfe einer DB-Subnetzgruppe können Sie beim Erstellen von DB-Instances eine bestimmte VPC festlegen.

Erstellen einer DB-Sicherheitsgruppe

1. Identifizieren Sie die privaten Subnetze für Ihre Datenbank in der VPC.

   a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
   b. Wählen Sie VPC Dashboard (VPC-Dashboard) und dann Subnets (Subnetze) aus.
   c. Notieren Sie sich das Subnetz IDs der Subnetze mit den Namen tutorial-subnet-private 1-us-west-2a und 2-us-west-2b. tutorial-subnet-private

   Sie benötigen das Subnetz, wenn Sie Ihre DB-Subnetzgruppe erstellen IDs .

2. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

   Stellen Sie sicher, dass Sie eine Verbindung mit der Amazon-RDS-Konsole herstellen, nicht mit der Amazon-VPC-Konsole.

3. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
4. Wählen Sie Create DB subnet group (DB-Subnetzgruppe erstellen) aus.
5. Legen Sie auf der Seite DB-Subnetzgruppe erstellen unter Subnetzgruppendetails die folgenden Werte fest:

   - Name (Name: **`tutorial-db-subnet-group`**
   - Description (Beschreibung: **`Tutorial DB Subnet Group`**
   - VPC: Tutorial-vpc (vpc -) *`identifier`*

6. Wählen Sie im Abschnitt Subnetze hinzufügen die Availability Zones und Subnetze aus.

   Wählen Sie für dieses Tutorial us-west-2a und us-west-2b als Availability Zones aus. Wählen Sie für Subnets (Subnetze) die privaten Subnetze aus, die Sie im vorherigen Schritt identifiziert haben.

7. Wählen Sie Erstellen aus.

   Ihre neue DB-Subnetzgruppe wird in der Liste der DB-Subnetzgruppen in der RDS-Konsole angezeigt. Sie können die DB-Subnetzgruppe auswählen und unten im Detailbereich ausführliche Informationen anzeigen. Diese Informationen umfassen alle Subnetze, die der Gruppe zugeordnet sind.

> **Note**
>
> Wenn Sie diese VPC zum Vervollständigen von Tutorial: Erstellen eines Webservers und einer Amazon RDS-DB-Instance erstellt haben, erstellen Sie die DB-Instance, indem Sie die Anweisungen unter Erstellen einer DB-Instance von Amazon RDS befolgen.

Löschen der VPC

Nachdem Sie die VPC und andere Ressourcen für dieses Tutorial erstellt haben, können Sie sie löschen, wenn sie nicht mehr benötigt werden.

> **Note**
>
> Wenn Sie in der VPC, die Sie für dieses Tutorial erstellt haben, Ressourcen hinzugefügt haben, müssen Sie diese möglicherweise löschen, bevor Sie die VPC löschen können. Zu diesen Ressourcen können beispielsweise EC2 Amazon-Instances oder Amazon gehören. Weitere Informationen finden Sie unter Löschen Ihrer VPC im Amazon VPC User Guide.

So löschen Sie eine VPC und zugehörige Ressourcen

1. Löschen Sie die DB-Subnetzgruppe.

   a. Öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.

   b. Wählen Sie im Navigationsbereich Subnetzgruppe aus.

c. Wählen Sie die DB-Subnetzgruppe aus, die Sie löschen möchten, z. B. tutorial-db-subnet-group

d. Wählen Sie Löschen, und wählen Sie dann im Bestätigungsfenster Löschen.

2. Notieren Sie die VPC ID.

a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

b. Wählen Sie VPC Dashboard und dann. VPCs

c. Suchen Sie in der Liste die von Ihnen erstellte VPC, z. B.tutorial-vpc.

d. Notieren Sie die VPC ID (VPC-ID) der VPC, die Sie erstellt haben. Sie benötigen die VPC-ID in späteren Schritten.

3. Löschen der Sicherheitsgruppe.

a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

b. Wählen Sie VPC Dashboard und dann Sicherheitsgruppen.

c. Wählen Sie die Sicherheitsgruppe für die Amazon RDS-DB-Instance aus, z. tutorial-db-securitygroupB.

d. Wählen Sie unter Actions (Aktionen) Delete security groups (Sicherheitsgruppen löschen) und dann Delete (Löschen) auf der Bestätigungsseite aus.

e. Wählen Sie auf der Seite Sicherheitsgruppen die Sicherheitsgruppe für die EC2 Amazon-Instance aus, z. B. tutorial-securitygroup.

f. Wählen Sie unter Actions (Aktionen) Delete security groups (Sicherheitsgruppen löschen) und dann Delete (Löschen) auf der Bestätigungsseite aus.

4. Löschen Sie die VPC.

a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

b. Wählen Sie VPC Dashboard und dann. VPCs

c. Wählen Sie die VPC aus, die Sie löschen möchten, z. B.tutorial-vpcaus.

d. Wählen Sie unter Actions (Aktionen) die Option Delete VPC (VPC löschen) aus.

Auf der Bestätigungsseite werden weitere Ressourcen angezeigt, die der VPC zugeordnet sind, die ebenfalls gelöscht werden, einschließlich der damit verknüpften Subnetze.

e. Geben Sie auf der Bestätigungsseite **`delete`** ein, und wählen Sie die Option Delete (Löschen) aus.

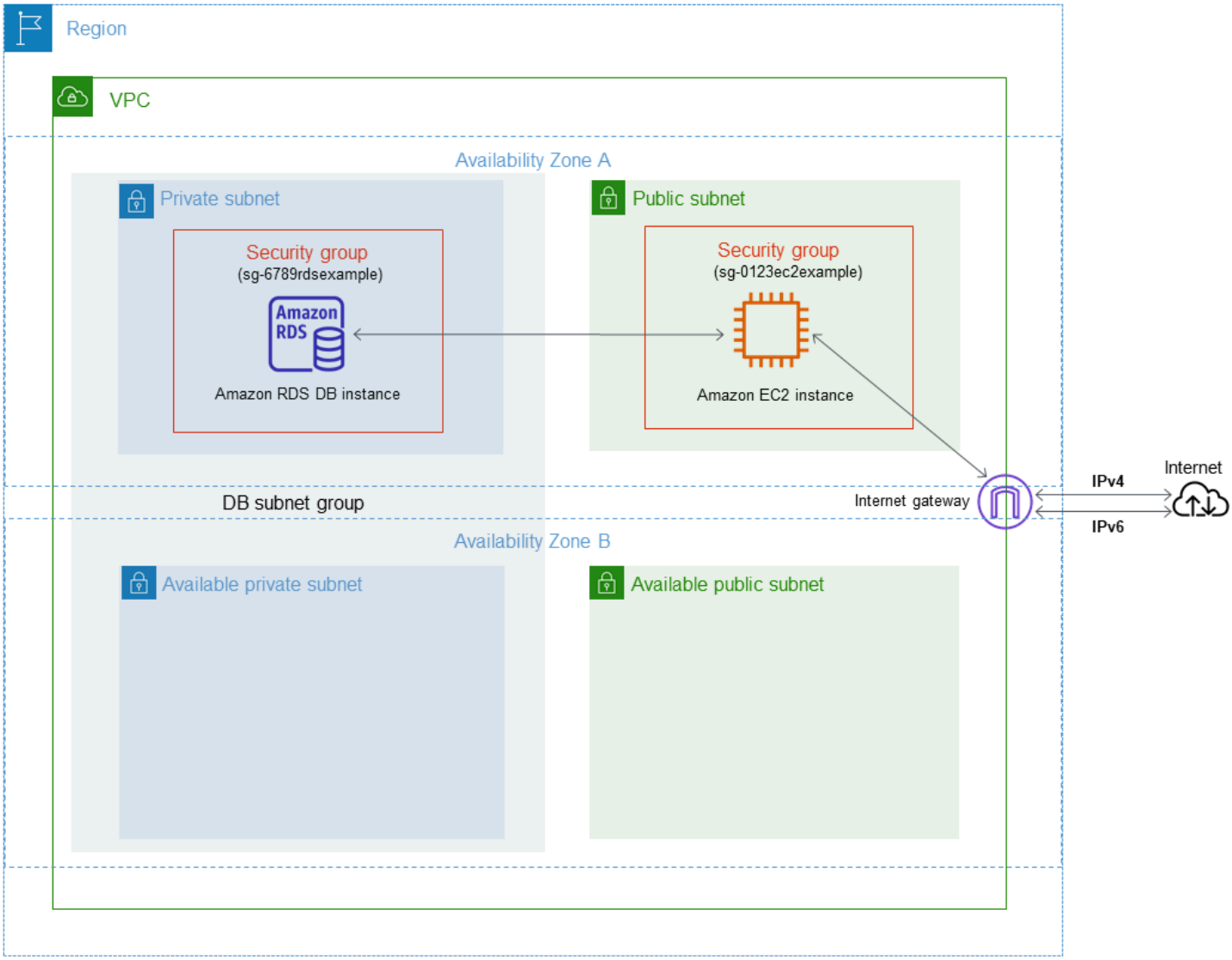
Tutorial: Erstellen einer VPC zur Verwendung mit einer DB–Instance (Dual-Stack-Modus)

Ein häufiges Szenario umfasst eine DB-Instance in einer Virtual Private Cloud (VPC), die auf dem Amazon-VPC-Service basiert. Diese VPC teilt Daten mit einer öffentlichen Amazon-EC2-Instance, die in derselben VPC ausgeführt wird.

In diesem Tutorial erstellen Sie die VPC für dieses Szenario, die mit einer Datenbank arbeitet, die im Dual-Stack-Modus ausgeführt wird. Dual-Stack-Modus, um eine Verbindung über das IPv6-Adressierungsprotokoll zu ermöglichen. Weitere Informationen über die IP-Adressierung finden Sie unter Amazon-RDS-IP-Adressierung.

Dual-Stack-Netzwerk-Instances werden in den meisten Regionen unterstützt. Weitere Informationen finden Sie unter Verfügbarkeit von Regionen und Versionen. Informationen zu den Einschränkungen des Dual-Stack-Modus finden Sie unter Einschränkungen für Dual-Stack-Netzwerk-DB-Instances.

Im folgenden Diagramm wird dieses Szenario veranschaulicht.



Weitere Informationen zu anderen Szenarien finden Sie unter Szenarien für den Zugriff auf einen in einem VPC.

Ihre DB-Instance muss nur für Ihre Amazon-EC2-Instance verfügbar sein und nicht im öffentlichen Internet. Daher erstellen Sie eine VPC sowohl mit öffentlichen als auch mit privaten Subnetzen. Die Amazon-EC2-Instance wird im öffentlichen Subnetz gehostet, damit sie im öffentlichen Internet erreicht werden kann. Die DB-Instance wird in einem privaten Subnetz gehostet. Die Amazon-EC2-Instance kann eine Verbindung mit der DB-Instance herstellen, da sie innerhalb derselben VPC gehostet wird. Die DB-Instance ist jedoch nicht für das öffentliche Internet verfügbar und bietet so mehr Sicherheit.

In diesem Tutorial wird ein zusätzliches öffentliches und ein privates Subnetz in einer separaten Availability Zone konfiguriert. Diese Subnetze werden im Tutorial nicht verwendet. Eine RDS DB-

Subnetzgruppe erfordert ein Subnetz in mindestens zwei Availability Zones. Das zusätzliche Subnetz erleichtert den späteren Wechsel zu einer Multi-AZ-Bereitstellung der DB-Instance.

Zum Erstellen einer DB-Instance, die den Dual-Stack-Modus verwendet, geben Sie Dual-stack mode (Dual-Stack-Modus) für die Einstellung Network type (Netzwerktyp) ein. Sie können eine DB-Instance mit der gleichen Einstellung auch ändern. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance und Ändern einer Amazon RDS DB-Instance.

In diesem Tutorial wird das Konfigurieren einer VPC für Amazon RDS-DB-Instances beschrieben. Weitere Informationen zur Amazon VPC-Sicherheit finden Sie im Amazon VPC-Benutzerhandbuch.

Erstellen einer VPC mit privaten und öffentlichen Subnetzen

Verwenden sie die folgenden Vorgänge, um eine VPC sowohl mit öffentlichen als auch mit privaten Subnetzen zu erstellen.

So erstellen Sie eine VPC und Subnetze

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie in der oberen rechten Ecke von die Region aus AWS Management Console, in der Sie Ihre VPC erstellen möchten. In diesem Beispiel wird die Region USA Ost (Ohio) verwendet.
3. Wählen Sie links oben die Option VPC-Dashboard aus. Wählen Sie Create VPC (VPC erstellen) aus, um mit dem Erstellen einer VPC zu beginnen.
4. Wählen Sie unter VPC Settings (VPC-Einstellungen) für Resources to create (Zu erstellende Ressourcen) VPC and more (VPC und mehr) aus.
5. Legen Sie für die übrigen VPC settings (VPC-Einstellungen) folgende Werte fest:
   - Name tag auto-generation (Namens-Tag automatisch erstellen) – **`tutorial-dual-stack`**
   - IPv4 CIDR block (IPv4-CIDR-Block) – **`10.0.0.0/16`**
   - IPv6 CIDR block (IPv6-CIDR-Block) – Amazon-provided IPv6 CIDR block (Von Amazon bereitgestellter IPv6-CIDR-Block)
   - Tenancy – Default (Standard)
   - Number of Availability Zones (AZs) (Anzahl der Availability Zones (AZs) – 2
   - Customize AZs (AZs anpassen) – Übernehmen Sie die Standardwerte.
   - Number of public subnet (Anzahl der öffentlichen Subnetze) – 2
   - Number of private subnets (Anzahl der privaten Subnetze) – 2

- Customize subnets CIDR blocks (CIDR-Blöcke für Subnetze anpassen) – Übernehmen Sie die Standardwerte.
- NAT gateways ($) (NAT-Gateways ($)) – None (Keine)
- Egress only internet gateway (Internet-Gateway nur für ausgehenden Datenverkehr) – No (Nein)
- VPC endpoints (VPC-Endpunkte) – None (Keine)
- DNS options (DNS-Optionen) – Übernehmen Sie die Standardwerte.

> **Note**
>
> Amazon RDS erfordert mindestens zwei Subnetze in zwei verschiedenen Availability Zones, um Multi-AZ-Bereitstellungen von DB-Instances zu unterstützen. In diesem Tutorial wird eine Single-AZ-Bereitstellung erstellt, aber die Anforderung erleichtert die spätere Konvertierung in eine Multi-AZ-Bereitstellung von DB-Instances.

6. Wählen Sie VPC erstellen aus.

Erstellen einer VPC-Sicherheitsgruppe für eine öffentliche Amazon-EC2-Instance

Als Nächstes erstellen Sie eine Sicherheitsgruppe für öffentlichen Zugriff. Wenn Sie eine Verbindung mit öffentlichen EC2-Instances in Ihrer VPC herstellen möchten, fügen Sie Ihrer VPC-Sicherheitsgruppe eingehende Regeln hinzu, die Verbindungen aus dem Internet zulassen.

So erstellen Sie eine VPC-Sicherheitsgruppe

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie VPC Dashboard (VPC-Dashboard), Security Groups (Sicherheitsgruppen) und anschließend Create security group (Sicherheitsgruppe erstellen).
3. Legen Sie auf der Seite Create security group (Sicherheitsgruppe erstellen) die folgenden Werte fest:

   - Security group name (Name der Sicherheitsgruppe: **`tutorial-dual-stack-securitygroup`**
   - Description (Beschreibung: **`Tutorial Dual-Stack Security Group`**

- VPC: Wählen Sie die VPC aus, die Sie zuvor erstellt haben, zum Beispiel: vpc-***identifier*** (tutorial-dual-stack-vpc)

4. Fügen Sie der Sicherheitsgruppe Regeln für den eingehenden Datenverkehr hinzu.

   a. Legen Sie die IP-Adresse fest, die für die Verbindung mit EC2-Instances in Ihrer VPC mit Secure Shell (SSH) verwendet werden soll.

      Ein Beispiel für eine Internet Protocol Version 4 (IPv4)-Adresse ist `203.0.113.25/32`. Ein Beispiel für einen Internet Protocol Version 6 (IPv6)-Adressbereich ist `2001:db8:1234:1a00::/64`.

      In vielen Fällen können Sie eine Verbindung über einen Internetdienstanbieter (ISP) oder hinter Ihrer Firewall ohne statische IP-Adresse herstellen. Suchen Sie in diesem Fall den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

      > ⚠ Warning
      >
      > Wenn Sie `0.0.0.0/0` für IPv4 oder `::0` für IPv6 verwenden, lassen Sie für alle IP-Adressen den Zugriff auf Ihre öffentlichen Instances mit SSH zu. Dieser Ansatz ist zwar für kurze Zeit in einer Testumgebung zulässig, aber für Produktionsumgebungen sehr unsicher. Autorisieren Sie in Produktionsumgebungen nur eine bestimmte IP-Adresse bzw. einen bestimmten Adressbereich für den Zugriff auf Ihre Instances.

   b. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.

   c. Legen Sie die folgenden Werte für Ihre neue eingehende Regel fest, um Secure Shell (SSH) den Zugriff auf Ihre Amazon-EC2-Instance zu erlauben. In diesem Fall können Sie eine Verbindung mit Ihrer EC2-Instance herstellen, um SQL-Clients und andere Anwendungen zu installieren. Geben Sie eine IP-Adresse an, damit Sie auf Ihre EC2-Instance zugreifen können:

      - Typ: **`SSH`**
      - Source (Quelle): Die IP-Adresse bzw. der IP-Bereich aus Schritt a. Ein Beispiel für eine IPv4-IP-Adresse ist **`203.0.113.25/32`**. Ein Beispiel für eine IPv6-IP-Adresse ist **`2001:DB8::/32`**.

5. Wählen Sie Create security group (Sicherheitsgruppe erstellen) aus, um die Sicherheitsgruppe zu erstellen.

Notieren Sie sich die Sicherheitsgruppen-ID, da Sie sie später in diesem Tutorial benötigen.

Erstellen einer VPC-Sicherheitsgruppe für eine private DB-Instance

Erstellen Sie eine zweite Sicherheitsgruppe für privaten Zugriff, um Ihre DB-Instance privat zu halten. Um eine Verbindung mit privaten DB-Instances in Ihrer VPC herzustellen, fügen Sie Regeln für eingehenden Datenverkehr zu Ihrer VPC-Sicherheitsgruppe hinzu. Diese lassen ausschließlich Datenverkehr von Ihrer Amazon-EC2-Instance zu.

So erstellen Sie eine VPC-Sicherheitsgruppe

1. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie VPC Dashboard (VPC-Dashboard), Security Groups (Sicherheitsgruppen) und anschließend Create security group (Sicherheitsgruppe erstellen).
3. Legen Sie auf der Seite Create security group (Sicherheitsgruppe erstellen) die folgenden Werte fest:

   - Security group name (Name der Sicherheitsgruppe: **`tutorial-dual-stack-db-securitygroup`**
   - Description (Beschreibung: **`Tutorial Dual-Stack DB Instance Security Group`**
   - VPC: Wählen Sie die VPC aus, die Sie zuvor erstellt haben, zum Beispiel: vpc-***`identifier`*** (tutorial-dual-stack-vpc)
4. Fügen Sie der Sicherheitsgruppe Regeln für den eingehenden Datenverkehr hinzu:

   a. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.

   b. Legen Sie die folgenden Werte für Ihre neue eingehende Regel fest, um MySQL-Datenverkehr von Ihrer Amazon-EC2-Instance an Port 3306 zuzulassen. In diesem Fall können Sie von Ihrer EC2-Instance eine Verbindung mit Ihrer DB-Instance herstellen. Dies bedeutet, dass Sie Daten aus Ihrer EC2-Instance an Ihre Datenbank senden können.

      - Type (Typ) MySQL/Aurora
      - Source (Quelle): Die Kennung der Sicherheitsgruppe tutorial-dual-stack-securitygroup, die Sie zuvor in diesem Tutorial erstellt haben, z. B. sg-9edd5cfb.
5. Um die Sicherheitsgruppe zu erstellen, wählen Sie Sicherheitsgruppe erstellen aus.

Erstellen einer DB-Subnetzgruppe

Eine DB-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie in einer VPC erstellen und anschließend den DB-Instances zuweisen. Mit einer DB-Subnetzgruppe können Sie beim Erstellen von DB-Instances eine bestimmte VPC angeben. Wenn Sie eine DB-Sicherheitsgruppe erstellen möchten, die mit DUAL kompatibel ist, müssen alle Subnetze mit DUAL kompatibel sein. Um mit DUAL kompatibel zu sein, muss einem Subnetz ein IPv6 CIDR zugeordnet sein.

Erstellen einer DB-Sicherheitsgruppe

1. Identifizieren Sie die privaten Subnetze für Ihre Datenbank in der VPC.

   a. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

   b. Wählen Sie VPC Dashboard (VPC-Dashboard) und dann Subnets (Subnetze) aus.

   c. Beachten Sie die Subnetz-IDs der Subnetze mit den Namen tutorial-dual-stack-subnet-private1-us-west-2a und tutorial-dual-stack-subnet-private2-us-west-2b.

      Sie benötigen die Subnetz-IDs, wenn Sie Ihre DB-Subnetzgruppe erstellen.

2. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.

   Stellen Sie sicher, dass Sie eine Verbindung mit der Amazon-RDS-Konsole herstellen, nicht mit der Amazon-VPC-Konsole.

3. Wählen Sie im Navigationsbereich Subnetzgruppe aus.

4. Wählen Sie Create DB subnet group (DB-Subnetzgruppe erstellen) aus.

5. Legen Sie auf der Seite DB-Subnetzgruppe erstellen unter Subnetzgruppendetails die folgenden Werte fest:

   - Name (Name: **`tutorial-dual-stack-db-subnet-group`**
   - Description (Beschreibung: **`Tutorial Dual-Stack DB Subnet Group`**
   - VPC: tutorial-dual-stack-vpc ***`(vpc-Kennung)`***

6. Wählen Sie im Abschnitt Add subnets (Subnetze hinzufügen) Werte für die Availability Zones und Subnets (Subnetze) aus.

   Wählen Sie für dieses Tutorial us-east-2a und us-east-2b als Availability Zones aus. Wählen Sie für Subnets (Subnetze) die privaten Subnetze aus, die Sie im vorherigen Schritt identifiziert haben.

7. Wählen Sie Create (Erstellen) aus.

Ihre neue DB-Subnetzgruppe wird in der Liste der DB-Subnetzgruppen in der RDS-Konsole angezeigt. Sie können die DB-Subnetzgruppe auswählen, um ihre Details anzuzeigen. Dazu gehören die unterstützten Adressierungsprotokolle und alle Subnetze, die mit der Gruppe verknüpft sind, sowie der von der DB-Subnetzgruppe unterstützte Netzwerktyp.

Erstellen einer Amazon-EC2-Instance im Dual-Stack-Modus

Um eine Amazon EC2 EC2-Instance zu erstellen, folgen Sie den Anweisungen unter Starten einer Instance mithilfe des Assistenten zum Starten einer Instance im Amazon EC2 EC2-Benutzerhandbuch.

Legen Sie auf der Seite Configure Instance Details (Konfigurieren von Instance-Detail)s wie im Folgenden gezeigt die folgenden Werte fest und behalten Sie die Standardwerte für die anderen Werte bei:

- Network (Netzwerk) – Wählen Sie eine vorhandene VPC mit öffentlichen und privaten Subnetzen aus, z. B. tutorial-dual-stack-vpc (vpc-*identifier*), die in Erstellen einer VPC mit privaten und öffentlichen Subnetzen erstellt wurde.
- Subnetz — Wählen Sie ein vorhandenes öffentliches Subnetz aus, z. B. subnet- ***identifier*** | tutorial-dual-stack-subnet -public1-us-east-2a | us-east-2a, erstellt in. Erstellen einer VPC-Sicherheitsgruppe für eine öffentliche Amazon-EC2-Instance
- Auto-assign Public IP (Öffentliche IP-Adresse automatisch zuweisen) – Wählen Sie Enable (Aktivieren) aus.
- Auto-assign IPv6 IP (IPv6-IP automatisch zuweisen) – Wählen Sie Enable (Aktivieren) aus.
- Firewall (security groups) (Firewall (Sicherheitsgruppen)) – Wählen Sie Select an existing security group (Eine vorhandene Sicherheitsgruppe auswählen) aus.
- Common security groups (Allgemeine Sicherheitsgruppen) – Wählen Sie eine vorhandene Sicherheitsgruppe aus, z. B. die Gruppe `tutorial-securitygroup`, die Sie in Erstellen einer VPC-Sicherheitsgruppe für eine öffentliche Amazon-EC2-Instance erstellt haben. Stellen Sie sicher, dass die von Ihnen gewählte Sicherheitsgruppe Regeln für eingehenden Datenverkehr für Secure Shell (SSH) und HTTP-Zugriff enthält.

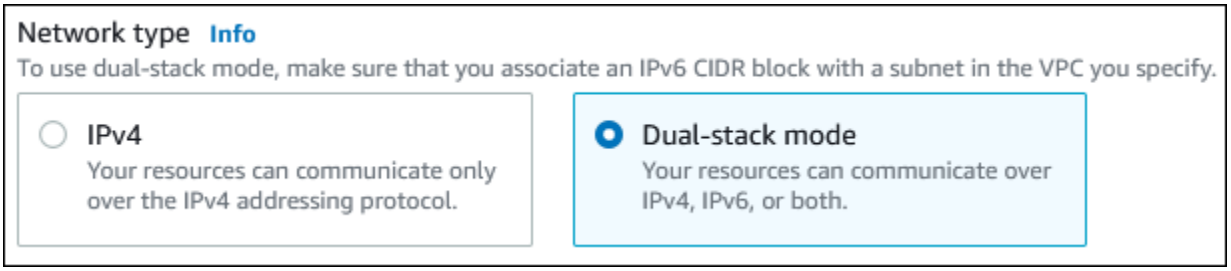
Erstellen einer DB-Instance im Dual-Stack-Modus

In diesem Schritt erstellen Sie eine DB-Instance zum Ausführen im Dual-Stack-Modus.

So erstellen Sie eine DB-Instance

1. Melden Sie sich bei der Amazon RDS-Konsole an AWS Management Console und öffnen Sie sie unter https://console.aws.amazon.com/rds/.
2. In diesem Beispiel wird die Region USA Ost (Ohio) verwendet.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Create database (Datenbank erstellen) aus.
5. Stellen Sie auf der Seite Create database (Datenbank erstellen), die nachfolgend dargestellt ist, sicher, dass die Option Standard create (Standarderstellung) ausgewählt ist, und wählen Sie dann den DB-Engine-Typ MySQL aus.
6. Legen Sie im Abschnitt Connectivity folgende Werte fest:

   - Network type (Netzwerktyp) – Wählen Sie Dual-stack mode (Dual-Stack-Modus) aus.



   - Virtual Private Cloud (VPC) Wählen Sie eine vorhandene VPC mit öffentlichen und privaten Subnetzen aus, z. B. tutorial-dual-stack-vpc (vpc-*identifier*), die in Erstellen einer VPC mit privaten und öffentlichen Subnetzen erstellt wurde.

     Die VPC muss Subnetze in verschiedenen Availability Zones haben.
   - DB Subnet group (DB-Subnetzgruppe) – Wählen Sie eine DB-Subnetzgruppe für die VPC aus, z. B. tutorial-dual-stack-db-subnet-group, die in Erstellen einer DB-Subnetzgruppe erstellt wurde.
   - Public access (Öffentlicher Zugriff) – Wählen Sie No (Nein) aus.
   - VPC security group (firewall) (VPC-Sicherheitsgruppe (Firewall)) – Wählen Sie Choose existing (Vorhandene auswählen) aus.
   - Existing VPC security groups (Vorhandene VPC-Sicherheitsgruppen) – Wählen Sie eine vorhandene VPC-Sicherheitsgruppe aus, die für den privaten Zugriff konfiguriert ist, z. B. tutorial-dual-stack-db-securitygroup, die in Erstellen einer VPC-Sicherheitsgruppe für eine private DB-Instance erstellt wurde.

Entfernen Sie andere Sicherheitsgruppen wie die Standardsicherheitsgruppe, indem Sie das jeweilige X wählen.

- Availability Zone – Wählen Sie us-west-2a aus.

  Um AZ-übergreifenden Datenverkehr zu vermeiden, stellen Sie sicher, dass sich die DB-Instance und die EC2-Instance in derselben Availability Zone befinden.

7. Geben Sie für die restlichen Abschnitte die gewünschten Einstellungen für die DB-Instance an. Weitere Informationen zu den einzelnen Einstellungen finden Sie unter Einstellungen für DB-Instances.

Herstellen einer Verbindung mit Ihrer Amazon-EC2-Instance und der DB-Instance

Nachdem Ihre Amazon-EC2-Instance und die DB-Instance im Dual-Stack-Modus erstellt wurden, können Sie sich über das IPv6-Protokoll mit jeder einzelnen Instance verbinden. Um mithilfe des IPv6-Protokolls eine Verbindung zu einer Amazon EC2 EC2-Instance herzustellen, folgen Sie den Anweisungen unter Connect to your Linux Instance im Amazon EC2 EC2-Benutzerhandbuch.

Um von der Amazon-EC2-Instance aus eine Verbindung zu Ihrer RDS-for-MySQL-DB-Instance herzustellen, folgen Sie den Anweisungen unter Verbindung zu einer MySQL-DB-Instance herstellen.

Löschen der VPC

Nachdem Sie die VPC und andere Ressourcen für dieses Tutorial erstellt haben, können Sie sie löschen, wenn sie nicht mehr benötigt werden.

Wenn Sie in der VPC, die Sie für dieses Tutorial erstellt haben, Ressourcen hinzugefügt haben, müssen Sie diese möglicherweise löschen, bevor Sie die VPC löschen können. Beispiele für Ressourcen sind Amazon-EC2-Instances oder DB-Instances. Weitere Informationen finden Sie unter Löschen Ihrer VPC im Amazon VPC User Guide.

So löschen Sie eine VPC und zugehörige Ressourcen

1. Löschen Sie die DB-Subnetzgruppe:

   a. Öffnen Sie die Amazon-RDS-Konsole unter https://console.aws.amazon.com/rds/.
   b. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
   c. Wählen Sie die DB-Subnetzgruppe aus, die Sie löschen möchten, z. B. tutorial-db-subnet-group.

d. Wählen Sie Löschen, und wählen Sie dann im Bestätigungsfenster Löschen.

2. Notieren Sie sich die VPC-ID.

   a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

   b. Wählen Sie VPC Dashboard und dann VPCs.

   c. Suchen Sie die von Ihnen erstellte VPC in der Liste, z. B. tutorial-dual-stack-vpc.

   d. Notieren Sie den Wert VPC ID (VPC-ID) der VPC, die Sie erstellt haben. Sie benötigen diese VPC-ID in den nachfolgenden Schritten.

3. Löschen der Sicherheitsgruppen:

   a. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

   b. Wählen Sie VPC Dashboard und dann Sicherheitsgruppen.

   c. Wählen Sie die Sicherheitsgruppe für die Amazon-RDS-DB-Instance aus, z. B. tutorial-dual-stack-db-securitygroup.

   d. Wählen Sie unter Actions (Aktionen) Delete security groups (Sicherheitsgruppen löschen) und dann Delete (Löschen) auf der Bestätigungsseite aus.

   e. Wählen Sie auf der Seite Security Groups (Sicherheitsgruppen) die Sicherheitsgruppe für die Amazon-EC2-Instance aus, z. B. tutorial-dual-stack-securitygroup.

   f. Wählen Sie unter Actions (Aktionen) Delete security groups (Sicherheitsgruppen löschen) und dann Delete (Löschen) auf der Bestätigungsseite aus.

4. Löschen des NAT-Gateways:

   a. Öffnen Sie die Amazon VPC-Konsole unter https://console.aws.amazon.com/vpc/.

   b. Wählen Sie VPC Dashboard und dann NAT Gateways.

   c. Wählen Sie das NAT-Gateway der VPC aus, die Sie erstellt haben. Verwenden Sie die VPC-ID, um das richtige NAT-Gateway zu identifizieren.

   d. Wählen Sie unter Actions (Aktionen) die Option Delete NAT gateway (NAT-Gateway löschen) aus.

   e. Geben Sie auf der Bestätigungsseite **`delete`** ein, und wählen Sie die Option Delete (Löschen) aus.

5. Löschen der VPC:

   a. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.

   b. Wählen Sie VPC Dashboard und dann VPCs.

c. Wählen Sie die VPC aus, die Sie löschen möchten, z. B.tutorial-dual-stack-vpc.

d. Wählen Sie unter Actions (Aktionen) die Option Delete VPC (VPC löschen) aus.

   Auf der Bestätigungsseite werden weitere Ressourcen angezeigt, die der VPC zugeordnet sind, die ebenfalls gelöscht werden, einschließlich der damit verknüpften Subnetze.

e. Geben Sie auf der Bestätigungsseite **`delete`** ein, und wählen Sie die Option Delete (Löschen) aus.

6. Freigeben der Elastic-IP-Adressen:

   a. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Klicken Sie aufEC2-DashboardKlicken Sie auf und danach aufElastische IP-Adressenaus.

   c. Wählen Sie die Elastic-IP-Adresse aus, die Sie freigeben möchten.

   d. Wählen Sie unter Actions (Aktionen) die Option Release Elastic IP addresses (Elastic-IP-Adressenfreigeben) aus.

   e. Wählen Sie auf der Bestätigungsseite Freigeben.

Verschieben einer DB-Instance von außerhalb einer VPC in eine VPC

Einige ältere DB-Instances auf der EC2-Classic-Plattform sind nicht in einer VPC. Falls sich Ihre DB-Instance nicht in einer VPC befindet, können Sie sie einfach mithilfe der AWS Management Console in eine VPC verschieben. Sie müssen die VPC erst erstellen, bevor Sie eine DB-Instance von außerhalb in diese VPC verschieben können.

EC2-Classic wurde am 15. August 2022 außer Betrieb genommen. Wir empfehlen Ihnen, so bald wie möglich zu migrieren, falls noch nicht von EC2-Classic zu einer VPC migriert. Weitere Informationen finden Sie unter Migration von EC2-Classic zu einer VPC im Benutzerhandbuch für Amazon EC2 und auf dem Blog EC2-Classic Networking geht in den Ruhestand – So bereiten Sie sich vor.

Important

Wenn Sie neu bei Amazon RDS sind, wenn Sie noch nie zuvor eine DB-Instance erstellt haben oder wenn Sie eine DB-Instance in einer AWS-Region erstellen, in der Sie diese noch nie zuvor verwendet haben, nutzen Sie höchstwahrscheinlich die Plattform EC2-VPC sowie eine Standard-VPC. Für Informationen über die Arbeit mit DB-Instanzen in einer VPC siehe Arbeiten mit einer DB-Instance in einer VPC.

Gehen Sie wie folgt vor, um eine VPC für Ihre DB-Instance zu erstellen.

- Schritt 1: Erstellen einer VPC
- Schritt 2: Erstellen einer DB-Subnetzgruppe
- Schritt 3: Erstellen einer VPC-Sicherheitsgruppe

Nachdem Sie die VPC erstellt haben, führen Sie die folgenden Schritte aus, um die DB-Instance in die VPC zu verschieben.

- Aktualisierung der VPC für eine DB-Instance

Es wird dringend empfohlen, dass Sie unmittelbar vor der Migration ein Backup Ihrer DB-Instance erstellen. Dadurch wird sichergestellt, dass Sie die Daten bei Fehlschlagen der Migration

wiederherstellen können. Weitere Informationen finden Sie unter Daten sichern, wiederherstellen und exportieren.

Im Folgenden werden einige Einschränkungen beim Verschieben der DB-Instance in die VPC beschrieben.

- DB-Instance-Klassen der vorherigen Generation – DB-Instance-Klassen der vorherigen Generation werden auf der VPC-Plattform möglicherweise nicht unterstützt. Wenn Sie eine DB-Instance in eine VPC verschieben, wählen Sie eine DB-Instance-Klasse db.m3 oder db.r3. Nachdem Sie die DB-Instance in eine VPC verschoben haben, können Sie die DB-Instance skalieren, um eine höhere DB-Instance-Klasse zu verwenden. Eine vollständige Liste der von VPC unterstützten Instance-Klassen finden Sie unter Amazon RDS-Instance-Typen.
- Multi-AZ – Derzeit ist es nicht möglich, eine Multi-AZ-DB-Instance, die sich außerhalb einer VPC befindet, in eine VPC zu verschieben. Um Ihre DB-Instance in eine VPC zu verschieben, modifizieren Sie zunächst die DB-Instance so, dass es sich um eine Single-AZ-Bereitstellung handelt. Ändern Sie die Einstellung Multi-AZ-Bereitstellung auf Keine. Nachdem Sie die DB-Instance in eine VPC verschoben haben, ändern Sie sie erneut, um sie zu einer Multi-AZ-Bereitstellung zu machen. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.
- Read Replicas – Derzeit ist es nicht möglich, eine DB-Instance mit Read Replicas, die sich außerhalb einer VPC befinden, in eine VPC zu verschieben. Um Ihre DB-Instance in eine VPC zu verschieben, löschen Sie zunächst alle Read Replicas. Nachdem Sie die DB-Instance in eine VPC verschoben haben, erstellen Sie die Read Replicas neu. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.
- Optionsgruppen – Wenn Sie Ihre DB-Instance in eine VPC verschieben und die DB-Instance eine benutzerdefinierte Optionsgruppe verwendet, ändern Sie die Optionsgruppe, die mit Ihrer DB-Instance verknüpft ist. Optionsgruppen sind plattformspezifisch und das Verschieben in eine VPC bedeutet eine Änderung der Plattform. Um hier eine benutzerdefinierte Optionsgruppe zu nutzen, weisen Sie der DB-Instance die Optionsgruppe der Standard-VPC zu oder weisen Sie eine Optionsgruppe zu, die von anderen DB-Instances in der aktuellen VPC verwendet wird. Alternativ erstellen Sie eine neue Optionsgruppe und weisen Sie diese der DB-Instance zu. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen.

Alternativen zum Verschieben einer DB-Instance, die sich nicht in einer VPC befindet, in eine VPC, mit minimaler Ausfallzeit

Mit den folgenden Alternativen können Sie eine DB-Instance, die sich nicht in einer VPC befindet, mit minimaler Ausfallzeit in eine VPC verschieben. Diese Alternativen führen zu minimalen Unterbrechungen der Quell-DB-Instance und ermöglichen es ihr, den Benutzerdatenverkehr während der Migration bereitzustellen. Die Zeit, die für die Migration in eine VPC benötigt wird, hängt jedoch von der Datenbankgröße und den Live-Workload-Eigenschaften ab.

- AWS Database Migration Service (AWS DMS) – AWS DMS ermöglicht die Live-Migration von Daten, während die Quell-DB-Instance voll funktionsfähig bleibt, repliziert jedoch nur einen begrenzten Satz von DDL-Anweisungen repliziert. AWS DMS verbreitet keine Elemente wie Indizes, Benutzer, Berechtigungen, gespeicherte Prozeduren und andere Datenbankänderungen, die sich nicht direkt auf die Tabellendaten beziehen. Darüber hinaus verwendet AWS DMS nicht automatisch RDS-Snapshots für die erste Erstellung der DB-Instance, was die Migrationszeit verlängern kann. Weitere Informationen finden Sie unter AWS Database Migration Service.
- DB-Snapshot-Wiederherstellung oder Zeitpunktbezogene Wiederherstellung – Sie können eine DB-Instance in eine VPC verschieben, indem Sie einen Snapshot der DB-Instance oder eine DB-Instance auf einen bestimmten Zeitpunkt wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance und Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Kontingente und Beschränkungen für Amazon RDS

Im Folgenden finden Sie eine Beschreibung der Ressourcenkontingente und Benennungseinschränkungen für Amazon RDS.

Themen

- Kontingente in Amazon RDS
- Benennungseinschränkungen in Amazon RDS
- Maximale Anzahl von Datenbankverbindungen
- Limits für Dateigrößen in Amazon RDS

Kontingente in Amazon RDS

Jedes AWS Konto hat für jede AWS Region Kontingente für die Anzahl der Amazon RDS Amazon , die erstellt werden können. Nachdem das Kontingent für eine Ressource erreicht wurde, schlagen zusätzliche Aufrufe zum Erstellen dieser Ressource mit einer Ausnahme fehl.

In der folgenden Tabelle sind die Ressourcen und ihre Kontingente pro AWS Region aufgeführt.

Name	Standard	Anpas	Beschreibung
Autorisierungen pro DB-Sicherheitsgrup pe	Jede unterstützte Region: 20	Nein	Anzahl der Sicherhei tsgruppenberechtigungen pro DB-Sicherheitsgruppe
Benutzerdefinierte Endpunkte pro DB-Cluster	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von benutzerdefinierte n Endpunkten, die Sie pro Aurora-DB-Cluster in diesem Konto in der aktuellen Region erstellen können. Dieser Wert spiegelt die höchste Anzahl von benutzerd efinierten Endpunkte n in einem DB-Cluster

Name	Standard	Anpas	Beschreibung
			im Konto wider. Andere DB-Cluster im Konto haben möglicherweise eine geringere Anzahl von benutzerdefinierten Endpunkten.
Kundenspezifische Motorversionen	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von benutzerdefinierte n Engine-Versionen, die in diesem Konto in der aktuellen Region zulässig sind
DB-Cluster-Parametergruppen	Jede unterstützte Region: 50	Ja	Die maximale Anzahl von DB-Cluster-Paramet ergruppen
DB-Cluster	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Aurora-Clustern in diesem Konto in der aktuellen Region
DB-Instances	ap-south-1:20 Jede der anderen unterstützten Regionen: 40	Ja	Die maximale Anzahl von DB-Instances, die in diesem Konto in der aktuellen Region zulässig ist
DB-Shard-Gruppen	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von DB-Shard-Gruppen für Aurora Limitless Database in diesem Konto in der aktuellen Region

Name	Standard	Anpas	Beschreibung
DB-Subnetzgruppen	ap-south-1:20 Jede der anderen unterstützten Regionen: 50	Ja	Die maximale Anzahl von DB-Subnetzgruppen
Größe des HTTP-Anforderungstexts der Daten-API	Jede unterstützte Region: 4 Megabyte	Nein	Die maximal zulässige Größe für den HTTP-Anforderungstext.
Maximale Anzahl gleichzeitiger Cluster-Geheimnis-Paare der Daten-API	Jede unterstützte Region: 30	Nein	Die maximale Anzahl eindeutiger Paare von Aurora Serverless v1-DB-Clustern und Geheimnissen in gleichzeitigen Daten-API-Anfragen für dieses Konto in der aktuellen AWS Region.
Maximale Anzahl gleichzeitiger Daten-API-Anforderungen	Jede unterstützte Region: 500	Nein	Die maximale Anzahl von Daten-API-Anfragen an einen Aurora Serverless v1-DB-Cluster, die dasselbe Geheimnis verwenden und gleichzeitig verarbeitet werden können. Zusätzliche Anforderungen werden in die Warteschlange gestellt und verarbeitet, sobald in Bearbeitung befindliche Anforderungen abgeschlossen sind.

Name	Standard	Anpas	Beschreibung
Maximale Ergebnissatzgröße der Daten-API	Jede unterstützte Region: 1 Megabyte	Nein	Die maximale Größe der Datenbank-Ergebnismenge, die von der Daten-API zurückgegeben werden kann.
Maximale Daten-API-Größe der JSON-Antwortzeichenfolge	Jede unterstützte Region: 10 Megabyte	Nein	Die maximale Größe der vereinfachten JSON-Antwortzeichenfolge, die von der RDS-Daten-API zurückgegeben wird.
Daten-API-Anforderungen pro Sekunde	Jede unterstützte Region: 1000 pro Sekunde	Nein	Die maximale Anzahl von Anfragen an die Daten-API pro Sekunde, die für dieses Konto in der aktuellen AWS Region zulässig ist. Dieses Kontingent gilt nur für Amazon Aurora Serverless v1-Cluster.
Ereignisabonnements	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von Ereignisabonnements
IAM-Rollen pro DB-Cluster	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von IAM-Rollen, die mit einem DB-Cluster verknüpft werden können
IAM-Rollen pro DB-Instance	Jede unterstützte Region: 5	Yes (Ja)	Die maximale Anzahl von IAM-Rollen, die mit einer DB-Instance verknüpft werden können

Name	Standard	Anpas	Beschreibung
Integrationen	Jede unterstützte Region: 100	Nein	Die maximale Anzahl von Integrationen, die in diesem Konto in der aktuellen Region zulässig sind AWS
Manuelle DB-Cluster-Snapshots	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl manueller DB-Cluster-Snapshots
Manuelle DB-Instance-Snapshots	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl manueller DB-Instance-Snapshots
Optionsgruppen	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von Optionsgruppen
Parametergruppen	ap-south-1:20 Jede der anderen unterstützten Regionen: 50	Ja	Die maximale Anzahl von Parametergruppen
Proxys	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von Proxys, die für dieses Konto in der aktuellen Region zulässig sind AWS
Read Replicas pro Primary	Jede unterstützte Region: 15	Ja	Die maximale Anzahl von Lesereplikaten pro primäre DB-Instance. Dieses Kontingent kann für Amazon Aurora nicht angepasst werden.

Name	Standard	Anpas	Beschreibung
Reservierte DB-Instances	Jede unterstützte Region: 40	Ja	Die maximale Anzahl reservierter DB-Instances, die für dieses Konto in der aktuellen Region zulässig sind AWS
Sicherheitsgruppen	ap-south-1:20 Jede der anderen unterstützten Regionen: 25	Ja	Die maximale Anzahl von DB-Sicherheitsgruppen
Subnetze pro DB-Subnetzgruppe	Jede unterstützte Region: 20	Nein	Die maximale Anzahl von Subnetzen pro DB-Subnetzgruppe
Gesamtspeicher für alle DB-Instances	Jede unterstützte Region: 100 000 Gigabyte	Ja	Der maximale Gesamtspeicher (in GB) von EBS-Volumes für alle Amazon-RDS-DB-Instances zusammen. Dieses Kontingent gilt nicht für Amazon Aurora, das ein maximales Cluster-Volumen von 128 TiB für jeden DB-Cluster hat.

> **Note**
>
> Standardmäßig ist das Verwenden von bis zu insgesamt 40 DB-Instances möglich. RDS-DB-Instances, Aurora DB-Instances, Amazon Neptune-Instances und Amazon DocumentDB-Instances werden zu diesem Kontingent hinzugerechnet.
> Die folgenden Einschränkungen gelten für die Amazon RDS-DB-Instances:

- 10 für jede SQL Server-Edition (Enterprise, Standard, Web und Express) im Rahmen des Modells "License-included".
- 10 für Oracle im Rahmen des Modells "license-included"
- 40 für Db2 im Rahmen des Lizenzmodells "bring-your-own-license" (BYOL)
- 40 für MySQL, MariaDB oder PostgreSQL.
- 40 für Oracle im Rahmen des Lizenzmodells "bring-your-own-license" (BYOL)

Wenn Ihre Anwendung mehr DB-Instances benötigt, können Sie zusätzliche DB-Instances anfordern, indem Sie die Konsole für Service Quotas öffnen. Wählen Sie im Navigationsbereich AWS -Services aus. Wählen Sie Amazon Relational Database Service (Amazon RDS) sowie ein Kontingent aus und folgen Sie den Anweisungen, um eine Kontingenterhöhung anzufordern. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service Quotas-Benutzerhandbuch.
Für RDS for Oracle können Sie bis zu 15 Read Replicas pro Quell-DB-Instance in jeder Region erstellen. Wir empfehlen jedoch, Repliken auf 5 zu beschränken, um die Replikationsverzögerung zu minimieren.
Von verwaltete Backups AWS Backup gelten als manuelle , werden aber nicht auf das Kontingent für manuelle angerechnet. Informationen dazu AWS Backup finden Sie im AWS Backup Entwicklerhandbuch.

Beachten Sie, dass das Standardkontingent für regionsübergreifende automatische Backups 20 für eine AWS-Konto beträgt und dass das Standardkontingent für die Anzahl gleichzeitiger Snapshot-Kopieranforderungen 20 für jede Region für eine beträgt. AWS-Konto RDS kann dieses Kontingent nicht für Folgendes anpassen: AWS-Regionen

- Afrika (Kapstadt)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Melbourne)
- Kanada West (Calgary)
- Europa (Zürich)

Wenn Sie RDS-API-Operationen verwenden und das Standardkontingent für die Anzahl von Aufrufen pro Sekunde überschreiten, gibt die Amazon-RDS-API eine Fehlermeldung ähnlich der folgenden aus.

ClientError: Beim Aufrufen des *API_name* Vorgangs ist ein Fehler aufgetreten (ThrottlingException): Rate überschritten.

Reduzieren Sie in diesem Fall die Anzahl der Aufrufe pro Sekunde. Das Kontingent soll die meisten Anwendungsfälle abdecken. Wenn höhere Kontingente benötigt werden, können Sie eine Erhöhung des Kontingents beantragen, indem Sie eine der folgenden Optionen verwenden:

- Öffnen Sie von der Konsole aus die Service Quotas Quotas-Konsole.
- Verwenden Sie von der AWS CLI aus den request-service-quota-increase AWS CLI Befehl.

Weitere Informationen zu diesem Service finden Sie im Benutzerhandbuch für Service Quotas.

Benennungseinschränkungen in Amazon RDS

Die Namensbeschränkungen in Amazon RDS lauten wie folgt:

- der DB-Instance:
  - Sie müssen 1–63 alphanumerische Zeichen oder Bindestriche enthalten.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle DB-Instances pro AWS Konto und AWS Region eindeutig sein.
- Ursprünglicher Datenbankname:
  - Die für Datenbanknamen geltenden Einschränkungen sind bei jeder Datenbank-Engine unterschiedlich. Weitere Informationen finden Sie in den verfügbaren Einstellungen beim Erstellen der einzelnen DB-Instances.
  - SQL Server — Erstellen Sie Ihre Datenbanken, nachdem Sie Ihre DB-Instance erstellt haben.
- Master-Benutzername — Die Einschränkungen für den Master-Benutzernamen sind für jede Datenbank-Engine unterschiedlich. Weitere Informationen finden Sie in den verfügbaren Einstellungen bei der Erstellung des .
- Master-Passwort:
  - Das Passwort für den Master-Benutzer der Datenbank kann jedes druckbare ASCII-Zeichen außer `/`, `'`, `"`, `@` oder Leerzeichen enthalten.

Für Oracle ist & eine zusätzliche Zeichenbeschränkung.

  - Das Passwort kann je nach DB-Engine die folgende Anzahl druckbarer ASCII-Zeichen enthalten:
    - Db2:8—255
    - MariaDB und MySQL: 8–41
    - Oracle: 8–30
    - SQL Server und PostgreSQL: 8–128
- DB-Parametergruppe:
  - Sie müssen 1–255 alphanumerische Zeichen enthalten.
  - Muss mit einem Buchstaben beginnen.
  - Bindestriche sind erlaubt, aber der Name darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- DB-Subnetzgruppe:
  - Müssen 1–255 Zeichen enthalten.
  - Alphanumerische Zeichen, Leerzeichen, Bindestriche, Unterstriche und Punkte sind erlaubt.

Maximale Anzahl von Datenbankverbindungen

Die maximale Anzahl gleichzeitiger Datenbankverbindungen variiert je nach DB-Engine-Typ und Speicherzuweisung für die DB-Instance-Klasse. Die maximale Anzahl von Verbindungen wird im Allgemeinen in der Parametergruppe festgelegt, die der DB-Instance zugeordnet ist. Eine Ausnahme bildet Microsoft SQL Server. Hier wird sie in den Servereigenschaften für die DB-Instance in SQL Server Management Studio (SSMS) festgelegt.

Datenbankverbindungen verbrauchen Speicher. Wenn Sie einen dieser Parameter zu hoch festlegen, kann dies zu einem niedrigen Speicherzustand führen, der dazu führen kann, dass eine DB-Instance in den incompatible-parameters Status kommt. Weitere Informationen finden Sie unter Diagnostizieren und Auflösen des Status "incompatible-parameters" für ein Speicherlimit.

Wenn Ihre Anwendungen häufig Verbindungen öffnen und schließen oder langlebige Verbindungen in großer Zahl offen lassen, empfehlen wir Ihnen, Amazon-RDS-Proxy zu verwenden. RDS-Proxy ist ein vollständig verwalteter, hochverfügbarer Datenbank-Proxy, der Datenbankverbindungen sicher und effizient per Verbindungspooling freigibt. Weitere Informationen zu RDS Proxy finden Sie unter Amazon RDS-Proxy .

> **Note**
>
> Für Oracle legen Sie die maximale Anzahl von Benutzerprozessen sowie Benutzer- und Systemsitzungen fest.
> Für Db2 können Sie keine maximalen Verbindungen festlegen. Das Limit liegt bei 64000.

Die folgende Tabelle enthält Informationen zu den maximalen Datenbankverbindungen für verschiedene DB-Engines.

DB-Engine	Parameter	Zulässige Werte	Standardwert	Beschreibung
MariaDB	`max_connections`	1–100000	• Für MariaDB 10.5 und höhere Versionen ist die Standardeinstellung: AM WENIGSTEN ({DBInstanceClassMemory/25165760} ,12000) Die Formel entspricht praktisch MB/25. Wenn die Standardwertberechnung zu einem Wert über 12.000 führt, legt Amazon RDS das Limit auf 12.000 fest. • Für MariaDB Version 10.4: {/12582880} DBInstance ClassMemory	Anzahl der zulässigen gleichzeitigen Clientverbindungen

DB-Engine	Parameter	Zulässige Werte	Standardwert	Beschreibung
			Die Formel entspricht praktisch MB/12.	
MySQL	`max_connections`	1–100000	{/12582880} DBInstanceClassMemory Die Formel entspricht praktisch MB/12.	Anzahl der zulässigen gleichzeitigen Clientverbindungen
Oracle	`processes`	80–20000	AM WENIGSTEN ({DBInstanceClassMemory/9868951}, 20000)	Benutzerprozesse
Oracle	`sessions`	100–65535	Nicht zutreffend	Benutzer- und Systemsitzungen
PostgreSQL	`max_connections`	6—262143	AM WENIGSTEN ({DBInstanceClassMemory/9531392}, 5000)	Maximale Anzahl gleichzeitiger Verbindungen
SQL Server	`user connections`	0–32767	0 (unbegrenzt)	Maximale Anzahl gleichzeitiger Verbindungen. Weitere Informationen finden Sie unter Benutzerverbindungen konfigurieren (Serverkonfigurationsoption).

`DBInstanceClassMemory` wird in Byte angegeben. Weitere Informationen zur Berechnung dieses Werts finden Sie unter Festlegen von DB-Parametern. Aufgrund des Speichers, der für das Betriebssystem und die RDS-Verwaltungsprozesse reserviert ist, ist diese Speichergröße kleiner als

der Wert in Gibibyte (GiB), der in Hardware-Spezifikationen für DB-Instance-Klassen angegeben wird.

Beispielsweise haben einige DB-Instance-Klassen 8 GiB Speicher, was 8.589.934.592 Byte entspricht. Für eine MySQL-DB-Instance, die auf einer DB-Instance-Klasse mit 8 GiB Arbeitsspeicher ausgeführt wird, wie beispielsweise db.m7g.large, würde die Gleichung, die den Gesamtspeicher verwendet, `8589934592/12582880=683` lauten. Die Variable `DBInstanceClassMemory` subtrahiert jedoch automatisch die Beträge, die für das Betriebssystem und die RDS-Prozesse reserviert sind, die die Instance verwalten. Der Rest der Subtraktion wird dann durch 12 582 880 geteilt. Diese Berechnung ergibt für den Wert von `max_connections` ungefähr 630 statt 683. Dieser Wert hängt von der DB-Instance-Klasse und der DB-Engine ab.

Wenn eine MariaDB- oder MySQL-DB-Instance auf einer kleinen DB-Instance-Klasse wie db.t3.micro oder db.t3.small ausgeführt wird, ist der verfügbare Gesamtspeicher gering. Für diese DB-Instance-Klassen reserviert RDS einen erheblichen Teil des verfügbaren Speichers, was sich auf den Wert von `max_connections` auswirkt. Die standardmäßige maximale Anzahl von Verbindungen für eine MySQL-DB-Instance, die in einer DB-Instance-Klasse des Typs „db.t3.micro“ ausgeführt wird, beträgt ungefähr 60. Sie können den `max_connections`-Wert für Ihre MariaDB- oder MySQL-DB-Instance ermitteln, indem Sie eine Verbindung zu ihr herstellen und den folgenden SQL-Befehl ausführen:

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

Limits für Dateigrößen in Amazon RDS

Dateigrößenbeschränkungen gelten für bestimmte Amazon RDS DB-Instances. Weitere Informationen finden Sie in den folgenden Engine-spezifischen Beschränkungen:

- MariaDB-Dateigrößenlimits in Amazon RDS
- MySQL-Dateigrößenlimits in Amazon RDS
- Oracle-Dateigrößenbeschränkungen in Amazon RDS

Fehlerbehebung für Amazon RDS

Verwenden Sie die folgenden Abschnitte, um Probleme zu beheben, die Sie mit DB-Instances in Amazon RDS und Amazon Aurora haben.

Themen

- Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden
- Amazon RDS-Sicherheitsprobleme
- Problembehandlung bei inkompatiblem Netzwerkstatus
- Zurücksetzen des Besitzerpassworts der DB-Instance
- Ausfall oder Neustart einer Amazon RDS-DB-Instance
- Amazon RDS-DB-Parameter Änderungen wirken sich nicht aus
- Amazon RDS-DB-Instance hat zu wenig Speicher
- Amazon RDS: unzureichende DB-Instances verfügbar
- Probleme mit freisetzbarem Speicher in Amazon RDS
- Probleme mit MySQL und MariaDB
- Der Aufbewahrungszeitraum für Backups kann nicht auf 0 gesetzt werden

Informationen über das Beheben von Problemen mithilfe der Amazon RDS-API finden Sie unter Fehlerbehebung für Anwendungen in Amazon RDS.

Verbindung zur Amazon RDS-DB-Instance kann nicht hergestellt werden

Wenn Sie keine Verbindung zu einer DB-Instance herstellen können, sind die folgenden Punkte häufige Ursachen:

- Regeln für eingehenden Datenverkehr – Die von Ihrer lokalen Firewall erzwungenen Zugriffsregeln und die für den Zugriff auf Ihre DB-Instance autorisierten IP-Adressen stimmen möglicherweise nicht überein. Das Problem sind höchstwahrscheinlich die Regeln für eingehenden Datenverkehr in Ihrer Sicherheitsgruppe.

  Standardmäßig erlauben DB-Instances keinen Zugriff. Zugriff wird über eine Sicherheitsgruppe gewährt, die der VPC zugeordnet ist und Datenverkehr in die und aus der DB-Instance zulässt.

Fügen Sie der Sicherheitsgruppe bei Bedarf Regeln für eingehenden und ausgehenden Datenverkehr für Ihre besondere Situation hinzu. Sie können eine IP-Adresse, einen IP-Adressbereich oder eine andere VPC-Sicherheitsgruppe angeben.

> **Note**
>
> Wenn Sie eine neue Regel für eingehenden Datenverkehr hinzufügen, können Sie für Source (Quelle) die Option My IP (Meine IP) auswählen, um den Zugriff auf die DB-Instance von der in Ihrem Browser erkannten IP-Adresse zu ermöglichen.

Weitere Informationen zum Einrichten von Sicherheitsgruppen finden Sie unter Ermöglichen des Zugriffs auf Ihre DB-Instance in der VPC durch Erstellen einer Sicherheitsgruppe.

> **Note**
>
> Client-Verbindungen von IP-Adressen im Bereich 169.254.0.0/16 sind nicht erlaubt. Dies ist der APIPA-Bereich (Automatic Private IP Addressing), der für die Local-Link-Adressierung verwendet wird.

- Öffentliche Zugänglichkeit– Um eine Verbindung mit Ihrer DB-Instance von außerhalb der VPC herzustellen, z. B. mithilfe einer Client-Anwendung, muss der Instance eine öffentliche IP-Adresse zugewiesen sein.

  Um die Instance öffentlich zugänglich zu machen, ändern Sie sie und wählen Sie unter Public accessibility (Öffentlicher Zugriff) die Option Yes (Ja) aus. Weitere Informationen finden Sie unter Ausblenden einer DB-Instance in einer VPC vor dem Internet.
- Port — Der Port, den Sie bei der Erstellung der DB-Instance angegeben haben, kann aufgrund Ihrer lokalen Firewall-Einschränkungen nicht zum Senden oder Empfangen von Kommunikation verwendet werden. Wenden Sie sich an Ihren Netzwerkadministrator, um herauszufinden, ob Ihr Netzwerk den angegebenen Port für eingehende und ausgehende Kommunikation zulässt.
- Verfügbarkeit – Bei einer neu erstellten DB-Instance lautet ihr Status `creating`, bis die DB-Instance bereit für die Verwendung ist. Wenn sich der Status in `available` ändert, können Sie die Verbindung zur DB-Instance herstellen. Abhängig von der Größe Ihrer DB-Instance kann es bis zu 20 Minuten dauern, bevor eine Instance verfügbar ist.
- Internet-Gateway – Damit öffentlich auf eine DB-Instance zugegriffen werden kann, müssen die Subnetze in seiner DB-Subnetzgruppe über ein Internet-Gateway verfügen.

So konfigurieren Sie ein Internet-Gateway für ein Subnetz

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie im Navigationsbereich Databases (Datenbanken) und dann den Namen der DB-Instance aus.
3. Notieren Sie auf der Registerkarte Connectivity & security (Konnektivität und Sicherheit) die Werte der VPC-ID unter VPC und die Subnetz-ID unter Subnets (Subnetze).
4. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
5. Wählen Sie im Navigationsbereich Internet Gateways aus. Überprüfen Sie, ob Ihrer VPC ein Internet-Gateway angefügt ist. Falls nicht, wählen Sie Create Internet Gateway, um ein Internet-Gateway zu erstellen. Wählen Sie das Internet-Gateway aus, klicken Sie auf die Option Attach to VPC, und folgen Sie den Anleitungen, um das Gateway Ihrer VPC anzufügen.
6. Wählen Sie im Navigationsbereich die Option Subnets und dann Ihr Subnetz aus.
7. Überprüfen Sie, ob auf der Registerkarte Route Table eine Route mit `0.0.0.0/0` unter "Destination" und das Internet-Gateway für Ihre VPC unter „Target“ vorhanden ist.

   Wenn Sie über deren IPv6 Adresse eine Verbindung zu Ihrer Instance herstellen, stellen Sie sicher, dass es eine Route für den gesamten IPv6 Traffic (`::/0`) gibt, die auf das Internet-Gateway verweist. Andernfalls gehen Sie wie folgt vor:

   a. Wählen Sie die ID der Routing-Tabelle (rtb-xxxxxxxx) aus, um zur Routing-Tabelle zu gelangen.
   b. Klicken Sie auf der Registerkarte Routes (Routen) auf Edit routes (Routen bearbeiten). Wählen Sie Add route (Route hinzufügen) aus, verwenden Sie `0.0.0.0/0` als Ziel und das Internet-Gateway als Ziel.

      Wählen Sie für IPv6 Route hinzufügen, `::/0` als Ziel verwenden und das Internet-Gateway als Ziel aus.
   c. Wählen Sie Save Rules (Routen speichern) aus.

   Wenn Sie versuchen, eine Verbindung zum IPv6 Endpunkt herzustellen, stellen Sie außerdem sicher, dass der IPv6 Client-Adressbereich autorisiert ist, eine Verbindung zur DB-Instance herzustellen.

Weitere Informationen finden Sie unter Arbeiten mit einer DB-Instance in einer VPC.

Informationen zu Engine-spezifischen Verbindungsproblemen finden Sie in den folgenden Themen:

- Problembehandlung bei Verbindungen zu Ihrer SQL Server-DB-Instance
- Fehlerbehebung bei Verbindungen mit Ihrer Oracle-DB-Instance
- Fehlerbehebung bei Verbindungen mit Ihrer RDS für PostgreSQL-Instance
- Maximale Anzahl von MySQL- und MariaDB-Verbindungen

Testen der Verbindung zu einer DB-Instance

Sie können Ihre Verbindung zu einer DB-Instance mit gängigen Linux- oder Microsoft Windows-Tools testen.

Sie können die Verbindung über ein Linux- oder Unix-Terminal testen, indem Sie Folgendes eingeben. Ersetzen Sie *DB-instance-endpoint* durch den Endpunkt und *port* durch den Port Ihrer DB-Instance.

```
nc -zv DB-instance-endpoint port
```

Das folgende Beispiel zeigt einen Beispielbefehl und den Rückgabewert.

```
nc -zv postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299

  Connection to postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299 port [tcp/
vvr-data] succeeded!
```

Windows-Benutzer können Telnet verwenden, um die Verbindung zu einer DB-Instance zu testen. Telnet-Aktionen werden nur zum Testen der Verbindung unterstützt. Wenn eine Verbindung erfolgreich ist, gibt die Aktion keine Nachricht zurück. Wenn eine Verbindung nicht erfolgreich ist, erhalten Sie eine Fehlermeldung wie die folgende.

```
C:\>telnet sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com 819

  Connecting To sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com...Could not
 open
```

```
connection to the host, on port 819: Connect failed
```

Wenn Telnet-Aktionen erfolgreich sind, wurde Ihre Sicherheitsgruppe ordnungsgemäß konfiguriert.

> **Note**
>
> Internet Control Message Protocol (ICMP)-Datenverkehr, einschließlich Ping, wird von Amazon RDS nicht akzeptiert.

Fehlerbehebung bei der Verbindungsauthentifizierung

In einigen Fällen können Sie eine Verbindung mit Ihrer DB-Instance herstellen, erhalten jedoch Authentifizierungsfehler. In diesen Fällen sollten Sie das Hauptbenutzerpasswort für die DB-Instance zurücksetzen. Sie können dies tun, indem Sie die RDS-Instance ändern.

Weitere Informationen über das Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Amazon RDS-Sicherheitsprobleme

Um Sicherheitsprobleme zu vermeiden, sollten Sie niemals die E-Mail-Adresse und das Passwort Ihres AWS-Konto Root-Benutzers für ein Benutzerkonto verwenden. Es empfiehlt sich, Ihren Root-Benutzer zu verwenden, um Benutzer zu erstellen und diese DB-Benutzerkonten zuzuweisen. Sie können Ihren Root-Benutzer auch verwenden, um bei Bedarf weitere Benutzerkonten zu erstellen.

Informationen zum Erstellen von Benutzern finden Sie unter Erstellen eines IAM-Benutzers in Ihrem AWS-Konto. Informationen zum Erstellen von Benutzern in AWS IAM Identity Center finden Sie unter Identitäten verwalten im IAM Identity Center.

Fehlermeldung „Failed to retrieve account attributes, certain console functions may be impaired (Fehler beim Abrufen von Kontoattributen, bestimmte Konsolenfunktionen können beeinträchtigt sein).“

Dieser Fehler kann verschiedene Ursachen haben. Es kann daran liegen, dass Ihrem Konto Berechtigungen fehlen oder Ihr Konto nicht ordnungsgemäß eingerichtet wurde. Wenn Ihr Konto neu ist, haben Sie möglicherweise nicht abgewartet, bis das Konto bereit ist. Wenn dies ein vorhandenes

Konto ist, könnten Berechtigungen in Ihren Zugriffsrichtlinien fehlen, um bestimmte Aktionen auszuführen, wie z. B. das Erstellen einer DB-Instance. Um das Problem zu beheben, muss Ihr Administrator die erforderlichen Rollen für Ihr Konto bereitstellen. Weitere Informationen finden Sie in der IAM-Dokumentation.

Problembehandlung bei inkompatiblem Netzwerkstatus

Der Status „Inkompatibles Netzwerk“ bedeutet, dass auf Datenbankebene möglicherweise weiterhin auf die Datenbank zugegriffen werden kann, Sie sie jedoch nicht ändern oder neu starten können.

Ursachen

Der Status „Inkompatibles Netzwerk“ Ihrer DB-Instance könnte das Ergebnis einer der folgenden Aktionen sein:

- Ändern der DB-Instance-Klasse.
- Ändern der DB-Instance, um die Multi-AZ-DB-Cluster-Bereitstellung zu verwenden.
- Ersetzen eines Hosts aufgrund eines Wartungsereignisses.
- Starten einer Ersatz-DB-Instance.
- Wiederherstellung aus einer Snapshot-Sicherung.
- Starten einer angehaltenen DB-Instance

Auflösung

Verwenden Sie start-db-instance den Befehl

Gehen Sie folgendermaßen vor, um eine Datenbank zu reparieren, die sich im Status „inkompatibles Netzwerk“ befindet:

1. Öffnen Sie den https://console.aws.amazon.com/rds/und wählen Sie im Navigationsbereich Datenbanken aus.
2. Wählen Sie die DB-Instance aus, die sich im Status „Inkompatibles Netzwerk“ befindet, und notieren Sie sich die DB-Instance-ID, die VPC-ID und das Subnetz auf der Registerkarte Konnektivität und IDs Sicherheit.
3. Verwenden Sie den, um den Befehl auszuführen AWS CLI . `start-db-instance` Geben Sie den `--db-instance-identifier`-Wert an.

> **Note**
>
> Wenn Sie diesen Befehl ausführen, wenn sich Ihre Datenbank im inkompatiblen Modus befindet, kann dies zu Ausfallzeiten führen.
> Der `start-db-instance`-Befehl behebt dieses Problem für RDS für SQL Server-DB-Instances nicht.

Ihr Datenbankstatus ändert sich zu Verfügbar, wenn der Befehl erfolgreich ausgeführt wird.

Wenn Ihre Datenbank neu gestartet wird, führt die DB-Instance möglicherweise den letzten Vorgang aus, der auf der Instance ausgeführt wurde, bevor sie in den Status „Inkompatibles Netzwerk“ versetzt wurde. Dadurch könnte die Instance wieder in den Status „Inkompatibles Netzwerk“ versetzt werden.

Wenn der `start-db-instance`-Befehl nicht erfolgreich ist oder die Instance in den Status „Inkompatibles Netzwerk“ zurückkehrt, öffnen Sie die Seite Datenbanken in der RDS-Konsole, und wählen Sie die Datenbank aus. Navigieren Sie zum Abschnitt Protokolle und Ereignisse. Der Abschnitt Aktuelle Ereignisse zeigt weitere Lösungsschritte an, die Sie befolgen können. Die Nachrichten sind wie folgt klassifiziert:

- INTERNER RESSOURCENCHECK: Möglicherweise liegen Probleme mit Ihren internen Ressourcen vor.
- DNS-PRÜFUNG: Überprüfen Sie die DNS-Auflösung und die Hostnamen für die VPC in der VPC-Konsole.
- ENI-PRÜFUNG: Die Elastic-Network-Schnittstelle (ENI) für Ihre Datenbank ist möglicherweise nicht vorhanden.
- GATEWAY-PRÜFUNG: Das Internet-Gateway für Ihre öffentlich verfügbare Datenbank ist nicht mit der VPC verbunden.
- IP-PRÜFUNG: Es gibt keine freien IP-Adressen in Ihren Subnetzen.
- PRÜFUNG DER SICHERHEITSGRUPPE: Ihrer Datenbank sind keine Sicherheitsgruppen zugeordnet, oder die Sicherheitsgruppen sind ungültig.
- SUBNETZPRÜFUNG: Es gibt keine gültigen Subnetze in Ihrer DB-Subnetzgruppe, oder es gibt Probleme mit Ihrem Subnetz.
- VPC-PRÜFUNG: Die mit Ihrer Datenbank verknüpfte VPC ist ungültig.

Führen Sie die point-in-time Wiederherstellung durch

Es empfiehlt sich, eine Sicherungskopie (Snapshot oder logisch) zu erstellen, falls Ihre Datenbank in den Zustand „Inkompatibles Netzwerk“ übergeht. Siehe Einführung in Backups. Wenn Sie automatische Backups aktiviert haben, beenden Sie vorübergehend alle Schreibvorgänge in die Datenbank und führen Sie eine point-in-time Wiederherstellung durch.

> **Note**
>
> Wenn eine Instance in den Status „Inkompatibles Netzwerk“ übergeht, kann möglicherweise nicht mehr auf die DB-Instance zugegriffen werden, um ein logisches Backup durchzuführen.

Wenn Sie automatische Backups nicht aktiviert haben, erstellen Sie eine neue DB-Instance. Migrieren Sie dann die Daten mit AWS Database Migration Service (AWS DMS) oder mithilfe eines Sicherungs- und Wiederherstellungstools.

Wenn das Problem dadurch nicht behoben wird, wenden Sie sich an uns, Support um weitere Unterstützung zu erhalten.

Zurücksetzen des Besitzerpassworts der DB-Instance

Wenn Sie aus Ihrem DB-Instance- ausgesperrt werden, können Sie sich als Hauptbenutzer anmelden. Anschließend können Sie die Anmeldeinformationen für andere administrative Benutzer oder Rollen zurücksetzen. Wenn Sie sich nicht als Hauptbenutzer anmelden können, kann der AWS Kontoinhaber das Masterbenutzer-Passwort zurücksetzen. Weitere Informationen zu den Administratorkonten oder -rollen, die Sie möglicherweise zurücksetzen müssen, finden Sie unter Berechtigungen von Hauptbenutzerkonten.

Sie können das DB-Instance-Passwort ändern, indem Sie die Amazon RDS-Konsole, den AWS CLI Befehl modify-db-instanceoder den Vorgang „DBInstanceAPI modifizieren“ verwenden. Weitere Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Ausfall oder Neustart einer Amazon RDS-DB-Instance

Ein Ausfall der DB-Instance kann auftreten, wenn eine DB-Instance neu gestartet wird. Er kann auch auftreten, wenn die DB-Instance in einen Zustand versetzt wird, der den Zugriff darauf verhindert,

und wenn die Datenbank neu gestartet wird. Ein Neustart kann erfolgen, wenn Sie Ihre DB-Instance manuell neu starten. Ein Neustart kann auch auftreten, wenn Sie eine DB-Instance-Einstellung ändern, für die ein Neustart erforderlich ist, um wirksam zu werden.

Ein Neustart der DB-Instance tritt auf, wenn Sie eine Einstellung ändern, für die ein Neustart erforderlich ist, oder wenn Sie einen Neustart manuell durchführen. Ein Neustart kann sofort erfolgen, wenn Sie eine Einstellung ändern und die Änderung sofort wirksam werden soll. Oder er kann während des Wartungsfensters der DB-Instance auftreten.

Ein Neustart der DB-Instance wird sofort ausgeführt, wenn eine der folgenden Situationen eintritt:

- Sie ändern den Aufbewahrungszeitraum für Backups für eine DB-Instance von 0 auf einen Wert ungleich Null oder von einem Wert ungleich Null auf 0. Anschließend legen Sie Apply Immediately (Sofort anwenden) auf `true` fest.
- Sie ändern die DB-Instance-Klasse und Apply Immediately (Direkt anwenden) ist auf `true` eingestellt.
- Sie ändern den Speichertyp von Magnetic (Standard) (Magnetisch (Standard)) zu General Purpose (SSD) (Allzweck (SSD)) oder Provisioned IOPS (SSD) (Bereitgestellte IOPS (SSD)) oder von Provisioned IOPS (SSD) (Bereitgestellte IOPS (SSD)) oder General Purpose (SSD) (Allzweck (SSD)) zu Magnetic (Standard) (Magnetisch (Standard)).

Ein Neustart der DB-Instance tritt während des Wartungsfensters auf, wenn eine der folgenden Situationen eintritt:

- Sie ändern den Aufbewahrungszeitraum für Backups für eine DB-Instance von 0 auf einen Wert ungleich Null oder von einem Wert ungleich Null auf 0 und Apply Immediately (Direkt anwenden) ist auf `false` festgelegt.
- Sie ändern die DB-Instance-Klasse und Apply Immediately (Direkt anwenden) ist auf `false` eingestellt.

Wenn Sie einen statischen Parameter in einer DB-Parametergruppe ändern, wird die Änderung erst wirksam, wenn die der Parametergruppe zugeordnete DB-Instance neu gestartet wird. Die Änderung erfordert einen manuellen Neustart. Die DB-Instance wird während des Wartungsfensters nicht automatisch neu gestartet.

Um eine Tabelle mit den DB-Instance-Aktionen und dem Effekt der Einstellung des Wertes Apply Immediately anzuzeigen, siehe Ändern einer Amazon RDS DB-Instance.

Amazon RDS-DB-Parameter Änderungen wirken sich nicht aus

In einigen Fällen können Sie einen Parameter in einer DB-Parametergruppe ändern, aber die Änderungen werden nicht wirksam. Wenn dies der Fall ist, müssen Sie wahrscheinlich die DB-Instance, die der DB-Parametergruppe zugeordnet ist, neu starten. Wenn Sie einen dynamischen Parameter ändern, wird die Änderung sofort wirksam. Wenn Sie einen statischen Parameter ändern, wird die Änderung erst wirksam, wenn Sie die der Parametergruppe zugeordnete DB-Instance neu starten.

Sie können eine DB-Instance mithilfe der RDS-Konsole neu starten. Oder Sie können die API-Operation `RebootDBInstance` explizit aufrufen. Sie können ohne Failover neu starten, wenn sich die DB-Instance in einer Multi-AZ-Bereitstellung befindet. Da die zugeordnete DB-Instance nach der Änderung eines statischen Parameters neu gestartet werden muss, wird das Risiko einer fehlerhaften Konfiguration gesenkt, die API-Aufrufe beeinträchtigen könnte. Ein Beispiel hierfür ist der Aufruf von `ModifyDBInstance` zur Änderung der DB-Instance-Klasse. Weitere Informationen finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

Amazon RDS-DB-Instance hat zu wenig Speicher

Wenn Ihre DB-Instance zu wenig Speicherplatz hat, ist sie möglicherweise nicht mehr verfügbar. Wir empfehlen Ihnen dringend, die in veröffentlichte `FreeStorageSpace` Metrik ständig zu überwachen CloudWatch , um sicherzustellen, dass Ihre DB-Instance über ausreichend freien Speicherplatz verfügt.

Wenn Ihre Datenbank-Instance nicht mehr über ausreichend Speicher verfügt, ändert sich der Status in `storage-full`. Zum Beispiel gibt das Aufrufen der `DescribeDBInstances`-API-Operation für eine DB-Instance, deren Speicher aufgebraucht ist, Folgendes aus.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Um dieses Szenario zu überwinden, fügen Sie Ihrer Instance mithilfe der `ModifyDBInstance` API-Operation oder des folgenden AWS CLI Befehls mehr Speicherplatz hinzu.

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --allocated-storage 60 \
    --apply-immediately
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 60 ^
    --apply-immediately
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Nun werden Sie feststellen, dass Ihre DB-Instance den Status `modifying` aufweist. Das bedeutet, dass der Speicherplatz skaliert wird.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
modifying  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com
3306  us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Nachdem die Speicherskalierung abgeschlossen ist, ändert sich der Status Ihrer DB-Instance in `available`.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  60  sa
available  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
```

```
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Sie können Benachrichtigungen erhalten, wenn Ihr Speicherplatz durch die `DescribeEvents`-Operation erschöpft ist. Wenn Sie in einem solchen Fall nach diesen Operationen beispielsweise `DescribeEvents` aufrufen, wird Ihnen Folgendes angezeigt werden.

```
aws rds describe-events --source-type db-instance --source-identifier mydbinstance
```

```
2009-12-22T23:44:14.374Z  mydbinstance  Allocated storage has been exhausted db-
instance
2009-12-23T00:14:02.737Z  mydbinstance  Applying modification to allocated storage db-
instance
2009-12-23T00:31:54.764Z  mydbinstance  Finished applying modification to allocated
 storage
```

Amazon RDS: unzureichende DB-Instances verfügbar

Der Fehler `InsufficientDBInstanceCapacity` kann zurückgegeben werden, wenn Sie versuchen, eine DB-Instance zu erstellen, zu starten oder zu ändern. Er kann auch zurückgegeben werden, wenn Sie versuchen, eine DB-Instance aus einem DB-Snapshot wiederherzustellen. Wenn dieser Fehler zurückgegeben wird, besteht die Ursache häufig darin, dass die spezifische DB-Instance-Klasse in der angeforderten Availability Zone nicht verfügbar ist. Sie können folgende Aktionen ausführen, um zu versuchen, das Problem zu beheben:

- Wiederholen Sie die Anforderung mit einer anderen FB-Instance-Klasse.
- Wiederholen Sie die Anforderung mit einer anderen Availability Zone.
- Wiederholen Sie die Anforderung ohne explizite Angabe einer Availability Zone.

Informationen zur Behebung von Problemen mit der Instance-Kapazität für Amazon EC2 finden Sie unter Unzureichende Instance-Kapazität im EC2 Amazon-Benutzerhandbuch.

Informationen zum Ändern einer DB-Instance finden Sie unter Ändern einer Amazon RDS DB-Instance.

Probleme mit freisetzbarem Speicher in Amazon RDS

Freisetzbarer Speicher ist der gesamte Random Access Memory (RAM, Arbeitsspeicher) einer DB-Instance, der der Datenbank-Engine zur Verfügung gestellt werden kann. Es ist die Summe aus dem freien Arbeitsspeicher des Betriebssystems und dem verfügbaren Puffer- und Seitencache-Speicher. Die Datenbank-Engine verwendet den größten Teil des Speichers auf dem Host, aber Betriebssystemprozesse verbrauchen ebenfalls RAM. Speicher, der derzeit der Datenbank-Engine zugewiesen oder von Betriebssystemprozessen verwendet wird, ist nicht im freisetzbaren Speicher enthalten. Wenn der Datenbank-Engine der Speicher ausgeht, kann die DB-Instance den temporären Speicherplatz nutzen, der normalerweise zum Puffern und Zwischenspeichern verwendet wird. Wie bereits erwähnt, ist dieser temporäre Speicherplatz im freisetzbaren Speicher enthalten.

Sie verwenden die `FreeableMemory` Metrik in Amazon CloudWatch , um den freien Speicher zu überwachen. Weitere Informationen finden Sie unter Überwachungstools für Amazon RDS.

Wenn Ihre DB-Instance ständig wenig möglichen freien Speicher hat oder Auslagerungsbereiche verwendet, erwägen Sie, auf eine größere DB-Instance-Klasse hochzuskalieren. Weitere Informationen finden Sie unter .

Sie können auch die Speichereinstellungen ändern. Beispiel: Bei RDS for MySQL können Sie die Größe des Parameters `innodb_buffer_pool_size` anpassen. Dieser Parameter ist standardmäßig auf 75 Prozent des physischen Speichers festgelegt. Weitere Tipps zur MySQL-Fehlerbehebung finden Sie unter Wie kann ich Probleme mit geringem freisetzbaren Speicher in einer Datenbank von Amazon RDS für MySQL beheben?

Probleme mit MySQL und MariaDB

Sie können Probleme mit MySQL- und MariaDB-DB-Instances diagnostizieren und beheben.

Themen

- Maximale Anzahl von MySQL- und MariaDB-Verbindungen
- Diagnostizieren und Auflösen des Status "incompatible-parameters" für ein Speicherlimit
- Diagnose und Lösung bei Verzögerungen zwischen Read Replicas (Lesereplikaten)
- Diagnose und Lösung eines Fehlers bei einer MySQL oder MariaDB Read Replica
- Erstellen von Auslösern mit aktivierter Binärprotokollierung erfordert SUPER-Berechtigung
- Diagnose und Behebung von Wiederherstellungsfehlern point-in-time

- Fehler „Replication stopped (Replikation gestoppt)“
- Erstellung von Read Replica fehlgeschlagen oder Replikationsunterbrechungen mit schwerwiegendem Fehler 1236
- Die Read Replica-Replikation kann die Metadatenstruktur nicht initialisieren

Maximale Anzahl von MySQL- und MariaDB-Verbindungen

Die maximale Anzahl der für eine RDS für MySQL- oder RDS für MariaDB-DB-Instance zulässigen Verbindungen basiert auf der Menge des für die DB-Instance-Klasse verfügbaren Arbeitsspeichers. Eine DB-Instance Klasse mit mehr verfügbarem Arbeitsspeicher resultiert in einer größeren Anzahl von verfügbaren Verbindungen. Weitere Informationen zu DB-Instance-Klassen finden Sie unter .

Das Verbindungslimit einer DB-Instance ist standardmäßig festgelegt auf die maximale Verbindungsanzahl für die DB-Instance-Klasse. Sie können die Anzahl gleichzeitiger Verbindungen auf einen beliebigen Wert bis zur maximal zulässigen Anzahl von Verbindungen beschränken. Verwenden Sie den `max_connections`-Parameter in der Parametergruppe für die DB-Instance. Weitere Informationen erhalten Sie unter Maximale Anzahl von Datenbankverbindungen und Parametergruppen für Amazon RDS.

Sie können die maximal zulässige Anzahl von Verbindungen für eine MySQL- oder MariaDB-DB-Instance abrufen, indem Sie die folgende Abfrage ausführen.

```
SELECT @@max_connections;
```

Sie können die Anzahl der aktiven Verbindungen zu einer MySQL- oder MariaDB-DB-Instance abrufen, indem Sie die folgende Abfrage ausführen.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

Diagnostizieren und Auflösen des Status "incompatible-parameters" für ein Speicherlimit

Eine MariaDB- oder MySQL-DB-Instance kann in den Status incompatible-parameters für ein Speicherlimit versetzt werden, wenn die folgenden Bedingungen erfüllt sind:

- Die DB-Instance wird entweder mindestens dreimal innerhalb einer Stunde oder mindestens fünfmal an einem Tag neu gestartet, wenn der Status der DB-Instance Verfügbar lautet.

- Ein Versuch, die DB-Instance neu zu starten, schlägt fehl, weil eine Wartungsaktion oder ein Überwachungsprozess die DB-Instance nicht neu starten konnte.
- Die potenzielle Arbeitsspeicherauslastung der DB-Instance übersteigt das 1,2-Fache des Arbeitsspeichers, der ihrer DB-Instance-Klasse zugewiesen ist.

Wenn eine DB-Instance zum dritten Mal innerhalb einer Stunde oder zum fünften Mal an einem Tag neu gestartet wird, wird eine Prüfung der Arbeitsspeicherauslastung durchgeführt. Die Prüfung erstellt eine Berechnung der potenziellen Arbeitsspeicherauslastung der DB-Instance. Der von der Berechnung gelieferte Wert ist die Summe der folgenden Werte:

- Wert 1 – Die Summe der folgenden Parameter:
  - `innodb_additional_mem_pool_size`
  - `innodb_buffer_pool_size`

    Sie können den Wert für änder`innodb_buffer_pool_size`. Der Wert entspricht jedoch nicht immer dem, was Sie eingegeben haben. Diese Nichtübereinstimmung tritt aus mehreren Gründen auf. Wenn es sich bei der DB-Instance um eine Micro-DB-Instance handelt, überschreiben wir zunächst den Standardwert und setzen ihn auf 256 MB. Weitere Informationen finden Sie unter Überschreiben von innodb_buffer_pool_size.

    Zweitens stellen wir sicher, dass 500 MB Speicher auf der DB-Instance für den Hostmanager, die Engine, das Betriebssystem und den Kernel reserviert sind.

    Schließlich optimieren wir es, `innodb_buffer_pool_size` indem wir es in Einheiten unterteilen. Der Host-Manager rundet auf das nächste Vielfache dieser Einheiten ab. Die Einheiten werden durch Multiplikation mit `innodb_buffer_pool_chunk_size` berechnet. `innodb_buffer_pool_instances` Weitere Informationen finden Sie unter Configuring InnoDB Buffer Pool Size in der MySQL-Dokumentation.

    Die Standardeinstellung für `innodb_buffer_pool_instances` ist 8, sofern sie nicht `innodb_buffer_pool_size` weniger als 1 GB beträgt. Wenn weniger als 1 GB `innodb_buffer_pool_size` ist, `innodb_buffer_pool_instances` ist die Standardeinstellung für 1. Die Standardeinstellung für `innodb_buffer_pool_chunk_size` ist 128 MB.
  - `innodb_log_buffer_size`
  - `key_buffer_size`

  - `query_cache_size` (nur MySQL Version 5.7)
  - `tmp_table_size`
- Wert 2 – Der Parameter `max_connections` multipliziert mit der Summe der folgenden Parameter:
  - `binlog_cache_size`
  - `join_buffer_size`
  - `read_buffer_size`
  - `read_rnd_buffer_size`
  - `sort_buffer_size`
  - `thread_stack`
- Wert 3 – Wenn der Parameter `performance_schema` aktiviert ist, multiplizieren Sie den Parameter `max_connections` mit `429498`.

  Wenn der Parameter `performance_schema` deaktiviert ist, ist dieser Wert 0.

Der von der Berechnung gelieferte Wert ist also folgender:

```
Value 1 + Value 2 + Value 3
```

Wenn dieser Wert das 1,2-Fache des Arbeitsspeichers überschreitet, der der von der DB-Instance verwendeten DB-Instance-Klasse zugewiesen ist, wird die DB-Instance in den Status incompatible-parameters versetzt. Weitere Informationen über den Arbeitsspeicher, der DB-Instance-Klassen zugewiesen ist, finden Sie unter Hardware-Spezifikationen für DB-Instance-Klassen .

Die Berechnung multipliziert den Wert des Parameters `max_connections` mit der Summe mehrerer Parameter. Wenn der Parameter `max_connections` auf einen großen Wert festgelegt ist, kann das dazu führen, dass die Prüfung einen übermäßig hohen Wert für die potenzielle Arbeitsspeicherauslastung der DB-Instance zurückgibt. In diesem Fall sollten Sie erwägen, den Wert des Parameters `max_connections` zu senken.

Um das Problem zu lösen, führen Sie die folgenden Schritte aus:

1. Passen Sie die Arbeitsspeicherparameter der DB-Parametergruppe, die der DB-Instance zugeordnet ist, entsprechend an. Die potenzielle Arbeitsspeicherauslastung sollte niedriger sein als das 1,2-fache des Arbeitsspeichers, der der DB-Instance-Klasse zugewiesen ist.

   Weitere Informationen zum Festlegen von Parametern finden Sie unter Ändern von Parametern in einer DB-Parametergruppe in Amazon RDS.

2. Starten Sie die DB-Instance neu.

   Weitere Informationen zum Festlegen von Parametern finden Sie unter Starten einer Amazon RDS DB-Instance, die zuvor gestoppt wurde.

Diagnose und Lösung bei Verzögerungen zwischen Read Replicas (Lesereplikaten)

Nachdem Sie ein MySQL- oder MariaDB-Lesereplikat erstellt haben und das Replikat verfügbar ist, repliziert Amazon RDS zunächst die an der Quell-DB-Instance vorgenommen Änderungen ab dem Zeitpunkt der Operation der Lesereplikaterstellung. Während dieser Phase ist die Verzögerungszeit der Replikation für das Lesereplikat größer als 0. Sie können diese Verzögerungszeit in Amazon überwachen, CloudWatch indem Sie sich die Amazon `ReplicaLag`RDS-Metrik ansehen.

Die`ReplicaLag` Metrik gibt den Wert des`Seconds_Behind_Master` Feldes des MariaDB- oder`SHOW REPLICA STATUS` MySQL-Befehls an. Weitere Informationen finden Sie unter SHOW REPLICA STATUS-Anweisung in der MySQL-Dokumentation.

Wenn die Metrik `ReplicaLag` 0 erreicht, hat das Replica den Stand der Quell-DB-Instance erreicht. Wenn die `ReplicaLag` Metric auf -1 zurückgeht, ist die Replikation möglicherweise nicht aktiv. Um einen Replikationsfehler zu beheben, lesen Sie Diagnose und Lösung eines Fehlers bei einer MySQL oder MariaDB Read Replica. Ein `ReplicaLag`-Wert von -1 kann auch bedeuten, dass der `Seconds_Behind_Master`-Wert nicht bestimmt werden kann oder `NULL` ist.

> **Note**
>
> Frühere Versionen von MariaDB wurden`SHOW SLAVE STATUS` anstelle von verwendet`SHOW REPLICA STATUS`. Wenn Sie eine MariaDB-Version unter 10.5 verwenden, verwenden Sie. `SHOW SLAVE STATUS`

Die Metrik `ReplicaLag` gibt während eines Netzwerkausfalls den Wert -1 an oder wenn ein Patch während des Wartungsfensters angewendet wird. Warten Sie in diesem Fall, bis die Netzwerkverbindung wiederhergestellt ist oder die Wartung beendet ist, bevor Sie die Metrik `ReplicaLag` wieder überprüfen.

Die MySQL- und MariaDB-Lesereplikationstechnologie ist asynchron. Daher können Sie gelegentliche Erhöhungen für die `BinLogDiskUsage`-Metrik in der Quell-DB-Instance und für die

`ReplicaLag` -Metrik auf dem Lesereplikat erwarten. Betrachten Sie beispielsweise eine Situation, in der ein hohes Volumen von Schreiboperationen auf die Quell-DB-Instance parallel ausgeführt wird. Gleichzeitig werden Schreiboperationen in das Lesereplikat mit einem einzelnen I/O-Thread serialisiert. Eine solche Situation kann zu einer Verzögerung zwischen der Quell-Instance und dem Lesereplikat führen.

Weitere Informationen zu Read Replicas und MySQL finden Sie in unter Replication Implementation Details in der MySQL-Dokumentation. Weitere Informationen zu Read Replicas und MariaDB finden Sie unter Replication Overview in der MariaDB-Dokumentation.

Sie können die Verzögerung zwischen den Updates einer Quell-DB-Instance und den nachfolgenden Updates der Lesereplikate folgendermaßen reduzieren:

- Legen Sie für die DB-Instance-Klasse des Lesereplikats eine Speichergröße fest, die mit der Größe der Quell-DB-Instance vergleichbar ist.
- Stellen Sie sicher, dass die Parametereinstellungen in den DB-Parametergruppen, die von der Quell-DB-Instance und dem Lesereplikat verwendet werden, kompatibel sind. Weitere Informationen und ein Beispiel finden Sie zum Thema der `max_allowed_packet` Parameter im nächsten Abschnitt.
- Deaktivieren Sie den Anfrage-Cache. Bei Tabellen, die häufig geändert werden, kann die Verwendung des Anfragecaches die Replikationsverzögerung erhöhen, da der Cache häufig gesperrt und aktualisiert wird. Wenn dies der Fall ist, reduzieren Sie möglicherweise die Replikationsverzögerung durch die Deaktivierung des Anfragecaches. Sie können den Anfrage-Cache deaktivieren, indem Sie den `query_cache_type parameter` in der DB-Parametergruppe für die DB-Instance auf 0 setzen. Weitere Informationen zum Abfragecache finden Sie unter Abfragecache-Konfiguration.
- Wärmen Sie den Pufferpool auf dem Lesereplikat für InnoDB für MySQL oder MariaDB auf. Nehmen wir beispielsweise an, Sie haben eine kleine Gruppe von Tabellen, die häufig aktualisiert werden, und Sie verwenden das InnoDB- oder XTraDB-Tabellenschema. Legen Sie in diesem Fall diese Tabellen auf dem Lesereplikat ab. Dies führt dazu, dass die Datenbank-Engine die Zeilen dieser Tabellen auf dem Datenträger durchsucht und sie dann im Pufferpool zwischenspeichert. Dieser Ansatz kann Replikatverzögerung reduzieren. Es folgt ein Beispiel.

  Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
PROMPT> mysqldump \
    -h <endpoint> \
    --port=<port> \
```

```
    -u=<username> \
    -p <password> \
    database_name table1 table2 > /dev/null
```

Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
PROMPT> mysqldump ^
    -h <endpoint> ^
    --port=<port> ^
    -u=<username> ^
    -p <password> ^
    database_name table1 table2 > /dev/null
```

Diagnose und Lösung eines Fehlers bei einer MySQL oder MariaDB Read Replica

Amazon RDS überwacht den Replikationsstatus Ihrer Lesereplikate. RDS aktualisiert das Feld Replication State (Replikationsstatus) der Lesereplikat-Instance auf `Error`, wenn die Replikation aus irgendeinem Grund angehalten wird. Die Einzelheiten des von den MySQL- oder MariaDB-Engines ausgelösten Fehlers finden Sie im Feld Replication Error (Replikationsfehler). Ereignisse, die den Status des Lesereplikats angeben, werden ebenfalls generiert, einschließlich RDS-EVENT-0045, RDS-EVENT-0046 und RDS-EVENT-0057. Weitere Informationen über Ereignisse und Abonnements zu Ereignissen finden Sie unter Mit RDS Amazon-Event-Benachrichtigungen arbeiten. Wenn eine MySQL-Fehlermeldung zurückgegeben wird, lesen Sie den Fehler in der MySQL Fehlermeldungsdokumentation nach. Wenn eine MariaDB-Fehlermeldung ausgegeben wird, überprüfen Sie den Fehler in der MariaDB-Fehlermeldungsdokumentation.

Die folgenden, allgemeinen Situationen können häufig zu Replikationsfehlern führen:

- Der Wert für den `max_allowed_packet`-Parameter für ein Lesereplikat ist niedriger als der `max_allowed_packet`-Parameter für die Quell-DB-Instance.

  Der `max_allowed_packet`-Parameter ist ein benutzerdefinierter Parameter, den Sie in einer DB-Parametergruppe festlegen können. Der `max_allowed_packet`-Parameter wird verwendet, um die maximale Größe der Data Manipulation Language (DML) anzugeben, die in der Datenbank ausgeführt werden kann. In einigen Fällen ist der `max_allowed_packet`-Wert für die Quell-DB-Instance möglicherweise größer als der `max_allowed_packet`-Wert des Lesereplikats. In diesen Fällen kann der Replikationsprozess einen Fehler ausgeben und die Replikation anhalten. Der

häufigste Fehler ist `packet bigger than 'max_allowed_packet' bytes`. Sie können den Fehler beheben, indem Sie die DB-Parametergruppe der Quelle und des Lesereplikats mit denselben Parameterwerten für `max_allowed_packet` verwenden.

- Schreibvorgänge auf Tabellen in einem Lesereplikat. Wenn Sie Indizes für ein Lesereplikat erstellen, müssen Sie den `read_only`-Parameter auf 0 setzen, um die Indizes zu erstellen. Wenn Sie in Tabellen auf dem Lesereplikat schreiben, kann die Replikation unterbrochen werden.
- Die Verwendung einer nicht-transaktionalen Speicher-Engine wie MyISAM: Read Replicas erfordern eine transaktionale Speicher-Engine. Die Replikation wird nur für die folgenden Speicher-Engines unterstützt: InnoDB for MySQL oder MariaDB.

  Führen Sie zum Umwandeln einer MyISAM-Tabelle in eine InnoDB-Tabelle den folgenden Befehl aus:

  ```
  alter table <schema>.<table_name> engine=innodb;
  ```
- Verwenden von nicht-deterministischen Abfragen wie `SYSDATE()`. Weitere Informationen finden Sie unter Determination of Safe and Unsafe Statements in Binary Logging in der MySQL-Dokumentation.

Mit den folgenden Schritten können Sie Ihren Replikationsfehler beheben:

- Wenn Sie einen logischen Fehler feststellen und den Fehler sicher überspringen können, befolgen Sie die Schritte in Überspringen des aktuellen Replikationsfehlers für RDS für My SQL. Ihre MySQL- oder MariaDB DB-Instance muss eine Version ausführen, die die `mysql_rds_skip_repl_error` Verfahren umfasst. Weitere Informationen finden Sie unter mysql.rds_skip_repl_error.
- Wenn Sie auf ein Problem mit der Position des Binärprotokolls (Binlog) stoßen, können Sie die Position der Replikatwiedergabe mit dem Befehl oder ändern.
- Möglicherweise tritt aufgrund der hohen DML-Last ein vorübergehendes Leistungsproblem auf. In diesem Fall können Sie den Parameter `innodb_flush_log_at_trx_commit` in der DB-Parametergruppe für das Lesereplikat auf 2 festlegen. Dies kann das Aufholen des Lesereplikats unterstützen, auch wenn es vorübergehend die Atomizität, die Konsistenz, die Isolation und die Haltbarkeit (Atomicity, Consistency, Isolation und Durability – ACID) verringert.
- Sie können das Lesereplikat löschen und eine Instance mit derselben DB-Instance-Kennung erstellen. Wenn Sie dies tun, bleibt der Endpunkt derselbe wie der Ihres alten Lesereplikats.

Wenn ein Replikationsfehler korrigiert wird, ändert sich der Wert unter Replication State (Replikationsstatus) zu Replicating (Replizierend). Weitere Informationen finden Sie unter Behebung eines My SQL Read Replica-Problems.

Erstellen von Auslösern mit aktivierter Binärprotokollierung erfordert SUPER-Berechtigung

Wenn Sie versuchen, Auslöser in einer RDS für MySQL- oder RDS für MariaDB-DB-Instance zu erstellen, wird möglicherweise der folgende Fehler angezeigt.

```
"You do not have the SUPER privilege and binary logging is enabled"
```

Um Trigger zu verwenden, wenn die Binärprotokollierung aktiviert ist, ist die SUPER-Berechtigung erforderlich, die auf RDS für MySQL- und RDS für MariaDB-DB-Instancen beschränkt ist. Sie können Trigger erstellen, wenn die Binärprotokollierung ohne die SUPER-Berechtigung aktiviert wird, indem Sie die `log_bin_trust_function_creators` Parameter auf true setzen. Um den `log_bin_trust_function_creators` auf wahr zu stellen, legen Sie eine neue DB-Parametergruppe an oder ändern Sie eine vorhandene DB-Parametergruppe.

Sie können eine neue DB-Parametergruppe anlegen, mit der Sie Auslöser in Ihrer DB-Instance von RDS für MySQL oder RDS für MariaDB mit aktivierter Binärprotokollierung erstellen können. Verwenden Sie dazu die folgenden CLI-Befehle. Um eine vorhandene Parametergruppe zu ändern, beginnen Sie mit Schritt 2.

Um eine neue Parametergruppe zu erstellen, um Trigger mit der Binärprotokollierung zuzulassen, die über die CLI aktiviert ist

1. Neue Parametergruppe erstellen.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

   ```
   aws rds create-db-parameter-group \
       --db-parameter-group-name allow-triggers \
       --db-parameter-group-family mysql8.0 \
       --description "parameter group allowing triggers"
   ```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

   ```
   aws rds create-db-parameter-group ^
   ```

```
    --db-parameter-group-name allow-triggers ^
    --db-parameter-group-family mysql8.0 ^
    --description "parameter group allowing triggers"
```

2. Ändern Sie die DB-Parametergruppe, um Trigger zuzulassen.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name allow-triggers \
    --parameters "ParameterName=log_bin_trust_function_creators,
 ParameterValue=true, ApplyMethod=pending-reboot"
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name allow-triggers ^
    --parameters "ParameterName=log_bin_trust_function_creators,
 ParameterValue=true, ApplyMethod=pending-reboot"
```

3. Ändern Sie Ihre DB-Instance, um die neue DB-Parametergruppe zu verwenden.

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Linux, macOS, oder Unix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name allow-triggers \
    --apply-immediately
```

   Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name allow-triggers ^
    --apply-immediately
```

4. Starten Sie die DB-Instance manuell neu, damit die Änderungen wirksam werden.

```
aws rds reboot-db-instance --db-instance-identifier mydbinstance
```

Diagnose und Behebung von Wiederherstellungsfehlern point-in-time

Wiederherstellung einer DB-Instance, die temporäre Tabellen enthält

Wenn Sie versuchen, Ihre MySQL- oder MariaDB-DB-Instance point-in-time wiederherzustellen (PITR), tritt möglicherweise der folgende Fehler auf.

```
Database instance could not be restored because there has been incompatible database
 activity for restore
functionality. Common examples of incompatible activity include using temporary tables,
 in-memory tables,
or using MyISAM tables. In this case, use of Temporary table was detected.
```

PITR benötigt sowohl Sicherungs-Snapshots als auch Binärprotokolle (Binlogs) von MySQL oder MariaDB, um Ihre DB-Instance zu einer bestimmten Zeit wiederherzustellen. Temporäre Tabelleninformationen können in binlogs unzuverlässig sein und einen PITR-Fehler verursachen. Wenn Sie temporäre Tabellen in Ihrer MySQL- oder MariaDB-DB-Instance verwenden, können Sie die Wahrscheinlichkeit eines PITR-Fehlers verringern, indem Sie häufigere Backups durchführen. Ein PITR-Fehler ist in der Zeit zwischen der Erstellung einer temporären Tabelle und dem nächsten Sicherungs-Snapshot am wahrscheinlichsten.

Wiederherstellen einer DB-Instance, die In-Memory-Tabellen enthält

Beim Wiederherstellen einer Datenbank mit speicherinternen Tabellen kann ein Problem auftreten. In-Memory-Tabellen werden während eines Neustarts gelöscht. Infolgedessen sind Ihre speicherinternen Tabellen möglicherweise nach einem Neustart leer. Wir empfehlen, dass Sie bei der Verwendung von speicherinternen Tabellen Ihre Lösung für die Behandlung leerer Tabellen im Falle eines Neustarts erstellen. Wenn Sie In-Memory-Tabellen mit replizierten DB-Instances verwenden, müssen Sie möglicherweise die Lesereplikate nach einem Neustart neu erstellen. Dies kann erforderlich sein, wenn ein Lesereplikat neu gestartet wird und Daten aus einer leeren In-Memory-Tabelle nicht wiederherstellen kann.

Weitere Informationen zu Sicherungen und PITR finden Sie unter Einführung in Backups und Wiederherstellung einer DB-Instance zu einem bestimmten Zeitpunkt für Amazon RDS.

Fehler „Replication stopped (Replikation gestoppt)“

Wenn Sie den `mysql.rds_skip_repl_error` Befehl aufrufen, wird möglicherweise eine Fehlermeldung angezeigt, die besagt, dass die Replikation heruntergefahren oder deaktiviert ist.

Diese Fehlermeldung wird angezeigt, wenn die Replikation angehalten wird und nicht mehr neu gestartet werden kann.

Wenn Sie eine größere Anzahl von Fehlern ignorieren müssen, kann die Dauer der Verzögerung der Replikation den standardmäßigen Aufbewahrungszeitraum für binäre Protokolldateien überschreiten. In diesem Fall kann ein schwerwiegender Fehler auftreten, da binäre Protokolldateien gelöscht werden, bevor sie auf dem Replikat wiedergegeben wurden. Diese Bereinigung führt zur Beendigung der Replikation, und Sie können den Befehl `mysql.rds_skip_repl_error` nicht mehr aufrufen, um Replikationsfehler zu überspringen und zu ignorieren.

Sie können dieses Problem umgehen, indem Sie die Anzahl der Stunden erhöhen, für die binäre Protokolldateien in der Replikationsquelle beibehalten werden. Nachdem Sie die Aufbewahrungsdauer für binäre Protokolldateien verlängert haben, können Sie die Replikation neu starten und nach Bedarf den Befehl `mysql.rds_skip_repl_error` aufrufen.

Verwenden Sie das mysql.rds_set_configuration-Verfahren, um den Aufbewahrungszeitraum für Binärprotokolle festzulegen. Geben Sie einen Konfigurationsparameter namens "binlog retention hours" und einen zugehörigen Wert in Stunden an, um festzulegen, wie viele Stunden binäre Protokolldateien auf dem DB-Cluster vorgehalten werden sollen (maximal 720 Stunden = 30 Tage). Beim folgenden Beispiel wird die Aufbewahrungszeit für binäre Protokolle auf 48 Stunden festgelegt.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

Erstellung von Read Replica fehlgeschlagen oder Replikationsunterbrechungen mit schwerwiegendem Fehler 1236

Nach dem Ändern der Standardparameterwerte für eine MySQL- oder MariaDB DB-Instance kann eines der folgenden Probleme auftreten:

- Sie können kein Lesereplikat für die DB-Instance erstellen.
- Replication fails with `fatal error 1236`.

Einige Standardparameterwerte für MySQL- und MariaDB-DB-Instances helfen dabei sicherzustellen, dass die Datenbank ACID-konform ist und Lesereplikate absturzsicher sind. Sie tun dies, indem sie sicherstellen, dass jeder Commit vollständig synchronisiert wird, indem sie die Transaktion in das Binärprotokoll schreiben, bevor sie festgeschrieben wird. Wenn Sie diese Parameter so ändern, dass sie von ihren Standardwerten abweichen, um die Leistung zu verbessern, kann die Replikation fehlschlagen, wenn eine Transaktion nicht in das Binärprotokoll geschrieben wurde.

Legen Sie die folgenden Parameterwerte fest, um dieses Problem zu beheben:

- `sync_binlog = 1`
- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

Die Read Replica-Replikation kann die Metadatenstruktur nicht initialisieren

Beim Versuch, die Replikation zu starten, wurde die folgende Fehlermeldung angezeigt:

```
Read Replica Replication Error - SQLError: 13124, reason: Replica failed to initialize
 applier metadata structure from the repository
```

Dieser Fehler tritt auf, wenn ein Problem mit der Metadatenstruktur des Replikats vorliegt. Um die Metadatenstruktur zu korrigieren, müssen Sie ein neues Replikat erstellen.

Um zu verhindern, dass dies in future passiert, führen Sie eine der folgenden Aktionen aus:

- Deaktivieren Sie nach Möglichkeit Multithreading auf Ihren Replikaten. Ab MySQL 8.0.27 ist Multithreading standardmäßig aktiviert.
- Wenn Sie Multithreading für Ihre Replikate verwenden müssen, empfehlen wir Ihnen, die GTID-basierte Replikation zu verwenden. Weitere Informationen finden Sie unter Verwenden der GTID basierten Replikation.

Der Aufbewahrungszeitraum für Backups kann nicht auf 0 gesetzt werden

Es gibt mehrere Gründe, warum Sie möglicherweise den Aufbewahrungszeitraum für Backups auf 0 setzen müssen. Beispielsweise können Sie automatische Backups unmittelbar deaktivieren, indem Sie den Aufbewahrungszeitraum 0 setzen.

In einigen Fällen können Sie den Wert auf 0 festlegen und erhalten eine Meldung, dass der Aufbewahrungszeitraum zwischen 1 und 35 liegen muss. Stellen Sie in diesen Fällen sicher, dass Sie kein Lesereplikat für die Instance eingerichtet haben. Lesereplikate benötigen Sicherungen für die Verwaltung der Lesereplikatsprotokolle, deshalb können Sie den Aufbewahrungszeitraum nicht auf 0 setzen.

Amazon RDS-API-Referenz

Zusätzlich zur AWS Management Console und zur AWS Command Line Interface (AWS CLI) bietet Amazon RDS auch eine API. Mithilfe der API können Sie Aufgaben zur Verwaltung Ihrer DB-Instances und anderer Objekte in Amazon RDS automatisieren.

- Eine alphabetische Liste der API-Operationen finden Sie unter Aktionen.
- Eine alphabetische Liste der Datentypen finden Sie unter Datentypen.
- Eine Liste der häufigen Abfrageparameter finden Sie unter Häufige Parameter.
- Beschreibungen der Fehlercodes finden Sie unter Häufige Fehler.

Weitere Informationen zur AWS CLI finden Sie in der AWS Command Line Interface-Amazon-RDS-Referenz.

Themen

- Verwenden der Abfrage-API
- Fehlerbehebung für Anwendungen in Amazon RDS

Verwenden der Abfrage-API

In den folgenden Abschnitten werden die Parameter und die Abfrageauthentifizierung beschrieben, die für die Abfrage-API verwendet werden.

Allgemeine Informationen zur Funktionsweise der Abfrage-API finden Sie unter Abfrageanforderungen im Amazon EC2 API Reference.

Abfrageparameter

HTTP-Query-basierte Anfragen sind HTTP-Anfragen, die das HTTP-Verb GET oder POST und einen Query-Parameter namens verwende `Action`.

Jede Query-Anfrage muss einige allgemeine Parameter enthalten, um die Authentifizierung und Auswahl einer Aktion zu bearbeiten.

Einige Operationen verwenden Parameterlisten. Diese Listen werden mit der Notation `param.`*n* definiert. Werte von *n* sind Ganzzahlen ab 1.

Informationen zu Amazon-RDS-Regionen und -Endpunkten finden Sie unter Amazon Relational Database Service (RDS) im Abschnitt zu Regionen und Endpunkten der Allgemeine Amazon Web Services-Referenz.

Authentifizierung von Abfrageanforderungen

Sie können Query-Anfragen nur über HTTPS senden und müssen in jede Query-Anfrage eine Signatur einschließen. Sie müssen entweder AWS-Signature Version 4 oder -Signature Version 2 verwenden. Weitere Informationen finden Sie unter Signaturprozess mit Signature Version 4 und Signaturprozess mit Signature Version 2.

Fehlerbehebung für Anwendungen in Amazon RDS

Amazon RDS stellt spezifische und beschreibende Fehlermeldungen bereit, um Sie bei der Behebung von Problemen während der Interaktion mit der Amazon RDS-API zu unterstützen.

Themen

- Fehler bei Abrufen
- Tipps zur Problembehebung

Weitere Informationen zur Fehlerbehebung bei Amazon RDS-DB-Instances finden Sie unter Fehlerbehebung für Amazon RDS.

Fehler bei Abrufen

In der Regel sollte Ihre Anwendung überprüfen, ob eine Anforderung einen Fehler verursacht hat, bevor Sie Zeit für die Verarbeitung von Ergebnissen aufwenden. Die einfachste Möglichkeit, herauszufinden, ob ein Fehler aufgetreten ist, besteht darin, nach einem `Error`-Knoten in der Antwort aus der Amazon RDS-API zu suchen.

Die XPath-Syntax bietet eine einfache Möglichkeit, nach einem `Error`-Knoten zu suchen. Darüber hinaus vereinfacht sie den Abruf von Fehlercode und Fehlermeldung. Der folgende Codeausschnitt verwendet Perl und das XML::XPath-Modul, um zu ermitteln, ob während einer Anfrage ein Fehler aufgetreten ist. Wenn ein Fehler aufgetreten ist, gibt der Code den ersten Fehlercode und die erste Fehlermeldung in der Antwort an.

```
use XML::XPath;
    my $xp = XML::XPath->new(xml =>$response);
```

```
    if ( $xp->find("//Error") )
    {print "There was an error processing your request:\n", " Error code: ",
    $xp->findvalue("//Error[1]/Code"), "\n", " ",
    $xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Tipps zur Problembehebung

Die folgenden Prozesse werden empfohlen, um Probleme mit der Amazon-RDS-API zu diagnostizieren und zu beheben.

- Überprüfen Sie, ob Amazon RDS in der AWS-Region normal ausgeführt wird, indem Sie http://status.aws.amazon.com aufrufen.
- Überprüfen Sie die Struktur Ihrer Anforderung.

  Jede Amazon RDS-Operation verfügt über eine Referenzseite in der Amazon RDS-API-Referenz. Prüfen Sie nochmals, dass Sie die Parameter korrekt verwenden. Betrachten Sie die Beispielanforderungen oder Benutzerszenarien, um zu sehen, ob ähnliche Operationen ausgeführt werden, und um eine Vorstellung von möglichen Fehlern zu erhalten.
- Überprüfen Sie AWS re:Post.

  Amazon RDS besitzt ein Entwickler-Community, in der Sie nach Lösungen für Probleme suchen können, die andere Entwickler bereits hatten. Zum Anzeigen der Themen navigieren Sie zu AWS re:Post.

Dokumentverlauf

Aktuelle API-Version:2014-10-31

In der folgenden Tabelle sind wichtige Änderungen der einzelnen Versionen des Amazon RDS Benutzerhandbuchs nach Mai 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

> **Note**
>
> Sie können neue Amazon RDS Funktionen auf der Was ist neu mit Datenbank?-Seite filtern. Wählen Sie für Produkte Amazon RDS aus. Suchen Sie dann mit Schlüsselwörtern wie **`RDS Proxy`** oder **`Oracle 2023`**.

Änderung	Beschreibung	Datum
Amazon RDS unterstützt MariaDB 11.4.8, 10.11.14 und 10.6.23	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 11.4.8, 10.11.14 und 10.6.2 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	14. August 2025
Die Amazon RDS-Richtlinie für benutzerdefinierte, serviceverknüpfte Rollen wurde aktualisiert	Die von Amazon RDSCustomServiceRolePolicy verwaltete Richtlinie wurde aktualisiert, um dem CopySnapshot EBS-Autorisierungsverhalten zu entsprechen. Das Update `ec2:CopySnapshot` entfernt eine Anweisung und fügt zwei neue Anweisungen für Quell- und Ziel-Snapshot-Berechtigungen hinzu.	7. August 2025

	Weitere Informationen finden Sie unter Aktualisierungen der AWS verwalteten Richtlinien für Amazon RDS.	
Amazon RDS unterstützt MySQL 8.4.6 und 8.0.43	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL 8.4.6 und 8.0.43 ausgeführt wird. MySQL 8.4.6 enthält das neue Datenbank-Plugin rds_security_audit, ein internes RDS-MySQL-Plugin, das sicherheitsrelevante Metriken sammelt. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	1. August 2025
RDS for Oracle unterstützt die DB-Instance-Klassen db.m6in und db.r6in	Sie können die DB-Instance-Klassen db.m6in und db.r6in mit RDS for Oracle mit dem Bring Your Own License (BYOL) -Modell verwenden. Weitere Informationen finden Sie unter RDS für Oracle-DB-Instance-Klassen.	29. Juli 2025
Zero-ETL-Integrationen von RDS für PostgreSQL und RDS für Oracle sind jetzt verfügbar	Sie können jetzt Zero-ETL-Integrationen mit RDS für PostgreSQL und RDS für Oracle als Quelle erstellen. Weitere Informationen finden Sie unter Amazon RDS Zero-ETL-Integrationen.	23. Juli 2025

Amazon RDS for Db2 unterstützt die Gruppenautorisierung mit Kerberos	Sie können jetzt die Gruppenautorisierung mithilfe von Kerberos mit Ihrem selbstverwalteten Active Directory durchführen. Weitere Informationen finden Sie unter Einrichten der Kerberos-Authentifizierung für RDS für Db2-DB-Instances.	21. Juli 2025
Zero-ETL-Integrationen mit Amazon SageMaker AI sind allgemein verfügbar	Zero-ETL-Integrationen machen Transaktionsdaten innerhalb von Sekunden, nachdem sie in eine RDS for MySQL-DB-Instance geschrieben wurden, in einem Amazon SageMaker Lakehouse verfügbar. Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS Zero-ETL-Integrationen.	30. Juni 2025
Amazon RDS unterstützt MySQL 9.3 in der Database Preview-Umgebung	MySQL 9.3 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL Version 9.3 in der Database Preview-Umgebung.	16. Juni 2025

Amazon RDS for Db2 unterstützt regionsübergreifende Replikate im Standby-Modus	Sie können jetzt regionsübergreifende RDS für Db2-Replikate im Standby-Modus für die Notfallwiederherstellung erstellen. Weitere Informationen finden Sie unter Arbeiten mit Replikaten für RDS for Db2.	11. Juni 2025
Amazon RDS unterstützt MariaDB 10.11.13 und 11.4.7	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen die MariaDB-Versionen 10.11.13 und 11.4.7 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	4. Juni 2025
End-of-life Informationen für Performance Insights	AWS hat den end-of-life Termin für Performance Insights bekannt gegeben: 30. November 2025. Nach diesem Datum unterstützt Amazon RDS die Performance Insights Insights-Konsole, die flexiblen Aufbewahrungsfristen (1—24 Monate) und die damit verbundenen Preise nicht mehr. Wir empfehlen, Ihre DB-Instances vor diesem Datum auf Database Insights zu aktualisieren. Weitere Informationen finden Sie unter Überwachen der DB-Auslastung mit Performance Insights auf Amazon RDS.	30. Mai 2025

Amazon RDS Extended Support Version 5.7.44-RDS.20250508 für RDS für MySQL	Die RDS Extended Support Version 5.7.44-RDS.20250508 ist jetzt für RDS for MySQL verfügbar. Weitere Informationen finden Sie unter Amazon RDS Extended Support-Versionen für RDS for MySQL.	27. Mai 2025
RDS Custom for Oracle unterstützt die Instanzklassen db.m7i und db.r7i	Sie können jetzt die Instanzklassen db.m7i und db.r7i für RDS Custom for Oracle verwenden. Weitere Informationen finden Sie unter Support für DB-Instance-Klassen für RDS Custom für Oracle.	22. Mai 2025
Support-Daten für Open-Source-Engine-Hauptversionen anzeigen	Sie können jetzt Informationen zu den Start- und Enddaten für Open-Source-Engine-Hauptversionen in RDS Standard Support und RDS Extended Support anzeigen, indem Sie die AWS CLI und die RDS-API verwenden. Weitere Informationen finden Sie unter Supportdaten für Engine-Versionen anzeigen in Amazon RDS Extended Support.	20. Mai 2025
Amazon RDS unterstützt MariaDB 10.5.29 und 10.6.22	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 10.5.29 und 10.6.22 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	20. Mai 2025

Amazon RDS unterstützt MySQL 8.0.42	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL 8.0.42 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	12. Mai 2025
Amazon RDS unterstützt MySQL 8.4.5	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL 8.4.5 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	1. Mai 2025
Amazon RDS unterstützt die Backup-Replikation für Asien-Pazifik (Hongkong), Asien-Pazifik (Melbourne), Asien-Pazifik (Malaysia), Kanada West (Calgary) und Europa (Zürich)	Die Backup-Replikation ist jetzt für Datenbanken in den Regionen Asien-Pazifik (Hongkong), Asien-Pazifik (Melbourne), Asien-Pazifik (Malaysia), Kanada West (Calgary) und Europa (Zürich) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter Automatisierte Backups in eine andere Region replizieren. AWS-Region	30. April 2025

Amazon RDS unterstützt verwaltete Master-Benutzerkennwörter in der Oracle-Multitenant-Architektur.	Sie können es verwenden AWS Secrets Manager , um Master-Benutzerkennwörter für RDS for Oracle-Container-Datenbanken (CDBs) und austauschbare Datenbanken () zu verwalten . PDBs Sowohl die Single-Tenant- als auch die Multi-Tenant-Konfigurationen der CDB-Architektur unterstützen verwaltete Passwörter. Weitere Informationen finden Sie unter Benutzerkonten und Rechte in einer CDB.	24. April 2025
Amazon RDS for Oracle unterstützt die Instance-Klassen m6id und r6id für BYOL	Sie können jetzt RDS für Oracle-DB-Instances mithilfe der Instance-Klassen m6id und r6id im BYOL-Modell erstellen. Enterprise und Standard Edition 2 werden beide unterstützt. Weitere Informationen finden Sie unter Unterstützte Edition, Instanzklasse und Lizenzkombinationen in RDS for Oracle.	9. April 2025
Amazon RDS unterstützt MariaDB 11.8 in der Database Preview-Umgebung	MariaDB 11.8 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informationen finden Sie unter MariaDB Version 11.8 in der Database Preview-Umgebung.	2. April 2025

Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die `AmazonRDSCustomServiceRolePolicy` Richtlinie gewährt jetzt zusätzliche Berechtigungen zur Verwaltung von EC2 Schlüsselpaaren und zur Integration mit Amazon SQS. Weitere Informationen finden Sie unter Amazon RDS-Updates für AWS verwaltete Richtlinien.	25. März 2025
Amazon RDS unterstützt MySQL 9.2 in der Database Preview-Umgebung	MySQL 9.2 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL Version 9.2 in der Database Preview-Umgebung.	24. März 2025
Aktualisierung der bestehenden Richtlinie	Amazon RDS hat der verwalteten Richtlinie neue Berechtigungen hinzugefügt`AmazonRDSCustomInstanceProfileRolePolicy` , um die Verwendung von RDS Custom Managed Secrets auf einer RDS Custom-Instance zu ermöglichen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSCustomInstanceProfileRoleI-Richtlinie.	20. März 2025

Amazon RDS Extended Support Version 5.7.44-RDS.20250213 für RDS für MySQL	Die RDS Extended Support Version 5.7.44-RDS.20250213 ist jetzt für RDS for MySQL verfügbar. Weitere Informationen finden Sie unter Amazon RDS Extended Support-Versionen für RDS for MySQL.	12. März 2025
Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die `AmazonRDSCustomServiceRolePolicy` Richtlinie gewährt jetzt zusätzliche Berechtigungen zum Auflisten und Wiederherstellen von Secrets Manager Manager-Geheimnissen. Weitere Informationen finden Sie unter Amazon RDS-Updates für AWS verwaltete Richtlinien.	6. März 2025
Amazon RDS for Db2 unterstützt die Instance-Klassen db.m7i und db.r7i	Sie können jetzt die Instance-Klassen db.m7i und db.r7i für RDS for Db2 verwenden. Weitere Informationen finden Sie unter DB-Instance-Klassenarten.	27. Februar 2025

Amazon RDS for Oracle unterstützt Spatial Patch Bundles () SPBs	RDS for Oracle veröffentlicht jetzt Spatial Patch Bundles (SPB) Engine-Versionen für Oracle Database 19c. Eine SPB-Engine-Version enthält Release-Update-Patches (RU) sowie spezielle Patches für Oracle Spatial. In der Regel wird ein SPB 2-3 Wochen nach der entsprechenden RU veröffentlicht. Sie können Ihre DB-Instance von einer Standard-RU auf ein SPB mit derselben DB-Engine-Version oder höher aktualisieren. Weitere Informationen finden Sie unter Upgrades für kleinere Oracle-Versionen.	25. Februar 2025
Amazon RDS unterstützt MariaDB 11.4.5, 10.11.11, 10.6.21 und 10.5.28	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 11.4.5, 10.11.11, 10.6.21 und 10.5.28 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	24. Februar 2025

Amazon RDS unterstützt CloudWatch Database Insights	Sie können jetzt Database Insights verwenden, um Amazon RDS-Datenbanken zu überwachen. Weitere Informationen finden Sie unter Überwachen von Amazon RDS-Datenbanken mit CloudWatch Database Insights.	20. Februar 2025
Amazon RDS unterstützt MySQL 8.4.4 und MySQL 8.0.41	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL 8.4.4 und 8.0.41 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	19. Februar 2025
Amazon RDS Extended Support Version 5.7.44-RDS.20250103 für RDS für MySQL	Die RDS Extended Support Version 5.7.44-RDS.20250103 ist jetzt für RDS for MySQL verfügbar. Weitere Informationen finden Sie unter Amazon RDS Extended Support-Versionen für RDS for MySQL.	13. Februar 2025
Amazon RDS unterstützt MariaDB 11.7 in der Database Preview-Umgebung	MariaDB 11.7 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informationen finden Sie unter MariaDB Version 11.7 in der Database Preview-Umgebung.	9. Januar 2025

Amazon RDS unterstützt MariaDB 11.4.4, 10.11.10, 10.6.20 und 10.5.27	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 11.4.4, 10.11.10, 10.6.20 und 10.5.27 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	20. Dezember 2024
Amazon RDS for Db2 unterstützt mehrere Datenbanken	Sie können einer RDS for Db2-DB-Instance bis zu 50 Db2-Datenbanken hinzufügen. Weitere Informationen finden Sie unter Mehrere Datenbanken auf einer Amazon RDS for Db2-DB-Instance.	20. Dezember 2024
Amazon RDS unterstützt MySQL 9.1 in der Database Preview-Umgebung	MySQL 9.1 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL Version 9.1 in der Database Preview-Umgebung.	19. Dezember 2024
Amazon RDS unterstützt die Instance-Klassen db.m8g und db.r8g	Sie können jetzt die Instanzklassen db.m8g und db.r8g für RDS für MySQL, RDS für PostgreSQL und RDS für MariaDB verwenden. Weitere Informationen finden Sie unter DB-Instance-Klassenarten.	21. November 2024

Amazon RDS unterstützt MySQL 8.4	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL Version 8.4 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	21. November 2024
Amazon Relational Database Service unterstützt die automatische EC2 Migration von Datenbanken	Sie können die RDS-Konsole verwenden, um eine EC2 Datenbank zu Amazon RDS zu migrieren. Amazon RDS verwendet AWS Database Migration Service (AWS DMS), um Ihre EC2 Quelldatenbank zu migrieren. AWS DMS ermöglicht es Ihnen, relationale Datenbanken in Ihre AWS Cloud zu migrieren. Weitere Informationen finden Sie unter Automatische Migration von EC2 Datenbanken zu Amazon Relational Database Service mithilfe von. AWS Database Migration Service	20. November 2024

Amazon blue/green RDS-Bereitstellungen unterstützen die Skalierung von Speicher	Sie können jetzt Amazon blue/green RDS-Bereitstellungen verwenden, um die Speichereinstellungen in der grünen Umgebung anzupassen, um die Ressourcenzuweisung zu optimieren. Weitere Informationen finden Sie unter Änderungen beim Erstellen einer blue/green Bereitstellung angeben.	20. November 2024
Amazon blue/green RDS-Bereitstellungen fügen Speicherinitialisierung hinzu	Amazon RDS unterstützt jetzt die Speicherinitialisierung in umweltfreundlichen Umgebungen für blue/green Bereitstellungen und verbessert so die Volume-Leistung ab der ersten Verwendung, ohne die Verfügbarkeit zu beeinträchtigen. Weitere Informationen finden Sie unter Lazy Loading und Speicherinitialisierung für Bereitstellungen. blue/green	20. November 2024

Amazon RDS for Oracle unterstützt die Instance-Klassen db.m7i und db.r7i für BYOL	Sie können die Instance-Klassen db.m7i und db.r7i für RDS for Oracle im BYOL-Lizenzmodell verwenden . Alle Editionen von Oracle Database werden unterstützt. Weitere Informationen finden Sie unter DB-Instance-Klassentypen und Unterstützte RDS für Oracle-DB-Instance-Klassen.	18. November 2024
Amazon RDS Proxy unterstützt RDS für PostgreSQL 17	Sie können jetzt Proxys mit RDS Proxy for RDS für PostgreSQL 17-DB-Instances erstellen. Weitere Informationen finden Sie unter Amazon RDS Proxy.	15. November 2024
Amazon RDS unterstützt MySQL 8.0.40	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL Version 8.0.40 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	13. November 2024
Amazon RDS unterstützt Backup-Replikation für Afrika (Kapstadt) und Asien-Pazifik (Hyderabad)	Die Backup-Replikation ist jetzt für Datenbanken in den Regionen Afrika (Kapstadt) und Asien-Pazifik (Hyderabad) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter Automatisierte Backups in eine andere Region replizieren. AWS-Region	1. November 2024

Amazon RDS unterstützt die Instance-Klassen db.m7i und db.r7i	Sie können jetzt die Instanzklassen db.m7i und db.r7i für RDS für MySQL, RDS für PostgreSQL und RDS für MariaDB verwenden. Weitere Informationen finden Sie unter DB-Instance-Klassenarten.	29. Oktober 2024
Amazon RDS for Oracle unterstützt Oracle APEX Version 24.1.v1	Sie können Oracle APEX 24.1.v1 mit Oracle Database 19c und höher verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	22. Oktober 2024
Multi-AZ-DB-Cluster unterstützen die IAM-Datenbankauthentifizierung	Sie können sich jetzt mithilfe der AWS Identity and Access Management (IAM-) Datenbankauthentifizierung bei Ihrem Multi-AZ-DB-Cluster authentifizieren. Weitere Informationen finden Sie unter Einstellungen für die Erstellung von Multi-AZ-DB-Clustern.	17. Oktober 2024
Amazon RDS unterstützt MariaDB 11.4	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB Version 11.4 ausgeführt wird. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	15. Oktober 2024

Amazon RDS unterstützt Console-to-Code	Sie können es jetzt verwenden Console-to-Code, um Code aus Aktionen zu generiere n, die Sie in der RDS-Konso le ausführen. Der generiert e Code kann Ihnen beim Schreiben von Code helfen, um Ihre Nutzung anderer AWS Dienste zu automatis ieren. Weitere Informationen finden Sie unter Verwenden , Console-to-Code um Code für Ihre Amazon RDS-Konso lenaktionen zu generieren.	3. Oktober 2024
Amazon RDS for Oracle unterstützt Oracle Managemen t Agent 13.5.0.0.v2	RDS unterstützt Oracle Management Agent 13.5.0.0.v2, für den Oracle Management Service (OMS) 13.5.0.23 erforderlich ist. Weitere Informationen finden Sie unter Anforderungen für den Management Agent.	25. September 2024

Zero-ETL-Integrationen mit Amazon Redshift sind allgemein verfügbar	Zero-ETL-Integrationen machen Transaktionsdaten in Amazon Redshift innerhalb von Sekunden verfügbar, nachdem sie in eine RDS for MySQL-DB-Instance geschrieben wurden. Die Funktion ist jetzt allgemein verfügbar. Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS Zero-ETL-Integrationen mit Amazon Redshift.	12. September 2024
Amazon RDS unterstützt MariaDB 10.11.9, 10.6.19 und 10.5.26	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 10.11.9, 10.6.19 und 10.5.26 ausgeführt werden. Weitere Informationen finden Sie unter Anforderungen für den Management Agent.	4. September 2024
Amazon RDS for Oracle unterstützt die OEM-, OEM-AGENT- und OLS-Optionen für die CDB-Architektur	Sie können jetzt Oracle Enterprise Manager und Oracle Label Security mit RDS für Oracle CDB-Instances verwenden. Weitere Informationen finden Sie unter Oracle Enterprise Manager und Oracle Label Security.	4. September 2024

RDS Custom for SQL Server ist in weiteren Regionen verfügbar	RDS Custom for SQL Server ist jetzt in den Regionen USA West (Nordkalifornien), Asien-Pazifik (Osaka) und Europa (Paris) verfügbar. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom.	29. August 2024
Amazon RDS Extended Support Version 5.7.44-RDS.20240808 für RDS für MySQL	Die RDS Extended Support Version 5.7.44-RDS.20240808 ist jetzt für RDS for MySQL verfügbar. Weitere Informationen finden Sie unter Amazon RDS Extended Support-Versionen für RDS for MySQL.	29. August 2024
Amazon RDS ist in der Region Asien-Pazifik (Malaysia) verfügbar	Amazon RDS ist jetzt in der Region Asien-Pazifik (Malaysia) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	22. August 2024
Amazon RDS unterstützt MySQL 8.0.39	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL Version 8.0.39 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	12. August 2024

Aktualisierung der bestehenden Richtlinie	Amazon RDS hat der `AmazonRDSPreviewServiceRolePolicy` `AWSServiceRoleForRDSPreview` serviceverknüpften Rolle die `sns:Publish` Berechtigung entzogen. Weitere Informationen finden Sie unter Amazon RDS-Updates für AWS verwaltete Richtlinien.	7. August 2024
Aktualisierung der bestehenden Richtlinie	Amazon RDS hat der `AmazonRDSBetaServiceRolePolicy` `AWSServiceRoleForRDSBeta` serviceverknüpften Rolle die `sns:Publish` Berechtigung entzogen. Weitere Informationen finden Sie unter Amazon RDS-Updates für AWS verwaltete Richtlinien.	7. August 2024
Amazon RDS unterstützt MySQL 8.4 in der Database Preview-Umgebung	MySQL 8.4 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL Version 8.4 in der Database Preview-Umgebung.	1. August 2024

Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die `AmazonRDSCustomServiceRolePolicy` Richtlinie gewährt jetzt zusätzliche Berechtigungen, um mit Amazon RDS-Diensten in einem anderen zu kommunizieren AWS-Region und EC2 Bilder zu kopieren. Weitere Informationen finden Sie unter Amazon RDS-Updates für AWS verwaltete Richtlinien.	18. Juli 2024
Amazon RDS unterstützt MariaDB 11.4 in der Database Preview-Umgebung	MariaDB 11.4 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informationen finden Sie unter MariaDB Version 11.4 in der Database Preview-Umgebung.	18. Juli 2024
AWS ODBC-Treiber für MySQL allgemein verfügbar	Der Amazon Web Services (AWS) ODBC-Treiber für MySQL ist ein Client-Treiber, der für die hohe Verfügbarkeit von RDS für MySQL entwickelt wurde. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu RDS für MySQL mit dem Amazon Web Services (AWS) ODBC-Treiber für MySQL.	18. Juli 2024

Aktualisierung der bestehenden Richtlinie	Amazon RDS hat der `AmazonRDSServiceRolePolicy` `AWSServiceRoleForRDS` serviceverknüpften Rolle die `sns:Publish` Berechtigung entzogen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: Amazon RDSService RolePolicy.	2. Juli 2024
AWS Marketplace privates Angebot für Db2	AWS Marketplace unterstützt jetzt private Angebote für eine Db2-Lizenz bis hin AWS Marketplace zu Amazon RDS for Db2. Weitere Informationen finden Sie unter Privates Angebot einholen.	1. Juli 2024
Multi-AZ-DB-Cluster-Snapshot-Daten nach Amazon S3 exportieren	Sie können jetzt Multi-AZ-DB-Cluster-Snapshot-Daten nach Amazon S3 exportieren. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3.	27. Juni 2024

Amazon RDS for Oracle unterstützt vorkonfigurierte, speicheroptimierte R6i-Instance-Klassen	Die db.r6i Oracle-DB-Instance-Klassen sind für Workloads optimiert, die zusätzlichen Arbeitsspeicher und Speicherplatz pro vCPU benötigen. I/O Beispielsweise ist für db.r6i.8xlarge.tpc2.mem4x Multithreading aktiviert und bietet viermal so viel Speicher wie db.r6i.8xlarge. Weitere Informationen finden Sie unter RDS for Oracle-DB-Instance-Klassen.	21. Juni 2024
Amazon RDS Extended Support Version 5.7.44-RDS.20240529 für RDS für MySQL	Die RDS Extended Support Version 5.7.44-RDS.20240529 ist jetzt für RDS for MySQL verfügbar. Weitere Informationen finden Sie unter Amazon RDS Extended Support-Versionen für RDS for MySQL.	20. Juni 2024
Amazon RDS unterstützt MySQL 8.0.37	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL Version 8.0.37 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	18. Juni 2024

Amazon RDS unterstützt MariaDB 10.11.8, 10.6.18, 10.5.25 und 10.4.34	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 10.11.8, 10.6.18, 10.5.25 und 10.4.34 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	14. Juni 2024
Amazon RDS beendet die Unterstützung für die DB-Instance-Klassen db.m4, db.r4 und db.t2	Für die DB-Engines RDS for MariaDB, RDS for MySQL und RDS for PostgreSQL können Sie keine DB-Instances mehr erstellen, die die Instance-Klassen db.m4, db.r4 und db.t2 verwenden. RDS aktualisiert automatisch bestehende DB-Instances, die diese Klassen verwenden, auf eine neuere Generation. Weitere Informationen finden Sie unter DB-Instance-Klassenarten.	4. Juni 2024
Multi-AZ-DB-Cluster sind zusätzlich erhältlich AWS-Regionen	In mehreren Fällen können Sie Multi-AZ-DB-Cluster erstellen. AWS-Regionen Eine Tabelle mit allen unterstützten Regionen finden Sie unter Unterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS.	29. Mai 2024

AWS Python-Treiber allgemein verfügbar	Der Amazon Web Services (AWS) Python-Treiber ist als fortschrittlicher Python-Wrapper konzipiert. Dieser Wrapper ergänzt den Open-Source-Treiber Psycopg und erweitert dessen Funktionalität. Weitere Informationen finden Sie unter Mit den Treibern eine Verbindung zu DB-Instances herstellen. AWS	23. Mai 2024
RDS Proxy ist in mehr Regionen verfügbar	RDS Proxy ist jetzt in den Regionen Asien-Pazifik (Hyderabad), Asien-Pazifik (Melbourne), Naher Osten (VAE), Israel (Tel Aviv), Kanada West (Calgary) und Europa (Zürich) verfügbar . Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	21. Mai 2024
Db2-Lizenz bis AWS Marketplace	Wenn die Db2-Lizenz abgeschlossen ist AWS Marketplace, können Sie jetzt einen Stundensatz zahlen, um Db2-Lizenzen für Amazon RDS for Db2 zu abonnieren. Weitere Informationen finden Sie unter Amazon RDS for Db2-Lizenzierungsoptionen.	21. Mai 2024

Amazon RDS unterstützt differenzierten Zugriff für Performance Insights	Sie können jetzt den Zugriff auf einzelne Dimensionen in Performance Insights zulassen oder verweigern. Dieser fein abgestufte Zugriff kann für `GetResourceMetrics` `DescribeDimensionKeys` , und `GetDimensionKeyDetails` Aktionen verwendet werden. Weitere Informationen finden Sie unter Granularzugriff für Performance Insights gewähren.	21. Mai 2024
Amazon RDS-Versionen mit erweitertem Support für RDS für MySQL	Sie können sich alle Versionen von RDS Extended Support for RDS for MySQL ansehen. Weitere Informationen finden Sie unter Amazon RDS Extended Support-Versionen für RDS for MySQL.	16. Mai 2024
Amazon RDS unterstützt MySQL 8.3 in der Database Preview-Umgebung	MySQL 8.3 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL Version 8.3 in der Database Preview-Umgebung.	30. April 2024

Amazon RDS for Db2 unterstützt Zeitzonen	RDS for Db2 unterstützt jetzt die Einstellung lokaler Zeitzonen für neue RDS für Db2-DB-Instances. Weitere Informationen finden Sie unter Lokale Zeitzonen für Amazon RDS für Db2-DB-Instances.	25. April 2024
Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die `AmazonRDSCustomServiceRolePolicy` Richtlinie gewährt jetzt zusätzliche Berechtigungen, um einer benutzerdefinierten RDS-Instance eine Servicerolle als Instance-Profil zuzuordnen. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	19. April 2024
Amazon RDS for Oracle unterstützt den Oracle Data Guard-Switchover in allen Bereichen AWS-Regionen	Sie können Oracle Data Guard Switchover jetzt in allen unterstützten Regionen verwenden. Weitere Informationen finden Sie unter Überblick über Oracle Data Guard Switchover.	16. April 2024

RDS Custom für Oracle unterstützt Oracle Standard Edition 2	Sie können jetzt DB-Instances mit Standard Edition 2 auf Oracle Database 12c Release 1 (12.1), 12c Release 2 (12.2), 18c und 19c erstellen. Sie können sowohl als auch nicht-erstellen. CDBs CDBs Weitere Informationen finden Sie unter Editions- und Lizenzierungsunterstützung für RDS Custom for Oracle.	11. April 2024
Amazon RDS for Oracle unterstützt Oracle APEX Version 23.2.v1	Sie können Oracle APEX 23.2.v1 mit Oracle Database 19c und höher verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	11. April 2024
Aktualisierung der benutzerspezifischen RDS-Rollenberechtigungen für serviceverknüpfte Rollen	Das gewährt `AmazonRDSCustomServiceRolePolicy` jetzt zusätzliche Berechtigungen, damit RDS Custom for SQL Server Informationen zum EC2 Instance-Typ abrufen und den DB-Host-Instance-Typ ändern kann. Weitere Informationen finden Sie unter Updates für AWS verwaltete Richtlinien.	8. April 2024

Amazon RDS Custom for Oracle unterstützt die DB-Instance-Klasse db.x2iezn	Sie können jetzt die db.x2iezn -Instance-Klasse für RDS Custom für Oracle-DB-Instances verwenden. Weitere Informationen finden Sie unter Support für DB-Instance-Klassen für RDS Custom für Oracle.	26. März 2024
Amazon RDS unterstützt die db.c6gd-Instance-Klassen für Multi-AZ-DB-Cluster	Sie können jetzt die db.c6gd-Instance-Klassen für Multi-AZ-DB-Cluster-Bereits tellungen verwenden. Weitere Informationen finden Sie unter Verfügbarkeit von Instance-Klassen für Multi-AZ-DB-Cluster.	21. März 2024
Amazon RDS Extended Support	Beim Erstellen oder Wiederher stellen einer RDS for MySQL 5.7- oder RDS for PostgreSQ L 11-Datenbank wird diese Datenbank jetzt automatisch bei Amazon RDS Extended Support registriert, sodass Ihre vorhandenen Anwendung en weiterhin so funktionieren, wie sie sind. Sie können sich vom RDS Extended Support abmelden, um Gebühren nach Ablauf des RDS-Standard-Supports für Ihre Datenbank-Engine zu vermeiden. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Extended Support.	21. März 2024

Integration von RDS für Db2 mit AWS License Manager	RDS für Db2 ist jetzt in integriert. AWS License Manager Wenn Sie das Modell „Bring Your Own License“ (BYOL) verwenden , hilft Ihnen die AWS License Manager Integration dabei, die Nutzung Ihrer Db2-Lizen z in Ihrem Unternehmen zu überwachen. Weitere Informati onen finden Sie unter Integrier en mit. AWS License Manager	20. März 2024
Rotation der CA-Zertifikate für Multi-AZ-DB-Cluster	Sie können jetzt die CA-Zertif ikate für Ihre Multi-AZ-DB-Cluster rotieren. Erwägen Sie die Verwendung eines der neuen CA-Zertifikate rds-ca-rsa 2048-g1, rds-ca-rsa 4096-g1 oder 384-g1. rds-ca-ecc Weitere Informationen finden Sie unter Rotation Ihres SSL/TLS Zertifikats.	6. März 2024
Amazon RDS unterstützt io2 Block Express-Speicher	Sie können jetzt RDS-DB-Instances erstellen, die den Speichertyp io2 Block Express verwenden. Weitere Informati onen finden Sie unter io2 Block Express-Speicher.	6. März 2024

RDS Custom for SQL Server unterstützt die DB-Instance-Klassen db.r5b und db.x2iedn	Sie können jetzt die Instanzklassen db.r5b und db.x2iedn für RDS Custom for SQL Server-DB-Instances verwenden. Weitere Informationen finden Sie unter Unterstützung von DB-Instance-Klassen für RDS Custom for SQL Server.	4. März 2024
RDS Custom for Oracle ist in der Region Naher Osten (VAE) verfügbar	Sie können RDS Custom für Oracle-DB-Instances in der Region Naher Osten (VAE) erstellen. Eine Tabelle mit allen unterstützten AWS-Regionen Regionen und DB-Engines finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	4. März 2024
Neue AWS verwaltete Richtlinie	Amazon RDS hat eine neue verwaltete Richtlinie hinzugefügt `AmazonRDSCustomInstanceProfileRolePolicy`, mit der RDS Custom Automatisierungsaktionen und Datenbankverwaltungsaufgaben über ein EC2 Instance-Profil ausführen kann. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	27. Februar 2024

Amazon RDS unterstützt MariaDB 10.11.7, 10.6.17, 10.5.24 und 10.4.33	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 10.11.7, 10.6.17, 10.5.24 und 10.4.33 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	26. Februar 2024
Amazon RDS Multi-AZ-DB-Cluster unterstützen das Amazon EBS gp3-Speichervolume	Multi-AZ-DB-Cluster unterstützen jetzt GP3-SSD-basierte EBS-Volumes. Weitere Informationen finden Sie unter GP3-Speicher.	26. Februar 2024
Amazon RDS-Unterstützung für AWS Secrets Manager die Region Israel (Tel Aviv)	Amazon RDS unterstützt Secrets Manager in der Region Israel (Tel Aviv). Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS und AWS Secrets Manager.	21. Februar 2024
Amazon RDS for Db2 unterstützt die Audit-Protokollierung	RDS for Db2 unterstützt jetzt die Audit-Protokollierung auf Datenbankebene. Wenn Sie die Audit-Protokollierung für eine RDS for Db2-Datenbank aktivieren, zeichnet Amazon RDS die Datenbankaktivität auf und speichert die Audit-Logs in Amazon S3. Weitere Informationen finden Sie unter Db2-Auditprotokollierung.	15. Februar 2024

Amazon RDS Extended Support	Amazon RDS aktiviert jetzt automatisch Amazon RDS Extended Support, wenn die wichtigsten Engine-Versionen von RDS for MySQL und RDS for PostgreSQL in Ihren DB-Instances und Multi-AZ-DB-Clustern das Ende des Standard-Supports für RDS erreichen. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Extended Support.	15. Februar 2024
Amazon RDS unterstützt MySQL 8.0.36	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL Version 8.0.36 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	12. Februar 2024
Amazon RDS unterstützt EBCDIC-Kollation für RDS für Db2	Sie können jetzt Db2-Datenbanken erstellen, die EBCDIC-Kollatierungssequenzen verwenden, um Inhalte in den Datenbanken zu sortieren. Weitere Informationen finden Sie unter EBCDIC-Kollation für Db2-Datenbanken auf Amazon RDS.	29. Januar 2024

Aktualisieren Sie auf das Standard-CA-Zertifikat	Das Standard-CA-Zertifikat ist auf festgelegt. `rds-ca-rsa2048-g1` . Weitere Informationen finden Sie unter Verwenden, SSL/TLS um eine Verbindung zu einer DB-Instance zu verschlüsseln.	26. Januar 2024
Amazon RDS for PostgreSQL unterstützt zwei neue Crates für PL/Rust, croaring-rs und num-bigint	Sie können zwei neue Crates in Amazon RDS for PostgreSQL verwenden. Weitere Informationen finden Sie unter Verwenden von Crates mit PL/Rust.	24. Januar 2024
Amazon RDS for PostgreSQL unterstützt TLS Version 1.3	Sie können Transport Layer Security (TLS) Version 1.3 in RDS für PostgreSQL verwenden. Weitere Informationen finden Sie unter Verwenden von SSL mit einer PostgreSQL-DB-Instance.	24. Januar 2024
RDS Custom für SQL Server unterstützt Microsoft SQL Server 2022	Sie können jetzt RDS Custom für SQL Server-DB-Instances erstellen, die SQL Server 2022 verwenden. Weitere Informationen finden Sie unter Arbeiten mit RDS Custom für SQL Server.	22. Januar 2024

Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Die Rolle, `AmazonRDSServiceRolePolicy` die mit dem `AWSServiceRoleForRDS` Dienst verknüpft ist, hat eine neue Erklärung IDs. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	19. Januar 2024
RDS Custom for Oracle unterstützt die Region Europa (Paris)	Sie können RDS Custom für Oracle-DB-Instances in der Region Europa (Paris) erstellen. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	18. Januar 2024
Amazon RDS for MySQL unterstützt die Replikation mehrerer Quellen	Sie können jetzt Multisource-Replikation auf RDS für MySQL-DB-Instances verwenden. Weitere Informationen finden Sie unter Konfiguration der Multiquellenreplikation auf RDS for MySQL.	16. Januar 2024
Amazon RDS unterstützt MySQL 8.2 in der Database Preview-Umgebung	MySQL 8.2 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL Version 8.2 in der Database Preview-Umgebung.	11. Januar 2024

RDS Proxy ist in der Region Europa (Spanien) verfügbar	RDS Proxy ist jetzt in der Region Europa (Spanien) verfügbar. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	8. Januar 2024
Amazon RDS ist in der Region Kanada West (Calgary) verfügbar	Amazon RDS ist jetzt in der Region Kanada West (Calgary) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	20. Dezember 2023
Amazon RDS for Db2 unterstützt 5.000 lokale Benutzer	Sie können jetzt bis zu 5.000 lokale Benutzer zu einer Autorisierungsliste hinzufügen. Weitere Informationen finden Sie unter rdsadmin.add_user.	20. Dezember 2023
Amazon RDS unterstützt das Anzeigen und Beantworten von Empfehlungen	Amazon RDS-Empfehlungen umfassen jetzt schwellenwertbasierte proaktive und auf maschinellem Lernen basierende reaktive Empfehlungen für RDS for PostgreSQL. Weitere Informationen finden Sie unter Empfehlungen von Amazon RDS.	19. Dezember 2023

Amazon RDS unterstützt MariaDB 10.11.6, 10.6.16, 10.5.23 und 10.4.32	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB-Versionen 10.11.6, 10.6.16, 10.5.23 und 10.4.32 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	12. Dezember 2023
Amazon RDS führt Zero-ETL-Integrationen mit Amazon Redshift ein (Vorschau)	Zero-ETL-Integrationen bieten eine vollständig verwaltete Lösung, mit der Transaktionsdaten innerhalb von Sekunden nach dem Schreiben in eine RDS for MySQL-DB-Instance in Amazon Redshift verfügbar gemacht werden. Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS Zero-ETL-Integrationen mit Amazon Redshift (Vorschau).	28. November 2023
Amazon RDS unterstützt IBM Db2 Datenbank-Engines	Sie können jetzt IBM Db2 Datenbank-Engines in Amazon RDS ausführen. Weitere Informationen finden Sie unter Amazon RDS for Db2.	8. November 2023

RDS for PostgreSQL unterstützt Hauptversions-Upgrades auf PostgreSQL 16.1 und Nebenversions-Upgrades auf 15.5, 14.10, 13.13, 12.17 und 11.22	Mit RDS for PostgreSQL können Sie jetzt die DB-Engine auf die Hauptversion 16.1 und die Nebenversions-Upgrades auf 15.5, 14.10, 13.13, 12.17 und 11.22 aktualisieren. Weitere Informationen finden Sie unter Upgrade der PostgreSQL-DB-Engine für Amazon RDS.	17. November 2023
RDS Custom for Oracle unterstützt Optionsgruppen	Sie können eine Optionsgruppe erstellen oder ändern und sie einer RDS Custom for Oracle DB-Instance zuordnen. Die `Timezone` Option wird jetzt unterstützt. Weitere Informationen finden Sie unter Arbeiten mit Optionsgruppen in RDS Custom for Oracle.	17. November 2023
Amazon RDS for MySQL unterstützt das Group Replication Plugin	Sie können jetzt einen Active-Active-Cluster mit DB-Instances von RDS für MySQL Version 8.0.35 oder höher einrichten, indem Sie das von der MySQL-Community entwickelte und verwaltete Group Replication-Plugin verwenden. Weitere Informationen finden Sie unter Konfiguration von aktiv-aktiven Clustern für RDS for MySQL.	17. November 2023

Amazon RDS Proxy unterstützt RDS für PostgreSQL 16.1	Sie können jetzt Proxys mit RDS Proxy for RDS for PostgreSQL 16.1 DB-Instances erstellen. Weitere Informationen finden Sie unter Amazon RDS Proxy.	17. November 2023
RDS Custom für SQL Server unterstützt die Microsoft SQL Server 2019 Developer Edition	Sie können RDS Custom für SQL Server-DB-Instances erstellen, die die SQL Server 2019 Developer Edition verwenden. Weitere Informationen finden Sie unter Bring Your Own Media mit RDS Custom für SQL Server.	16. November 2023
Upgrades kleinerer Versionen von Multi-AZ-DB-Clustern mit minimaler Ausfallzeit	Wenn Sie ein kleineres Versions-Upgrade eines Multi-AZ-DB-Clusters durchführen, aktualisiert Amazon RDS jetzt die Reader-DB-Instances vor der Writer-Instance, wodurch Ausfallzeiten erheblich reduziert werden. Mithilfe von RDS Proxy können Sie die Ausfallzeit weiter auf eine Sekunde oder weniger reduzieren. Weitere Informationen finden Sie unter Upgrade der Engine-Version eines Multi-AZ-DB-Clusters.	16. November 2023

RDS für SQL Server unterstützt Microsoft SQL Server 2022	Sie können jetzt RDS-DB-Instances erstellen, die SQL Server 2022 verwenden. Weitere Informationen finden Sie unter Microsoft SQL Server-Versionen auf Amazon RDS.	15. November 2023
RDS for MySQL unterstützt das Upgrade von Snapshots von Version 5.7 auf 8.0	Sie können jetzt die Engine-Version eines RDS for MySQL-Snapshots von Version 5.7 auf Version 8.0 aktualisieren. Sie können dies tun, indem Sie die AWS Management Console, oder den `ModifyDBSnapshot` Betrieb der RDS-API oder verwenden AWS CLI. Weitere Informationen finden Sie unter Upgraden einer MySQL-DB-Snapshot-Engine-Version.	15. November 2023
RDS Custom for SQL Server unterstützt die Point-in-Time-Wiederherstellung von 1.000 Datenbanken	Sie können jetzt bis zu 1.000 Datenbanken auf Ihrer RDS Custom for SQL Server-DB-Instance einrichten, die für ein vollständiges Backup und eine Point-in-Time-Wiederherstellung in Frage kommen. Weitere Informationen finden Sie unter Wiederherstellen einer RDS Custom for SQL Server-Instanz auf einen bestimmten Zeitpunkt.	15. November 2023

RDS Custom for SQL Server unterstützt die Verwendung eines Service Master Keys	RDS Custom for SQL Server unterstützt jetzt die Verwendung eines Service Master Key (SMK). Ein SMK ermöglicht es Ihnen, Objekte wie Anmeldeinformationen zu verschlüsseln und SQL Server-Funktionen wie TDE und Spaltenverschlüsselung zu verwenden. Weitere Informationen finden Sie unter Verwenden eines Service-Hauptschlüssels mit RDS Custom für SQL Server.	13. November 2023
Amazon RDS unterstützt MySQL 8.1 in der Datenbank-Preview-Umgebung	MySQL 8.1 ist jetzt in der Database Preview-Umgebung im Osten der USA (Ohio) verfügbar AWS-Region. Weitere Informationen finden Sie unter MySQL-Version 8.1 in der Datenbank-Preview-Umgebung.	10. November 2023
RDS unterstützt MySQL 8.0.35 und MySQL 5.7.44	Sie können jetzt DB-Instances von Amazon RDS erstellen, auf denen die MySQL-Versionen 8.0.35 und 5.7.44 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	9. November 2023

RDS-Proxy unterstützt Multi-AZ-DB-Cluster	RDS-Proxy unterstützt jetzt die Verbindung zu Multi-AZ-DB-Clustern. Weitere Informationen finden Sie unter Arbeiten mit Amazon-RDS-Proxy-Endpunkten .	9. November 2023
RDS Custom for Oracle ist verfügbar in AWS GovCloud (US) Regions	Amazon RDS ist jetzt in den AWS GovCloud (US) Regions verfügbar. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	9. November 2023
Amazon-RDS-optimierte Schreibvorgänge unterstützt die DB-Instance-Klasse db.m5	Amazon-RDS-optimierte Schreibvorgänge unterstützt jetzt die DB-Instance-Klasse db.m5. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB und Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MySQL.	9. November 2023

Amazon RDS für Oracle unterstützt die Multi-Tenant-Konfiguration der CDB-Architektur	Mit dem Multi-Tenant-Feature von RDS für Oracle bietet RDS eine vollständig verwaltete Oracle-Multitenant-Architektur und -Umgebung für Ihre Oracle-Datenbanken. Sie können RDS verwenden PDBs, APIs um mehrere sogenannte Tenant-Datenbanken in einer CDB zu erstellen. RDS bietet die Multi-Tenant-Konfiguration der CDB-Architektur als Alternative zur älteren Single-Tenant-Konfiguration. Weitere Informationen finden Sie unter Multi-Tenant-Konfiguration der CDB-Architektur.	8. November 2023

Amazon RDS exportiert Performance Insights Insights-Metriken nach Amazon CloudWatch	Mit Performance Insights können Sie die vorkonfig urierten oder benutzerd efinierten Metrik-Dashboards nach Amazon exportieren. CloudWatch Die exportier ten Metrik-Dashboards können in der Konsole angezeigt werden. CloudWatc h Sie können auch ein ausgewähltes Performan ce Insights Insights-Metrik-Widget exportieren und die Metrikdaten in der CloudWatc h Konsole anzeigen. Weitere Informationen finden Sie unter Performance Insights Insights-Metriken exportieren nach CloudWatch.	8. November 2023
Amazon RDS Custom für Oracle ermöglicht Ihnen, das Betriebssystem auf einer DB-Instance zu aktualisieren	Sie können jetzt die Datenbank oder das Betriebss ystem (OS) für eine RDS Custom für Oracle-DB-Instance mithilfe des CLI-Befehls `modify-db-instance` aktualisieren. Weitere Informationen finden Sie unter Upgrade einer DB-Instance für Amazon RDS Custom für Oracle.	7. November 2023

RDS-Proxy unterstützt das erweiterte Protokoll für RDS für PostgreSQL	Sie können jetzt erweiterte Abfrageprotokolle auf einer RDS für PostgreSQL-DB-Instance ausführen. Weitere Informationen finden Sie unter Amazon RDS Proxy.	6. November 2023
Unterstützung von RDS für PostgreSQL für RDS-Bereitstellungen Blue/Green	Sie können jetzt eine blue/green Bereitstellung aus einer RDS for PostgreSQL-DB-Instance erstellen. Weitere Informationen finden Sie unter Verwenden von Amazon Blue/Green RDS-Bereitstellungen für Datenbank-Updates.	26. Oktober 2023
Aktualisierung der AWS verwalteten Richtlinien	Die von `AmazonRDSPerformanceInsightsReadOnly` und `AmazonRDSPerformanceInsightsFullAccess` verwalteten Richtlinien enthalten jetzt `Sid` (Statement-ID) als Bezeichner in der Richtlinienerklärung. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	23. Oktober 2023
RDS Custom für Oracle unterstützt die Region Europa (Mailand)	Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	23. Oktober 2023

Aktivieren RDS-optimierter Schreibvorgänge für bestehende Datenbanken	Sie können jetzt RDS-optimierte Schreibvorgänge für eine bestehende DB-Instance aktivieren, auch wenn diese mit einer Engine-Version, DB-Instance-Klasse oder Dateisystemkonfiguration erstellt wurde, die diese Funktion nicht unterstützt. Weitere Informationen finden Sie unter Aktivieren von RDS-optimierten Schreibvorgängen auf einer vorhandenen Datenbank für RDS für MySQL und Aktivieren von RDS-optimierten Schreibvorgängen auf einer vorhandenen Datenbank für RDS für MariaDB.	19. Oktober 2023
Amazon RDS unterstützt die Verwendung eines dedizierten Protokoll-Volumes (DLV).	Sie können jetzt ein dediziertes Log-Volume (DLV) mit RDS für MariaDB, RDS für MySQL und RDS für PostgreSQL verwenden. DLVs sind ideal für Datenbanken mit großem zugewiesenem Speicher, hohen I/O-pro-Sekunden-Anforderungen (IOPS) oder latenzsensitiven Workloads. Weitere Informationen finden Sie unter Verwenden eines dedizierten Protokoll-Volumes (DLV).	17. Oktober 2023

Amazon RDS für PostgreSQL, MySQL und MariaDB unterstützen neue DB-Instance-Klassen	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen PostgreSQL, MySQL und MariaDB mit den DB-Instance-Klassen db.m6.in, db.m6idn, db.r6.in und db.r6.idn ausgeführt werden. Weitere Informationen finden Sie unter Unterstützte DB-Engines für alle verfügbaren DB-Instance-Klassen.	12. Oktober 2023
Amazon RDS für PostgreSQL unterstützt „pgactive“	Die „pgactive“-Erweiterung ist in Amazon RDS für PostgreSQL verfügbar. Weitere Informationen finden Sie unter Verwenden von PostgreSQL-Erweiterungen mit Amazon RDS für PostgreSQL.	09. Oktober 2023
RDS Custom für Oracle ist in der Region Asien-Pazifik (Jakarta) verfügbar	Sie können RDS Custom für Oracle-DB-Instances in der Region Asien-Pazifik (Jakarta) erstellen. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	05. Oktober 2023

RDS Custom für SQL Server unterstützt neue Sortierungen auf Serverebene	RDS Custom für SQL Server unterstützt jetzt eine Vielzahl von Serversortierungen , sowohl in traditioneller als auch in UTF-8-Kodierung, für die Gebietsschemas SQL_Latin, Japanisch , Deutsch und Arabisch. Weitere Informationen finden Sie unter Sortierungs- und Zeichenunterstützung für DB-Instances von RDS Custom für SQL Server.	26. September 2023
Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Die Rolle `AmazonRDSCustomServiceRolePolicy` der `AWSServiceRoleForRDSCustom` serviceverknüpften Rolle verfügt über neue Berechtigungen, die es RDS Custom ermöglichen, EventBridge verwaltete Regeln zu erstellen , zu ändern und zu löschen. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	20. September 2023

Amazon RDS veröffentlicht Performance Insights Insights-Zählermetriken auf Amazon CloudWatch	Die metrische Rechenfunktion DB_PERF_INSIGHTS in der CloudWatch Konsole ermöglicht es Ihnen, Amazon RDS nach Performance Insights Insights-Zählermetriken abzufragen. Weitere Informationen finden Sie unter CloudWatch Alarme zur Überwachung von Amazon RDS erstellen.	20. September 2023
Performance Insights unterstützt Statistiken auf Digest-Ebene für SQL Server	Wenn Sie Performance Insights verwenden, können Sie SQL-Statistiken sowohl auf Anweisung- als auch auf Digest-Ebene für Amazon RDS für SQL Server anzeigen. Weitere Informationen finden Sie unter Analysieren von laufenden Abfragen in SQL Server.	18. September 2023

Amazon RDS für PostgreSQL, MySQL und MariaDB unterstützen die DB-Instance-Klassentypen db.m6.id und db.r6.id	Sie können jetzt DB-Instances von Amazon RDS erstellen, die PostgreSQL, MySQL und MariaDB ausführen, bei denen die arbeitsspeicheroptimierten DB-Instance-Klassentypen db.m6.id und db.r6.id verwendet werden. Diese Typen bieten lokalen NVMe SSD-Speicher. Weitere Informationen finden Sie unter Unterstützte DB-Engines für alle verfügbaren DB-Instance-Klassen.	11. September 2023
Unterstützung von Hauptversions-Upgrades für Multi-AZ-DB-Cluster von RDS für PostgreSQL	Sie können jetzt Hauptversions-Upgrades Ihrer Multi-AZ-DB-Cluster von RDS für PostgreSQL durchführen. Weitere Informationen finden Sie unter Upgrade der Engine-Version eines Multi-AZ-DB-Clusters.	07. September 2023
Amazon RDS unterstützt MariaDB 10.11.5, 10.6.15, 10.5.22 und 10.4.31	Sie können jetzt DB-Instances von Amazon RDS erstellen, die MariaDB Version 10.11.5, 10.6.15, 10.5.22 und 10.4.31 ausführen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	07. September 2023

Amazon RDS Extended Support	Amazon RDS kündigt die bevorstehende Möglichkeit an, die Engine-Hauptversionen von RDS für MySQL und RDS für PostgreSQL in Ihren DB-Instances auch nach dem Ende des Standard-Supports von RDS weiter auszuführen. Weitere Informationen finden Sie unter Verwenden von Amazon RDS Extended Support.	1. September 2023
RDS Custom unterstützt das Starten und Anhalten einer DB-Instance von RDS Custom für SQL Server	RDS Custom unterstützt jetzt das Starten und Anhalten einer DB-Instance von RDS Custom für SQL. Weitere Informationen finden Sie unter Eine DB-Instance von RDS Custom für SQL Server starten und anhalten.	31. August 2023
Amazon-RDS-optimierte Schreibvorgänge unterstützt die DB-Instance-Klasse db.r5	Amazon-RDS-optimierte Schreibvorgänge unterstützt jetzt die DB-Instance-Klasse db.r5. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB und Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MySQL.	31. August 2023

Amazon RDS for Oracle unterstützt das automatische Upgrade von Zeitzonendateien für CDBs	Mit der Option `TIMEZONE_FILE_AUTOUPGRADE` können Sie die aktuelle Zeitzonendatei auf die neueste Version Ihrer Container-Datenbank (CDB) von RDS für Oracle aktualisieren. Weitere Informationen finden Sie unter Autoupgrade der Oracle-Zeitzonendatei.	29. August 2023
Amazon-RDS-optimierte Schreibvorgänge unterstützt die DB-Instance-Klassen db.m6g und db.m6i	Amazon-RDS-optimierte Schreibvorgänge unterstützt jetzt die DB-Instance-Klassen db.m6g und db.m6i. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB und Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MySQL.	28. August 2023
Amazon RDS unterstützt MariaDB 10.11	Sie können jetzt DB-Instances von Amazon RDS erstellen, die MariaDB Version 10.11 ausführen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	21. August 2023

Aktualisierung der verwalteten Richtlinienberechtigungen AWS	Die `AmazonRDSCustomServiceRolePolicy` der serviceverknüpften Rolle `AWSServiceRoleForRDSCustom` verfügt über neue Berechtigungen, die es RDS Custom erlauben, Netzwerkschnittstellen zu erstellen. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	18. August 2023
Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Die `AmazonRDSFullAccess` -verwaltete Richtlinie umfasst neue Berechtigungen, mit denen Sie den Leistungsanalysebericht für einen bestimmten Zeitraum erstellen, anzeigen und löschen können. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	17. August 2023

Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Das Hinzufügen neuer Berechtigungen zur `AmazonRDSPerformanceInsightsReadOnly` -verwalteten Richtlinie und das Hinzufügen einer neuen verwalteten Richtlinie `AmazonRDSPerformanceInsightsFullAccess` ermöglicht es Ihnen, einen DB-Lastanalysebericht für einen bestimmten Zeitraum zu erstellen. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	16. August 2023
Amazon RDS unterstützt Leistungsanalysen für einen bestimmten Zeitraum	Performance Insights ermöglicht Ihnen, einen Leistungsanalysebericht für einen bestimmten Zeitraum zu erstellen und anzuzeigen. Der Bericht enthält die identifizierten Einblicke und Empfehlungen zum Beheben von Leistungsproblemen. Weitere Informationen finden Sie unter Analysieren der DB-Last über einen bestimmten Zeitraum.	16. August 2023

Amazon RDS Custom für Oracle unterstützt die DB-Instance-Klassen db.r5b und db.x2iedn	Sie können jetzt die Instance-Klassen db.r5b und db.x2iedn für DB-Instances von RDS Custom für Oracle verwenden. Weitere Informationen finden Sie unter Support für DB-Instance-Klassen für RDS Custom für Oracle.	16. August 2023
Amazon RDS Custom für Oracle unterstützt die DB-Instance-Klassen db.m6i, db.r6i und db.t3	Sie können jetzt die Instance-Klassen db.m6i, db.r6i und db.t3 für DB-Instances von RDS Custom für Oracle verwenden. Weitere Informationen finden Sie unter Support für DB-Instance-Klassen für RDS Custom für Oracle.	15. August 2023
Amazon RDS für PostgreSQL unterstützt jetzt die PostgreSQL-Version 16 Beta 3 in der Preview-Umgebung der Datenbank	PostgreSQL Version 16 Beta 3 ist jetzt in der Datenbank-Vorschauumgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informationen finden Sie unter Arbeiten in der Database Preview-Umgebung.	11. August 2023
Amazon RDS unterstützt MySQL 8.0.34 und 5.7.43	Sie können jetzt DB-Instances von Amazon RDS erstellen, auf denen die MySQL-Versionen 8.0.34 und 5.7.43 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	9. August 2023

RDS für SQL Server unterstützt die Ansicht der Betriebssystemmetriken für das Standby-Replikat	Sie können jetzt Betriebssystemmetriken für das Standby-Replikat für RDS für SQL Server anzeigen. Weitere Informationen finden Sie unter Anzeigen von Betriebssystemmetriken in der RDS-Konsole.	3. August 2023
RDS für Oracle unterstützt Oracle Data Guard für CDBs	RDS für Oracle unterstützt Data Guard-Read Replicas für die Container-Datenbanken Oracle Database 19c und 21c (). CDBs Sie können Read Replicas in einer CDB genauso wie in einer Nicht-CDB erstellen, verwalten und heraufstufen, indem Sie den vorhandenen RDS verwenden . APIs Weitere Informationen finden Sie unter Mehrmandantenfähige Lesereplikate.	1. August 2023
Amazon RDS ist in der Region Israel (Tel Aviv) verfügbar	Amazon RDS ist jetzt in der Region Israel (Tel Aviv) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	1. August 2023
Amazon RDS unterstützt Oracle APEX Version 23.1.v1	Sie können Oracle APEX 23.1.v1 mit Oracle Database 19c und höher verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	26. Juli 2023

Amazon RDS Custom für Oracle unterstützt eine nicht standardmäßige Oracle-SID	Wenn Sie mit Oracle Database 19c eine DB-Instance von RDS Custom für Oracle erstellen, können Sie einen nicht standardmäßigen Oracle-Systembezeichner (Oracle SID) angeben. Dieser Wert ist auch der Name der CDB. Weitere Informationen finden Sie unter Überlegungen zur Multi-Tenant-Architektur.	21. Juli 2023
RDS für SQL Server unterstützt selbstverwaltetes Active Directory	Sie können jetzt selbstverwaltetes Active Directory verwenden, um Ihre RDS-für-SQL Server-DB-Instances direkt Ihren Microsoft Active Directory (AD)-Domains hinzuzufügen. Selbstverwaltete AD-Domains können sich On-Premises oder in der Cloud befinden. Weitere Informationen finden Sie unter Arbeiten mit selbstverwaltetem Active Directory.	07. Juli 2023

Unterstützung der logischen PostgreSQL-Replikation für Multi-AZ-DB-Cluster	Sie können jetzt die logische PostgreSQL-Replikation mit Ihrem Multi-AZ-DB-Cluster verwenden, um einzelne Tabellen anstelle einer gesamten Datenbank -Instance zu replizieren und zu synchronisieren. Weitere Informationen finden Sie unter Einrichtung der logischen PostgreSQL-Replikation mit Multi-AZ-DB-Clustern für Amazon RDS.	6. Juli 2023
Amazon RDS für PostgreSQL unterstützt jetzt die PostgreSQL-Version 16 Beta 2 in der Preview-Umgebung der Datenbank	PostgreSQL Version 16 Beta 2 ist jetzt in der Datenbank-Vorschauumgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informationen finden Sie unter Arbeiten in der Database Preview-Umgebung.	6. Juli 2023
Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Die `AmazonRDSCustomServiceRolePolicy` der serviceverknüpften Rolle `AWSServiceRoleForRDSCustom` verfügt über neue Berechtigungen, die es RDS Custom für Oracle erlauben, Snapshots zu verwenden. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	23. Juni 2023

RDS unterstützt MariaDB 10.6.14, 10.5.21 und 10.4.30	Sie können jetzt DB-Instances von Amazon RDS erstellen, auf denen die MariaDB-Versionen 10.6.14, 10.5.21 und 10.4.30 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	22. Juni 2023
RDS unterstützt MySQL 8.0.33 und 5.7.42	Sie können jetzt DB-Instances von Amazon RDS erstellen, auf denen die MySQL-Versionen 8.0.33 und 5.7.42 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	15. Juni 2023
RDS unterstützt MariaDB 10.6.13, 10.5.20, 10.4.29 und 10.3.39	Sie können jetzt DB-Instances von Amazon RDS erstellen, auf denen die MariaDB-Versionen 10.6.13, 10.5.20, 10.4.29 und 10.3.39 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	15. Juni 2023

RDS für Oracle unterstützt Transportable Tablespaces	Mithilfe von Transportable Tablespaces können Sie Daten aus einer On-Premises-Oracle-Datenbank in eine DB-Instance von RDS für Oracle migrieren. Diese Methode erfordert keine zusätzlic he Lizenzierung und ist die Migrationstechnik mit den geringsten Ausfallzeiten. Weitere Informationen finden Sie unter Migrieren mithilfe von Oracle Transportable Tablespaces.	15. Juni 2023
Amazon RDS unterstützt RDS Proxy mit RDS für MariaDB Version 10.6	Sie können jetzt einen RDS-Proxy mit einer Datenbank von RDS für MariaDB Version 10.6 erstellen. Weitere Informati onen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	15. Juni 2023
RDS Custom für SQL Server unterstützt Bring Your Own Media (BYOM)	Sie können jetzt eine benutzerdefinierte Engine-Ve rsion (CEV) mit Ihren eigenen SQL-Server-Medien erstellen . Weitere Informationen finden Sie unter Bring Your Own Media mit RDS Custom für SQL Server.	08. Juni 2023

RDS für Oracle kann eine Nicht-CDB von Oracle Database 19c in eine CDB konvertieren	Wenn auf Ihrer DB-Instanz Oracle Database 19c mit der RU von April 2021 oder höher ausgeführt wird, können Sie eine Nicht-CDB in eine CDB (Container-Datenbank) konvertieren. Nachdem Sie die Architektur konvertiert haben, können Sie ein Upgrade Ihrer 19c-CDB auf eine 21c-CDB durchführen. Dieser Schritt ist notwendig, da Sie nicht mit einem einzigen Befehl Ihre Datenbank aktualisieren und die Architektur konvertieren können. Weitere Informationen finden Sie unter Konvertierung einer Nicht-CDB von RDS für Oracle in eine CDB.	31. Mai 2023
Multi-AZ-DB-Cluster sind in den China-Regionen verfügbar	Multi-AZ-DB-Cluster sind jetzt in AWS-Regionen China (Peking) und China (Ningxia) verfügbar. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS.	30. Mai 2023

Unterstützung von Amazon-RDS-optimierten Lesevorgängen für Multi-AZ-DB-Cluster	Amazon-RDS-optimierte Lesevorgänge unterstützen jetzt Multi-AZ-DB-Cluster. Weitere Informationen finden Sie unter Verbesserung der Abfrageleistung für RDS für MySQL mit Amazon-RDS-optimierten Lesevorgängen und Verbesserung der Abfrageleistung für RDS für PostgreSQL mit Amazon-RDS-optimierten Lesevorgängen.	30. Mai 2023
RDS Custom for Oracle unterstützt die Region Asien-Pazifik (Jakarta)	Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	29. Mai 2023
Erstellen eines DB-Instance-Lesereplikats mit einem Multi-AZ-Quell-DB-Cluster von RDS für PostgreSQL	Sie können jetzt ein DB-Instance-Lesereplikat mit einem Multi-AZ-DB-Cluster von RDS für PostgreSQL als Quelle erstellen. Bisher wurde nur RDS für MySQL unterstützt. Weitere Informationen finden Sie unter Erstellen eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster.	24. Mai 2023

Amazon RDS bietet kombinierte Performance Insights und CloudWatch Metriken im Performance Insights Insights-Dashboard	Amazon RDS bietet jetzt eine konsolidierte Ansicht von Performance Insights und CloudWatch Metriken im Performance Insights Insights-Dashboard. Weitere Informationen finden Sie unter Kombinierte Metriken mit dem Performance Insights Insights-Dashboard anzeigen.	24. Mai 2023
Amazon-RDS-optimierte Lesevorgänge sind in den chinesischen Regionen verfügbar	Amazon-RDS-optimierte Lesevorgänge sind jetzt in den AWS-Regionen China (Peking) und China (Ningxia) verfügbar. Weitere Informationen finden Sie unter Verbesserung der Abfrageleistung für RDS für MariaDB mit Amazon-RDS-optimierten Lesevorgängen und Verbesserung der Abfrageleistung für RDS für MySQL mit Amazon-RDS-optimierten Lesevorgängen.	24. April 2023
Amazon RDS-Unterstützung für AWS Secrets Manager die Regionen China	Amazon RDS unterstützt Secrets Manager in den Regionen China (Peking) und China (Ningxia). Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS und AWS Secrets Manager.	20. April 2023

RDS Custom for Oracle unterstützt die Wiederverwendung von AMI IDs für neue CEVs	Wenn Sie eine benutzerdefinierte Engine-Version (CEV) erstellen, verwendet RDS Custom für Oracle standardmäßig das neueste verfügbare Amazon Machine Image (AMI). Jetzt können Sie eine AMI-ID angeben, die in einer früheren CEV verwendet wurde. Weitere Informationen finden Sie unter Erstellen einer CEV.	19. April 2023
Amazon RDS unterstützt das Veröffentlichen von Ereignissen mit Tags für Themen-Subscriber	Amazon RDS-Ereignisbenachrichtigungen, die an Amazon Simple Notification Service (Amazon SNS) oder Amazon gesendet werden, enthalten EventBridge jetzt Ereignis-Tags im Nachrichtentext. Diese Tags liefern Daten über die Ressource, die vom Serviceereignis betroffen war. Weitere Informationen finden Sie unter Tags und Attribute von Amazon-RDS-Ereignisbenachrichtigungen.	17. April 2023
Kaufen von Reserved Instances für einen Multi-AZ-DB-Cluster	Sie können jetzt Reserved DB-Instances für einen Multi-AZ-DB-Cluster erwerben. Weitere Informationen finden Sie unter Reserved DB-Instances für einen Multi-AZ-DB-Cluster.	12. April 2023

Amazon RDS unterstützt die Instance-Klassen db.m7g und db.r7g	Sie können jetzt die Instance-Klassen db.m7g und db.r7g für DB-Instances von RDS für MySQL, RDS für MariaDB und RDS für PostgreSQL verwenden. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	12. April 2023
Aktualisieren auf serviceverknüpfte Rollenberechtigungen für Amazon RDS Custom	Die `AmazonRDSCustomServiceRolePolicy` gewährt nun zusätzliche Berechtigungen, um RDS Custom für SQL Server zu erlauben, Amazon SQS zu verwenden und Snapshots zu erstellen. Weitere Informationen finden Sie unter Aktualisierungen auf von AWS verwaltete Richtlinien.	06. April 2023

Migrieren zu einem Multi-AZ-DB-Cluster von RDS für MySQL mithilfe eines Lesereplikats	Sie können jetzt ein Leserepli kat verwenden, um eine Single-AZ-Bereitstellung oder eine Multi-AZ-Bereitste llung einer DB-Instance von RDS für MySQL zu einer Multi-AZ-Bereitstellung eines DB-Clusters von RDS für MySQL mit geringerer Ausfallzeit migrieren. Weitere Informationen finden Sie unter Migrieren zu einem Multi-AZ-DB-Cluster mithilfe eines Lesereplikats.	06. April 2023
Erstellen eines DB-Instance-Lesereplikats aus einem Multi-AZ-DB-Cluster	Sie können jetzt ein DB-Instan ce-Lesereplikat aus einem Multi-AZ-DB-Cluster erstellen , um über die Rechenkapazität des Quell-Clusters hinaus zu skalieren. Weitere Informati onen finden Sie unter Erstellen eines DB-Instance-Lesere plikats aus einem Multi-AZ-DB-Cluster.	06. April 2023
Amazon RDS Custom für SQL Server unterstützt Multi-AZ	Sie können eine Multi-AZ-Bereitstellung mit RDS Custom für SQL Server erstellen. Weitere Informati onen finden Sie unter Verwalten einer Multi-AZ-Bereitstellung für RDS Custom für SQL Server.	06. April 2023

Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Die `AmazonRDSReadOnlyAccess` Richtlinien `AmazonRDSFullAccess` und gewähren jetzt zusätzliche Berechtigungen, um die Anzeige von Amazon DevOps Guru-Ergebnissen in der RDS-Konsole zu ermöglichen. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	30. März 2023
Amazon RDS unterstützt Oracle APEX Version 22.2.v1	Sie können Oracle APEX 22.2.v1 mit allen unterstützten Versionen von Oracle Database verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	30. März 2023

Amazon DevOps Guru für RDS für PostgreSQL verfügbar	RDS for PostgreSQL warnt Sie vor kürzlich von Amazon Guru entdeckten Anomalien. DevOps Auf der Seite mit den Datenbankdetails der Konsole werden Sie über aktuelle Entwicklungen und Anomalien , die in den letzten 24 Stunden aufgetreten sind, informier t. DevOpsGuru veröffent licht proaktive Einblicke mit Empfehlungen, um Probleme in Ihren RDS für PostgreSQ L-Datenbanken zu beheben, bevor sie voraussichtlich auftreten. Weitere Informati onen finden Sie unter So funktioniert DevOps Guru for RDS.	30. März 2023
RDS Custom unterstützt das Amazon-EBS-Speichervolume gp3	RDS Custom für Oracle und RDS Custom für SQL Server unterstützen beide die SSD-basierten EBS-Volumes io1, gp2 und gp3. Weitere Informationen finden Sie unter Allgemeine Anforderungen für RDS Custom für Oracle und Allgemeine Anforderungen für RDS Custom für SQL Server.	29. März 2023

Aktualisierung der AWS verwalteten Richtlinienberechtigungen	Die `AmazonRDSReadOnlyAccess` Richtlinien `AmazonRDSFullAccess` und gewähren Amazon jetzt zusätzliche Berechtigungen CloudWatch. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	16. März 2023
RDS Proxy ist in den chinesischen Regionen verfügbar	RDS Proxy ist jetzt in den Regionen China (Peking) und China (Ningxia) verfügbar. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	15. März 2023
RDS Proxy ist in der Region Asien-Pazifik (Jakarta) verfügbar	RDS Proxy ist jetzt in der Region Asien-Pazifik (Jakarta) verfügbar. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	08. März 2023
Amazon-RDS-optimierte Schreibvorgänge verbessern die Leistung von Schreibtransaktionen für RDS für MariaDB	Mit Amazon-RDS-optimierten Schreibvorgängen können Sie die Leistung von Schreibtransaktionen für DB-Instances von RDS für MariaDB verbessern. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MariaDB.	7. März 2023

Amazon RDS für PostgreSQL Version 15.2	Zu den neuen Funktionen in Amazon RDS für PostgreSQL 15.2 gehören der SQL-Standardbefehl „MERGE“ für bedingte SQL-Abfragen, Leistungsverbesserungen sowohl für die speicherinterne als auch für die festplattenbasierte Sortierung sowie Unterstützung für zweiphasigen Commit und für die Zeilen-/Spaltenfilterung für die logische Replikation.	27. Februar 2023
RDS Custom für Oracle ist in den Regionen Kanada (Zentral) und Südamerika (São Paulo) verfügbar.	Eine Tabelle mit allen unterstützten AWS-Regionen Regionen und DB-Engines für RDS Custom for Oracle finden Sie unter Unterstützte Regionen und DB-Engines.	22. Februar 2023
Amazon RDS unterstützt regionsübergreifende automatisierte Backups für RDS für MariaDB und RDS für MySQL	Sie können jetzt DB-Snapshots und Transaktionsprotokolle zwischen AWS-Regionen für DB-Instances von RDS für MariaDB und RDS für MySQL replizieren. Weitere Informationen finden Sie unter Replizieren von automatisierten Backups in eine andere AWS-Region.	22. Februar 2023

Amazon RDS für Oracle unterstützt Vorankündigung von automatischen Nebenversions-Upgrades	RDS informiert Sie im Voraus darüber, wann eine neue Nebenversion der RDS für Oracle-Engine verfügbar sein wird. RDS beginnt mit der Planung automatischer Nebenversions-Upgrades Ihrer DB-Instances von RDS für Oracle am Verfügbarkeitsdatum. Weitere Informationen finden Sie unter Vor Planung eines automatischen Nebenversions-Upgrades.	21. Februar 2023

Amazon RDS für SQL Server unterstützt Datenbankaktivitäts-Streams	Sie können jetzt eine SQL-Server-DB-Instance mithilfe von Datenbankaktivitäts-Streams überwachen. Eine SQL-Server-Datenbank-Instance umfasst das von Amazon RDS verwaltete Server-Audit. Sie können die Richtlinien zur Aufzeichnung von Serverereignissen in der Server-Audit-Spezifikation definieren. Sie können eine Datenbank-Audit-Spezifikation erstellen und die Richtlinien für die Aufzeichnung von Datenbankereignissen definieren. Der Aktivitäts-Stream wird erfasst und an Amazon Kinesis übertragen. Von Kinesis aus können Sie den Aktivitäts-Stream zur weiteren Analyse überwachen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mithilfe von Datenbankaktivitäts-Streams.	15. Februar 2023
RDS unterstützt MySQL 8.0.32 und 5.7.41	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen die MySQL-Versionen 8.0.32 und 5.7.41 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	07. Februar 2023

Amazon RDS für Oracle unterstützt neue Verschlüsselungssuiten für SSL	Wenn Sie Oracle Database 19c oder 21c ausführen, können Sie in der SSL-Option sechs neue Verschlüsselungssuiten für RDS für Oracle angeben. Diese Suiten unterstützen FIPS und sind FedRAMP-konform. Weitere Informationen finden Sie unter Oracle Secure Sockets Layer.	3. Februar 2023
Amazon RDS für Oracle unterstützt neue Verschlüsselungssuiten für Oracle Enterprise Manager	Sie können für die OEM-Option vier neue FedRAMP-konforme Verschlüsselungssuiten verwenden. Weitere Informationen finden Sie unter Oracle Management Agent for Enterprise Manager Cloud Control.	3. Februar 2023
RDS für Oracle unterstützt Datenbankaktivitäts-Streams in den Regionen Asien-Pazifik (Hyderabad), Europa (Spanien) und Naher Osten (VAE).	Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für Datenbankaktivitätsstreams in Amazon RDS.	27. Januar 2023

Migrieren zu einem Multi-AZ-DB-Cluster von RDS für PostgreSQL mithilfe eines Lesereplikats	Mithilfe eines Lesereplikats können Sie eine Single-AZ -Bereitstellung oder eine Multi-AZ-Bereitstellung einer DB-Instance von RDS für PostgreSQL zu einer Multi-AZ-Bereitstellung eines DB-Clusters von RDS für PostgreSQL mit geringerer Ausfallzeit migrieren. Weitere Informationen finden Sie unter Migrieren zu einem Multi-AZ-DB-Cluster mithilfe eines Lesereplikats.	23. Januar 2023
Amazon RDS ist in der Region Asien-Pazifik (Melbourne) verfügbar	Amazon RDS ist jetzt in der Region Asien-Pazifik (Melbourne) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	23. Januar 2023
RDS für MariaDB unterstützt das Erzwingen von Verbindungen SSL/TLS	RDS for MariaDB unterstützt jetzt das Erzwingen von SSL/TLS Verbindungen, indem der `require_secure_transport` Parameter auf gesetzt wird. `ON` Weitere Informationen finden Sie unter SSL/TLS Für alle Verbindungen zu einer MariaDB-DB-Instance erforderlich.	19. Januar 2023

Amazon RDS Optimized Reads verbessert die Abfrageleistung für RDS für MariaDB	Mit Amazon RDS Optimized Reads können Sie eine schnellere Abfrageverarbeitung für DB-Instances von RDS für MariaDB erreichen. Weitere Informationen finden Sie unter Verbesserung der Abfrageleistung für RDS für MariaDB mit Amazon RDS Optimized Reads.	11. Januar 2023
Wiederherstellen eines Snapshots des Multi-AZ-DB-Clusters auf einer DB-Instance	Sie können jetzt einen Snapshot eines Multi-AZ-DB-Clusters in einer Single-AZ-Bereitstellung oder Multi-AZ-Bereitstellung der DB-Instance wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellen von einem Snapshot in einem Multi-AZ-DB-Cluster.	10. Januar 2023
Angeben der Zertifizierungsstelle (CA) bei der DB-Instance-Erstellung	Sie können jetzt bei der Erstellung der DB-Instance angeben, welche CA für das Serverzertifikat einer DB-Instance verwendet werden soll. Weitere Informationen finden Sie unter Zertifizierungsstellen.	5. Januar 2023

RDS Custom für SQL Server unterstützt benutzerdefinierte Engine-Versionen	Eine benutzerdefinierte Engine-Version (CEV) für RDS Custom für SQL Server ist ein Amazon Machine Image (AMI) mit vorinstalliertem Microsoft SQL Server. Sie wählen ein Amazon EC2 Windows AMI als Basis-Image und können andere Software auf dem Betriebssystem (OS) installieren. Sie können die Konfiguration des Betriebss ystems und des SQL Servers an Ihre Unternehmensanford erungen anpassen. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierte n Engine-Versionen für RDS Custom für SQL Server.	28. Dezember 2022
Verwenden Sie Amazon Blue/ Green RDS-Bereitstellungen, die zusätzlich verfügbar sind AWS-Regionen	Die Blue/Green Bereitste llungsfunktion ist jetzt in den Regionen China (Peking) und China (Ningxia) verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon Blue/Green RDS-Bereitstellungen für Datenbank -Updates.	22. Dezember 2022

Aktualisieren auf Berechtigungen für serviceverknüpfte IAM-Rollen	Die RDSService RolePolicy Amazon-Richtlinie gewährt jetzt zusätzliche Berechtigungen für AWS Secrets Manager. Weitere Informationen finden Sie unter Änderungen von Amazon RDS an von AWS verwalteten Richtlinien.	22. Dezember 2022
Amazon RDS unterstützt die Umbenennung eines Multi-AZ-DB-Clusters	Sie können jetzt einen Multi-AZ-DB-Cluster umbenennen. Weitere Informationen finden Sie unter Umbenennen von Multi-AZ-DB-Clustern.	22. Dezember 2022
Amazon RDS lässt sich AWS Secrets Manager für die Passwortverwaltung integrieren	Amazon RDS kann das Hauptbenutzerpasswort in Secrets Manager für eine DB-Instance oder einen Multi-AZ-DB-Cluster verwalten. Weitere Informationen finden Sie unter Passwortverwaltung mit Amazon RDS und AWS Secrets Manager.	22. Dezember 2022
Amazon RDS Optimized Writes unterstützt die DB-Instance-Klassen db.r6g und db.r6gd	Amazon RDS Optimized Writes unterstützt jetzt die DB-Instance-Klassen db.r6g und db.r6gd. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon RDS Optimized Writes.	22. Dezember 2022

Amazon RDS Custom für Oracle unterstützt neue AWS-Regionen	Sie können DB-Instances von RDS Custom für Oracle in den Regionen Asien-Pazifik (Seoul) und Asien-Pazifik (Osaka) erstellen. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für RDS Custom for Oracle.	21. Dezember 2022
Amazon RDS on AWS Outposts unterstützt Read Replicas	Sie können jetzt ein Lesereplikat aus einer DB-Instance von RDS on Outposts MySQL oder PostgreSQL erstellen. Weitere Informationen finden Sie unter Erstellen von Lesereplikaten für Amazon RDS on AWS Outposts.	19. Dezember 2022
RDS Custom für Oracle unterstützt das Ändern der DB-Instance-Klasse	Sie können jetzt die Instance-Klasse Ihrer DB-Instance von RDS Custom für Oracle ändern. Weitere Informationen finden Sie unter Ändern der DB-Instance von RDS Custom für Oracle.	16. Dezember 2022
RDS für MySQL und RDS für PostgreSQL unterstützen die DB-Instance-Klassen db.x2iedn	Sie können jetzt die DB-Instance-Klassen db.x2iedn für RDS für MySQL und RDS für PostgreSQL verwenden. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	14. Dezember 2022

Amazon RDS Optimized Writes unterstützt die DB-Instance-Klassen db.x2iedn	Amazon RDS Optimized Writes unterstützt jetzt die DB-Instance-Klassen db.x2iedn. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon RDS Optimized Writes.	14. Dezember 2022
Amazon RDS unterstützt das Kopieren von DB-Optionsgruppen beim Kopieren von DB-Snapshots	Sie können jetzt eine Optionsgruppe AWS-Konten als Teil einer Snapshot-Kopieranforderung auf RDS für Oracle-Datenbanken kopieren. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.	13. Dezember 2022
Amazon RDS unterstützt RDS Proxy mit RDS für PostgreSQL Version 13	Sie können jetzt einen RDS-Proxy mit einer Datenbank von RDS für PostgreSQL Version 14 erstellen. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	13. Dezember 2022
Amazon RDS für Oracle unterstützt die Instance-Klassen db.x2idn, db.x2iedn und db.x2iezn	Sie können jetzt die Instance-Klassen db.x2idn, db.x2iedn und db.x2iezn für DB-Instances von Amazon RDS für Oracle verwenden. Weitere Informationen finden Sie unter Unterstützte DB-Engines für DB-Instance-Klassen und Unterstützte Instance-Klassen von RDS für Oracle).	12. Dezember 2022

DB-Instances von RDS für PostgreSQL unterstützen Trusted Language Extensions für PostgreSQL	Trusted Language Extensions für PostgreSQL ist ein Open-Source-Entwicklungskit, mit dem Sie leistungsstarke PostgreSQL-Erweiterungen erstellen und diese sicher auf Ihrer DB-Instance von RDS für PostgreSQL ausführen können. Weitere Informationen finden Sie unter Arbeiten mit Trusted Language Extensions für PostgreSQL.	30. November 2022
Verwenden Sie Amazon RDS Blue/Green Deployments für Datenbank-Updates	Sie können Änderungen an einer DB-Instance in einer Staging-Umgebung vornehmen und die Änderungen testen, ohne dass sich dies auf Ihre Produktions-DB-Instance auswirkt. Wenn Sie bereit sind, können Sie die Staging-Umgebung zur neuen Produktionsdatenbankumgebung hochstufen, wobei die Ausfallzeit minimal ist. Weitere Informationen finden Sie unter Verwenden von Amazon Blue/Green RDS-Bereitstellungen für Datenbank-Updates.	27. November 2022

Amazon RDS Optimized Writes verbessert die Leistung von Schreibtransaktionen für RDS für MySQL	Mit Amazon RDS Optimized Writes können Sie die Leistung von Schreibtransaktionen für DB-Instances von RDS für MySQL verbessern. Weitere Informationen finden Sie unter Verbesserung der Schreibleistung mit Amazon-RDS-optimierten Schreibvorgängen für MySQL.	27. November 2022
Amazon RDS Optimized Reads verbessert die Abfrageleistung für RDS für MySQL	Mit Amazon RDS Optimized Reads können Sie eine schnellere Abfrageverarbeitung für DB-Instances von RDS für MySQL erreichen. Weitere Informationen finden Sie unter Verbesserung der Abfrageleistung mit Amazon RDS Optimized Reads.	27. November 2022
Amazon RDS ist in der Region Asien-Pazifik (Hyderabad) verfügbar	Amazon RDS ist jetzt in der Region Asien-Pazifik (Hyderabad) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	22. November 2022
RDS unterstützt MariaDB 10.6.11, 10.5.18, 10.4.27 und 10.3.37	Sie können jetzt DB-Instances von Amazon RDS erstellen, auf denen die MariaDB-Versionen 10.6.11, 10.5.18, 10.4.27 und 10.3.37 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	18. November 2022

RDS Custom für Oracle unterstützt das Festlegen von nicht standardmäßigen Installationsparametern in einer benutzerdefinierten Engine-Version (CEV)	Wenn Sie eine CEV erstellen, können Sie nicht standardmäßige Werte für die Oracle-Basis, das Oracle-Standardverzeichnis, den UNIX-Benutzernamen und die -ID sowie den UNIX-Gruppennamen und die -ID festlegen. Auf diese Weise erhalten Sie mehr Kontrolle über die Datenbankinstallation auf Ihrer DB-Instance von RDS Custom für Oracle. Weitere Informationen finden Sie unter Vorbereiten des CEV-Manifests.	18. November 2022
Amazon RDS unterstützt Oracle APEX Version 22.1.v1	Sie können Oracle APEX 22.1.v1 mit allen unterstützten Versionen von Oracle Database verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	18. November 2022
RDS für SQL Server unterstützt regionsübergreifende Lesereplikate	Sie können jetzt ein regionsübergreifendes Lesereplikat erstellen, um die Notfallwiederherstellungsfunktionen zu verbessern, die Anwendungsleselatenz zu reduzieren und Lese-Workloads von der primären DB-Instance auszulagern. Weitere Informationen finden Sie unter Eine Read Replica in einer anderen Umgebung erstellen. AWS-Region	16. November 2022

Amazon RDS ist in der Region Europa (Spanien) verfügbar	Amazon RDS ist jetzt in der Region Europa (Spanien) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	16. November 2022
RDS für SQL Server unterstützt verknüpfte Server für Oracle-Datenbanken	Sie können jetzt einen verknüpften Server erstellen, um auf externe Oracle-Datenbanken zuzugreifen und Daten zu lesen sowie SQL-Befehle auszuführen. Weitere Informationen finden Sie unter Verknüpfte Server mit Oracle OLEDB und RDS für SQL Server.	15. November 2022
RDS Custom für Oracle unterstützt Oracle Multitenant	Sie können eine DB-Instance von RDS Custom für Oracle als Container-Datenbank (CDB) erstellen. Nach der Erstellung enthält die CDB das CDB-Stammverzeichnis, den PDB-Seed und eine PDB. Sie können weitere PDBs manuell mit Oracle SQL hinzufügen. Weitere Informationen finden Sie unter Übersicht über die Architektur von Amazon RDS Custom für Oracle.	15. November 2022

Amazon RDS für Oracle unterstützt Amazon-EFS-Integration	Wenn Sie Ihrer Optionsgruppe die `EFS_INTEGRATION` -Option hinzufügen, können Sie Dateien zwischen Ihrer DB-Instance von RDS für Oracle und einem Amazon-EFS-Dateisystem übertragen. Weitere Informationen finden Sie unter Amazon EFS.	15. November 2022
RDS unterstützt MySQL 8.0.31 und 5.7.40	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen die MySQL-Versionen 8.0.31 und 5.7.40 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	10. November 2022
Amazon RDS ist in der Region Europa (Zürich) verfügbar	Amazon RDS ist jetzt in der Region Europa (Zürich) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	9. November 2022
Zugriff auf Transaktionsprotokoll-Backups ist jetzt für RDS für SQL Server verfügbar	Sie können nun Datenbank transaktionsprotokoll-Backups anzeigen und in einen Amazon-S3-Bucket kopieren. Weitere Informationen finden Sie unter Zugriff auf Transaktionsprotokoll-Backups.	7. November 2022

Multi-AZ-DB-Cluster werden zusätzlich unterstützt AWS-Regionen	Multi-AZ-DB-Cluster sind jetzt zusätzlich verfügbar . AWS-Regionen Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für Multi-AZ-DB-Cluster in Amazon RDS.	04. November 2022
Amazon RDS unterstützt gp3-Speicher	Sie können jetzt DB-Instances von Amazon RDS erstellen , die Amazon EBS General Purpose SSD (gp3)-Spe ichervolumes verwenden , sodass Sie die Speicherl eistung unabhängig von der Speicherkapazität anpassen können. Weitere Informationen finden Sie unter Allzweck-SSD-Speicher.	04. November 2022
Amazon RDS unterstützt ein neues Ereignis für Betriebss ystemaktualisierungen	Amazon RDS unterstützt jetzt ein neues DB-Instance-Ereign is, RDS-EVENT-0230, in der Ereigniskategorie „Ausführe n von Sicherheits-Patches“. Dieses neue Ereignis informier t Sie, wenn eine Betriebss ystemaktualisierung für Ihre DB-Instance verfügbar ist. Weitere Informationen finden Sie unter Überwachen von Amazon-RDS-Ereignissen und Arbeiten mit Betriebss ystemaktualisierungen.	28. Oktober 2022

Amazon RDS für Oracle unterstützt vorkonfigurierte arbeitsspeicheroptimierte r5b-Instance-Klassen	Die Oracle-DB-Instance -Klassen db.r5b sind für Workloads optimiert, die zusätzlichen Arbeitsspeicher und Speicherplatz pro vCPU benötigen. I/O Beispielsweise ist bei db.r5b.4xlarge.tpc 2.mem2x Multithreading aktiviert und bietet doppelt so viel Speicher wie db.r5b.4x large. Weitere Informationen finden Sie unter RDS for Oracle-DB-Instance-Klassen.	27. Oktober 2022
Amazon RDS unterstützt 15 Lesereplikate für DB-Instan ces von RDS für MariaDB, MySQL und PostgreSQL	Sie können jetzt bis zu 15 Lesereplikate für DB-Instan ces von RDS für MariaDB, MySQL und PostgreSQL erstellen. Weitere Informati onen zu Lesereplikaten finden Sie unter Arbeiten mit Lesereplikaten.	20. Oktober 2022
Amazon RDS für PostgreSQ L unterstützt jetzt PostgreSQ L-Version 15 RC 3 in der Preview-Umgebung der Datenbank.	PostgreSQL Version 15 Beta 3 ist jetzt in der Datenbank-Vorschauumgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informati onen finden Sie unter Arbeiten in der Database Preview-U mgebung.	18. Oktober 2022

Amazon RDS unterstützt die automatische Einrichtung von Konnektivität zwischen einer RDS-Datenbank und einer EC2 Instance.	Sie können den verwenden AWS Management Console , um Konnektivität zwischen einer vorhandenen RDS-DB-Instance oder einem Multi-AZ-DB-Cluster und einer EC2 Instance einzurichten. Weitere Informationen finden Sie unter Automatisches Verbinden einer EC2 Instance und einer RDS-Datenbank.	14. Oktober 2022
AWS JDBC-Treiber für PostgreSQL allgemein verfügbar	Der AWS JDBC-Treiber für PostgreSQL ist ein Client-Treiber, der für RDS for PostgreSQL entwickelt wurde. Der AWS -JDBC-Treiber für PostgreSQL ist jetzt allgemein verfügbar. Weitere Informationen finden Sie unter Verbindung mit dem AWS JDBC-Treiber für PostgreSQL herstellen.	6. Oktober 2022
Amazon RDS für Oracle unterstützt Oracle APEX Version 21.2.v1	Oracle APEX 21.2 enthält Patch 33420059. Weitere Informationen finden Sie unter Oracle APEX-Versionsanforderungen.	3. Oktober 2022
RDS unterstützt MySQL 5.7.39	Sie können jetzt DB-Instances von Amazon RDS erstellen , auf denen die MySQL-Version 5.7.39 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	29. September 2022

RDS unterstützt MariaDB 10.6.10	Sie können jetzt DB-Instances von Amazon RDS erstellen, die mit der MariaDB-Version 10.6.10 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	29. September 2022
RDS Proxy unterstützt RDS für SQL Server	Sie können jetzt einen RDS-Proxy für eine RDS-DB-Instance erstellen, auf der die Microsoft-SQL-Server-Version 2014 oder höher ausgeführt wird. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	19. September 2022
RDS unterstützt MariaDB 10.5.17, 10.4.26 und 10.3.36	Sie können jetzt DB-Instances von Amazon RDS erstellen, die mit den MariaDB-Versionen 10.5.17, 10.4.26 und 10.3.36 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	15. September 2022

Amazon RDS für SQL Server unterstützt lokale Instance-Speicher für temporäre Daten	Sie können Amazon RDS for Oracle jetzt auf den Instance-Typen Amazon EC2 db.r5d und db.m5d starten, wobei der temporäre Tablespac e und der Database Smart Flash Cache (der Flash-Cache) für die Verwendun g eines Instance-Speichers konfiguriert sind. Durch das lokale Speichern von temporären Daten können Sie im Vergleich zum Standards peicher basierend auf Amazon EBS niedrigere Lese- und Schreiblatenzen erzielen. Weitere Informationen finden Sie unter Speichern temporäre r Oracle-Daten im Instance-Speicher.	14. September 2022
Performance Insights zeigt die 25 wichtigsten SQL-Abfragen	Auf der Registerkarte Top SQL (Top-SQL) werden die 25 SQL-Abfragen angezeigt , die am meisten zur DB-Last beitragen. Weitere Informationen finden Sie unter Übersicht über die Registerk arte „Top SQL“ (Top-SQL).	13. September 2022

RDS unterstützt MySQL 8.0.30	Sie können jetzt DB-Instances von Amazon RDS erstellen , auf denen die MySQL-Version 8.0.30 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	09. September 2022
Amazon RDS ist in der Region Naher Osten (UAE) verfügbar	Amazon RDS ist jetzt in der Region Naher Osten (UAE) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	30. August 2022
Amazon RDS for SQL Server unterstützt SSRS E-Mail-Abonnements	Sie können jetzt die E-Mail-Erweiterung SQL Server Reporting Services (SSRS) verwenden, um Berichte an Benutzer zu senden und Berichte auf dem Berichtsserver zu abonnieren. Weitere Informationen finden Sie unter Unterstützung für SQL Server Reporting Services in RDS für SQL Server.	26. August 2022
RDS für Oracle unterstützt Lesereplikat-Backups	Sie können automatische Backups aktivieren und manuelle Snapshots von RDS-für-Oracle-Replikate erstellen . Weitere Informationen finden Sie unter Arbeiten mit RDS-für-Oracle-Replikat-Backups.	23. August 2022

RDS für Oracle unterstützt Oracle Data Guard Switchover	Ein Switchover ist ein Rollentausch zwischen einer Primärdatenbank und einem bereitgestellten oder offenen Oracle-Replikat. Während eines Switchovers wechselt die ursprüngliche Primärdatenbank in eine Standby-Rolle, während die ursprüngliche Standby-Datenbank in die primäre Rolle übergeht. Weitere Informationen finden Sie unter Durchführen eines Oracle Data Guard Switchover.	23. August 2022
Amazon RDS unterstützt das automatische Einrichten von Konnektivität mit einer EC2 Instance.	Wenn Sie eine DB-Instance oder einen Multi-AZ-DB-Cluster erstellen, können Sie den verwenden, AWS Management Console um die Konnektivität zwischen einer Amazon Elastic Compute Cloud-Instance und der neuen DB-Instance oder dem neuen DB-Cluster einzurichten. Weitere Informationen finden Sie unter Automatische Netzwerkkonnektivität mit einer EC2 Instance für eine neue DB-Instance konfigurieren und Automatische Netzwerkkonnektivität mit einer EC2 Instance für einen neuen DB-Cluster konfigurieren.	22. August 2022

RDS for Oracle unterstützt die Heraufstufung von Oracle-Replikaten	Wenn Sie RDS Custom for Oracle verwenden, können Sie Ihre verwalteten Oracle-Replikate mit dem `promote-read-replica` -CLI-Befehl verwenden. Sie können auch Ihre primäre DB-Instance löschen, was dazu führt, dass RDS Custom for Oracle Ihre verwalteten Oracle-Replikate zu eigenständigen Instances hochstuft. Weitere Informationen finden Sie unter Arbeiten mit Oracle-Replikaten für RDS Custom for Oracle.	5. August 2022
RDS für MySQL unterstützt das Erzwingen von Verbindungen SSL/TLS	RDS for MySQL unterstützt jetzt das Erzwingen von SSL/TLS Verbindungen, indem der `require_secure_transport` Parameter auf `ON` gesetzt wird. Weitere Informationen finden Sie unter SSL/TLS Verbindung zu einer MySQL-DB-Instance erforderlich.	1. August 2022

Amazon RDS hat den Support für Oracle Database 12c Release 1 (12.1.0.2) eingestellt	Support für Version 12.1.0.2 ist sowohl für das BYOL als auch für LI-Lizensierungsmodelle veraltet. Am 1. August 2022 beginnt RDS für Oracle mit automatischen Upgrades von 12c Release 1 (12.1.0.2) und stellt 12.1.0.2-Snapshots in Oracle Database 19c wieder her. Weitere Informationen finden Sie am Ende der Supportzeitleiste unter AWS re:Post.	1. August 2022
RDS Proxy unterstützt RDS	Sie können jetzt einen RDS-Proxy für eine RDS-DB-Instance erstellen, auf der die MariaDB-Version 10.2, 10.3, 10.4 oder 10.5 ausgeführt wird. Der MariaDB-Support ist in der MySQL-Engine-Familie enthalten. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	26. Juli 2022
RDS für MariaDB unterstützt die DB-Instanzklassen db.r5b	Sie können jetzt RDS für MariaDB DB-Instances erstellen, die die db.r5b DB-Instanzklassen verwenden. Weitere Informationen finden Sie unter Unterstützte DB-Engines für DB-Instance-Klassen.	25. Juli 2022

RDS für Oracle unterstützt die Änderung von Datenbank aktivitätsströmen	Wenn Sie RDS für Oracle verwenden, können Sie den Status der Überwachungsrichtlinie eines Datenbank-Aktivitäts-Streams entweder in gesperrt (Standard) oder entsperrt ändern. Anstatt einen Aktivitätsstream zu stoppen, können Sie seinen Richtlinienstatus entsperren, Ihre Überwachungsrichtlinie anpassen und dann den Richtlinienstatus erneut sperren. Weitere Informationen finden Sie unter Ändern eines Datenbankaktivitätsstroms.	22. Juli 2022
Performance Insights unterstützt die Region Asien-Pazifik (Jakarta)	Bisher konnten Sie Performance Insights in der Region Asien-Pazifik (Jakarta) nicht verwenden. Diese Einschränkung wurde entfernt. Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	21. Juli 2022

Microsoft SQL Server 2012 hat das Ende des Supports für Amazon RDS erreicht	Microsoft SQL Server 2012 hat das Ende des Supports erreicht, was mit dem Plan von Microsoft zusammenfällt, die erweiterte Unterstützung für diese Version am 12. Juli 2022 zu beenden. Alle vorhandenen Microsoft SQL Server 2012-Instances werden ab dem 1. Juni 2022 automatisch auf die neueste Unterversion von Microsoft SQL Server 2014 aktualisiert. Weitere Informationen finden Sie unter Microsoft SQL Server 2012 Support auf Amazon RDS.	12. Juli 2022
RDS unterstützt MariaDB 10.6.8, 10.5.16, 10.4.25, 10.3.35 und 10.2.44	Sie können jetzt Amazon RDS DB-Instances mit den MariaDB-Versionen 10.6.8, 10.5.16, 10.4.25, 10.3.35 und 10.2.44 erstellen. Weitere Informationen finden Sie unter Unterstützte MariaDB-Versionen auf Amazon RDS.	8. Juli 2022

RDS Performance Insights unterstützt zusätzliche Aufbewahrungsfristen	Bisher bot Performance Insights nur zwei Aufbewahrungsfristen an: 7 Tage (Standard) oder 2 Jahre (731 Tage). Wenn Sie Ihre Leistungsdaten jetzt länger als 7 Tage aufbewahren müssen, können Sie zwischen 1 und 24 Monaten angeben. Weitere Informationen finden Sie unterPreisgestaltung und Datenspeicherung für Performance Insights.	01. Juli 2022
RDS Custom unterstützt die Regionen Asien-Pazifik (Mumbai) und Europa (London)	Sie können RDS Custom for Oracle und RDS Custom for SQL Server DB-Instances in zwei neuen Versionen erstellen AWS-Regionen: Asien-Pazifik (Mumbai) und Europa (London). Weitere Informationen finden Sie unter AWS-Region -Support von RDS Custom for Oracle und AWS-Region -Support von RDS Custom for SQL Server.	21. Juni 2022

RDS Custom for Oracle unterstützt Oracle Database 18c und 12c Release 2 (12.2)	Sie können jetzt eine CEV for RDS Custom for Oracle erstellen, indem Sie Installat ionsdateien für Oracle Database 18c und 12c Release 2 (12.2) verwenden. Sie können diese verwenden CEVs , um eine RDS Custom for Oracle DB-Instance zu erstellen. Weitere Informati onen finden Sie unter Arbeiten mit benutzerdefinierten Engine-Versionen für Amazon RDS Custom für Oracle.	21. Juni 2022
Multi-AZ-DB-Cluster unterstüt zen die DB-Instance-Klassen db.m5d und db.r5d	Sie können jetzt Multi-AZ-DB-Cluster erstellen, die die DB-Instance-Klassen db.m5d und db.r5d verwenden. Weitere Informationen finden Sie unter Multi-AZ-DB-Cluster-Bereits tellungen und DB-Instance-Klassentypen.	21. Juni 2022
Multi-AZ-DB-Cluster sind zusätzlich erhältlich AWS-Regionen	Sie können jetzt Multi-AZ-DB-Cluster in den folgenden Regionen erstellen: Europa (Frankfurt) und Europa (Stockholm). Weitere Informati onen finden Sie unter Multi-AZ-DB-Cluster-Bereits tellungen.	21. Juni 2022

RDS for Microsoft SQL Server unterstützt die Migration von Datenbanken, die die transparente Datenverschlüsselung (TDE) verwenden	RDS for SQL Server unterstützt jetzt die Migration von Microsoft-SQL-Server-Datenbanken bei aktivierter TDE unter Verwendung von nativer Sicherung und Wiederherstellung. Weitere Informationen finden Sie unter Unterstützung für transparente Datenverschlüsselung in SQL Server.	14. Juni 2022
Amazon RDS unterstützt das Veröffentlichen von Ereignissen in verschlüsselten Amazon-SNS-Themen	Amazon RDS kann jetzt Ereignisse in Amazon Simple Notification Service (Amazon SNS)-Themen veröffentlichen, bei denen serversei tige Verschlüsselung (SSE) aktiviert ist, um Ereignisse, die sensible Daten enthalten, zusätzlich zu schützen. Weitere Informationen finden Sie unter Abonnieren von Amazon-RDS-Ereignisbenachrichtigungen.	1. Juni 2022
RDS unterstützt MySQL 5.7.38	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen MySQL Version 5.7.38 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	31. Mai 2022

RDS for PostgreSQL unterstützt kaskadierende Lesereplikate	Sie können jetzt kaskadierende Read Replicas mit RDS for PostgreSQL Version 14.1 und höher verwenden. Weitere Informationen finden Sie unter Arbeiten mit PostgreSQL-Lesereplikaten in Amazon RDS.	4. Mai 2022
Amazon RDS on AWS Outposts unterstützt skalierbare Speicher- und Autoscaling-Operationen	Sie können jetzt die Speichergröße von DB-Instances in Ihrem Outpost ändern und die automatische Speicherskalierung verwenden. Weitere Informationen finden Sie unter Amazon RDS auf AWS Outposts Outposts Unterstützung für Amazon RDS-Funktionen.	2. Mai 2022
Multi-AZ-DB-Cluster sind zusätzlich erhältlich AWS-Regionen	Sie können jetzt Multi-AZ-DB-Cluster in den folgenden Regionen erstellen: Asien-Pazifik (Singapur) und Asien-Pazifik (Sydney). Weitere Informationen finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen.	29. April 2022

Amazon RDS unterstützt den Dual-Stack-Modus	DB-Instances können jetzt im Dual-Stack-Modus ausgeführt werden. Im Dual-Stack-Modus können Ressourcen über, oder beides IPv4 IPv6, mit der DB-Instance kommunizieren. Weitere Informationen finden Sie unter Amazon-RDS-IP-Adressierung.	29. April 2022
Amazon RDS veröffentlicht Nutzungsmetriken auf Amazon CloudWatch	Der `AWS/Usage` Namespace in Amazon CloudWatch enthält Nutzungsmetriken auf Kontoebene für Ihre Amazon RDS-Servicekonting enten. Weitere Informationen finden Sie unter CloudWatch Amazon-Nutzungsmetriken für Amazon RDS.	28. April 2022
Amazon RDS for MySQL unterstützt die DB-Instance-Klassen db.m6i und db.r6i	Sie können nun die DB-Instance-Klassen db.m6i und db.r6i für Amazon-RDS-DB-Instances verwenden, die MySQL ausführen. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	28. April 2022

Amazon RDS for PostgreSQL unterstützt die DB-Instance-Klassen db.m6i und db.r6i	Sie können nun die DB-Instance-Klassen db.m6i und db.r6i für Amazon-RDS-DB-Instances verwenden, die PostgreSQL ausführen. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	27. April 2022
Amazon RDS for MariaDB unterstützt die DB-Instance-Klassen db.m6i und db.r6i	Sie können nun die DB-Instance-Klassen db.m6i und db.r6i für Amazon-RDS-DB-Instances verwenden, die MariaDB ausführen. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	26. April 2022
Amazon RDS on AWS Outposts unterstützt Multi-AZ-Bereitstellungen	Sie können jetzt eine Standby-DB-Instance für einen anderen Outpost erstellen. Weitere Informationen finden Sie unter Amazon RDS zur AWS Outposts Unterstützung von Amazon RDS-Funktionen.	19. April 2022

Amazon RDS for Oracle unterstützt die Instance-Klassen db.m6i und db.r6i	Wenn Sie Oracle Database 19c ausführen , können Sie die Instance-Klassen db.m6i und db.r6i verwenden. Die db.m6i-Klassen sind Allzweck-Instance-Klassen, die sich gut für eine breite Palette von Workloads eignen. Weitere Informationen finden Sie unter RDS for Oracle-DB-Instance-Klassen.	8. April 2022
Amazon RDS for SQL Server unterstützt die Jobreplikation von SQL Server Agent	Wenn Sie diese Funktion aktivieren, werden auf dem primären Host erstellte, geänderte oder gelöschte SQL-Server-Agent-Aufträge automatisch mit dem sekundären Host in einer Multi-AZ-Konfiguration synchronisiert. Weitere Informationen finden Sie unter Verwenden von SQL Server Agent.	7. April 2022
Amazon RDS unterstützt RDS Proxy mit RDS for PostgreSQL Version 13	Sie können jetzt einen RDS-Proxy mit einer RDS for PostgreSQL Version 13 Datenbank erstellen. Weitere Informationen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	4. April 2022

Amazon RDS plant, Oracle Database 12c einzustellen	Oracle Database 12c befindet sich auf einem Veraltungspfad. Die Oracle Corporation wird nach diesen end-of-support Terminen keine Patches mehr für Oracle Database 12c-Versionen bereitstellen. Amazon RDS plant, mit dem automatischen Upgrade von Oracle-Database-12c-DB-Instances auf Oracle Database 19c zu beginnen.	22. März 2022
Versionshinweise zu Amazon RDS for PostgreSQL	Es gibt jetzt einen separaten Leitfaden für die Versionshinweise von Amazon RDS for PostgreSQL. Weitere Informationen finden Sie in den Versionshinweisen zu Amazon RDS for PostgreSQL.	22. März 2022
Versionshinweise zu Amazon RDS for Oracle	Es gibt jetzt einen separaten Leitfaden für die Versionshinweise von Amazon RDS for Oracle. Weitere Informationen finden Sie in den Versionshinweisen zu Amazon RDS for Oracle.	22. März 2022
Multi-AZ-DB-Cluster sind zusätzlich erhältlich AWS-Regionen	Sie können jetzt Multi-AZ-DB-Cluster in den folgenden Regionen erstellen: USA Ost (Ohio) und Asien-Pazifik (Tokio). Weitere Informationen finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen.	15. März 2022

Amazon-RDS-for-PostgreSQL-Versionen 14.2, 13.6, 12.10, 11.15 und 10.20	RDS for PostgreSQL unterstützt jetzt die Versionen 14.2, 13.6, 12.10, 11.15 und 10.20. Version 14.2 und 13.6 bieten Unterstützung für zwei neue Fremddaten-Wrapper. Die Erweiterung `mysql_fdw` ermöglicht PostgreSQL, mit Daten zu arbeiten, die in MySQL-, MariaDB- und Aurora MySQL-Datenbanken gespeichert sind. Die Erweiterung `tds_fdw` ermöglicht PostgreSQL, mit Daten zu arbeiten, die in SQL-Server-Datenbanken gespeichert sind. Weitere Informationen finden Sie unter Unterstützte PostgreSQL-Datenbankversionen.	12. März 2022
RDS unterstützt MySQL 5.7.37	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen MySQL Version 5.7.37 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	11. März 2022

Amazon RDS for SQL Server unterstützt neue DB-Instance-Klassen	Sie können nun Amazon-RDS-DB-Instances mit Microsoft SQL Server erstellen, die die DB-Instance-Klassen db.m6i und db.r6i verwenden. Weitere Informationen finden Sie unter DB-Instance-Klassenunterstützung für Microsoft SQL Server.	9. März 2022
Amazon RDS for Oracle unterstützt Oracle Database 21c	Sie können nun Amazon-RDS-DB-Instances erstellen, auf denen Oracle Database 21c (21.0.0.0) ausgeführt wird. Dies ist die erste Oracle-Database-Version, die nur die Multitenant-Architektur (CDB) unterstützt. Weitere Informationen finden Sie unter Oracle Database 21c mit Amazon RDS.	7. März 2022
RDS unterstützt MariaDB 10.6.7, 10.5.15, 10.4.24, 10.3.34 und 10.2.43	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen die MariaDB-Versionen 10.6.7, 10.5.15, 10.4.24, 10.3.34, und 10.2.43 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	3. März 2022

AWS JDBC-Treiber für MySQL allgemein verfügbar	Der AWS JDBC-Treiber für MySQL ist ein Client-Treiber, der für RDS for MySQL entwickelt wurde. Der AWS JDBC-Treiber für MySQL ist jetzt allgemein verfügbar. Weitere Informationen finden Sie unter Herstellen einer Verbindung mit dem Amazon Web Services JDBC-Treiber für MySQL.	2. März 2022
Multi-AZ-DB-Cluster allgemein verfügbar	Eine Multi-AZ-DB-Cluster-Bereitstellung ist ein Bereitstellungsmodus für Hochverfügbarkeit von Amazon RDS mit zwei lesbaren Standby-DB-Instances. Multi-AZ-DB-Cluster sind jetzt allgemein verfügbar. Weitere Informationen finden Sie unter Multi-AZ-DB-Cluster-Bereitstellungen.	1. März 2022
RDS unterstützt MySQL 8.0.28	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0.28 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	28. Februar 2022

Amazon RDS for Oracle unterstützt neue Einstellungen für die native Netzwerkverschlüsselung (NNE)	Um zu steuern, ob Clients eine Verbindung mit nicht sicheren Verschlüsselungs- und Prüfsummierungsmethoden herstellen können, legen Sie `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` und `SQLNET.ALLOW_WEAK_CRYPTO` in der NNE-Option fest. Beispiele für unsichere Methoden sind DES, 3DES und. RC4 MD5 Weitere Informationen finden Sie unter Einstellungen für NNE-Optionen.	25. Februar 2022
Amazon RDS for SQL Server unterstützt Always-On-Availability-Groups für Microsoft SQL Server 2017 Standard Edition	Wenn Sie eine DB-Instance mit der Multi-AZ-Konfiguration auf SQL Server 2017 Standard Edition-Datenbank-Engine 14.00.3401.7 und höhere Versionen erstellen, verwendet RDS automatisch Verfügbarkeitsgruppen. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Microsoft SQL Server.	18. Februar 2022
RDS for Oracle unterstützt Datenbank-Aktivitäts-Streams in der Region Asien-Pazifik (Jakarta)	Weitere Informationen finden Sie unter Support AWS-Regionen für Datenbankaktivitätsstreams.	16. Februar 2022

Amazon-RDS-Custom-for-Oracle-Unterstützung für Oracle Database 12.1	Sie können jetzt benutzerdefinierte Engine-Versionen für RDS Custom for Oracle erstellen, die Oracle Database 12.1 Enterprise Edition verwenden. Weitere Informationen finden Sie unter Arbeiten mit benutzerdefinierten Engine-Versionen für Amazon RDS Custom für Oracle.	4. Februar 2022
Amazon RDS for MariaDB unterstützt eine neue Hauptversion	Sie können jetzt Amazon-RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.6 laufen. Weitere Informationen finden Sie unter MariaDB-10.6-Unterstützung auf Amazon RDS.	3. Februar 2022

Performance Insights unterstützt Planerfassung für Oracle-Abfragen	Die Performance-Insights-Konsole unterstützt eine neue Plandimension für Top-SQL. Wenn Sie nach Plan aufteilen, können Sie sehen, welche Pläne Ihre Top-Oracle-Abfragen verwenden. Wenn eine Abfrage mehrere Pläne verwendet, können Sie die Pläne nebeneinander in der Konsole vergleichen und ermitteln, welcher Plan am effizientesten ist. Sie können auch einen Drilldown durchführen, um zu sehen, welche Schritte in einem Plan die höchsten Kosten aufweisen. Weitere Informationen finden Sie unter Analysieren von Oracle-Ausführungsplänen über das Performance-Insights-Dashboard.	27. Januar 2022

Performance Insights unterstützt neue APIs	Performance Insights unterstützt Folgendes APIs: `GetResourceMetadata` `ListAvailableResourceDimensions` , und`ListAvailableResourceMetrics` . Weiteren Informationen finden Sie unter Abrufen von Metriken mit der Performance Insights API in diesem Handbuch und in der Referenz zur Amazon RDS Performance Insights API.	12. Januar 2022
RDS Proxy unterstützt Ereignisse	RDS Proxy generiert jetzt Ereignisse, die Sie abonnieren und unter CloudWatch Ereignisse anzeigen oder so konfigurieren können, dass sie an Amazon gesendet EventBridge werden. Weitere Informationen finden Sie unter Arbeiten mit RDS-Proxy-Ereignissen.	11. Januar 2022
Amazon RDS for SQL Server unterstützt SSAS im mehrdimensionalen Modus	RDS for SQL Server unterstützt die Ausführung von SQL Server Analysis Services (SSAS) im tabellarischen oder mehrdimensionalen Modus. Weitere Informationen finden Sie unter Unterstützung für SQL Server Analysis Services in RDS for SQL Server.	7. Januar 2022

RDS Proxy ist zusätzlich erhältlich AWS-Regionen	RDS Proxy ist jetzt in den folgenden Regionen erhältlich: Afrika (Kapstadt), Asien-Pazifik (Hongkong, Osaka), Europa (Milan, Paris, Stockholm), Naher Osten (Bahrain) und Südamerika (São Paulo). Weitere Informati onen zu RDS Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	5. Januar 2022
RDS unterstützt MySQL 8.0.27	Sie können jetzt Amazon-RD S-DB-Instances erstellen, auf denen MySQL-Version 8.0.27 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	21. Dezember 2021
Amazon RDS ist in der Region Asien-Pazifik (Jakarta) verfügbar	Amazon RDS ist jetzt in der Region Asien-Pazifik (Jakarta) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	13. Dezember 2021
Amazon RDS unterstützt MariaDB 10.5.13, 10.4.22, 10.3.32 und 10.2.41	Sie können jetzt Amazon-RDS-DB-Instances erstellen , auf denen die MariaDB-V ersionen 10.5.13, 10.4.22, 10.3.32 und 10.2.41 ausgeführ t werden. Weitere Informati onen finden Sie unter MariaDB auf Amazon RDS-Versionen.	8. Dezember 2021

Amazon RDS Custom für SQL Server	Amazon RDS Custom ist ein verwalteter Datenbankdienst für ältere, benutzerdefinierte und gepackte Anwendungen, die Zugriff auf das zugrunde liegende Betriebssystem und die Datenbankumgebung erfordern. Mit Amazon RDS Custom erhalten Sie die Automatisierung von Amazon RDS und die Flexibilität von Amazon EC2. Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS Custom.	1. Dezember 2021
Multi-AZ-DB-Cluster (Vorschau)	Sie können jetzt Multi-AZ-DB-Cluster für RDS for MySQL und RDS for PostgreSQ L erstellen. Eine Multi-AZ-DB-Cluster-Bereitstellung ist ein Bereitstellungsmod us für Hochverfügbarkeit von Amazon RDS mit zwei lesbaren Standby-DB-Instanz en. Multi-AZ DB-Cluster befinden sich in der Vorschau. Weitere Informationen finden Sie unter Multi-AZ DB-Cluster-Bereitstellungen (Vorschau).	23. November 2021

Amazon RDS unterstützt RDS Proxy mit RDS for PostgreSQL Version 12	Sie können jetzt einen RDS-Proxy mit einer RDS for PostgreSQL Version 12 Datenbank erstellen. Weitere Informationen über RDS-Proxy finden Sie unter Verwenden von Amazon RDS Proxy.	22. November 2021
Amazon RDS on AWS Outposts unterstützt lokale Backups	Sie können automatische Backups und manuelle Snapshots in Ihrem AWS-Region oder lokal auf Ihrem Outpost speichern. Weitere Informationen finden Sie unter Amazon RDS zur AWS Outposts Unterstützung von Amazon RDS-Funktionen.	22. November 2021
Amazon RDS-Unterstützung für kontoübergreifende Nutzung AWS KMS keys	Sie können einen KMS-Schlüssel von einem anderen AWS Konto für die Verschlüsselung verwenden, wenn Sie DB-Snapshots nach Amazon S3 exportieren. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3.	3. November 2021
Amazon RDS on AWS Outposts unterstützt die Veröffentlichung von Datenbank-Engine-Protokollen in CloudWatch Logs	RDS on Outposts unterstützt jetzt die Veröffentlichung von Datenbank-Engine-Protokollen in CloudWatch Logs. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	2. November 2021

Amazon RDS Custom für Oracle	Amazon RDS Custom ist ein verwalteter Datenbankdienst für ältere, benutzerdefinierte und gepackte Anwendungen, die Zugriff auf das zugrunde liegende Betriebssystem und die Datenbankumgebung erfordern. Mit Amazon RDS Custom erhalten Sie die Automatisierung von Amazon RDS und die Flexibilität von Amazon EC2. Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS Custom.	26. Oktober 2021
Support für verzögerte Replikation für RDS for MySQL Version 8.0	Ab RDS for MySQL Version 8.0.26 können Sie die verzögerte Replikation für RDS for MySQL Version 8.0 DB-Instances konfigurieren. Weitere Informationen finden Sie unter Konfigurieren der verzögerten Replikation mit MySQL.	25. Oktober 2021
Unterstützung für MySQL 8.0.26	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL-Ver sion 8.0.26 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	25. Oktober 2021

Support für GTID-basierte Replikation für RDS for MySQL Version 8.0	Ab RDS for MySQL Version 8.0.26 können Sie die GTID-basierte Replikation für RDS for MySQL Version 8.0 DB-Instanzen konfigurieren. Weitere Informationen finden Sie unter Verwenden der GTID-basierten Replikation für RDS for MySQL.	25. Oktober 2021
Amazon RDS unterstützt RDS Proxy mit RDS for MySQL 8.0	Sie können jetzt einen RDS-Proxy für eine RDS for MySQL 8.0-Datenbankinstanz erstellen. Weitere Informationen finden Sie unter Amazon RDS Proxy.	21. Oktober 2021
Amazon RDS on AWS Outposts unterstützt zusätzliche RDS für MySQL-Versionen	RDS on Outposts unterstützt jetzt die RDS for PostgreSQL Versionen 8.0.23 und 8.0.25. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	20. Oktober 2021
Amazon RDS for PostgreSQL unterstützt jetzt PostgreSQL-Version 14 RC 1 in der Database Preview-Umgebung.	PostgreSQL Version 14 RC 1 ist jetzt in der Datenbank-Vorschauumgebung im Osten der USA (Ohio) verfügbar. AWS-Region Weitere Informationen finden Sie unter Arbeiten in der Database Preview-Umgebung.	19. Oktober 2021

Amazon RDS unterstützt Performance Insights zusätzlich AWS-Regionen	Performance Insights ist in den Regionen Naher Osten (Bahrain), Afrika (Kapstadt), Europa (Mailand) und Asien-Pazifik (Osaka) verfügbar . Weitere Informationen finden Sie unter Unterstützte Regionen und DB-Engines für Performance Insights in Amazon RDS.	5. Oktober 2021
Performance Insights unterstützt Statistiken auf Verdauungsebene für Oracle	Wenn Sie Performance Insights verwenden, können Sie SQL-Statistiken sowohl auf Anweisung- als auch auf Digest-Ebene für Amazon RDS for Oracle anzeigen. Weitere Informationen finden Sie unter Analysieren von Statistiken zu laufenden Abfragen.	4. Oktober 2021
Amazon RDS on AWS Outposts unterstützt zusätzliche RDS für PostgreSQL-Versionen	RDS on Outposts unterstützt jetzt die RDS ffor PostgreSQL Versionen 12.8 und 13.4. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	1. Oktober 2021
Amazon RDS unterstützt Oracle APEX Version 21.1.v1	Sie können Oracle APEX 21.1.v1 mit allen unterstützten Versionen von Oracle Database verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	24. September 2021

Amazon RDS for Oracle unterstützt die clientseitige - Verschlüsselung für NNE	Wenn Sie NNE konfiguri eren, möchten Sie möglicher weise vermeiden, dass die Verschlüsselung auf der Serverseite erzwungen wird. Beispielsweise möchten Sie möglicherweise nicht alle Clientkommunikationen dazu zwingen, die Verschlüsselung zu verwenden, da der Server dies erfordert. In diesem Fall können Sie die Verschlüs selung auf der Clientseite mit dem`SQLNET.*CLIENT`-Optionen. Weitere Informati onen finden Sie unter Oracle native Netzwerkverschlüss elung.	24. September 2021
Amazon RDS for MySQL und RDS for PostgreSQL unterstüt zen neue DB-Instanzklassen	Sie können jetzt die Instanzkl assen db.r5b, db.t4g und db.x2g verwenden, um Amazon RDS-DB-Instanzen mit MySQL oder PostgreSQ L zu erstellen. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	15. September 2021

Amazon RDS for Microsoft SQL Server unterstützt Java Database Connectivity (JDBC) mit Microsoft Distributed Transaction Coordinator (MSDTC)	JDBC XA-Transaktionen werden jetzt mit MSDTC für SQL Server 2017 Version 14.00.3223.3 und höher sowie SQL Server 2019 unterstützt. Weitere Informationen finden Sie unter Unterstützung für Microsoft Distributed Transaction Coordinator in RDS for SQL Server.	7. September 2021
Amazon RDS unterstützt MariaDB 10.5.12, 10.4.21, 10.3.31 und 10.2.40	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MariaDB-Versionen 10.5.12, 10.4.21, 10.3.31 und 10.2.40 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	2. September 2021
Amazon RDS unterstützt Oracle Database 18c nicht mehr	Sie können DB-Instances nur für Oracle Database 12c und Oracle Database 19c erstellen. Wenn Sie über Snapshots von Oracle Database 18c verfügen, führen Sie ein Upgrade auf eine spätere Version durch. Weitere Informationen finden Sie unter Aktualisieren eines Oracle-DB-Snapshots.	17. August 2021

Amazon RDS for PostgreSQL unterstützt jetzt die PostgreSQL-Version 14 Beta 2 in der Database Preview-Umgebung	Weitere Informationen zu PostgreSQL-Version 14, Beta 1, finden Sie in den Versionshinweisen zu PostgreSQL 14 Beta 1. Weitere Informationen zu PostgreSQL-Version 14, Beta 2, finden Sie in den Versionshinweisen zu PostgreSQL 14 Beta 2. Weitere Informationen über die Database Preview-Umgebung finden Sie unter Arbeiten mit der Database Preview-Umgebung.	9. August 2021
Amazon RDS unterstützt RDS Proxy in einer gemeinsam genutzten VPC	Jetzt können Sie einen RDS Proxy in einer gemeinsam genutzten VPC erstellen Weitere Informationen zu RDS Proxy finden Sie unter „Verwalten von Verbindungen mit Amazon RDS Proxy“ im Amazon RDS-Benutzerhandbuch oder im Aurora-Benutzerhandbuch.	6. August 2021
Amazon RDS unterstützt MariaDB 10.2.39	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.2.39 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	4. August 2021

Amazon RDS for Oracle bietet die `TIMEZONE_FILE_AUTOUPGRADE` Option	Mit dieser Option können Sie die aktuelle Zeitzonendatei auf die neueste Version Ihrer DB-Instance aktualisieren. Weitere Informationen finden Sie unter Autoupgrade der Oracle-Zeitzonendatei.	30. Juli 2021
Amazon RDS erweitert Unterstützung für regionsübergreifende automatisierte Backups	Sie können jetzt DB-Snapshots und Transaktionsprotokolle zwischen weiteren AWS-Regionen replizieren. Weitere Informationen finden Sie unter Automatisierte Backups in eine andere Region replizieren. AWS	19. Juli 2021
Unterstützung für MySQL 5.7.34	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen MySQL-Version 5.7.34 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	8. Juli 2021
Amazon RDS on AWS Outposts unterstützt zusätzliche RDS für PostgreSQL-Versionen	RDS on Outposts unterstützt jetzt die RDS ffor PostgreSQL Versionen 12.7 und 13.3. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	8. Juli 2021

Amazon RDS for PostgreSQL unterstützt oracle_fdw	Sie können jetzt die Erweiterung oracle_fdw verwenden, um einen fremden Daten-Wrapper für den Zugriff auf Oracle-Datenbanken bereitzustellen. Weitere Informationen finden Sie unter Zugreifen auf externe Daten mit der Erweiterung oracle_fdw.	8. Juli 2021
Amazon RDS unterstützt Oracle Management Agent (OMA) Version 13.5	Sie können Oracle Management Agent (OMA) Version 13.5. mit Oracle Enterprise Manager (OEM) Cloud Control 13c Version 5 und höher. Amazon RDS for Oracle installiert OMA, das daraufhin mit Ihrem Oracle Management Service (OMS) kommuniziert, um Überwachungsinformationen bereitzustellen. Wenn Sie OMS 13.5 ausführen, können Sie Datenbanken verwalten, indem Sie OMA 13.5 installieren. Weitere Informationen finden Sie unter Oracle Management Agent for Enterprise Manager Cloud Control.	7. Juli 2021

Amazon RDS for Oracle unterstützt das Herunterladen von Protokollen aus Amazon S3	Wenn sich archivierte Redo-Logs nicht auf Ihrer Instance befinden, sondern durch den Aufbewahrungszeitraum für das Backup geschützt sind, können Sie `rdsadmin.rdsadmin_archive_log_download` verwenden, um sie von Amazon S3 herunterzuladen. RDS for Oracle speichert die Protokolle im `/rdsdbdata/log/arch` -Verzeichnis auf Ihrer DB-Instance. Weitere Informationen finden Sie unter Herunterladen archivierter Redo-Logs aus Amazon S3.	2. Juli 2021
Amazon RDS unterstützt die MySQL-Versionen 10.4.18 und 10.5.9	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MariaDB-Versionen 10.4.18 und 10.5.9 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	30. Juni 2021

Amazon RDS for Oracle unterstützt Datenbank-Aktivitäts-Streams	Sie können jetzt eine Oracle DB-Instance mithilfe von Datenbank-Aktivitäts-Streams überwachen. Eine Oracle-Datenbank schreibt Prüfungs-Datensätze in den einheitlichen Prüfungs-Trail. Wenn Sie einen Datenbank-Aktivitätsstream auf einer Oracle-DB-Instance starten, streamt Amazon Kinesis alle Aktivitäten, die den Prüfungs-Richtlinien von Oracle Database entsprechen. Weitere Informationen finden Sie unter Überwachung von Amazon RDS mithilfe von Datenbankaktivitäts-Streams.	23. Juni 2021
Amazon RDS for Oracle führt arbeitsspeicheroptimierte Instance-Klassen ein	Neue Oracle-DB-Instance-Klassen sind für Workloads optimiert, die zusätzlichen Arbeitsspeicher und Speicherplatz I/O pro vCPU benötigen. Weitere Informationen finden Sie unter RDS for Oracle-DB-Instance-Klassen.	23. Juni 2021
Unterstützung für MySQL 8.0.25	Sie können jetzt Amazon-RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0.25 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	18. Juni 2021

Amazon RDS on AWS Outposts unterstützt zusätzliche RDS für PostgreSQL-Versionen	RDS on Outposts unterstützt jetzt die RDS for PostgreSQL Versionen 12.5, 12.6, 13.1 und 13.2. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	28. Mai 2021
Amazon RDS unterstützt die MySQL-Versionen 10.2.37 und 10.3.28	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MariaDB-Versionen 10.2.37 und 10.3.28 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	27. Mai 2021
Amazon RDS for Oracle unterstützt Mandanten-Container-Datenbank (CDB)	Eine mandantenseitige Architektur erlaubt es einer Oracle-Datenbank, eine CDB zu sein. In Oracle Database 19c kann Ihre CDB eine einzige PDB enthalten. Die Benutzererfahrung mit einer PDB ist größtenteils identisch mit der Benutzererfahrung mit einer Nicht-CDB. Weitere Informationen finden Sie unter RDS for Oracle-Architektur.	25. Mai 2021
Amazon RDS on AWS Outposts unterstützt Amazon RDS for SQL Server	RDS auf Outposts unterstützt jetzt Amazon RDS for SQL Server. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	11. Mai 2021

Amazon RDS erweitert Unterstützung für regionsübergreifende automatisierte Backups	Sie können jetzt Amazon RDS-Datenbank-Instances konfigurieren, auf denen Microsoft SQL Server ausgeführt wird, um DB-Snapshots und Transaktionsprotokolle in eine andere AWS Region zu replizieren. Weitere Informationen finden Sie unter Automatisierte Backups in eine andere Region replizieren. AWS	7. Mai 2021
Amazon RDS unterstützt regionsübergreifende automatisierte Backups für verschlüsselte DB-Instances	Sie können jetzt DB-Snapshots und Transaktionsprotokolle für verschlüsselte Amazon RDS Datenbank-Instances mit Oracle oder PostgreSQL in eine andere AWS Region replizieren. Weitere Informationen finden Sie unter Automatisierte Backups in eine andere AWS Region replizieren.	3. Mai 2021
Amazon RDS on AWS Outposts unterstützt die Amazon-Überwachung CloudWatch	RDS on Outposts unterstützt jetzt die CloudWatch Amazon-Überwachung. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	21. April 2021

RDS für PostgreSQL unterstützt AWS Lambda-Funktionen	Sie können jetzt AWS Lambda-Funktionen für Ihre RDS for PostgreSQL-DB-Instances aufrufen. Weitere Informationen finden Sie unter Aufrufen einer AWS -Lambda-Funktion von einem RDS for PostgreSQL DB-Instance.	13. April 2021
RDS for SQL Server unterstützt erweiterte Ereignisse	Sie können erweiterte SQL Server-Ereignisse verwenden, um Informationen zu Debugging und Fehlerbehebung zu erfassen. Weitere Informationen finden Sie unter Verwenden erweiterter Datenereignisse mit Amazon RDS für Microsoft SQL Server.	8. April 2021
Support für MySQL 8.0.23, 5.7.33 und 5.6.51	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen die MySQL-Versionen 8.0.23, 5.7.33 und 5.6.51 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	31. März 2021

Automatisches Rollback auf fehlgeschlagene Amazon RDS for MySQL-Upgrade	Wenn ein DB-Instance-Upgrade von MySQL Version 5.7 auf MySQL Version 8.0 fehlschlägt, rollt Amazon RDS die für das Upgrade durchgeführten Änderungen automatisch zurück. Nach dem Rollback läuft die MySQL-DB-Instance MySQL-Version 5.7. Weitere Informationen finden Sie unter Rollback nach fehlgeschlagenem Upgrade von MySQL 5.7 auf 8.0.	18. März 2021
Amazon RDS unterstützt regionsübergreifende Read Replicas in Opt-in-Regionen	Sie können jetzt DB-Instances in Opt-in-Regionen replizieren. Weitere Informationen finden Sie unter Erstellen einer Read Replica in einer anderen Region. AWS	18. März 2021

Amazon RDS plant, Oracle Database 18c einzustellen	Oracle Database 18c (18.0.0.0) befindet sich auf einem Veraltungspfad. Die Oracle Corporation wird nach diesem Datum keine Patches mehr für Oracle Database 18c bereitstellen. end-of-support Am 1. Juli 2021 plant Amazon RDS, mit dem automatischen Upgrade von Oracle Database 18c Instances auf Oracle Database 19c zu beginnen. Bevor die automatischen Upgrades beginnen, empfehlen wir Ihnen dringend, Ihre vorhandenen Oracle Database 18c Instances manuell auf Oracle Database 19c zu aktualisieren. Weitere Informationen finden Sie unter Vorbereiten für das automatische Upgrade von Oracle Database 18c.	11. März 2021
Amazon RDS unterstützt Oracle Database 11g nicht mehr	Sie können nur DB-Instances für Oracle Database 12c Release 1 (12.1.0.2) und höher erstellen. Wenn Sie über Snapshots von Oracle Database 11g verfügen, führen Sie ein Upgrade auf eine spätere Version durch. Weitere Informationen finden Sie unter Aktualisieren eines Oracle-DB-Snapshots.	11. März 2021

Amazon RDS unterstützt kontinuierliche Backups von DB-Instances in AWS Backup	Sie können jetzt automatische Backups in diesen Backups erstellen AWS Backup und DB-Instances aus diesen Backups bis zu einem bestimmten Zeitpunkt wiederherstellen. Weitere Informationen finden Sie unter Verwendung AWS Backup zur Verwaltung automatisierter Backups.	10. März 2021
Amazon RDS unterstützt Oracle Management Agent (OMA) Version 13.4	Sie können Oracle Management Agent (OMA) Version 13.4. mit Oracle Enterprise Manager (OEM) Cloud Control 13c Version 4 Update 9 verwenden. Amazon RDS for Oracle installiert OMA, das daraufhin mit Ihrem Oracle Management Service (OMS) kommuniziert, um Überwachungsinformationen bereitzustellen. Wenn Sie OMS 13.4 ausführen, können Sie Datenbanken verwalten, indem Sie OMA 13.4 installieren. Weitere Informationen finden Sie unter Oracle Management Agent for Enterprise Manager Cloud Control.	10. März 2021

Erweiterungen des RDS-Proxy-Endpunkts	Sie können zusätzliche Endpunkte erstellen, die jedem RDS-Proxy zugeordnet sind. Das Erstellen eines Endpunkts in einer anderen VPC ermöglicht den VPC-übergreifenden Zugriff für den Proxy. Proxies für Aurora MySQL-Cluster können auch schreibgeschützte Endpunkte haben. Diese Reader-Endpunkte stellen eine Verbindung zu Reader-DB-Instances in den Clustern her und können die Leseskalierbarkeit und Verfügbarkeit für abfragein tensive Anwendungen verbessern. Weitere Informationen zu RDS Proxy finden Sie unter „Verwalten von Verbindungen mit Amazon RDS Proxy“ im Amazon RDS-Benutzerhandbuch oder im Aurora-Benutzerhandbuch.	8. März 2021
Amazon RDS erweitert Unterstützung für regionsübergreifende automatisierte Backups	Sie können jetzt Amazon RDS-Datenbank-Instances konfigurieren, auf denen PostgreSQL ausgeführt wird, um DB-Snapshots und Transaktionsprotokolle in eine andere Region zu replizieren. AWS Weitere Informationen finden Sie unter Automatisierte Backups in eine andere Region replizieren. AWS	8. März 2021

Replikationsfilter für Amazon RDS for MariaDB und MySQL werden in den Regionen China (Peking) und China (Ningxia) unterstützt	Die Replikationsfilterung wird jetzt in den Regionen China (Peking) und China (Ningxia) unterstützt. Weitere Informationen finden Sie unter Konfigurieren von Replikationsfiltern mit MariaDB und Konfigurieren von Replikationsfiltern mit MySQL.	5. März 2021
Amazon RDS unterstützt regionsübergreifende DB-Snapshot-Kopien in Opt-in-Regionen	Sie können jetzt DB-Snapshots in und aus AWS Opt-in-Regionen kopieren. Weitere Informationen finden Sie unter Kopieren von Snapshots zwischen Regionen. AWS	4. März 2021
Amazon RDS for SQL Server unterstützt Always On Availability-Groups für Standard Edition	Wenn Sie eine DB-Instance mit der Multi-AZ-Konfiguration auf SQL Server 2019 für die Standard Edition-Datenbank-Engine erstellen, verwendet RDS automatisch Verfügbarkeitsgruppen. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Microsoft SQL Server.	23. Februar 2021

Amazon RDS for Oracle führt beratungsbezogene Verfahren ein	Das `rdsadmin_util` -Paket enthält die Prozeduren `advisor_task_set_parameter` , `advisor_task_drop` und `dbms_stats_init` . Sie können diese Verfahren verwenden , um Berateraufgaben wie zu ändern, stoppen und reaktiviere `AUTO_STATS_ADVISOR_TASK` . Weitere Informationen finden Sie unter Festlegen von Parametern für Berateraufgaben .	23. Februar 2021
Amazon RDS liefert Failover-Gründe für Multi-AZ-DB-Instances	Sie können jetzt detailliertere Erklärungen sehen, wenn eine Multi-AZ-DB-Instance per Failover zu einer Standby-Replika übergeht. Weitere Informationen finden Sie unter Failover-Prozess für Amazon RDS.	18. Februar 2021
Amazon RDS erweitert Unterstützung für den Export von Snapshots nach Amazon S3	Sie können jetzt DB-Snapshot-Daten zu Amazon S3 in China exportieren. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3.	17. Februar 2021

Replikationsfilter für Amazon RDS for MariaDB und MySQL	Sie können Replikationsfilter für MySQL- und MariaDB-Instances konfigurieren. Replikationsfilter spezifizieren, welche Datenbanken und Tabellen in einer gelesenen Replika repliziert werden. Sie können Listen mit Datensätzen und Tabellen erstellen, die für jedes Lesereplikat ein- oder ausgeschlossen werden sollen. Weitere Informationen finden Sie unter Konfigurieren von Replikationsfiltern mit MariaDB und Konfigurieren von Replikationsfiltern mit MySQL.	12. Februar 2021
RDS für Oracle unterstützt Oracle APEX 20.2v1	Sie können Oracle APEX 20.2.v1 mit allen unterstützten Versionen von Oracle Database verwenden. Weitere Informationen finden Sie unterOracle Application Express.	2. Februar 2021

Amazon RDS for SQL Server unterstützt lokale Instance-Speicher für die tempdb-Datenbank	Sie können jetzt Amazon RDS for SQL Server auf den Instance-Typen Amazon EC2 db.r5d und db.m5d starten, wobei die tempdb-Datenbank für die Verwendung eines Instance-Speichers konfiguriert ist. Durch das lokale Speichern von tempdb-Datendateien und -Protokolldateien können Sie im Vergleich zum Standardspeicher basierend auf Amazon EBS niedrigere Lese- und Schreiblatenzen erzielen. Weitere Informationen finden Sie unter Instance-Speicher-Support für die tempdb-Datenbank in Amazon RDS for SQL Server.	27. Januar 2021

Amazon RDS for PostgreSQL unterstützt pg_partman und pg_cron	Amazon RDS for PostgreSQL unterstützt jetzt die Erweiterungen pg_partman und pg_cron. Weitere Informationen zur pg_partman-Erweiterung finden Sie unter Managing PostgreSQL partitions with the pg_partman extension (Verwaltung von PostgreSQL-Partitionen mit der Erweiterung pg_partman). Weitere Informationen zur Erweiterung pg_cron finden Sie unter Scheduling maintenance with the PostgreSQL pg_cron extension (Planung der Wartung mit der Erweiterung pg_cron).	12. Januar 2021
Amazon RDS unterstützt die Veröffentlichung des Oracle Management Agent-Protokolls in Amazon CloudWatch Logs.	Das Oracle Management Agent-Protokoll besteht aus emctl.log, emdctlj.log, gcagent.log, gcagent_errors.log, emagent.nohup und secure.log. Amazon RDS veröffentlicht jedes dieser Protokolle als separaten CloudWatch Protokollstream. Weitere Informationen finden Sie unter Oracle-Logs in Amazon CloudWatch Logs veröffentlichen.	28. Dezember 2020

Amazon RDS on AWS Outposts unterstützt zusätzliche Datenbankversionen	RDS auf Outposts unterstützt jetzt zusätzliche MySQL- und PostgreSQL-Versionen. Weitere Informationen finden Sie unter Unterstützung von Amazon RDS on AWS Outposts für Amazon RDS-Funktionen.	23. Dezember 2020
Amazon RDS auf AWS Outposts unterstützt Co IPs	RDS on Outposts unterstützt jetzt kundeneigene IP-Adressen (CoIPs). Co IPs stellt über Ihr lokales Netzwerk lokale oder externe Konnektivität zu Ressourcen in Ihren Outpost-Subnetzen bereit. Weitere Informationen finden Sie unter Customer-owned IP addresses for RDS on Outposts (Kundeneigene IP-Adressen für RDS auf Outposts).	22. Dezember 2020

Amazon RDS for Oracle plant Upgrade von 11g BYOL-Instances auf 19c	Am 4. Januar 2021 planen wir, mit der automatischen Aktualisierung aller Editionen von Oracle Database 11g-Instances im Bring Your Own License (BYOL)-Modell auf Oracle Database 19c zu beginnen. Alle 11g-Instances, einschließlich Reserved Instances, werden auf das neueste verfügbare Oracle Release Update (RU) verschoben. Weitere Informationen finden Sie unter Preparing for the automatic upgrade of Oracle 11g BYOL (Vorbereiten des automatischen Oracle 11g BYOL-Upgrades).	11. Dezember 2020
Amazon RDS unterstützt die Replikation automatisierter Backups in eine andere Region AWS	Sie können Ihre Amazon RDS-Datenbank-Instances jetzt so konfigurieren, dass sie Snapshots und Transaktionsprotokolle in eine AWS Zielregion Ihrer Wahl replizieren. Weitere Informationen finden Sie unter Automatisierte Backups in eine andere Region replizieren. AWS	4. Dezember 2020

Amazon RDS for Oracle und Microsoft SQL Server unterstützen eine neue DB-Instance-Klasse	Sie können jetzt die db.r5b-Instance-Klasse verwenden, um Amazon RDS-DB-Instances zu erstellen, auf denen Oracle oder SQL Server ausgeführt wird. Weitere Informationen finden Sie unter Supported DB engines for DB instance classes (Unterstützte DB-Engines für DB-Instance-Klassen).	4. Dezember 2020
Unterstützung für MariaDB 10.2.32	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.2.32 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	25. November 2020
Amazon RDS for SQL Server unterstützt nun die Microsoft Business Intelligence Suite auf dem SQL Server 2019	Sie können jetzt SQL Server Analysis Services, SQL Server Integration Services und SQL Server Reporting Services auf DB-Instances mit der neuesten Hauptversion ausführen. Weitere Informationen finden Sie unter Options for the Microsoft SQL Server database engine (Optionen der Microsoft SQL Server-Datenbank-Engine).	24. November 2020

Amazon RDS for PostgreSQL Version 13 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt jetzt PostgreSQL-Version 13 in der Database Preview-Umgebung. Weitere Informationen finden Sie unter PostgreSQL 13-Versionen.	24. November 2020
Amazon RDS-Performance Insights führt neue Dimensionen ein	Sie können die Datenbanklast entsprechend den Dimensionsgruppen für Datenbanken (PostgreSQL, MySQL und MariaDB), Anwendungen (PostgreSQL) und den Sitzungstyp (PostgreSQL, MySQL und MariaDB) gruppieren. Amazon RDS unterstützt auch die Dimensionen db.name (PostgreSQL, MySQL, and MariaDB), db.application.name (PostgreSQL) und db.session_type.name (PostgreSQL). Weitere Informationen finden Sie unter Top load table (Hauptlast-Tabelle).	24. November 2020
Amazon RDS for MariaDB unterstützt eine neue Hauptversion	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.5 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	23. November 2020

Unterstützung für MySQL 5.6.49	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 5.6.49 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	20. November 2020
Unterstützung für MySQL 5.5.62	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 5.5.62 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	20. November 2020
Performance Insights unterstützt die Analyse von Statistiken zu laufenden PostgreSQL-Abfragen	Sie können jetzt Statistiken laufender Abfragen mit Performance Insights für PostgreSQL-DB-Instances analysieren. Weitere Informationen finden Sie unter Statistics for PostgreSQL (Statistiken für PostgreSQL).	18. November 2020

Amazon RDS erweitert Unterstützung für Speicher-Autoscaling	Sie können jetzt das Speicher-Autoscaling aktivieren, wenn Sie eine Read Replica erstellen, eine DB-Instance auf eine bestimmte Zeit wiederherstellen oder eine MySQL-DB-Instance aus einem Amazon S3-Backup wiederherstellen. Weitere Informationen finden Sie unter Managing capacity automatically with Amazon RDS storage autoscaling (Automatische Kapazitätsverwaltung mit Speicher-Autoscaling).	18. November 2020
Amazon RDS for SQL Server unterstützt Database Mail	Mit Database Mail können Sie E-Mail-Nachrichten von Ihrer Amazon RDS for SQL Server-Datenbank-Instance versenden. Nachdem Sie die E-Mail-Empfänger angegeben haben, können Sie der von Ihnen gesendeten Nachricht Dateien hinzufügen oder Ergebnisse abfragen. Weitere Informationen finden Sie unter Using Database Mail on Amazon RDS for SQL Server (Verwenden von Database Mail auf Amazon RDS für SQL Server).	4. November 2020

Unterstützung für MySQL 8.0.21	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL-Ver sion 8.0.21 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	22. Oktober 2020
Amazon RDS erweitert Unterstützung für den Export von Snapshots nach Amazon S3	Sie können jetzt in allen kommerziellen AWS Regionen DB-Snapshot-Daten nach Amazon S3 exportieren. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3.	22. Oktober 2020
Amazon RDS for PostgreSQL unterstützt Lesereplikat-Upgra des	Wenn Sie mit Amazon RDS for PostgreSQL ein Upgrade der Hauptversion der primären DB-Instance durchführen, werden auch Lesereplikate automatisch aktualisiert. Weitere Informationen finden Sie unter Aktualisieren der PostgreSQL-DB-Engine.	15. Oktober 2020

Amazon RDS for MariaDB, MySQL und PostgreSQL unterstützen die Graviton2 DB-Instance-Klassen	Sie können jetzt die Graviton2 DB-Instance-Klassen db.m6g.x und db.r6g.x verwenden, um Amazon RDS DB-Instances mit MariaDB, MySQL oder PostgreSQ L zu erstellen. Weitere Informationen finden Sie unter Unterstützte DB-Engines für alle verfügbaren DB-Instance-Klassen.	15. Oktober 2020
Amazon RDS for SQL Server unterstützt Upgrades auf SQL Server 2019	Sie können Ihre SQL Server DB-Instances auf SQL Server 2019 aktualisieren. Weitere Informationen finden Sie unter Upgraden der Microsoft SQL Server-DB-Engine.	6. Oktober 2020
Amazon RDS for Oracle unterstützt die Angabe des nationalen Zeichensatzes	Der nationale Zeichensa tz, auch NCHAR-Zei chensatz genannt, wird in den Datentypen `NCHAR`, `NVARCHAR2` und `NCLOB` verwendet. Wenn Sie eine Datenbank erstellen, können Sie entweder AL16 UTF16 (Standard) oder UTF8 den NCHAR-Zeichensatz angeben. Weitere Informationen finden Sie unter Oracle-Zeichensätz e, die in Amazon RDS unterstützt werden.	2. Oktober 2020

Unterstützung für MySQL 5.7.31	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL-Ver sion 5.7.31 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	1. Oktober 2020
Amazon RDS for PostgreSQ L unterstützt den Export von Daten zu Amazon S3	Sie können Daten aus einer PostgreSQL-DB-Instance abfragen und direkt in Dateien exportieren, die in einem Amazon S3-Bucket gespeiche rt sind. Weitere Informationen finden Sie unter Exportieren von Daten aus einer RDS für PostgreSQL DB-Instance zu Amazon S3.	24. September 2020
Amazon RDS for MySQL 8.0 unterstützt Percona XtraBacku p	Sie können jetzt Percona verwenden XtraBackup , um ein Backup in einer Amazon RDS for MySQL 8.0-DB-In stance wiederherzustellen. Weitere Informationen finden Sie unter Wiederherstellen einer Sicherung zu einer MySQL-DB-Instance.	17. September 2020
Amazon RDS for SQL Server unterstützt die native Sicherung und Wiederher stellung auf DB-Instances mit Lesereplikaten	Sie können ein natives SQL Server-Backup auf einer DB-Instance wiederherstellen, für die Read Replicas konfiguri ert sind. Weitere Informati onen finden Sie unter Importier en und Exportieren von SQL Server-Datenbanken.	16. September 2020

Amazon RDS for SQL Server unterstützt zusätzliche Zeitzonen	Sie können Ihre DB-Instance-Zeitzone mit der gewählten Zeitzone abgleichen. Weitere Informationen finden Sie unter Lokale Zeitzone für Microsoft SQL Server-DB-Instances.	11. September 2020
Amazon RDS for PostgreSQL Version 13 Beta 3 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt jetzt die PostgreSQL-Version 13 Beta 3 in der Database Preview-Umgebung. Weitere Informationen finden Sie unter PostgreSQL 13-Versionen.	9. September 2020
Amazon RDS for SQL Server unterstützt Ablaufverfolgungs-Flag 692	Sie können nun das Ablaufverfolgungs-Flag 692 mittels DB-Parametergruppen als Startup-Parameter verwenden. Durch das Aktivieren dieses Ablaufverfolgungs-Flags werden schnelle Einfügungen beim Massenladen von Daten in Heap- oder Cluster-Indizes deaktiviert. Weitere Informationen finden Sie unter Deaktivieren schneller Einfügungen während des Massenladens.	27. August 2020
Amazon RDS for SQL Server unterstützt Microsoft SQL Server 2019	Sie können jetzt RDS-DB-Instances erstellen, die SQL Server 2019 verwenden. Weitere Informationen finden Sie unter Microsoft SQL Server-Versionen auf Amazon RDS.	26. August 2020

RDS for Oracle unterstützt die aufgespielte Replikat-Datenbank	Wenn Sie ein Oracle-Replikat erstellen oder ändern, können Sie es in den aufgespielten Modus versetzen. Da die Replikat-Datenbank keine Benutzerverbindungen akzeptiert, kann sie keinen schreibgeschützten Workload bereitstellen. Das aufgespielte Replikat löscht archivierte Redo-Protokolldateien, nachdem sie angewendet wurden. Die primäre Verwendung für aufgespielte Replikate ist die überregionale Notfallwiederherstellung. Weitere Informationen finden Sie unter Übersicht über Oracle-Replikate.	13. August 2020

RDS for Oracle plant ein Upgrade von 11g LI-Instances SE1	Am 1. November 2020 planen wir, mit dem automatischen Upgrade von Oracle Database 11g SE1 License Included (LI) -Instances auf Oracle Database 19c für Amazon RDS for Oracle zu beginnen. Alle 11g-Instances, einschließlich Reserved Instances, werden auf das neueste verfügbare Oracle Release Update (RU) verschoben. Weitere Informationen finden Sie unter Vorbereitung für das automatische Upgrade von Oracle Database 11g. SE1	31. Juli 2020
Amazon RDS unterstützt neue Graviton2 DB-Instance-Klassen in Vorversion für PostgreSQL und MySQL	Sie können jetzt Amazon RDS-DB-Instances erstellen, die PostgreSQL oder MySQL ausführen, bei denen die DB-Instance-Klassen db.m6g.x und db.r6g.x verwendet werden. Weitere Informationen finden Sie unter Unterstützte DB-Engines für alle verfügbaren DB-Instance-Klassen.	30. Juli 2020
RDS für Oracle unterstützt Oracle APEX 20.1v1	Sie können Oracle APEX 20.1v1 mit allen unterstützten Versionen von Oracle Database verwenden. Weitere Informationen finden Sie unter Oracle Application Express.	28. Juli 2020

Unterstützung für MySQL 8.0.20	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0.20 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	23. Juli 2020
Amazon RDS for MariaDB und MySQL unterstützen neue DB-Instance-Klassen	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MariaDB und MySQL mit den DB-Instance-Klassen db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge und db.r5.8xlarge ausgeführt werden. Weitere Informationen finden Sie unter Unterstützte DB-Engines für alle verfügbaren DB-Instance-Klassen.	23. Juli 2020
RDS for SQL Server unterstützt das Deaktivieren alter Versionen von TLS und Verschlüsselungen	Sie können bestimmte Sicherheitsprotokolle und Verschlüsselungen ein- und ausschalten. Weitere Informationen finden Sie unter Konfigurieren von Sicherheitsprotokollen und Verschlüsselungen.	21. Juli 2020
RDS unterstützt Oracle Spatial auf SE2	Sie können Oracle Spatial in Standard Edition 2 (SE2) für alle Versionen von 12.2, 18c und 19c verwenden. Weitere Informationen finden Sie unter Oracle Spatial.	9. Juli 2020

Amazon RDS unterstützt AWS PrivateLink	Amazon RDS unterstützt jetzt die Erstellung von Amazon VPC-Endpunkten für Amazon RDS-API-Aufrufe, um den Verkehr zwischen Anwendungen und Amazon RDS im AWS Netzwerk aufrechtzuerhalten. Weitere Informationen finden Sie unter Amazon RDS und Schnittstellen-VPC-Endpunkte (AWS PrivateLink).	9. Juli 2020
Die Versionen 9.4.x von Amazon RDS für PostgreSQL haben das Ende des Supports erreicht.	Amazon RDS for PostgreSQL unterstützt die Versionen 9.4.x nicht mehr. Informationen zu unterstützten Versionen finden Sie unter Unterstützte PostgreSQL-Datenbankversionen.	8. Juli 2020
Unterstützung für MariaDB 10.3.23 und 10.4.13	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen die MariaDB-Versionen 10.3.23 und 10.4.13 ausgeführt werden. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	6. Juli 2020
Amazon RDS auf AWS Outposts	Sie können auf Amazon RDS-DB-Instances auf AWS Outposts erstellen. Weitere Informationen finden Sie unter Arbeiten mit Amazon RDS auf AWS Outposts.	6. Juli 2020

Amazon RDS for Oracle erstellt Bestandsdateien automatisch	Um Serviceanfragen für BYOL-Kunden zu öffnen, fordert Oracle Support Bestandsdateien an, die von Opatch generiert wurden. Amazon RDS for Oracle erstellt automatisch stündlich Inventardateien im BDUMP-Verzeichnis. Weitere Informationen finden Sie unter Zugreifen auf Opatch-Dateien.	6. Juli 2020
Unterstützung für MySQL 5.7.30 und 5.6.48	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen die MySQL-Versionen 5.7.30 und 5.6.48 ausgeführt werden. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	25. Juni 2020

Amazon RDS for Oracle unterstützt ADRCI	Das Dienstprogramm „Automatic Diagnostic Repository Command Interpreter“ (ADRCI) ist ein Oracle-Befehlszeilentool für die Verwaltung von Diagnosedaten. Durch die Verwendung der Funktionen im Amazon RDS-Paket `rdsadmin_adrci_util` können Sie Probleme und Vorfälle auflisten und zusammenfassen sowie Ablaufverfolgungsdateien anzeigen. Weitere Informationen finden Sie unter Allgemeine DBA-Diagnoseaufgaben für Oracle DB-Instances.	17. Juni 2020
Unterstützung für MySQL 8.0.19	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0.19 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	2. Juni 2020
MySQL 8.0 unterstützt kleingeschriebene Tabellennamen	Sie können nun den `lower_case_table_names` -Parameter für Amazon RDS-DB-Instances mit MySQL Version 8.0.19 und höher 8.0 auf 1 setzen. Weitere Informationen finden Sie unter MySQL-Parameterausnahmen für Amazon RDS-DB-Instances.	2. Juni 2020

Amazon RDS for Microsoft SQL Server unterstützt SQL Server Integration Services (SSIS)	SSIS ist eine Plattform für Datenintegrations- und Workflow-Anwendungen. Sie können SSIS auf vorhandenen oder neuen DB-Instances aktivieren. Es wird auf derselben DB-Instance wie Ihre Datenbank-Engine installiert. Weitere Informationen finden Sie unter Unterstützung für SQL Server Integration Services in SQL Server.	19. Mai 2020
Amazon RDS for Microsoft SQL Server unterstützt SQL Server Reporting Services (SSRS)	SSRS ist eine serverbasierte Anwendung, die für die Erstellung und Verteilung von Berichten verwendet wird. Sie können SSRS auf vorhandenen oder neuen DB-Instances aktivieren. Es wird auf derselben DB-Instance wie Ihre Datenbank-Engine installiert. Weitere Informationen finden Sie unter Unterstützung für SQL Server Reporting Services in SQL Server.	15. Mai 2020

Amazon RDS for Microsoft SQL Server unterstützt die S3-Integration auf Multi-AZ-Instances	Sie können jetzt Amazon S3 mit SQL Server-Funktionen wie etwa der Masseneinfügung auf Multi-AZ-DB-Instances verwenden. Weitere Informationen finden Sie unter Integration einer Amazon RDS für SQL Server-DB-Instance mit Amazon S3.	15. Mai 2020
Amazon RDS for Oracle unterstützt das Bereinigen des Papierkorbs	Das `rdsadmin.rdsadmin_util.purge_dba_recyclebin` -Verfahren bereinigt den Papierkorb. Weitere Informationen finden Sie unter Bereinigen des Papierkorbs.	13. Mai 2020
Amazon RDS for Oracle verbessert die Verwaltbarkeit von Automatic Workload Repository (AWR)	Die `rdsadmin.rdsadmin_diagnostic_util` -Verfahren generieren AWR-Berichte und extrahieren AWR-Daten in Dump-Dateien. Weitere Informationen finden Sie unter Generieren von Leistungsberichten mit Automatic Workload Repository (AWR).	13. Mai 2020
Amazon RDS for Microsoft SQL Server unterstützt Microsoft Distributed Transaction Coordinator (MSDTC)	Amazon RDS for SQL Server unterstützt verteilte Transaktionen zwischen Hosts. Weitere Informationen finden Sie unter Unterstützung für Microsoft Distributed Transaction Coordinator in SQL Server.	4. Mai 2020

Amazon RDS for Microsoft SQL Server unterstützt neue Versionen	Sie können jetzt Amazon RDS-DB-Instances mit den SQL Server-Versionen 2017 CU19 14.00.3281.6, 2016 SP2 CU11 13.00.5598.27, 2014 12.00.6329.1 und 2012 GDR 11.0.7493.4 für alle Editionen erstellen. SP3 CU4 SP4 Weitere Informationen finden Sie unter Microsoft SQL Server-Versionen auf Amazon RDS.	28. April 2020
Amazon RDS in der Region Europa (Mailand) verfügbar	Amazon RDS ist jetzt in der Region Region Europa (Mailand) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	28. April 2020
Amazon RDS-Unterstützung für Local Zones	Sie können jetzt DB-Instances in einem Local Zones-Subnetz starten. Weitere Informationen finden Sie unter Regionen, Availability Zones und Local Zones.	23. April 2020
Amazon RDS in der Region Afrika (Kapstadt) verfügbar	Amazon RDS ist jetzt in der Region Region Afrika (Kapstadt) verfügbar. Weitere Informationen finden Sie unter Regionen und Availability Zones.	22. April 2020

Amazon RDS for Microsoft SQL Server unterstützt SQL Server Analysis Services (SSAS)	SSAS ist ein Online Analytical Processing (OLAP)- und Data Mining-Tool, das in SQL Server installiert ist. Sie können SSAS auf vorhandenen oder neuen DB-Instances aktivieren. Es wird auf derselben DB-Instance wie Ihre Datenbank-Engine installiert. Weitere Informationen finden Sie unter Unterstützung für SQL Server Analysis Services in SQL Server.	17. April 2020
Amazon RDS-Proxy für PostgreSQL	Amazon RDS-Proxy ist jetzt für PostgreSQL verfügbar. Sie können RDS-Proxy verwenden, um den Overhead der Verbindungsverwaltung in Ihrer DB-Instance zu reduzieren und auch die Wahrscheinlichkeit des Fehlers „zu viele Verbindungen“ zu reduzieren. Der RDS-Proxy ist derzeit als öffentliche Vorversion für PostgreSQL verfügbar. Weitere Informationen finden Sie unter Verwalten von Verbindungen mit Amazon RDS Proxy (Vorversion).	8. April 2020

Amazon RDS for Oracle unterstützt Oracle APEX Version 19.2.v1	Amazon RDS for Oracle unterstützt jetzt Oracle Application Express (APEX) Version 19.2.v1. Weitere Informationen finden Sie unter Oracle Application Express.	8. April 2020
Amazon RDS for MariaDB unterstützt eine neue Hauptversion	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit MariaDB-Version 10.4 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	6, 2020. April 2020
Amazon RDS Performance Insights ist für Amazon RDS for MariaDB 10.4 verfügbar	Amazon RDS Performance Insights ist jetzt Amazon RDS for MariaDB Version 10.4 verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	6, 2020. April 2020
Die Versionen 9.3.x von Amazon RDS für PostgreSQL haben das Ende des Supports erreicht	Amazon RDS for PostgreSQL unterstützt die Versionen 9.3.x nicht mehr. Informationen zu unterstützten Versionen finden Sie unter Unterstützte PostgreSQL-Datenbankversionen.	3. April 2020
Amazon RDS for Microsoft SQL Server unterstützt Lesereplikate	Sie können jetzt Lesereplikate für SQL Server-DB-Instances erstellen. Weitere Informationen finden Sie unter Arbeiten mit Lesereplikaten.	3. April 2020

Amazon RDS for Microsoft SQL Server unterstützt Backups mit mehreren Dateien	Sie können Datenbanken nun mit den systemeigenen Backups und Wiederher stellungen von SQL Server in mehreren Dateien sichern. Weitere Informationen finden Sie unter Sichern einer Datenbank.	2. April 2020
Integration von Amazon RDS for Oracle mit AWS License Manager	Amazon RDS for Oracle ist jetzt in integriert AWS License Manager. Wenn Sie das Bring Your Own License-M odell verwenden, erleichtert die AWS License Manager Integration die Überwachung Ihrer Oracle-Lizenznutzung in Ihrem Unternehmen. Weitere Informationen finden Sie unter Integrieren mit AWS License Manager.	23. März 2020
Unterstützung für 64 TiB auf Instances vom Typ „db.r5“ in Amazon RDS for MariaDB und MySQL	Sie können jetzt Amazon RDS-DB-Instances for MariaDB und MySQL erstellen , die die DB-Instance-Klasse vom Typ „db.r5“ mit bis zu 64 TiB Speicher verwenden. Weitere Informationen finden Sie unter Faktoren, die die Speicherleistung beeinflussen.	18. März 2020

Unterstützung für MySQL 8.0.17	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0.17 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	10. März 2020
Amazon RDS Performance Insights ist für Amazon RDS for MySQL 8.0 verfügbar	Amazon RDS Performance Insights ist ab sofort für Amazon RDS for MySQL-Version 8.0.17 und höheren 8.0 Versionen verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	10. März 2020
Unterstützung für MySQL 5.6.46	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 5.6.46 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	28. Februar 2020
Amazon RDS Performance Insights ist für Amazon RDS for MariaDB 10.3 verfügbar	Amazon RDS Performance Insights ist ab sofort für Amazon RDS for MariaDB-Version 10.3.13 und höheren 10.3 Versionen verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	26. Februar 2020

Unterstützung für MySQL 5.7.28	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 5.7.28 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	20. Februar 2020
Unterstützung für MariaDB 10.3.20	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.3.20 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	20. Februar 2020
Amazon RDS for Microsoft SQL Server unterstützt eine neue DB-Instance-Klasse	Sie können jetzt Amazon RDS-DB-Instances mit SQL Server erstellen, die die DB-Instance-Klasse vom Typ „db.z1d” verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse nunterstützung für Microsoft SQL Server.	19. Februar 2020
Unterstützung von kontoüber greifenden, VPC-überg reifenden Active Directory-Domänen in Amazon RDS for SQL Server	Amazon RDS for Microsoft SQL Server unterstützt jetzt die Verknüpfung von DB-Instances mit Active Directory-Domänen, die unterschiedlichen Konten gehören, und VPCs. Weitere Informationen finden Sie unter Verwenden der Windows-Authentifizierung mit einer Microsoft SQL Server-DB-Instance.	13. Februar 2020

Oracle OLAP-Option	Amazon RDS for Oracle unterstützt jetzt die Online Analytical Processing (OLAP)-Option für Oracle-DB-Instances. Mit Oracle OLAP können Sie große Datenmengen analysieren, indem Sie Dimensionsobjekte und Cubes gemäß dem OLAP-Standard erstellen. Weitere Informationen finden Sie unter Oracle OLAP.	13. Februar 2020
Unterstützung von FIPS 140-2 für Oracle	Amazon RDS for Oracle unterstützt die Federal Information Processing Standard Publication 140-2 (FIPS 140-2) für Verbindungen. SSL/TLS Weitere Informationen finden Sie unter FIPS-Unterstützung.	11. Februar 2020
Amazon RDS for PostgreSQL unterstützt neue DB-Instance-Klassen	Sie können jetzt Amazon RDS-DB-Instances mit PostgreSQL erstellen, die die DB-Instance-Klassen db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge und db.r5.8xlarge verwenden. Weitere Informationen finden Sie unter Unterstützte DB-Engines für alle verfügbaren DB-Instance-Klassen.	11. Februar 2020

Performance Insights unterstützt die Analyse von Statistiken zu laufenden MariaDB- und MySQL-Abfragen	Sie können jetzt Statistiken laufender Abfragen mit Performance Insights für MariaDB- und MySQL-DB-Instances analysieren. Weitere Informationen finden Sie unter Analysieren von Statistiken zu laufenden Abfragen.	4. Februar 2020
Unterstützung für den Export von DB-Snapshot-Daten nach Amazon S3 für MariaDB, MySQL und PostgreSQL	Amazon RDS unterstützt den Export von DB-Snapshot-Daten nach Amazon S3 für MariaDB, MySQL und PostgreSQL. Weitere Informationen finden Sie unter Exportieren von DB-Snapshot-Daten nach Amazon S3.	23. Januar 2020
Amazon RDS for MySQL unterstützt die Kerberos-Authentifizierung	Sie können jetzt die Kerberos-Authentifizierung verwenden, um Benutzer zu authentifizieren, wenn sie sich mit Ihren Amazon RDS for MySQL DB-Instances verbinden. Weitere Informationen finden Sie unter Verwenden der Kerberos-Authentifizierung für MySQL.	21. Januar 2020

Amazon RDS Performance Insights unterstützt das Anzeigen von mehr SQL-Text für Amazon RDS for Microsoft SQL Server	Amazon RDS Performance Insights unterstützt nun die Anzeige von mehr SQL-Text im Performance Insights-Dashboard für Amazon RDS for Microsoft SQL Server-DB-Instances. Weitere Informationen finden Sie unter Anzeigen von mehr SQL-Text im Performance-Insights-Dashboard.	17. Dezember 2019
Amazon RDS Proxy	Sie können den Overhead für die Verbindungsverwaltung in Ihrem Cluster reduzieren und die Wahrscheinlichkeit des Fehlers "zu viele Verbindungen" reduzieren, indem Sie Amazon RDS Proxy verwenden. Sie ordnen jeden Proxy einer RDS DB-Instance oder einem Aurora DB-Cluster zu. Dann verwenden Sie den Proxy-Endpunkt in der Verbindungszeichenfolge für Ihre Anwendung. Der Amazon RDS-Proxy ist derzeit als öffentliche Vorversion verfügbar. Es unterstützt die RDS for MySQL-Datenbank-Engine. Weitere Informationen finden Sie unter Verwalten von Verbindungen mit Amazon RDS Proxy (Vorversion).	3. Dezember 2019

Amazon RDS aktiviert AWS Outposts (Vorschau)	Wenn Amazon RDS aktiviert ist AWS Outposts, können Sie AWS verwaltete relationale Datenbanken in Ihren lokalen Rechenzentren erstellen. Mit RDS on Outposts können Sie RDS-Datenbanken auf AWS Outposts ausführen. Weitere Informationen finden Sie Amazon RDS on AWS Outposts (Vorschau).	03. Dezember 2019
Amazon RDS for Oracle unterstützt regionenübergreifende Lesereplikate	Amazon RDS for Oracle unterstützt jetzt regionenübergreifende Lesereplikate mit Active Data Guard. Weitere Informationen finden Sie unter Arbeiten mit Lesereplikaten und Arbeiten mit Oracle-Lesereplikaten.	26. November 2019
Performance Insights unterstützt die Analyse von Statistiken zu laufenden Oracle-Abfragen	Sie können jetzt Statistiken laufender Abfragen mit Performance Insights für Oracle-DB-Instances analysieren. Weitere Informationen finden Sie unter Analysieren von Statistiken zu laufenden Abfragen.	25. November 2019

Amazon RDS for Microsoft SQL Server unterstützt das Veröffentlichen von Protokollen in CloudWatch Logs	Sie können Ihre Amazon RDS for SQL Server-DB-Instance so konfigurieren, dass Protokollereignisse direkt in Amazon CloudWatch Logs veröffentlicht werden. Weitere Informationen finden Sie unter Veröffentlichen von SQL Server-Protokollen in Amazon CloudWatch Logs.	25. November 2019
Amazon RDS for Microsoft SQL Server unterstützt neue DB-Instance-Klassen	Sie können jetzt Amazon RDS-DB-Instances mit SQL Server erstellen, die die DB-Instance-Klassen db.x1e und db.x1 verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse nunterstützung für Microsoft SQL Server.	25. November 2019
Amazon RDS for Microsoft SQL Server unterstützt differentielle und Protokoll wiederherstellungen	Sie können differentielle Backups und Protokolle mit der nativen Sicherungs- und Wiederherstellungsfunktion von SQL Server wiederher stellen. Weitere Informationen finden Sie unter Verwendung der nativen Sicherungs- und Wiederherstellungsfunktion.	25. November 2019

Multi-AZ-Unterstützung auf Amazon RDS for Microsoft SQL Server in neuen Regionen	Multi-AZ auf SQL Server ist jetzt verfügbar in China, Naher Osten (Bahrain) und Europa (Stockholm). Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Microsoft SQL Server.	22. November 2019
Amazon RDS for Microsoft SQL Server unterstützt jetzt die Masseneinfügung und die S3-Integration	Sie können Dateien zwischen einer SQL Server-DB-Instance und einem Amazon S3-Bucket übertragen. Dann können Sie Amazon S3 mit SQL Server-Funktionen wie etwa der Masseneinfügung verwenden . Weitere Informationen finden Sie unter Integration einer Amazon RDS für SQL Server-DB-Instance mit Amazon S3.	21. November 2019
Performance-Insights-Zähler für Amazon RDS for Microsoft SQL Server	Sie können jetzt Performan ce-Zähler zu Ihren Performan ce Insights-Diagrammen für Microsoft SQL Server-DB-Instances hinzufügen. Weitere Informationen finden Sie unter Performance-Insights-Zähler für Amazon RDS for Microsoft SQL Server.	12. November 2019

Amazon RDS for Microsoft SQL Server unterstützt jetzt neue DB-Instance-Klasse ngrößen	Sie können jetzt Amazon RDS-DB-Instances mit SQL Server erstellen, die die Größen 8xlarge und 16xlarge für die DB-Instance-Klassen db.m5 und db.r5 verwenden . Instance-Größen von small bis 2xlarge sind jetzt für die Instance-Klasse db.t3 verfügbar. Weitere Informati onen finden Sie unter DB-Instance-Klassenunterstützung für Microsoft SQL Server.	11. November 2019
Unterstützung für PostgreSQL-Snapshot-Upgrades	Wenn Sie bereits über manuelle DB-Snapshots Ihrer Amazon RDS PostgreSQ L-DB-Instances verfügen, können Sie diese jetzt auf eine neuere Version der PostgreSQ L-Datenbank-Engine aktualisi eren. Weitere Informationen finden Sie unter Aktualisieren eines PostgreSQL-DB-Snap shots.	7. November 2019
Amazon RDS for Oracle unterstützt eine neue Hauptversion	Sie können nun Amazon RDS DB-Instances erstellen, auf denen Oracle Database 19c (19.0) ausgeführt wird. Weitere Informationen finden Sie unter Oracle Database 19c mit Amazon RDS.	7. November 2019

Amazon RDS for PostgreSQL Version 12.0 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt jetzt PostgreSQL-Version 12.0 in der Database Preview-Umgebung Weitere Informationen finden Sie unter PostgreSQL-Version 12.0 in der Database Preview-Umgebung.	1. November 2019
Amazon RDS for PostgreSQL unterstützt die Kerberos-Authentifizierung	Sie können nun die Kerberos-Authentifizierung verwenden, um Benutzer zu authentifizieren, wenn sie sich mit Ihrer Amazon RDS DB-Instance mit PostgreSQL verbinden. Weitere Informationen finden Sie unter Verwendung der Kerberos-Authentifizierung mit Amazon RDS für PostgreSQL.	28. Oktober 2019
OEM Management-Agent-Datenbankaufgaben für Oracle DB-Instances	Amazon RDS for Oracle DB-Instances unterstützen nun Verfahren zum Aufrufen bestimmter EMCTL-Befehle im Management-Agent. Weitere Informationen finden Sie unter OEM-Agent-Datenbankaufgaben.	24. Oktober 2019

Amazon RDS for PostgreSQL unterstützt PostgreSQL-Transportdatenbanken	PostgreSQL-Transportdatenbanken bieten eine extrem schnelle Methode zur Migration einer RDS PostgreSQL-Datenbank zwischen zwei DB-Instances. Weitere Informationen finden Sie unter Transport von PostgreSQL-Datenbanken zwischen DB-Instances.	8. Oktober 2019
Amazon RDS for Oracle unterstützt die Kerberos-Authentifizierung	Sie können jetzt die Kerberos-Authentifizierung verwenden, um Benutzer zu authentifizieren, wenn diese sich mit Ihrer Amazon RDS-DB-Instance auf Oracle verbinden. Weitere Informationen finden Sie unter Verwendung der Kerberos-Authentifizierung mit Amazon RDS für Oracle.	30. September 2019
Amazon RDS for PostgreSQL Version 12 Beta 3 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt nun PostgreSQL Version 12 Beta 3 in der Database Preview-Umgebung. Weitere Informationen finden Sie unter PostgreSQL-Version 12 Beta 3 auf Amazon RDS in der Database Preview-Umgebung.	28. August 2019

Unterstützung für MySQL 8.0.16	Sie können nun Amazon RDS DB-Instances mit MySQL-Version 8.0.16 erstellen. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	19. August 2019
Amazon RDS for Oracle unterstützt eine neue Hauptversion	Sie können nun Amazon RDS DB-Instances erstellen, auf denen Oracle Database 18c (18.0) ausgeführt wird. Weitere Informationen finden Sie unter Oracle 18c mit Amazon RDS.	15. August 2019
Management Agent for OEM 13c Release 3	Die Amazon RDS for Oracle-DB-Instances unterstützen jetzt den Management Agent für Oracle Enterprise Manager (OEM) Cloud Control 13c Release 3. Weitere Informationen finden Sie unter Oracle Management Agent für Enterprise Manager Cloud Control.	7. August 2019
Amazon RDS for PostgreSQL Version 12 Beta 2 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt jetzt die PostgreSQL-Version 12 Beta 2 in der Database Preview-Umgebung Weitere Informationen finden Sie unter PostgreSQL-Version 12 Beta 2 auf Amazon RDS in der Database Preview-Umgebung.	6. August 2019

Amazon RDS unterstützt Server-basierte Sortierungen für SQL Server	Amazon RDS for SQL Server unterstützt eine Reihe von Sortierungen für neue DB-Instances. Weitere Informationen finden Sie unter Sortierungen und Zeichensätze für Microsoft SQL Server.	29. Juli 2019
Amazon RDS for Oracle unterstützt Oracle APEX Version 19.1.v1	Amazon RDS for Oracle unterstützt jetzt Oracle Application Express (APEX) Version 19.1.v1. Weitere Informationen finden Sie unter Oracle Application Express.	28. Juni 2019
Amazon RDS for PostgreSQL Version 13 Beta 1 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt jetzt die PostgreSQL-Version 13 Beta 1 in der Database Preview-Umgebung Weitere Informationen finden Sie unter PostgreSQL 13-Versionen.	22. Juni 2019
Automatische Amazon RDS-Speicherskalierung	Durch automatische Speicherskalierung für Amazon RDS-DB-Instances kann Amazon RDS den mit einer DB-Instance verknüpften Speicher automatisch erweitern, um die Wahrscheinlichkeit von Störungen zu verringern out-of-space. Informationen zur automatischen Skalierung von Speichern finden Sie unter Arbeiten mit Speicher für Amazon RDS DB-Instances.	20. Juni 2019

Amazon RDS for Oracle unterstützt db.z1d DB-Instance-Klassen	Sie können jetzt Amazon RDS-DB-Instances mit Oracle erstellen, die DB-Instance-Klassen vom Typ „db.z1d” verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	13. Juni 2019
Amazon RDS Performance Insights unterstützt die Anzeige von mehr SQL-Text für Amazon RDS for Oracle	Amazon RDS Performance Insights unterstützt nun die Anzeige von mehr SQL-Text im Performance Insights-Dashboard für Amazon RDS for Oracle DB-Instances. Weitere Informationen finden Sie unter Anzeigen von mehr SQL-Text im Performance-Insights-Dashboard.	10. Juni 2019
Amazon RDS fügt Unterstützung für native Wiederherstellungen von SQL Server-Datenbanken mit einer Größe bis zu 16 TB hinzu	Sie können nun native Wiederherstellungen mit bis zu 16 TB von SQL Server zu Amazon RDS ausführen. Weitere Informationen finden Sie unter Amazon RDS für SQL Server: Grenzen und Empfehlungen.	4. Juni 2019

Amazon RDS fügt Unterstützung für Microsoft SQL Server-Audit hinzu	Mit Amazon RDS for Microsoft SQL Server können Sie Ereignisse auf Server- und Datenbankebene mit SQL Server Audit prüfen und die Ergebnisse auf Ihrer DB-Instance anzeigen oder die Audit-Protokolldateien direkt an Amazon S3 senden. Weitere Informationen finden Sie unter SQL Server Audit.	23. Mai 2019
Verbesserungen der Amazon RDS-Empfehlungen	Amazon RDS bietet verbesserte automatisierte Empfehlungen für Datenbankressourcen. So stellt Amazon RDS nun beispielsweise Empfehlungen für Datenbankparameter bereit. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Empfehlungen.	22. Mai 2019
Unterstützung für mehrere Datenbanken pro Instance für Amazon RDS for SQL Server	Sie können auf jeder Ihrer DB-Instances, die Microsoft SQL Server ausführen, bis zu 30 Datenbanken erstellen. Weitere Informationen finden Sie unter Begrenzungen für Microsoft SQL Server DB-Instances.	21. Mai 2019

Support für 64 TiB- und 80k-IOPS-Speicher für Amazon RDS for MariaDB, MySQL und PostgreSQL	Sie können nun Amazon RDS-DB-Instances für MariaDB, MySQL und PostgreSQL mit einem Speicher von 64 TiB und bis zu 80.000 bereitgestellten IOPS erstellen. Weitere Informationen finden Sie unter DB-Instance-Speicher.	20. Mai 2019
Amazon RDS for MySQL unterstützt Upgrade-Vorabprüfungen	Wenn Sie eine DB-Instance von MySQL 5.7 auf MySQL 8.0 upgraden, führt Amazon RDS vorab eine Prüfung auf mögliche Inkompatibilitäten durch. Weitere Informationen finden Sie über Vorabprüfungen bei Upgrades von MySQL 5.7 auf 8.0.	17. Mai 2019
Support für das MySQL-Plugin für die Passwortvalidierung	Sie können nun das `validate_password` -Plugin für eine verbesserte Sicherheit von Amazon RDS for MySQL-DB-Instances verwenden. Weitere Informationen finden Sie unter Verwenden des Plugin für die Passwortvalidierungn.	16. Mai 2019
Performance-Insights-Zähler für Amazon RDS for Oracle	Sie können nun Leistungszähler zu Ihren Performance Insights-Diagrammen für Oracle-DB-Instances hinzufügen. Weitere Informationen finden Sie unter Performance-Insights-Zähler for Amazon RDS for Oracle.	8. Mai 2019

Unterstützung der sekundengenauen Abrechnung	Amazon RDS wird jetzt in allen AWS Regionen außer AWS GovCloud (USA) für On-Demand-Instances in 1-Sekunden-Schritten abgerechnet. Weitere Informationen finden Sie unter Abrechnung von DB-Instances für Amazon RDS.	25. April 2019
Unterstützung des Imports von Daten von Amazon S3 für Amazon RDS for PostgreSQL	Sie können nun Daten von einer Amazon S3-Datei in eine Tabelle auf einer RDS PostgreSQL-DB-Instance importieren. Weitere Informationen finden Sie auf der Seite über Importieren von Amazon S3-Daten in eine RDS PostgreSQL-DB-Instance.	24. April 2019
Unterstützung der Wiederherstellung von 5.7-Backups aus Amazon S3	Sie können jetzt ein Backup Ihrer MySQL Version 5.7-Datenbank erstellen und auf Amazon S3 speichern und die Sicherungsdatei anschließend auf einer neuen Amazon RDS-DB-Instance mit MySQL wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellen einer Sicherung zu einer MySQL-DB-Instance.	17. April 2019

Unterstützung für mehrere Hauptversions-Upgrades für Amazon RDS for PostgreSQL	Bei Amazon RDS for PostgreSQL können Sie beim Upgrade der DB-Engine jetzt unter mehreren Hauptversionen wählen. Diese Funktion ermöglicht Ihnen, zu einer neueren Version zu springen, wenn Sie ausgewählte PostgreSQL-Engine-Versionen aktualisieren. Weitere Informationen finden Sie unter Aktualisieren der PostgreSQL-DB-Engine.	16. April 2019
Unterstützung für 64 TiB Speicherplatz für Amazon RDS for Oracle	Sie können jetzt Amazon RDS-DB-Instances für Oracle mit bis zu 64 TiB Speicherplatz und bis zu 80.000 bereitgestellten IOPS erstellen. Weitere Informationen finden Sie unter DB-Instance-Speicher.	4. April 2019
Unterstützung für MySQL 8.0.15	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0.15 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	3. April 2019
Unterstützung für MariaDB 10.3.13	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.3.13 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	3. April 2019

Microsoft SQL Server 2008 R2 hat das Ende des Supports für Amazon RDS erreicht	Microsoft SQL Server 2008 R2 hat das Ende seines Supports erreicht, was mit dem Plan von Microsoft zusammenfällt, den erweiterten Support für diese Version am 9. Juli 2019 zu beenden. Alle vorhanden en Microsoft SQL Server 2008 R2-Snapshots sollen am 1. Juni 2019 automatisch auf die neueste Nebenversion von Microsoft SQL Server 2012 aktualisiert werden. Weitere Informationen finden Sie unter Microsoft SQL Server 2008 R2-Unterstützung auf Amazon RDS.	2. April 2019
Unterstützung von AlwaysOn-Verfügbarkeitsgruppen in Microsoft SQL Server 2017	Sie können nun AlwaysOn-Verfügbarkeitsgruppen in SQL Server 2017 Enterpris e Edition 14.00.3049.1 oder höher verwenden. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Microsoft SQL Server.	29. März 2019

Anzeigen von Volumemetriken	Sie können jetzt Metriken für die Amazon Elastic Block Store (Amazon EBS)-Volumes anzeigen. Dabei handelt es sich um physische Geräte, die für die Datenbank - und Protokollspeicherung verwendet werden. Weitere Informationen finden Sie unter Anzeigen von Enhanced Monitoring.	20. März 2019
Unterstützung für MySQL 5.7.25	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen MySQL-Ver sion 5.7.25 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	19. März 2019
Amazon RDS for Oracle unterstützt RMAN-DBA-Aufgaben	Amazon RDS for Oracle unterstützt jetzt Oracle Recovery Manager (RMAN)-DBA-Aufgaben, einschlie ßlich RMAN-Backups. Weitere Informationen finden Sie unter Geläufige DBA Recovery Manager (RMAN)-Aufgaben für Oracle DB-Instances.	14. März 2019
Amazon RDS for PostgreSQL unterstützt Version 11.1	Sie können jetzt Amazon RDS-DB-Instances erstellen , die PostgreSQL Version 11.1 ausführen. Weitere Informationen finden Sie unter PostgreSQL Version 11.1 auf Amazon RDS.	12. März 2019

Wiederherstellung mehrerer Dateien in Amazon RDS for SQL Server verfügbar	Sie können nun mehrere Dateien gleichzeitig mit Amazon RDS for SQL Server wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellen einer Datenbank.	11. März 2019
MariaDB 10.2.21	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit der MariaDB-V ersion 10.2.21 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	11. März 2019
Amazon RDS for Oracle unterstützt Lesereplikate	Amazon RDS for Oracle unterstützt jetzt Leserepli kate mit Active Data Guard. Weitere Informationen finden Sie unter Arbeiten mit Lesereplikaten und Arbeiten mit Oracle-Lesereplikaten.	11. März 2019
Amazon RDS Performance Insights ist für Amazon RDS for MariaDB verfügbar	Amazon RDS Performance Insights ist jetzt für Amazon RDS for MariaDB verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-In sights.	11. März 2019

MySQL 8.0.13 und 5.7.24	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MySQL-Versionen 8.0.13 und 5.7.24 laufen. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	8. März 2019
Amazon RDS Performance Insights ist für Amazon RDS for SQL Server verfügbar	Amazon RDS Performance Insights ist jetzt für Amazon RDS for SQL Server verfügbar . Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	4. März 2019
Amazon RDS for Oracle unterstützt Amazon S3-Integration	Sie können jetzt Dateien zwischen einer Amazon RDS for Oracle-DB-Instance und einem Amazon S3-Bucket übertragen. Weitere Informationen finden Sie unter Integration von Amazon RDS für Oracle und Amazon S3.	26. Februar 2019
Amazon RDS for MySQL und Amazon RDS for MariaDB unterstützen jetzt DB-Instance-Klassen vom Typ db.t3	Sie können nun Amazon RDS-DB-Instances mit MySQL oder MariaDB erstellen, die DB-Instance-Klassen vom Typ "db.t3" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	20. Februar 2019

Amazon RDS for MySQL und Amazon RDS for MariaDB unterstützen DB-Instance-Klassen vom Typ db.r5	Sie können nun Amazon RDS-DB-Instances, die MySQL oder MariaDB ausführen und DB-Instance-Klassen vom Typ "db.r5" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	20. Februar 2019
Performance Insights-Zähler für RDS for MySQL und PostgreSQL	Sie können nun Leistungszähler zu Ihren Performance Insights-Diagrammen für MySQL- und PostgreSQL-DB-Instances hinzufügen. Weitere Informationen finden Sie unter Verwenden der Performance-Insights-Dashboard-Komponenten .	19. Februar 2019
Amazon RDS for PostgreSQL unterstützt jetzt die adaptive Optimierung der Selbstbereinigungsparameter	Die adaptive Optimierung von Selbstbereinigungsparametern mit Amazon RDS for PostgreSQL hilft Transaktions-ID-Wraparound zu vermeiden, indem die Werte der Selbstbereinigungsparameter automatisch angepasst werden. Weitere Informationen finden Sie unter Verringern der Wahrscheinlichkeit von Transaktions-ID-Wraparounds.	12. Februar 2019

Amazon RDS for Oracle unterstützt Oracle APEX-Versionen 18.1.v1 und 18.2.v1	Amazon RDS for Oracle unterstützt jetzt die Oracle Application Express (APEX)-Versionen 18.1.v1 und 18.2.v1 Weitere Informationen finden Sie unter Oracle Application Express.	11. Februar 2019
Amazon RDS Performance Insights unterstützt Anzeige von mehr SQL-Text für RDS for MySQL	Amazon RDS Performance Insights unterstützt jetzt die Anzeige von mehr SQL-Text im Performance Insights-Dashboard für MySQL-DB-Instances. Weitere Informationen finden Sie unter Anzeigen von mehr SQL-Text im Performance-Insights-Dashboard.	6. Februar 2019
Amazon RDS for PostgreSQL unterstützt DB-Instance-Klassen vom Typ db.t3	Sie können jetzt Amazon RDS-DB-Instances mit PostgreSQL erstellen, die DB-Instance-Klassen vom Typ "db.t3" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	25. Januar 2019
Amazon RDS for Oracle unterstützt DB-Instance-Klassen vom Typ db.t3	Sie können jetzt Amazon RDS-DB-Instances mit Oracle erstellen, die DB-Instance-Klassen vom Typ "db.t3" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	25. Januar 2019

Amazon RDS Performance Insights unterstützt die Anzeige von mehr SQL-Text für Amazon RDS PostgreSQL	Amazon RDS Performance Insights unterstützt jetzt die Anzeige von mehr SQL-Text im Performance Insights-Dashboard für Amazon RDS PostgreSQL-DB-Instances. Weitere Informationen finden Sie unter Anzeigen von mehr SQL-Text im Performance-Insights-Dashboard.	24. Januar 2019
Amazon RDS for Oracle unterstützt eine neue Version von SQLT	Amazon RDS for Oracle unterstützt jetzt die SQLT-Version 12.2.180725. Weitere Informationen finden Sie unter Oracle SQLT.	22. Januar 2019
Amazon RDS for PostgreSQL unterstützt DB-Instance-Klassen vom Typ db.r5	Sie können jetzt Amazon RDS-DB-Instances mit PostgreSQL erstellen, die DB-Instance-Klassen vom Typ "db.r5" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	19. Dezember 2018

Amazon RDS for PostgreSQL unterstützt jetzt die eingeschränkte Passwortverwaltung	Mit Amazon RDS for PostgreSQL können Sie einschränken, wer Änderungen der Benutzerpasswörter und des Passwortablaufs verwalten darf. Verwenden Sie hierfür den Parameter `rds.restrict_password_commands` und die Rolle `rds_password` . Weitere Informationen hierzu finden Sie unter Beschränken der Passwortverwaltung.	19. Dezember 2018
Amazon RDS for PostgreSQL unterstützt das Hochladen von Datenbankprotokollen zu Amazon Logs CloudWatch	Amazon RDS for PostgreSQL unterstützt das Hochladen von Datenbankprotokollen in Logs. CloudWatch Weitere Informationen finden Sie unter PostgreSQL-Protokolle in Logs veröffentlichen. CloudWatch	10. Dezember 2018
Amazon RDS for Oracle unterstützt DB-Instance-Klassen vom Typ db.r5	Sie können jetzt Amazon RDS-DB-Instances mit Oracle erstellen, die DB-Instance-Klassen vom Typ "db.r5" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	20. November 2018
Aufbewahren von Backups beim Löschen einer DB-Instance	Amazon RDS unterstützt die Aufbewahrung automatisierter Backups beim Löschen einer DB-Instance. Weitere Informationen finden Sie unter Arbeiten mit Sicherungen.	15. November 2018

Amazon RDS for PostgreSQL unterstützt DB-Instance-Klassen vom Typ db.m5	Sie können jetzt Amazon RDS-DB-Instances mit PostgreSQL erstellen, die DB-Instance-Klassen vom Typ "db.m5" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	15. November 2018
Amazon RDS for Oracle unterstützt eine neue Hauptversion	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit Oracle Version 12.2 laufen.	13. November 2018
Amazon RDS for SQL Server unterstützt AlwaysOn	Amazon RDS for SQL Server unterstützt AlwaysOn-Verfügbarkeitsgruppen. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Microsoft SQL Server.	8. November 2018
Amazon RDS for PostgreSQL unterstützt ausgehenden Netzwerkzugriff mit benutzerdefinierten DNS-Servern	Amazon RDS for PostgreSQL unterstützt ausgehenden Netzwerkzugriff mit benutzerdefinierten DNS-Servern. Weitere Informationen finden Sie unter Verwenden eines benutzerdefinierten DNS-Servers für ausgehenden Netzwerkzugriff.	8. November 2018

Amazon RDS for MariaDB, MySQL und PostgreSQL unterstützt 32 TiB Speicher	Sie können jetzt Amazon RDS DB-Instances mit bis zu 32 TiB Speicherplatz für MySQL, MariaDB und PostgreSQL. erstellen. Weitere Informationen finden Sie unter DB-Instance-Speicher.	7. November 2018
Amazon RDS for Oracle unterstützt erweiterte Datentypen	Sie können jetzt in Amazon RDS-DB-Instances, auf denen Oracle ausgeführt wird, erweiterte Datentypen aktivieren. Bei erweiterten Datentypen beträgt die maximale Größe 32.767 Byte für die Datentypen VARCHAR2NVARCHAR2, und RAW. Weitere Informationen finden Sie unter Verwenden erweiterter Datentypen.	6. November 2018
Amazon RDS for Oracle unterstützt DB-Instance-Klassen vom Typ db.m5	Sie können jetzt Amazon RDS-DB-Instances mit Oracle erstellen, die DB-Instance-Klassen vom Typ „db.m5” verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	2. November 2018

Migration von Amazon RDS for Oracle von SE SE1, oder SE2 zu EE	Sie können jetzt von jeder Oracle Database Standard Edition (SE, SE1, oderSE2) zu Oracle Database Enterprise Edition (EE) migrieren. Weitere Informationen finden Sie unter Migrieren zwischen Oracle-Editionen.	31. Oktober 2018
Amazon RDS kann jetzt Multi-AZ-Instances stoppen	Amazon RDS kann jetzt eine DB-Instance stoppen, die Teil einer Multi-AZ-Bereitstellung ist. Früher gab es für die Funktion "Instance beenden" eine Einschränkung in Bezug auf Multi-AZ-Instances. Weitere Informationen finden Sie unter Vorübergehendes Anhalten einer Amazon RDS-DB-Instance.	29. Oktober 2018
Amazon RDS-Performance Insights ist für Amazon RDS for Oracle verfügbar	Amazon RDS-Performance Insights ist jetzt für Amazon RDS for Oracle verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	29. Oktober 2018
Amazon RDS for PostgreSQL unterstützt die PostgreSQL-Version 11 in der Database Preview-Umgebung	Amazon RDS for PostgreSQL unterstützt jetzt PostgreSQL-Version 11 in der Database Preview-Umgebung Weitere Informationen finden Sie unter PostgreSQL-Version 11 auf Amazon RDS in der Database Preview-Umgebung.	25. Oktober 2018

MySQL unterstützt eine neue Hauptversion	Sie können jetzt Amazon RDS-DB-Instances erstellen, auf denen MySQL-Version 8.0 ausgeführt wird. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	23. Oktober 2018
MariaDB unterstützt eine neue Hauptversion	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.3 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	23. Oktober 2018
Amazon RDS for Oracle unterstützt Oracle JVM	Amazon RDS for Oracle unterstützt jetzt die Oracle Java Virtual Machine (JVM)-Option. Weitere Informationen finden Sie unter Oracle Java Virtual Machine	16. Oktober 2018
Benutzerdefinierte Parametergruppe für die Wiederherstellung und zeitpunktbezogene Wiederherstellung	Sie können jetzt bei einer Snapshot-Wiederherstellung oder einer zeitpunktbezogenen Wiederherstellung eine benutzerdefinierte Parametergruppe angeben. Weitere Informationen finden Sie unter Wiederherstellen aus einem DB-Snapshot und Wiederherstellen einer DB-Instance zu einer bestimmten Zeit.	15. Oktober 2018

Amazon RDS for Oracle unterstützt 32 TiB Speicherplatz	Sie können jetzt Oracle-RDS-DB-Instances mit bis zu 32 TiB Speicherplatz erstellen. Weitere Informationen finden Sie unter DB-Instance-Speicher.	15. Oktober 2018
Amazon RDS for MySQL unterstützt GTIDs	Amazon RDS for MySQL unterstützt jetzt globale Transaktions-Identifikatoren (GTIDs), die für alle DB-Instances und in einer Replikationskonfiguration eindeutig sind. Weitere Informationen finden Sie unter Verwenden der GTID-basierten Replikation für RDS für MySQL.	10. Oktober 2018
MySQL 5.7.23, 5.6.41 und 5.5.61	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MySQL-Versionen 5.7.23, 5.6.41 und 5.5.61 laufen. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	8. Oktober 2018
Amazon RDS for Oracle unterstützt eine neue Version von SQLT	Amazon RDS for Oracle unterstützt jetzt die SQLT-Version 12.2.180331. Weitere Informationen finden Sie unter Oracle SQLT.	4. Oktober 2018

Amazon RDS for PostgreSQL unterstützt jetzt die IAM-Authentifizierung	Amazon RDS for PostgreSQL unterstützt jetzt die IAM-Authentifizierung. Weitere Informationen finden Sie unter IAM-Datenbank-Authentifizierung für MySQL und PostgreSQL.	27. September 2018
Sie können Löschschutz für Ihre Amazon RDS-DB-Instances aktivieren	Wenn Sie für eine DB-Instance Löschschutz aktivieren, kann die Datenbank von keinem Benutzer gelöscht werden. Weitere Informationen finden Sie unter Löschen einer DB-Instance.	26. September 2018
Amazon RDS for MySQL und Amazon RDS for MariaDB unterstützen jetzt DB-Instance-Klassen vom Typ db.m5	Sie können nun Amazon RDS DB-Instances mit MySQL oder MariaDB erstellen, die DB-Instance-Klassen vom Typ "db.m5" verwenden. Weitere Informationen finden Sie unter DB-Instance-Klasse.	18. September 2018
Amazon RDS unterstützt jetzt Upgrades auf SQL Server 2017	Sie können Ihre vorhandene DB-Instance auf SQL Server 2017 aktualisieren. Bei SQL Server 2008 ist das allerdings nicht möglich. Wenn Sie SQL Server 2008 auf die neueste Version aktualisieren möchten, müssen Sie zunächst ein Upgrade auf eine frühere Version durchführen. Weitere Informationen finden Sie unter Upgraden der Microsoft SQL Server-DB-Engine.	11. September 2018

Amazon RDS for PostgreSQL unterstützt jetzt die PostgreSQL-Version 11 Beta 3 in der Database Preview-Umgebung	In dieser Version wurde die WAL-Segmentgröße von (Write-Ahead Log; wal_segment_size) auf 64 MB festgelegt. Weitere Informationen zu PostgreSQL-Version 11, Beta 3, finden Sie unter PostgreSQL 11 Beta 3 Released. Weitere Informationen über die Database Preview-Umgebung finden Sie unter Arbeiten mit der Database Preview-Umgebung.	7. September 2018
Amazon Aurora Benutzerhandbuch	Das Amazon Aurora-Benutzerhandbuch enthält detaillierte Beschreibungen der Konzepte von Amazon Aurora sowie Anleitungen zur Nutzung der Funktionen über die Konsole und Befehlszeilen-Schnittstelle. Das Amazon RDS-Benutzerhandbuch behandelt jetzt auch Nicht-Aurora-Datenbank-Engines.	31. August 2018
Amazon RDS Performance Insights ist für RDS for MySQL verfügbar	Amazon RDS-Performance Insights ist jetzt für RDS for MySQL verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	28. August 2018

Aurora PostgreSQL-kompatible Edition unterstützt jetzt das Auto Scaling von Aurora	Das Auto Scaling von Aurora-Repliken ist jetzt für Aurora PostgreSQL-kompatible Edition verfügbar. Weitere Informationen finden Sie unter Amazon Aurora Auto Scaling with Aurora Replicas.	16. August 2018
Aurora Serverless for Aurora MySQL	Aurora Serverless ist eine bedarfsabhängige Auto Scaling-Konfiguration für Amazon Aurora. Weitere Informationen finden Sie unter Verwendung von Amazon Aurora Serverless.	9. August 2018
MySQL 5.7.22 und 5.6.40	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MySQL-Versionen 5.7.22 und 5.6.40 laufen. Weitere Informationen finden Sie unter MySQL auf Amazon RDS-Versionen.	6. August 2018
Aurora ist jetzt in der Region China (Ningxia) verfügbar	Aurora MySQL und Aurora PostgreSQL sind jetzt in der Region China (Ningxia) verfügbar. Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora MySQL und Verfügbarkeit für Amazon Aurora PostgreSQL.	6. August 2018

Amazon RDS for MySQL unterstützt die verzögerte Replikation	Amazon RDS for MySQL unterstützt jetzt die verzögerte Replikation als Strategie für die Notfallwiederherstellung. Weitere Informationen finden Sie unter Konfigurieren der verzögerten Replikation mit MySQL.	6. August 2018
Amazon RDS-Performance Insights ist für Aurora MySQL verfügbar	Amazon RDS-Performance Insights ist jetzt für Aurora MySQL verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	6. August 2018
Integration von Amazon RDS Performance Insights mit Amazon CloudWatch	Amazon RDS Performance Insights veröffentlicht automatisch Metriken auf Amazon CloudWatch. Weitere Informationen finden Sie unter Performance Insights Insights-Metriken, die auf veröffentlicht wurden CloudWatch.	6. August 2018
Amazon RDS-Empfehlungen	Amazon RDS bietet nun automatisierte Empfehlungen für Datenbankressourcen. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Empfehlungen.	25. Juli 2018

Inkrementelle Snapshot-Kopien zwischen Regionen AWS	Amazon RDS unterstützt inkrementelle Snapshot-Kopien in allen AWS Regionen sowohl für unverschlüsselte als auch für verschlüsselte Instances. Weitere Informationen finden Sie unter Kopieren von Snapshots zwischen Regionen. AWS	24. Juli 2018
Amazon RDS-Performance Insights ist für Amazon RDS for PostgreSQL verfügbar	Amazon RDS-Performance Insights ist jetzt für Amazon RDS for PostgreSQL verfügbar. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	18. Juli 2018
Amazon RDS for Oracle unterstützt Oracle APEX Version 5.1.4.v1	Amazon RDS for Oracle unterstützt jetzt Oracle Application Express (APEX) Version 5.1.4.v1. Weitere Informationen finden Sie unter Oracle Application Express.	10. Juli 2018
Amazon RDS for Oracle unterstützt das Veröffentlichen von Protokollen in Amazon CloudWatch Logs	Amazon RDS for Oracle unterstützt jetzt die Veröffentlichung von Alert-, Audit-, Trace- und Listener-Protokolldaten in einer Protokollgruppe in CloudWatch Logs. Weitere Informationen finden Sie unter Oracle-Logs in Amazon CloudWatch Logs veröffentlichen.	9. Juli 2018

MariaDB 10.2.15, 10.1.34 und 10.0.35	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit den MariaDB-V ersionen 10.2.15, 10.1.34 und 10.0.35 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon RDS-Versionen.	5. Juli 2018
Aurora PostgreSQL 1.2 ist verfügbar und mit PostgreSQL 9.6.8 kompatibel	Aurora PostgreSQL 1.2 ist jetzt verfügbar und mit PostgreSQL 9.6.8 kompatibe l. Weitere Informationen finden Sie unter Version 1.2.	27. Juni 2018
Lesereplikate für Amazon RDS PostgreSQL unterstützen Multi-AZ-Bereitstellungen	RDS-Lesereplikate in Amazon RDS PostgreSQL unterstüt zen jetzt mehrere Availability Zones. Weitere Informationen finden Sie unter Arbeiten mit PostgreSQL-Lesereplikaten.	25. Juni 2018
Performance Insights ist für Aurora PostgreSQL verfügbar	Performance Insights ist generell für Aurora PostgreSQ L verfügbar, mit Unterstützung für die erweiterte Speicherung von Leistungsdaten. Weitere Informationen finden Sie unter Verwenden von Amazon RDS-Performance-Insights.	21. Juni 2018
Aurora PostgreSQL ist in der Region USA West (Nordkali fornien) verfügbar	Aurora PostgreSQL ist jetzt in der Region USA West (Nordkalifornien) verfügbar. Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora PostgreSQL.	11. Juni 2018

Amazon RDS for Oracle unterstützt jetzt eine CPU-Konfiguration	Amazon RDS for Oracle unterstützt die Konfiguration der Anzahl der CPU-Kerne und der Anzahl der Threads für jeden Kern für den Prozessor einer DB-Instance-Klasse. Weitere Informationen finden Sie unter Konfigurieren des Prozessors der DB-Instance-Klasse.	5. Juni 2018

Frühere Aktualisierungen

In der folgenden Tabelle sind die wichtigen Änderungen in jeder Version des Amazon RDS-Benutzerhandbuchs vor Juni 2018 beschrieben.

Änderungen	Beschreibung	Datum geändert
Amazon RDS for PostgreSQL unterstützt jetzt die PostgreSQL-Version 11 Beta 1 in der Database Preview-Umgebung	Die PostgreSQL-Version 11 Beta 1 enthält verschiedene Verbesserungen, die unter PostgreSQL 11 Beta 1 Released! beschrieben werden. Weitere Informationen zur Database Preview-Umgebung finden Sie unter Arbeiten mit der Datenbank-Vorschauumgebung.	31. Mai 2018
Amazon RDS for Oracle unterstützt jetzt die TLS-Versionen 1.0 und 1.2	Amazon RDS for Oracle unterstützt jetzt Transport Layer Security (TLS) in den Versionen 1.0 und 1.2. Weitere Informationen finden Sie unter TLSVersionen für die SSL Oracle-Option.	30. Mai 2018
Aurora MySQL unterstützt die Veröffentlichung von Protokoll	Aurora MySQL unterstützt jetzt die Veröffentlichung von allgemeinen, langsamen, Prüfungs- und Fehlerprotokolldaten in einer Protokollgruppe in CloudWatch	23. Mai 2018

Änderungen	Beschreibung	Datum geändert
en in Amazon CloudWatch Logs	Logs. Weitere Informationen finden Sie unter Aurora MySQL in CloudWatch Logs veröffentlichen.	
Database Preview-Umgebung für Amazon RDS PostgreSQL	Sie können jetzt eine neue Instance von Amazon RDS PostgreSQL in einem Vorschaumodus starten. Weitere Informationen zur Database Preview-Umgebung finden Sie unte, Arbeiten mit der Datenbank-Vorschauumgebung.	22. Mai 2018
Amazon RDS for Oracle DB-Instances unterstützen neue DB-Instance-Klassen	Oracle-DB-Instances unterstützen jetzt die DB-Instance-Klassen db.x1e und db.x1. Weitere Informationen erhalten Sie unter und RDS für Oracle-DB-Instance-Klassen.	22. Mai 2018
Amazon RDS PostgreSQL unterstützt jetzt postgres_fdw auf einem Leserepli kat.	Sie können jetzt mit postgres_fdw von einem Leserepli kat eine Verbindung zu einem Remote-Server herstellen. Weitere Informationen finden Sie unter Verwenden der postgres_fdw-Erweiterung für den Zugriff auf externe Daten.	17. Mai 2018
Amazon RDS for Oracle unterstützt jetzt das Einstellen von sqlnet.ora-Parametern	Sie können jetzt sqlnet.ora-Parameter mit Amazon RDS for Oracle einstellen. Weitere Informationen finden Sie unter Ändern von Verbindungseigenschaften mit sqlnet.ora-Parametern.	10. Mai 2018
Aurora PostgreSQL ist in der Region Asien-Pazifik (Seoul) verfügbar.	Aurora PostgreSQL ist jetzt in der Region Asien-Pazifik (Seoul) verfügbar. Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora PostgreSQL.	9. Mai 2018

Änderungen	Beschreibung	Datum geändert
Aurora MySQL unterstützt die Rückverfolgung	Aurora MySQL unterstützt jetzt das "Zurückspulen" eines DB-Clusters auf einen bestimmten Zeitpunkt , ohne dass die Daten aus einem Backup wiederher gestellt werden müssen. Weitere Informationen finden Sie unter Rückverfolgen eines Aurora-DB-Clusters.	9. Mai 2018
Aurora MySQL unterstützt die verschlüsselte Migration und Replikation aus einer externen MySQL-Dat enbank.	Aurora MySQL unterstützt jetzt die verschlüsselte Migration und Replikation aus einer externen MySQL-Datenbank. Weitere Informationen finden Sie unter Migrieren von Daten aus einer externen MySQL-Dat enbank zu einem Amazon Aurora MySQL-DB-Cluster und Replikation zwischen Aurora und MySQL oder zwischen Aurora und einem anderen Aurora-DB-Cluster.	25. April 2018
Aurora PostgreSQ L-Compatible Edition-Unterstütz ung für das Protokoll. Copy-on-Write	Sie können jetzt Datenbanken in einem Aurora PostgreSQL-Datenbank-Cluster klonen. Weitere Informationen finden Sie unter Klonen von Datenbank en in einem Aurora-DB-Cluster.	10. April 2018
MariaDB 10.2.12, 10.1.31 und 10.0.34	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit den MariaDB-Versionen 10.2.12, 10.1.31 und 10.0.34 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon-RDS-Versionen.	21. März 2018
Aurora PostgreSQ L-Unterstützung für neue Regionen	Aurora PostgreSQL ist jetzt in den Regionen EU (London) und Asien-Pazifik (Singapur) verfügbar. Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora PostgreSQL.	13. März 2018
MySQL 5.7.21, 5.6.39 und 5.5.59	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit den MySQL-Versionen 5.7.21, 5.6.39 und 5.5.59 laufen. Weitere Informationen finden Sie unter MySQL in Amazon RDS-Versionen.	9. März 2018

Änderungen	Beschreibung	Datum geändert
Amazon RDS for Oracle unterstützt jetzt Oracle REST Data Services.	Amazon RDS for Oracle unterstützt Oracle REST Data Services als Teil der APEX-Option. Weitere Informationen finden Sie unter Oracle Application Express (APEX).	9. März 2018
Amazon Aurora MySQL-Compatible Edition in neuer Region verfügbar AWS	Aurora MySQL ist jetzt in der Region Asien-Pazifik (Singapur) erhältlich. Die vollständige Liste der AWS Regionen für Aurora MySQL finden Sie unter Verfügbarkeit für Amazon Aurora MySQL.	6. März 2018
Amazon RDS DB-Instances mit Microsoft SQL Server unterstützen die Erfassung von Datenänderungen (Change Data Capture, CDC)	DB-Instances mit Amazon RDS for Microsoft SQL Server unterstützen jetzt die Erfassung von Datenänderungen (Change Data Capture, CDC). Weitere Informationen finden Sie unter Unterstützung für Change Data Capture für Microsoft SQL Server-DB-Instances.	6. Februar 2018
Aurora MySQL unterstützt jetzt eine neue Hauptversion	Sie können jetzt Aurora MySQL-DB-Cluster erstellen, die mit der MySQL-Version 5.7 laufen. Weitere Informationen finden Sie in den Updates der Amazon Aurora MySQL-Datenbank-Engine (2018-02-06).	6. Februar 2018
Veröffentlichen Sie MySQL- und MariaDB-Protokolle in Amazon Logs CloudWatch	Sie können jetzt MySQL- und MariaDB-Protokolldaten in Logs veröffentlichen. CloudWatch Weitere Informationen erhalten Sie unter Veröffentlichen von MySQL-Protokollen in Amazon CloudWatch Logs und Veröffentlichen von MariaDB-Protokollen in Amazon Logs CloudWatch .	17. Januar 2018

Änderungen	Beschreibung	Datum geändert
Multi-AZ-Unterstützung für Lesereplikate	Sie können jetzt ein Lesereplikat als Multi-AZ-DB-Instance erstellen. Amazon RDS erstellt eine Standby-Version des Replikats in einer anderen Availability Zone, um ein Failover für das Replikat zu unterstützen. Das Erstellen Ihres Lesereplikats als Multi-AZ-DB-Instance ist unabhängig davon, ob die Quelldatenbank eine Multi-AZ-DB-Instance ist. Weitere Informationen finden Sie unter Arbeiten mit DB-Instance-Lesereplikaten.	11. Januar 2018
Amazon RDS for MariaDB unterstützt eine neue Hauptversion	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit der MariaDB-Version 10.2 laufen. Weitere Informationen finden Sie im Support für MariaDB 10.2 in Amazon RDS.	3. Januar 2018
Amazon Aurora PostgreSQL-kompatible Edition in neuer AWS -Region verfügbar.	Aurora PostgreSQL ist jetzt in der Region EU (Paris) verfügbar. Die vollständige Liste der AWS Regionen für Aurora PostgreSQL finden Sie unter Verfügbarkeit für Amazon Aurora PostgreSQL.	22. Dezember 2017
Aurora PostgreSQL unterstützt neue Instance-Typen	Aurora PostgreSQL unterstützt jetzt neue Instance-Typen. Eine vollständige Liste von Instance-Typen finden Sie unter Auswahl von DB-Instance-Klassen.	20. Dezember 2017
Amazon Aurora MySQL-Compatible Edition in neuer Region verfügbar AWS	Aurora MySQL ist jetzt in der Region EU (Paris) verfügbar. Die vollständige Liste der AWS Regionen für Aurora MySQL finden Sie unter Verfügbarkeit für Amazon Aurora MySQL.	18. Dezember 2017
Aurora MySQL unterstützt Hash-Joins	Diese Funktion kann die Abfrageleistung verbessern, wenn Sie eine große Datenmenge mithilfe eines Equijoins verbinden müssen. Weitere Informationen finden Sie unter Arbeiten mit Hash-Joins in Aurora MySQL.	11. Dezember 2017

Änderungen	Beschreibung	Datum geändert
Aurora MySQL unterstützt native Funktionen zum Aufruf von AWS Lambda -F unktionen	Sie können die nativen Funktionen `lambda_sync` und `lambda_async` aufrufen, wenn Sie Aurora MySQL verwenden. Weitere Informationen finden Sie unter Aufrufen einer Lambda-Funktion aus einem Amazon Aurora MySQL-DB-Cluster.	11. Dezember 2017
Aurora PostgreSQ L HIPAA-Ber echtigung hinzugefügt.	Aurora PostgreSQL unterstützt jetzt die Erstellung HIPAA-konformer Anwendungen. Weitere Informati onen finden Sie unter Arbeiten mit Amazon Aurora PostgreSQL.	6. Dezember 2017
Zusätzliche AWS Regionen für Amazon Aurora mit PostgreSQ L-Kompatibilität verfügbar	Amazon Aurora mit PostgreSQL-Kompatibilität ist jetzt in vier neuen AWS Regionen verfügbar. Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora PostgreSQL.	22. November 2017
Ändern des Speichers für Amazon RDS DB-Instances, auf denen Microsoft SQL Server ausgeführt wird	Sie können nun den Speicher Ihrer Amazon RDS DB-Instances ändern, auf denen SQL Server läuft. Weitere Informationen finden Sie unter Ändern einer Amazon RDS DB-Instance.	21. November 2017
Amazon RDS unterstützt 16 TiB-Speicher für Linux-basierte Engines	Sie können jetzt MySQL, MariaDB, PostgreSQL und Oracle RDS DB-Instances mit bis zu 16 TiB Speicherp latz erstellen. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.	21. November 2017

Änderungen	Beschreibung	Datum geändert
Amazon RDS unterstützt schnelles Scale-up des Speichers	Sie können jetzt MySQL, MariaDB, PostgreSQL, PostgreSQL und Oracle RDS DB-Instances in wenigen Minuten mit mehr Speicherplatz versehen. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.	21. November 2017
Amazon RDS unterstützt die MySQL-Versionen 10.1.26 und 10.0.32	Sie können jetzt Amazon RDS-DB-Instances erstellen, die mit den MariaDB-Versionen 10.1.26 und 10.0.32 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon-RDS-Versionen.	20. November 2017
Amazon RDS for Microsoft SQL Server unterstützt jetzt neue DB-Instance-Klassen	Sie können nun Amazon RDS DB-Instances mit SQL Server erstellen, die die DB-Instance-Klassen db.r4 und db.m4.16xlarge verwenden. Weitere Informationen finden Sie unter Unterstützung für DB-Instance-Klassen für Microsoft SQL Server.	20. November 2017
Amazon RDS for MySQL und MariaDB unterstützt jetzt neue DB-Instance-Klassen	Sie können nun Amazon RDS DB-Instances mit MySQL und MariaDB erstellen, die die DB-Instance-Klassen db.r4, db.m4.16xlarge, db.t2.xlarge und db.t2.2xlarge verwenden. Weitere Informationen finden Sie unter .	20. November 2017
SQL Server 2017	Sie können jetzt Amazon RDS DB-Instances erstellen, die mit Microsoft SQL Server 2017 laufen. Sie können auch DB-Instances erstellen, auf denen SQL Server 2016 ausgeführt wird. SP1 CU5 Weitere Informationen finden Sie unter Amazon RDS for Microsoft SQL Server.	17. November 2017

Änderungen	Beschreibung	Datum geändert
Wiederherstellen von MySQL-Backups von Amazon S3 aus	Sie können jetzt ein Backup der Datenbank vor Ort erstellen, auf Amazon S3 speichern und die Sicherungsdatei anschließend auf einer neuen Amazon RDS-DB-Instance mit MySQL wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellung eines Backups in einer Amazon RDS for MySQL MySQL-DB-Instance.	17. November 2017
Auto Scaling mit Aurora-Repliken	Amazon Aurora MySQL unterstützt jetzt Aurora-Auto Scaling. Aurora-Auto Scaling passt die Anzahl der Aurora-Repliken dynamisch an, basierend auf der Zunahme oder Abnahme der Konnektivität oder der Workload. Weitere Informationen finden Sie unter Amazon Aurora Auto Scaling mit Aurora-Repliken.	17. November 2017
Unterstützung der Oracle Standardversion	Amazon RDS for Oracle DB-Instances unterstützt nun das Setzen der Standardedition für die DB-Instance. Weitere Informationen finden Sie unter Einrichten der Standardversion für eine DB-Instance.	3. November 2017
Validierung der Oracle DB-Instance-Datei	Amazon RDS for Oracle DB-Instances unterstützt jetzt die Validierung von DB-Instance-Dateien mit dem logischen Validierungsdienstprogramm des Oracle Recovery Manager (RMAN). Weitere Informationen finden Sie unter Datenbankdateien RDS für Oracle validieren .	3. November 2017
Management Agent für OEM 13c	Die Amazon RDS for Oracle DB-Instances unterstützen jetzt den Management Agent für Oracle Enterprise Manager (OEM) Cloud Control 13c. Weitere Informationen finden Sie unter Oracle Management Agent für Enterprise Cloud Control.	1. November 2017

Änderungen	Beschreibung	Datum geändert
Speicherr ekonfiguration für Microsoft SQL Server-Snapshots	Sie können nun den Speicher neu konfigurieren, wenn Sie einen Snapshot in einer Amazon RDS-DB-Instance mit Microsoft SQL Server wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellung auf einer DB-Instance.	26. Oktober 2017
Asynchrones Key Prefetch für Aurora MySQL-kompatible Edition	Asynchrones Key Prefetch (AKP) verbessert die Performance von nicht gepufferten Index-Joins, indem Schlüssel im Arbeitsspeicher vor dem benötigten Zeitpunkt vorgeladen werden. Weitere Informationen finden Sie unter Arbeiten mit einem asynchronen Key Prefetch in Amazon Aurora.	26. Oktober 2017
MySQL 5.7.19, 5.6.37 und 5.5.57	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit den MySQL-Versionen 5.7.19, 5.6.37 und 5.5.57 laufen. Weitere Informationen finden Sie unter MySQL in Amazon RDS-Versionen.	25. Oktober 2017
Allgemeine Verfügbarkeit von Amazon Aurora mit PostgreSQL-Kompatibilität	Amazon Aurora mit PostgreSQL-Kompatibilität macht es sehr einfach und kosteneffizient, bestehend e PostgreSQL-Bereitstellungen einzurichten, zu betreiben und zu skalieren, damit Sie sich voll und ganz auf Ihr Geschäft und Ihre Anwendungen konzentrieren können. Weitere Informationen finden Sie unter Arbeiten mit Amazon Aurora PostgreSQL.	24. Oktober 2017
Amazon RDS for Oracle DB-Instan ces unterstützen neue DB-Instance-Klassen	Die Amazon RDS for Oracle DB-Instances unterstüt zen nun Instance-Klassen der arbeitsspeicheropt imierten nächsten Generation (db.r4). Amazon RDS for Oracle DB-Instances unterstützen nun auch die folgenden Instance-Klassen der aktuellen Generatio n: db.m4.16xlarge, db.t2.xlarge und db.t2.2xlarge. Weitere Informationen erhalten Sie unter und RDS für Oracle-DB-Instance-Klassen.	23. Oktober 2017

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Ihre neuen und vorhandenen reservierten Instances können nun mehrere Größen in derselben DB-Instance-Klasse abdecken. Größenflexible Reserved Instances sind für DB-Instances mit derselben AWS Region, Datenbank-Engine und Instance-Familie sowie für alle AZ-Konfigurationen verfügbar. Größenflexible Reserved Instances sind für die folgenden Datenbank-Engines verfügbar: Amazon Aurora, MariaDB, MySQL, Oracle (Bring Your Own License), PostgreSQL. Weitere Informationen finden Sie unter Größenflexible Reservierte DB-Instances.	11. Oktober 2017
Neue Funktion	Sie können nun die Oracle SQLT-Option verwenden, um eine SQL-Anweisung auf optimale Performance abzustimmen. Weitere Informationen finden Sie unter Orakel SQLT.	22. September 2017
Neue Funktion	Wenn Sie bereits über manuelle DB-Snapshots Ihrer Amazon RDS for Oracle DB-Instances verfügen, können Sie diese jetzt auf eine neuere Version der Oracle Datenbank-Engine aktualisieren. Weitere Informationen finden Sie unter Aktualisieren eines Oracle-DB-Snapshots.	20. September 2017
Neue Funktion	Mit Oracle Spatial können Sie jetzt Geodaten in Ihren Amazon RDS DB-Instances mit Oracle speichern, abrufen, aktualisieren und abfragen. Weitere Informationen finden Sie unter Oracle Spatial.	15. September 2017
Neue Funktion	Mit Oracle Locator können Sie jetzt Internet- und Wireless-Service-basierte Anwendungen sowie GIS-Lösungen auf Partnerbasis mit Ihren Amazon RDS DB-Instances mit Oracle unterstützen. Weitere Informationen finden Sie unter Oracle Locator.	15. September 2017

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Mit Oracle Multimedia können Sie jetzt Bilder, Audio-, Video- und andere heterogene Mediendaten in Ihren Amazon RDS DB-Instances mit Oracle speichern, verwalten und abrufen.	15. September 2017
Neues Feature	Sie können jetzt Audit-Logs aus Ihren Amazon Aurora MySQL-DB-Clustern nach Amazon CloudWatch Logs exportieren. Weitere Informationen finden Sie unter Aurora MySQL-Protokolle in Amazon CloudWatch Logs veröffentlichen.	14. September 2017
Neue Funktion	Amazon RDS unterstützt jetzt mehrere Versionen von Oracle Application Express (APEX) für Ihre DB-Instances mit Oracle. Weitere Informationen finden Sie unter Oracle Application Express (APEX).	13. September 2017
Neue Funktion	Sie können jetzt Amazon Aurora verwenden, um einen unverschlüsselten oder verschlüsselten DB-Snapshot oder eine MySQL-DB-Instance in einen verschlüsselten Aurora-MySQL-DB-Cluster zu migrieren. Weitere Informationen finden Sie unter Migrieren eines RDS for MySQL-Snapshots zu Aurora und Datenmigration von einer MySQL-DB-Instance zu einem Amazon Aurora MySQL-DB-Cluster unter Verwendung eines Aurora-Lesereplikats.	5. September 2017
Neue Funktion	Sie können Amazon RDS for Microsoft SQL Server-Datenbanken für die Erstellung von HIPAA-kompatiblen Anwendungen verwenden. Weitere Informationen finden Sie unter Unterstützung zu Compliance-Programmen für Microsoft SQL Server-DB-Instances.	31. August 2017
Neue Funktion	Sie können jetzt Amazon RDS for MariaDB-Datenbanken für die Erstellung von HIPAA-kompatiblen Anwendungen verwenden. Weitere Informationen finden Sie unter Amazon RDS für MariaDB.	31. August 2017

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Sie können jetzt Amazon RDS-DB-Instances erstellen , auf denen Microsoft SQL Server mit zugewiesenem Speicher bis zu 16 TiB und bereitgestellten IOPS in Speicherbereichen von 1:1–50:1 ausgeführt wird. Weitere Informationen finden Sie unter Amazon RDS-DB-Instance-Speicher.	22. August 2017
Neue Funktion	Sie können jetzt Multi-AZ-Bereitstellungen für DB-Instances mit Microsoft SQL Server in der EU-Region (Frankfurt) verwenden. Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.	3. August 2017
Neue Funktion	Sie können jetzt Amazon RDS-DB-Instances erstellen , die mit den MariaDB-Versionen 10.1.23 und 10.0.31 laufen. Weitere Informationen finden Sie unter MariaDB auf Amazon-RDS-Versionen.	17. Juli 2017
Neues Feature	Amazon RDS unterstützt jetzt Microsoft SQL Server Enterprise Edition mit dem Modell „Lizenz enthalten“ in allen AWS Regionen. Weitere Informationen finden Sie unter Lizenzierung Microsoft SQL Server auf Amazon RDS.	13. Juli 2017
Neue Funktion	Amazon RDS for Oracle unterstützt jetzt Huge Pages von Linux Kernel für eine höhere Skalierbarkeit der Datenbank. Durch die Huge-Pages-Verwendung werden Page-Tabellen verkleinert und weniger CPU-Zeit für die Speicherverwaltung benötigt, sodass die Leistung von großen Datenbank-Instances erhöht wird. Sie können Huge Pages mit den Amazon RDS-DB-In stances aller Editionen der Oracle-Versionen 12.1.0.2 und 11.2.0.4 verwenden. Weitere Informationen finden Sie unter Aktivieren von HugePages für eine Instance von RDS für Oracle.	7. Juli 2017

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung der Verschlüsselung ruhender Daten für die DB-Instance-Klassen db.t2.small und db.t2.medium bei allen Nicht-Aurora-DB-Engines. Weitere Informationen finden Sie unter Verfügbarkeit der Amazon RDS-Verschlüsselung.	27. Juni 2017
Neue Funktion	Aktualisiert mit der Unterstützung von Amazon Aurora in der Region Europa (Frankfurt). Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora MySQL.	16. Juni 2017
Neues Feature	Sie können jetzt eine Optionsgruppe angeben, wenn Sie einen DB-Snapshot AWS regionsübergreifend kopieren. Weitere Informationen finden Sie unter Überlegungen zu Optionsgruppen.	12. Juni 2017
Neues Feature	Sie können jetzt DB-Snapshots, die von speziellen DB-Instances erstellt wurden, regionsübergreifend AWS kopieren. Sie können Snapshots aus DB-Instances kopieren, die Oracle TDE, Microsoft SQL Server TDE und Microsoft SQL Server Multi-AZ mit Spiegelung verwenden. Weitere Informationen finden Sie unter Kopieren eines DB-Snapshots.	12. Juni 2017
Neue Funktion	Mit Amazon Aurora können Sie nun schnell und kosteneffizient all Ihre Datenbanken in ein Amazon Aurora-DB-Cluster kopieren. Weitere Informationen finden Sie unter Klonen von Datenbanken in einem Aurora-DB-Cluster.	12. Juni 2017
Neues Feature	Amazon RDS unterstützt jetzt Microsoft SQL Server 2016 SP1 CU2. Weitere Informationen finden Sie unter Amazon RDS for Microsoft SQL Server.	7. Juni 2017

Änderungen	Beschreibung	Datum geändert
Vorversion	Öffentliche Vorversion von Amazon Aurora mit PostgreSQL-Kompatibilität. Weitere Informationen finden Sie unter Arbeiten mit Amazon Aurora PostgreSQL.	19. April 2017
Neue Funktion	Mit Amazon Aurora können Sie nun den Vorgang ALTER TABLE tbl_name ADD COLUMN col_name column_definition nahezu in Echtzeit ausführen. Die Operation wird abgeschlossen, ohne dass ein Kopieren der Tabelle erforderlich wäre und ohne eine materielle Auswirkung auf andere DML-Statements zu haben. Weitere Informationen finden Sie unter Ändern von Tabellen in Amazon Aurora mithilfe von schneller DDL.	5. April 2017
Neue Funktion	Wir haben den neuen Überwachungsbefehl SHOW VOLUME STATUS hinzugefügt, mit dem Sie die Anzahl der Knoten und Datenträger für ein Volume anzeigen können. Weitere Informationen finden Sie unter Anzeigen des Volume-Status für einen Aurora-DB-Cluster.	5. April 2017
Neues Feature	Sie können Ihre eigene benutzerdefinierte Logik in Ihren eigenen Passwortverifizierungsfunktionen für Oracle in Amazon RDS verwenden. Weitere Informationen finden Sie unter Erstellen von benutzerdefinierten Funktionen für das Überprüfen von Passwörtern.	21. März 2017
Neue Funktion	Sie können nun auf Ihre Online- und archivierten Redo-Log-Dateien auf den Oracle-DB-Instances in Amazon RDS zugreifen. Weitere Informationen finden Sie unter Zugriff auf Online- oder archivierte Redo-Protokolle.	21. März 2017

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Sie können nun sowohl verschlüsselte als auch unverschlüsselte DB-Cluster-Snapshots zwischen Konten derselben Region kopieren. Weitere Informationen finden Sie unter Kopieren eines DB-Cluster-Snapshots in ein anderes Konto.	7. März 2017
Neue Funktion	Sie können nun verschlüsselte und unverschlüsselte DB-Cluster-Snapshots zwischen Konten derselben Region teilen. Weitere Informationen finden Sie unter Freigeben eines DB-Cluster-Snapshots.	7. März 2017
Neue Funktion	Sie können jetzt verschlüsselte Amazon Aurora MySQL-DB-Cluster replizieren, um regionenübergreifende Aurora-Repliken zu erstellen. Weitere Informationen finden Sie unter AWS Regionale Replikation von Aurora MySQL-DB-Clustern.	7. März 2017
Neue Funktion	Sie können nun festlegen, dass alle Verbindungen zur DB-Instance, auf der Microsoft SQL Server ausgeführt wird, Secure Sockets Layer (SSL) verwenden müssen. Weitere Informationen finden Sie unter Verwendung SSL mit einer Microsoft SQL Server-DB-Instance.	27. Februar 2017
Neue Funktion	Sie können nun Ihre lokale Zeitzone in einer von 15 zusätzlichen Zeitzonen festlegen. Weitere Informationen finden Sie unter Unterstützte Zeitzonen.	27. Februar 2017
Neue Funktion	Sie können jetzt mit dem Amazon RDS-Verfahren `msdb.dbo.rds_shrink_tempdbfile` die tempdb-Datenbank auf den DB-Instances komprimieren, auf denen Microsoft SQL Server ausgeführt wird. Weitere Informationen finden Sie unter Verkleinern der Datenbank tempdb.	17. Februar 2017

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Sie können jetzt Ihre Sicherungsdateien komprimieren, wenn Sie eine Microsoft SQL Server-Datenbank (Enterprise/Standard Edition) von einer Amazon RDS-DB-Instance in Amazon S3 exportieren. Weitere Informationen finden Sie unter Komprimieren von Sicherungsdateien.	17. Februar 2017
Neue Funktion	Amazon RDS unterstützt nun benutzerdefinierte DNS-Server, um DNS-Namen aufzulösen, die bei ausgehendem Netzwerkzugriff auf Ihre DB-Instance (auf der Oracle ausgeführt wird) verwendet werden. Weitere Informationen finden Sie unter Einen benutzerdefinierten DNS Server einrichten.	26. Januar 2017
Neue Funktion	Amazon RDS unterstützt nun das Erstellen eines verschlüsselten Lesereplikats in einer anderen Region. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats in einer anderen AWS-Region und Create DBInstance ReadReplica.	23. Januar 2017
Neue Funktion	Amazon RDS unterstützt jetzt Upgrades eines MySQL-DB-Snapshots von MySQL 5.1 auf MySQL 5.5.	20. Januar 2017
Neue Funktion	Amazon RDS unterstützt nun das Kopieren eines verschlüsselten DB-Snapshots in eine andere Region für die Datenbank-Engines MariaDB, MySQL, PostgreSQL und Microsoft SQL Server. Weitere Informationen finden Sie unter Kopieren eines DB-Snapshots und Kopieren DBSnapshot.	20. Dezember 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Amazon Aurora MySQL unterstützt jetzt die räumliche Indizierung. Räumliche Indizierung verbessert die Abfrageleistung in großen Datensätzen für Abfragen, die räumliche Daten verwenden. Weitere Informationen finden Sie unter Amazon Aurora MySQL und raumbezogene Daten.	14. Dezember 2016
Neues Feature	Amazon RDS unterstützt jetzt ausgehenden Netzwerkzugriff auf Ihre DB-Instance in Oracle. Sie können utl_http, utl_tcp und utl_smtp verwenden, um von Ihrer DB-Instance eine Verbindung zum Netzwerk herzustellen. Weitere Informationen finden Sie unter Konfigurieren des UTL_HTTP-Zugriffs mit Zertifikaten und einer Oracle Wallet.	5. Dezember 2016
Neue Funktion	Amazon RDS unterstützt die MySQL-Version 5.1 nicht mehr Sie können jedoch bei Bedarf bestehende MySQL 5.1-Snapshots in einer Instance mit MySQL 5.5 wiederherstellen. Weitere Informationen finden Sie unter Unterstützte Speicher-Engines für RDS for MySQL.	15. November 2016
Neues Feature	Amazon RDS unterstützt jetzt Microsoft SQL Server 2016 RTM. CU2 Weitere Informationen finden Sie unter Amazon RDS for Microsoft SQL Server.	4. November 2016
Neue Funktion	Amazon RDS unterstützt jetzt Hauptversions-Upgrades für DB-Instances, auf denen Oracle ausgeführt wird. Sie können Ihre Oracle-DB-Instances nun von Version 11g auf Version 12c aktualisieren. Weitere Informationen finden Sie unter Aktualisieren der DB-Engine von RDS für Oracle.	2. November 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Sie können jetzt DB-Instances erstellen, die mit Microsoft SQL Server 2014 Enterprise Edition laufen. Amazon RDS unterstützt jetzt SQL Server 2014 SP2 für alle Editionen und alle Regionen. Weitere Informationen finden Sie unter Amazon RDS for Microsoft SQL Server.	25. Oktober 2016
Neues Feature	Amazon Aurora MySQL lässt sich jetzt in andere AWS Dienste integrieren: Sie können Text- oder XML-Daten aus einem Amazon S3 S3-Bucket in eine Tabelle laden oder eine AWS Lambda Funktion aus dem Datenbankcode aufrufen. Weitere Informationen finden Sie unter Integration von Aurora MySQL mit anderen AWS Diensten.	18. Oktober 2016
Neue Funktion	Sie können jetzt auf die tempdb-Datenbank in Ihren Amazon RDS-DB-Instances zugreifen, auf denen Microsoft SQL Server ausgeführt wird. Sie können auf die tempdb-Datenbank mit Transact-SQL über Microsoft SQL Server Management Studio (SSMS) oder über eine andere Standard-SQL-Clientanwendung zugreifen. Weitere Informationen finden Sie unter Zugreifen auf die tempdb-Datenbank auf Microsoft SQL Server-DB-Instances auf Amazon RDS.	29. September 2016
Neue Funktion	Sie können jetzt das Paket UTL_MAIL für Amazon RDS-DB-Instances, auf denen Oracle ausgeführt wird, verwenden. Weitere Informationen finden Sie unter Orakel UTL _ MAIL.	20. September 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktionen	Sie können jetzt die Zeitzone für Ihre neuen Microsoft SQL Server-DB-Instances auf eine lokale Zeitzone setzen, damit sie mit der Zeitzone Ihrer Anwendungen übereinstimmt. Weitere Informationen finden Sie unter Lokale Zeitzone für Microsoft SQL Server-DB-Instances.	19. September 2016
Neues Feature	Sie können nun die Oracle Label Security-Option für die Zugriffskontrolle auf einzelne Tabellenzeilen der Amazon RDS-DB-Instances mit Oracle Database 12c verwenden. Dank Oracle Label Security können Sie die Compliance mit gesetzlichen Vorschriften über ein richtlinienbasiertes Administrationsmodell erzwingen und den Zugriff auf vertrauliche Daten auf Benutzer mit einer entsprechenden Berechtigungsstufe beschränken. Weitere Informationen finden Sie unter Oracle Label Security.	8. September 2016
Neue Funktion	Sie können nun die Verbindung zu einem Amazon Aurora-DB-Cluster über einen Lese-Endpunkt herstellen. Dieser führt einen Lastausgleich für die Verbindungen aller Aurora Replicas aus, die im DB-Cluster verfügbar sind. Während Clients neue Verbindungsanfragen an den Reader-Endpunkt tätigen, verteilt Aurora die Verbindungsanfragen zwischen den Aurora Replicas in einem DB-Cluster. Diese Funktionalität kann dabei helfen, den Workload bei Lesevorgängen zwischen mehreren Aurora Replicas in Ihrem DB-Cluster auszugleichen. Weitere Informationen finden Sie unter Amazon Aurora-Endpunkte.	8. September 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Es wird jetzt Unterstützung für Oracle Enterprise Manager Cloud Control für Amazon RDS-DB-In stances, auf denen Oracle ausgeführt wird, geboten. Sie können Management Agent auf den DB-Instances aktivieren und Daten für Oracle Management Service (OMS) freigeben. Weitere Informationen finden Sie unter Oracle Management Agent für Enterprise Cloud Control.	1. September 2016
Neue Funktion	Mit diesem Release wird die Unterstützung für den ARN-Abruf einer Ressource hinzugefügt. Weitere Informationen finden Sie unter Ein vorhandenes ARN für Amazon erwerben RDS.	23. August 2016
Neue Funktion	Sie können nun bis zu 50 Tags für jede Amazon RDS-Ressource zuweisen, um Ihre Ressourcen zu verwalten und Kosten nachzuverfolgen. Weitere Informationen finden Sie unter Verschlagwortung von Amazon-Ressourcen RDS.	19. August 2016
Neue Funktion	Amazon RDS unterstützt jetzt das Modell der enthalten en Lizenz für Oracle Standard Edition Two. Weitere Informationen finden Sie unter Erstellen einer Amazon RDS-DB-Instance. Sie können nun das Lizenzmodell Ihrer Amazon RDS-DB-Instances, auf denen Microsoft SQL Server und Oracle ausgeführt werden, ändern. Weitere Informati onen erhalten Sie unter Lizenzierung Microsoft SQL Server auf Amazon RDS und RDSfür Oracle-Li zenzoptionen.	5. August 2016

Änderungen	Beschreibung	Datum geändert
Neues Feature	Amazon RDS unterstützt jetzt native Backups und Wiederherstellungen für Microsoft SQL Server-Datenbanken mit vollständigen Sicherungsdateien (BAK-Dateien). Sie können jetzt problemlos SQL Server-Datenbanken zu Amazon RDS migrieren und Datenbanken in einer einzigen, leicht portierbaren Datei importieren und exportieren, wobei Sie Amazon S3 für die Speicherung und AWS KMS Verschlüsselung verwenden. Weitere Informationen finden Sie unter Importieren und Exportieren von SQL-Server-Datenbanken mithilfe nativer Sicherung und Wiederherstellung.	27. Juli 2016
Neue Funktion	Sie können jetzt die Quelldateien von einer MySQL-Datenbank in ein Amazon Simple Storage Service (Amazon-S3)-Bucket kopieren und dann ein Amazon Aurora-DB-Cluster aus diesen Dateien wiederherstellen. Dieser Weg ist bedeutend schneller als eine Datenmigration mit `mysqldump` . Weitere Informationen finden Sie unter Migrieren von Daten aus einer externen MySQL-Datenbank in einen Aurora MySQL-DB-Cluster.	20. Juli 2016
Neues Feature	Sie können jetzt einen unverschlüsselten Amazon Aurora Aurora-DB-Cluster-Snapshot wiederherstellen, um einen verschlüsselten Amazon Aurora Aurora-DB-Cluster zu erstellen, indem Sie während des Wiederherstellungsvorgangs einen AWS Key Management Service (AWS KMS) Verschlüsselungsschlüssel angeben. Weitere Informationen finden Sie unter Verschlüsseln von Amazon RDS-Ressourcen.	30. Juni 2016

Änderungen	Beschreibung	Datum geändert
Neues Feature	Sie können Oracle Repository Creation Utility (RCU) verwenden, um ein Repository in Amazon RDS for Oracle zu erstellen. Weitere Informationen finden Sie unter Verwenden des Oracle Repository Creation Utility (RCU) in RDS for Oracle .	17. Juni 2016
Neue Funktion	Unterstützung für regionenübergreifende Lesereplikate von PostgreSQL hinzugefügt. Weitere Informationen finden Sie unter Erstellen eines Lesereplikats in einer anderen AWS-Region.	16. Juni 2016
Neues Feature	Sie können das jetzt verwenden AWS Managemen t Console , um Multi-AZ mit Spiegelung einfach zu einer Microsoft SQL Server-DB-Instance hinzuzufügen. Weitere Informationen finden Sie unter Hinzufügen von Multi-AZ zu einer Microsoft SQL Server-DB-Instance.	9. Juni 2016
Neues Feature	Sie können jetzt Multi-AZ-Bereitstellungen mit SQL Server-Spiegelung in folgenden weiteren Regionen verwenden: Asien-Pazifik (Sydney), Asien-Pazifik (Tokio) und Südamerika (São Paolo). Weitere Informati onen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.	9. Juni 2016
Neue Funktion	Aktualisiert für die Unterstützung von MariaDB-Version 10.1. Weitere Informationen finden Sie unter Amazon RDS für MariaDB.	1. Juni 2016
Neue Funktion	Aktualisiert für die Unterstützung der regionenü bergreifenden DB-Cluster von Amazon Aurora, bei denen es sich um Lesereplikate handelt. Weitere Informationen finden Sie unter Replizieren von Aurora-MySQL-DB-Clustern über AWS -Regionen hinweg.	1. Juni 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	„Enhanced Monitoring“ (Erweiterte Überwachung) ist jetzt für Oracle-DB-Instances verfügbar. Weitere Informationen erhalten Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung) und Ändern einer Amazon RDS DB-Instance.	27. Mai 2016
Neue Funktion	Aktualisiert für die Unterstützung der manuellen Snapshot-Freigabe für Amazon Aurora-DB-Cluster-Snapshots. Weitere Informationen finden Sie unter Freigeben eines DB-Cluster-Snapshots.	18. Mai 2016
Neue Funktion	Sie können jetzt das MariaDB-Audit-Plugin verwenden , um die Datenbankaktivität auf MariaDB- und MySQL-DB-Instances zu protokollieren. Weitere Informationen erhalten Sie unter Optionen für MariaDB-Datenbank-Engine und Optionen für Meine SQL DB-Instances.	27. April 2016
Neue Funktion	Nun sind direkte Hauptversions-Upgrades von MySQL-Version 5.6 auf Version 5.7 verfügbar. Weitere Informationen finden Sie unter Upgrades der DB-Engine RDS für MySQL.	26. April 2016
Neue Funktion	„Enhanced Monitoring“ (Erweiterte Überwachung) ist jetzt für Microsoft SQL Server-DB-Instances verfügbar . Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).	22. April 2016
Neues Feature	Aktualisiert für die Anzeige der Amazon Aurora-Cluster in der Amazon RDS-Konsole. Weitere Informationen finden Sie unter Anzeigen eines Aurora-DB-Clusters.	1. April 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung von SQL Server-Multi-AZ mit Spiegelung in der Region Asien-Pazifik (Seoul). Weitere Informationen finden Sie unter Multi-AZ-Bereitstellungen für Amazon RDS für Microsoft Server SQL.	31. März 2016
Neue Funktion	Aktualisiert für die Unterstützung von Amazon Aurora-Multi-AZ mit Spiegelung in der Region Asien-Paz ifik (Seoul). Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora MySQL.	31. März 2016
Neue Funktion	PostgreSQL-DB-Instances können nun die Nutzung von SSL für den Verbindungsaufbau erzwingen. Weitere Informationen finden Sie unter Verwenden von SSL mit einer PostgreSQL-DB-Instance.	25. März 2016
Neue Funktion	„Enhanced Monitoring“ (Erweiterte Überwachung) ist jetzt für PostgreSQL-DB-Instances verfügbar. Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).	25. März 2016
Neue Funktion	Microsoft SQL Server-DB-Instances können jetzt die Windows-Authentifizierung zur Benutzerauthentifi zierung verwenden. Weitere Informationen finden Sie unter Arbeiten mit AWS Managed Active Directory mit RDS für SQL Server.	23. März 2016
Neue Funktion	„Enhanced Monitoring“ (Erweiterte Überwachung) ist jetzt für die Region Asien-Pazifik (Seoul) verfügbar. Weitere Informationen finden Sie unter Überwachen von Betriebssystem-Metriken mithilfe von „Enhanced Monitoring“·(Erweiterte·Überwachung).	16. März 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Sie können nun die anpassen, in welcher Reihenfolge die Aurora Replicas während eines Failovers zur primären Instance hochgestuft werden. Weitere Informationen finden Sie unter Fehlertoleranz für einen Aurora-DB-Cluster.	14. März 2016
Neue Funktion	Aktualisiert für die Unterstützung der Verschlüsselung während einer Migration zu einem Aurora-DB-Cluster. Weitere Informationen finden Sie unter Migrieren von Daten zu einem Aurora-DB-Cluster.	2. März 2016
Neue Funktion	Aktualisiert für die Unterstützung von lokalen Zeitzonen für Aurora-DB-Cluster. Weitere Informationen finden Sie unter Lokale Zeitzone für Aurora-DB-Cluster.	1. März 2016
Neue Funktion	Aktualisiert für die Unterstützung von MySQL-Version 5.7 für die aktuelle Generation von Amazon RDS-DB-Instance-Klassen.	22. Februar 2016
Neues Feature	Es wurde aktualisiert, um die DB-Instance-Klassen db.r3 und db.t2 in der Region (US-West) zu unterstützen. AWS GovCloud	11. Februar 2016
Neue Funktion	Aktualisiert für die Unterstützung der Kopie-Verschlüsselung von DB-Snapshots und der Freigabe von verschlüsselten DB-Snapshots. Weitere Informationen erhalten Sie unter Kopieren eines DB-Snapshots für Amazon RDS und Einen DB-Snapshot für Amazon teilen RDS.	11. Februar 2016
Neue Funktion	Aktualisiert mit der Unterstützung von Amazon Aurora in der Region Asien-Pazifik (Sydney). Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora MySQL.	11. Februar 2016

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung von SSL für Oracle-DB-Instances. Weitere Informationen finden Sie unter Verwenden von SSL mit einer DB-Instance von RDS für Oracle.	9. Februar 2016
Neue Funktion	Aktualisiert für die Unterstützung der lokalen Zeitzone auf MySQL- und MariaDB-DB-Instances. Weitere Informationen erhalten Sie unter Lokale Zeitzone für My SQL DB-Instances und Lokale Zeitzone für MariaDB DB-Instances.	21. Dezember 2015
Neue Funktion	Aktualisiert für die Unterstützung von „Enhanced Monitoring“ (Erweiterte Überwachung) von Betriebss ystem-Metriken für MySQL- und MariaDB-Instances sowie Aurora-DB-Cluster. Weitere Informationen finden Sie unter Metriken in der RDS Amazon-Konsole anzeigen.	18. Dezember 2015
Neues Feature	Aktualisiert für die Unterstützung der DB-Instan ce-Klassen db.t2, db.r3 und db.m4 für MySQL-Ver sion 5.5. Weitere Informationen finden Sie unter .	4. Dezember 2015
Neue Funktion	Aktualisiert für die Unterstützung von Änderungen des Datenbankports bei einer bestehenden DB-Instance.	3. Dezember 2015
Neues Feature	Aktualisiert für die Unterstützung von Hauptversions-Upgrades der Datenbank-Engine für PostgreSQL-Instances. Weitere Informationen finden Sie unter Upgrades der RDS for PostgreSQL-DB-Engine.	19. November 2015
Neue Funktion	Aktualisiert für die Unterstützung von Änderungen des öffentlichen Zugriffs bei einer bestehenden DB-Instan ce. Aktualisiert für die Unterstützung von db.m4-Sta ndard-DB-Instance-Klassen.	11. November 2015

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung der manuellen DB-Snapshot-Freigabe. Weitere Informationen finden Sie unter Einen DB-Snapshot für Amazon teilen RDS.	28. Oktober 2015
Neue Funktion	Aktualisiert für die Unterstützung von Microsoft SQL Server 2014 für die Web Express und Standard Editionen.	26. Oktober 2015
Neue Funktion	Aktualisiert für die Unterstützung der MySQL-basierten MariaDB-Datenbank-Engine. Weitere Informationen finden Sie unter Amazon RDS für MariaDB.	7. Oktober 2015
Neue Funktion	Aktualisiert mit der Unterstützung von Amazon Aurora in der Region Asien-Pazifik (Tokio). Weitere Informationen finden Sie unter Verfügbarkeit für Amazon Aurora MySQL.	7. Oktober 2015
Neue Funktion	Aktualisiert für die Unterstützung der serienfähigen DB-Instance-Klassen "db-t2" für alle DB-Engines, zudem wurde die DB-Instance-Klasse "db.t2.large" hinzugefügt. Weitere Informationen finden Sie unter .	25. September 2015
Neue Funktion	Aktualisiert für die Unterstützung von Oracle-DB-Instances in R3- und T2-DB-Instance-Klassen. Weitere Informationen finden Sie unter .	5. August 2015
Neues Feature	Microsoft SQL Server Enterprise Edition ist jetzt mit dem Servicemodell der enthaltenen Lizenz verfügbar . Weitere Informationen finden Sie unter Lizenzierung Microsoft SQL Server auf Amazon RDS.	29. Juli 2015
Neue Funktion	Amazon Aurora wurde offiziell veröffentlicht. Die Amazon Aurora-DB-Engine unterstützt mehrere DB-Instances in einem DB-Cluster. Detaillierte Informationen finden Sie unter Was ist Amazon Aurora?.	27. Juli 2015

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung zum Kopieren von Tags in DB-Snapshots.	20. Juli 2015
Neue Funktion	Aktualisiert für die Unterstützung eines größeren Speichers für alle DB-Engines und höheren bereitgestellten IOPS für SQL Server.	18. Juni 2015
Neue Funktion	Aktualisierte Optionen für reservierte DB-Instances.	15. Juni 2015
Neues Feature	Aktualisiert für die Unterstützung der Verwendung von Amazon CloudHSM mit Oracle-DB-Instances mit TDE.	8. Januar 2015
Neue Funktion	Aktualisiert für die Unterstützung der Verschlüsselung von Daten im Ruhezustand und der neuen API-Version 2014-10-31.	6. Januar 2015
Neues Feature	Aktualisiert für die Einbindung der neuen Amazon-DB-Engine: Aurora. Die Amazon Aurora-DB-Engine unterstützt mehrere DB-Instances in einem DB-Cluster. Amazon Aurora befindet sich in der Vorschauversion und kann noch geändert werden. Detaillierte Informationen finden Sie unter Was ist Amazon Aurora?.	12. November 2014
Neue Funktion	Aktualisiert für die Unterstützung von PostgreSQL-Lesereplikaten.	10. November 2014
Neue API und Funktionen	Aktualisiert, um den GP2 Speichertyp und die neue API-Version 2014-09-01 zu unterstützen. Aktualisiert für die Unterstützung der Funktion, eine bestehende Options- oder Parametergruppe zu kopieren und daraus eine neue Options- oder Parametergruppe zu erstellen.	7. Oktober 2014
Neue Funktion	Aktualisiert für die Unterstützung von InnoDB-Cache-Warming für DB-Instances, auf denen MySQL-Version 5.6.19 und neuer ausgeführt wird.	3. September 2014

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung der SSL-Zerti fikatsüberprüfung beim Verbindungsaufbau zu MySQL- (Version 5.6), SQL Server- und PostgreSQL-Datenba nk-Engines.	5. August 2014
Neues Feature	Aktualisiert für die Unterstützung der DB-Instance-Klassen db.t2 mit Spitzenlastleistung.	4. August 2014
Neues Feature	Aktualisiert für die Unterstützung von arbeitssp eicheroptimierten DB-Instance-Klassen db.r3 für die MySQL- (Version 5.6), SQL Server- und PostgreSQL-Datenbank-Engines.	28. Mai 2014
Neue Funktion	Aktualisiert für die Unterstützung von SQL Server-Mu lti-AZ-Bereitstellungen mit SQL Server-Spiegelung.	19. Mai 2014
Neue Funktion	Aktualisiert für die Unterstützung der Upgrades von MySQL-Version 5.5 auf 5.6.	23. April 2014
Neues Feature	Zur Unterstützung von Oracle aktualisiert. GoldenGate	3. April 2014
Neue Funktion	Aktualisiert für die Unterstützung von M3-DB-Instance-Klassen.	20. Februar 2014
Neue Funktion	Aktualisiert für die Unterstützung der Oracle-Ze itzonenoption.	13. Januar 2014
Neue Funktion	Aktualisiert für die Unterstützung der Replikation von MySQL-DB-Instances in verschiedenen Regionen.	26. November 2013
Neue Funktion	Aktualisiert für die Unterstützung der PostgreSQL-DB-Engine.	14. November 2013
Neue Funktion	Aktualisiert für die Unterstützung von SQL Server Transparent Data Encryption (TDE).	7. November 2013

Änderungen	Beschreibung	Datum geändert
Neue API und neue Funktion	Aktualisiert für die Unterstützung regionenübergreifender DB-Snapshot-Kopien und neue API-Version 2013-09-09.	31. Oktober 2013
Neue Funktionen	Aktualisiert für die Unterstützung von Oracle Statspack.	26. September 2013
Neue Features	Aktualisiert, um die Verwendung der Replikation für den Import oder Export von Daten zwischen MySQL-Instances, die in Amazon RDS ausgeführt werden, und MySQL-Instances, die lokal oder auf Amazon EC2 laufen, zu unterstützen.	5. September 2013
Neue Funktionen	Aktualisiert für die Unterstützung der DB-Instance-Klasse "cr1.8xlarge" für MySQL 5.6.	4. September 2013
Neue Funktion	Aktualisiert für die Unterstützung der Replikation von Lesereplikaten.	28. August 2013
Neue Funktion	Aktualisiert für die Unterstützung der parallelen Erstellung von Lesereplikaten.	22. Juli 2013
Neue Funktion	Aktualisiert für die Unterstützung von differenzierten Berechtigungen und Tagging für alle Amazon RDS-Ressourcen.	8. Juli 2013
Neue Funktion	Aktualisiert für die Unterstützung von MySQL 5.6 bei neuen Instances, einschließlich Unterstützung für die Memcached-Schnittstelle von MySQL 5.6 und Zugriff auf Binärprotokolle.	1. Juli 2013
Neue Funktion	Aktualisiert für die Unterstützung von Hauptversions-Upgrades von MySQL-Version 5.1 auf MySQL 5.5.	20. Juni 2013
Neue Funktion	Aktualisierte DB-Parametergruppen, um Ausdrücke für Parameterwerte zuzulassen.	20. Juni 2013

Änderungen	Beschreibung	Datum geändert
Neue API und neue Funktion	Aktualisiert für die Unterstützung des Lesereplikat-Status und neue API-Version 2013-05-15.	23. Mai 2013
Neue Funktionen	Aktualisiert für die Unterstützung der Oracle Advanced Security-Funktionen für Native Network Encryption (NNE) und Oracle Transparent Data Encryption (TDE).	18. April 2013
Neue Funktionen	Aktualisiert für die Unterstützung von Hauptversions-Updates für SQL Server und zusätzlicher Funktionalität für bereitgestellte IOPS.	13. März 2013
Neue Funktion	Aktualisiert für die standardmäßige Unterstützung der Standard-VPC für RDS.	11. März 2013
Neue API und Funktion	Aktualisiert für die Unterstützung des Protokollzugriffs und neue API-Version 2013-02-12.	4. März 2013
Neue Funktion	Aktualisiert für die Unterstützung der Abonnements für RDS-Ereignisbenachrichtigungen.	4. Februar 2013
Neue API und Funktion	Aktualisiert für die Unterstützung der Umbenennung von DB-Instances und der Migration von DB-Sicher heitsgruppenmitgliedern in einer VPC in eine VPC-Sicherheitsgruppe.	14. Januar 2013
Neues Feature	Für Unterstützung AWS GovCloud (US-West) aktualisi ert.	17. Dezember 2012
Neue Funktion	Aktualisiert für die Unterstützung der DB-Instance-Klassen m1.medium und m1.xlarge.	6. November 2012
Neue Funktion	Aktualisiert für die Unterstützung einer Lesereplikat-Hochstufung.	11. Oktober 2012
Neue Funktion	Aktualisiert für die Unterstützung von SSL für Microsoft -SQL-Server-DB-Instances.	10. Oktober 2012

Änderungen	Beschreibung	Datum geändert
Neue Funktion	Aktualisiert für die Unterstützung von Oracle-micro-DB-Instances.	27. September 2012
Neue Funktion	Aktualisiert für die Unterstützung von SQL Server 2012.	26. September 2012
Neue API und Funktion	Aktualisiert für die Unterstützung von bereitgestellten IOPS. API-Version 2012-09-17.	25. September 2012
Neue Funktionen	Aktualisiert für SQL-Server-Unterstützung von DB-Instances in einer VPC und Oracle-Unterstützung von Data Pump.	13. September 2012
Neue Funktion	Aktualisiert für die Unterstützung von SQL Server Agent.	22. August 2012
Neue Funktion	Aktualisiert für die Unterstützung des Taggings von DB-Instances.	21. August 2012
Neue Funktionen	Aktualisiert für die Unterstützung von Oracle APEX und XML DB, Oracle-Zeitzonen und Oracle-DB-Instances in einer VPC.	16. August 2012
Neue Funktionen	Aktualisiert für die Unterstützung des SQL-Server-Datenbankoptimierungsratgebers und Oracle-DB-Instances in VPC.	18. Juli 2012
Neue Funktion	Aktualisiert für die Unterstützung von Optionsgruppen und Erstoption, Oracle Enterprise Manager Database Control.	29. Mai 2012
Neue Funktion	Aktualisiert für die Unterstützung von Lesereplikaten in Amazon Virtual Private Cloud.	17. Mai 2012
Neue Funktion	Aktualisiert für Microsoft SQL Server-Support.	8. Mai 2012

Änderungen	Beschreibung	Datum geändert
Neue Funktionen	Aktualisiert für die Unterstützung von erzwungen em Failover, Multi-AZ-Bereitstellung von Oracle-DB-Instances und nicht-standardmäßigen Zeichensätzen für Oracle-DB-Instances.	2. Mai 2012
Neue Funktion	Aktualisiert für die Unterstützung von Amazon Virtual Private Cloud (VPC).	13. Februar 2012
Aktualisierter Inhalt	Aktualisiert für neue Reserved Instance-Typen.	19. Dezember 2011
Neue Funktion	Aktualisiert für Oracle-Engine-Support.	23. Mai 2011
Aktualisierter Inhalt	Konsolen-Updates.	13. Mai 2011
Aktualisierter Inhalt	Bearbeiteter Inhalt für kürzere Sicherungs- und Wartungsfenster.	28. Februar 2011
Neue Funktion	Unterstützung für MySQL 5.5 hinzugefügt.	31. Januar 2011
Neue Funktion	Unterstützung für Lesereplikate hinzugefügt.	4. Oktober 2010
Neues Feature	Unterstützung für AWS Identity and Access Management (IAM) hinzugefügt.	2. September 2010
Neue Funktion	DB-Engine-Versionsverwaltung hinzugefügt.	16. August 2010
Neue Funktion	Reservierte DB-Instances hinzugefügt.	16. August 2010
Neue Funktion	Amazon RDS unterstützt jetzt SSL-Verbindungen für Ihre DB-Instances.	28. Juni 2010
Neues Handbuch	Dies ist die erste Version des Amazon RDS-Benut zerhandbuch.	7. Juni 2010

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.